BACnet to Leverage IT

A Whitepaper on BACnet/IT

(BACnet Addendum 135-2016bj)

Bernhard Isler
ASHRAE SSPC 135 BACnet Committee

Abstract

BACnet/IP is known to have some properties that may not always be accepted in today IP infrastructures. In particular in shared infrastructures that are tightly managed by IT departments; or when high-secure device to device communication is required.

BACnet/IT is enabling BACnet communication within such infrastructures, yet remaining fully compatible with BACnet devices implementing the BACnet network layer. In BACnet/IT, duties such as segmentation, network security, and network routing are left to IP application and lower layer protocols entirely. The BACnet network layer is no longer needed. For compatibility, the application layer is retained, yet simplified. Initially, secure WebSockets are used for transporting BACnet messages; and DNS service discovery (DNS-SD) is used for resolution and discovery. Usage of other IP application protocols can be added over time.

Contents

Figures

Tables

1. Overview

This whitepaper summarizes and outlines the requirements of the proposed BACnet/IT solution. Addendum 135-2016bj, referred to as the "BACnet/IT addendum", specifies the solution in detail. BACnet/IT in general is built on well-known IP application protocols, yet flexible enough to be extended for use of additional and future IP application protocols.

BACnet/IT introduces a new option in lieu of the BACnet network layer, called the BACnet/IT layer. The BACnet/IT layer essentially consists of transport and directory ports, which enable the transport of BACnet messages and directory information across well-known IP application protocols including their standard security mechanisms. Some new application layer functionality is defined as well, for support of functions like Directory, Device Group Coordination, and Device Proxying.

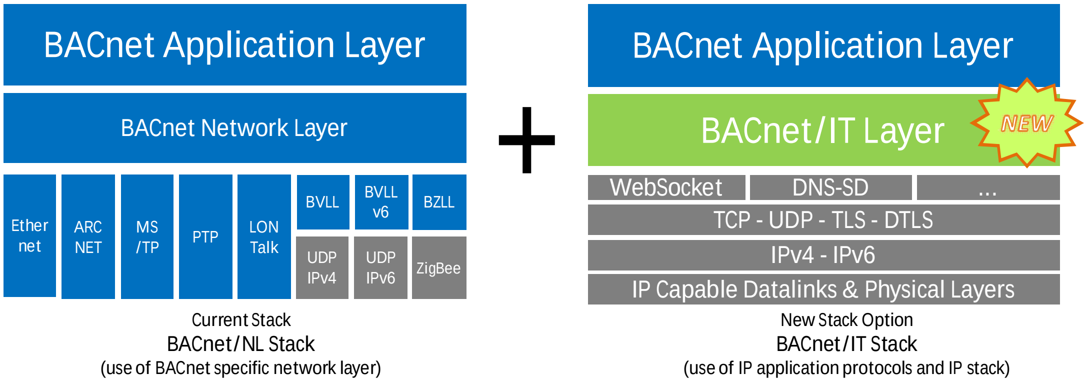


Figure 1. BACnet/NL Stack and BACnet/IT Stack

Initially, BACnet/IT defines the use of WebSockets (RFC 6455) for message transport, and DNS based Service Discovery (DNS-SD, RFC 6763) for resolution and discovery. The usage of other protocols can be added over time, without breaking the applications.

The Directory function enables BACnet to avoid broadcasts for resolution and discovery. The Device Group Coordination function allows messages being distributed to multiple recipient devices using a variety of mechanisms, suitable to the IP application protocols used. The Device Proxy function enables bridging among different IP application protocols used, but also connecting with BACnet devices using the BACnet/NL stack.

The following picture provides an example internetwork topology that includes BACnet/NL and BACnet/IT devices.

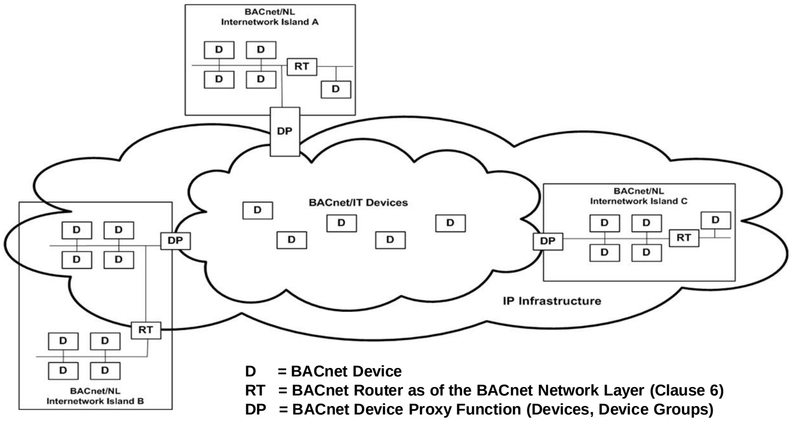

BACnet/NL Internetwork Island A
D
RT
DP
BACnet/IT Devices
BACnet/NL Internetwork Island C
IP Infrastructure
BACnet/NL Internetwork Island B
D = BACnet Device
RT = BACnet Router as of the BACnet Network Layer (Clause 6)
DP = BACnet Device Proxy Function (Devices, Device Groups)

Figure 2. BACnet Internetwork Topology with BACnet/IT

2. Current Situation

BACnet is more often required to use an existing IP infrastructure. The current approaches for BACnet/IP and BACnet/IPv6 are not suitable for all such infrastructures.

2.1 Networking Today and Trends

Some notable points regarding the today networking and trends to the future:

- 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.
- BACnet's own network layer stack is perceived by some IT organizations as exotic or less desirable.
- Similar network layer approaches have disappeared from the market (Novell Netware, DECnet, etc.).
- 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 controls domain, network standards are rapidly moving to IP-centric solutions.
- IPv6 is emerging.
- The Building Automation (BA) market is demanding simple plug-and-play devices.
- The size of internetworked systems is increasing.

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

2.2 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 and BACnet/IP 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 TLS.
- Demand for fixed IP addresses, in particular for BBMDs.
- 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.

2.3 Values of BACnet that Need to be Preserved

There are important characteristics of BACnet that need to be preserved. A fundamental goal is that any additions to BACnet must preserve interoperability with current BACnet devices.

- 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.

2.4 Features that Need to be Enabled for BACnet

A number of features were identified that are needed for BACnet to exist in today's and future IP infrastructures.

- Enable the use of IP networks in a way suitable for highly managed IP infrastructures, not just as a virtual LAN.
- Enable the use of standard IP application protocols, such as HTTP.
- 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).

3. Requirements Summary

The requirements on solutions for BACnet in IP infrastructures are the following. These requirements are written under the assumption that BACnet will be using IP application protocols for its communication.

3.1 Basic Requirements for Internet Application Protocol Bindings

For using IP application protocols for BACnet, here are a number of requirements when doing so.

- Keep the current data model and services of the application level that are related to BA functionality.
- Facilitate transmission and reception of NPDUs as an octet string
- Current network infrastructure related services may be modified, extended, or replaced to accommodate for IP infrastructure requirements.
- Enable integration and interoperation with current BACnet devices with a minimum of effort to develop and deploy in installations.
- Shield the application level from network specifics.
- Use standard IP connectivity.
- Use well known IP network functions for discovery and resolution.
- Use well known IP transport- and application protocols.
- Be agnostic to the IP version (IPv4 and IPv6).
- Minimize IP subnet broadcasts on IPv4 to well known link-local broadcasts, if needed.
- Enable the use of multicast where the infrastructure supports it, if needed.
- Ensure scalability (multiple issues to be considered, see below).
- Provide network security by using established IP security mechanisms of the protocols used.
- Enhanced reliability over BACnet/NL session functionality of the application layer.

IP/IT management functionality such as SNMP is a product feature that is not strictly a requirement for BACnet devices. Some feature sets may be needed to enable common IT network management functionality even in multi-vendor installations. The definition of the feature set required is out of scope for this addendum.

3.2 Scalability Aspects

Solutions for BACnet in IP infrastructures have to consider a number of scalability aspects:

- Device size (platforms, objects, price/costs ...).
- Device performance, from sleepy devices to high end computing platforms.
- Engineering and commissioning workflows.
- Single IP subnet networks with no IP routers or infrastructure services up to highly structured and wide area networks.
- Low performance constrained networks to high performance backbone networks
- Network management levels and policies (no network management to very strict policies)
- Security features and policies (plain to high security)
- Application functions (latency, throughput, bandwidth, ...)
- Network functions (resolution, discovery, ...) to scale from small to very large systems

3.3 Non-Functional Requirements

A number of non-functional quality requirements were identified.

- Solutions for BACnet in IP infrastructures must ensure that the normal lifetime of an installed BA system can be reached.
- In a given environment and given set of constraints, the overall lifecycle costs of a BACnet/IT based system should be comparable to a BACnet/NL based system.
- Sufficient availability through support of redundancy of network functions, network infrastructure, and media.
- BA systems that can be installed, engineered, and commissioned by the same field staff that is installing BACnet/NL based BACnet systems.

4. BACnet/IT for BACnet in IT Environments

For a solution to these issues, a more comprehensive approach was chosen than just expanding BACnet/IP to address just some of the problems, yet likely creating incompatibilities, and yet another datalink type. BACnet/IT, in contrast to BACnet/IP, or more general to BACnet with its network layer, delegates all segmentation, routing, and security functionality to IP application and lower layer IP protocols, known to and manageable by IT departments.

The protocol architecture enhancements for BACnet/IT are specified in section 1 of the BACnet/IT addendum.

4.1 BACnet/NL Stack

The current BACnet protocol stack, as shown in Figure 3 below, has its own network layer. Since a BACnet defined network layer is used, this stack is named the BACnet/NL stack. BACnet devices, when communicating through this stack, are considered BACnet/NL devices.

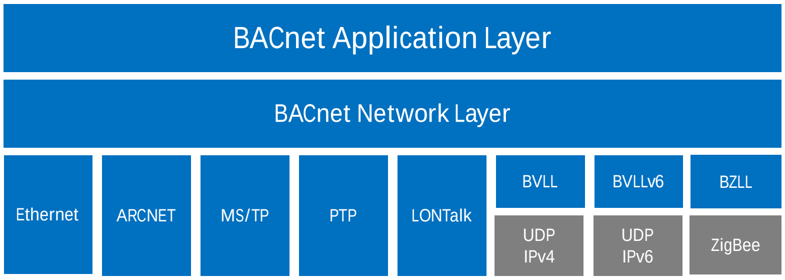


Figure 3. Current Stack: BACnet/NL Stack

4.2 BACnet/IT Stack

BACnet/IT has no need for the BACnet network layer. In lieu, the BACnet/IT Layer binds the BACnet application layer onto IP application protocols. This new stack is referred to as the BACnet/IT stack. BACnet devices communicating through this stack are considered BACnet/IT devices.

The BACnet/IT stack is not meant to replace the BACnet/NL stack. It is an addition to the standard. For specific application situations, it will be practical to use BACnet/NL devices, which tightly cooperate with BACnet/IT devices.

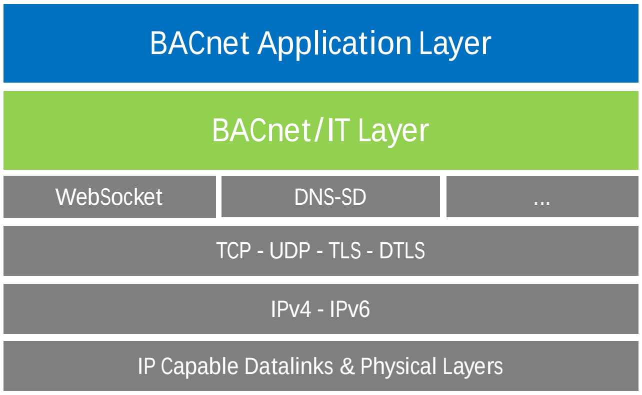


Figure 4. Additional: BACnet/IT Stack

The core element of the BACnet/IT stack is the BACnet/IT Layer. This new layer allows the BACnet application layer to exchange BACnet service messages and to discover devices and objects. The BACnet/IT layer includes the implementation of at least one transport binding. An instantiated transport binding implementation is referred to as a transport port which enables the transport of messages across the network. The BACnet/IT layer also includes the implementation of at least one directory binding. An instantiated directory binding implementation is referred to as a directory port, enabling network communication for resolution and discovery. Initially, the binding to the WebSocket protocol is defined for implementing transport ports. For directory functionality, the binding to DNS-SD over DNS or mDNS is defined for implementing directory ports. Future protocol bindings and respective ports may be added as needed.

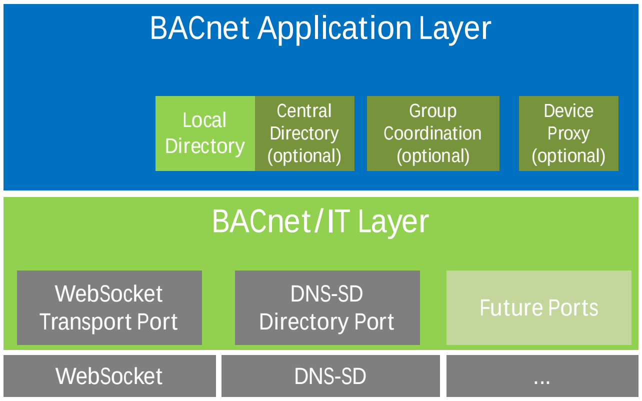


Figure 5. BACnet/IT Overview

Aside the BACnet/IT layer, application layer functionality is defined for supporting functions, such as for the directory, for device group communication, and for device proxying. These application layer functions are mostly optional and are not required to be implemented in all devices. The minimum requirement is the support of the local directory.

5. BACnet/IT Layer

The BACnet/IT layer essentially replaces the BACnet network layer. It includes transport and directory ports, and exposes these ports to the application layer, through abstracted interfaces called service access points (SAP). The BACnet/IT layer is introduced in section 4 of the BACnet/IT addendum, in the preamble of proposed new Annex ZZ.

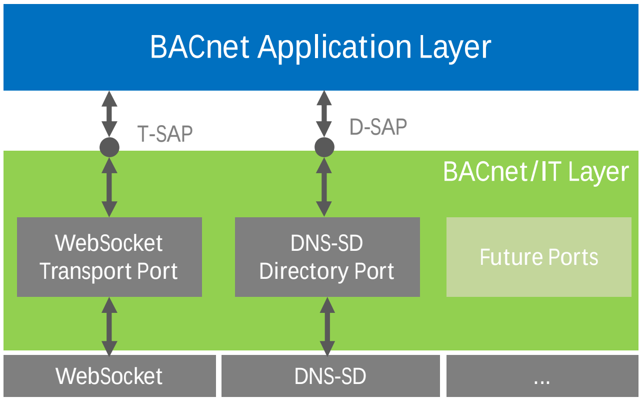


Figure 6. BACnet/IT Layer

The transport service access point (T-SAP) is defined for the application layer to have a network interface that is similar to the BACnet network layer. Transforming an implementation from BACnet/NL to BACnet/IT does not require significant modifications and may even simplify it.
The directory service access point (D-SAP) provides an abstract interface for the directory functionality of the application layer. It supports identifier resolution and entity discovery.

5.1 Abstracted Communication Model

The interface provided by the BACnet/IT layer is abstracted to hide specifics of the IP application protocols used, allowing extensibility of BACnet/IT for other IP application protocols. The application is assumed to work with logical BACnet device and BACnet device group identifiers which are not tied to or depend on lower protocol layers or network structures.

5.1.1. Communication Peers

At the service access points of the BACnet/IT layer, the communication model is abstracted such that for the application layer, communication is considered to take place between BACnet devices, logically identified by a numerical Device ID. For the definition of communication peers and their identification see new Clause ZZ.1 in section 4 of the BACnet/IT addendum.

BACnet broadcast communication is generalized and abstracted to be a communication from a device to a group of devices, referred to as BACnet Device Group, logically identified by a numerical Device Group ID. For distributing requests to a device group, the device group coordination function of the application layer distributes requests received for the group. See section 7 Device Group Coordination.

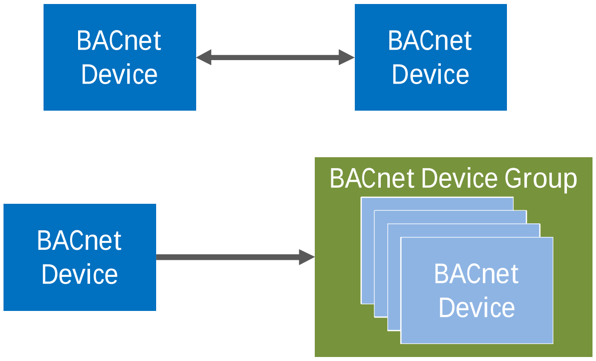


Figure 7. BACnet Communication Abstracted in BACnet/IT

BACnet device groups can be hierarchical. Device groups may be a member of a higher level device group. Devices and device groups may be member of multiple device groups.

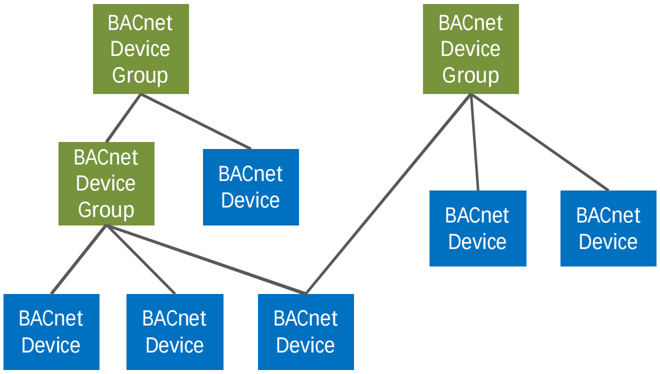


Figure 8. BACnet Device Group Hierarchy

A misconfiguration of complex device group hierarchies may lead to multiple paths and loops. Multiple hierarchy paths may lead to message duplications. This is critical for non-idempotent requests. For the sake of simplicity, no explicit protocol mechanism is in place to avoid such duplicates. The message sequence number may be used for identifying duplicates. Membership loops may cause messages to circulate indefinite. To stop these messages from circulating, a forwards credit mechanism is in place. If this credit is exhausted, the message is dropped instead of being forwarded again. Generally, it is important that device group configurations avoid multiple paths and loops.

Both BACnet devices and the device group coordination function for BACnet devices groups are hosted on some IP host. An IP host may contain one or multiple BACnet/IT devices and device groups, under a single or multiple URLs.

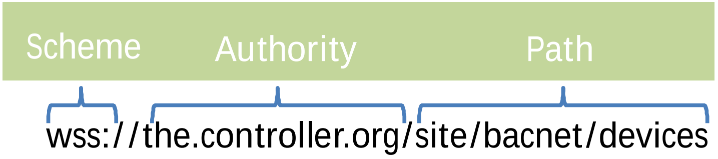
5.1.2. Logical Entity IDs and URLs

For flexibility and future extensibility, the logical identifiers for both BACnet Devices and BACnet device groups are considered BACnet Entity IDs, or EIDs.

Table 1. BACnet Entity ID

BACnet Entity ID Option	Range	Description
Unassigned	-	No entity ID is assigned
Device ID	0..4194302	Device Object Instance Number
Device Group ID	1..4194302	Device Group Object Instance Number

Before communication can take place, an EID is resolved to a URL. The local directory functionality includes a resolver for resolving an EID into a URL. The BACnet/IT layer T-SAP expects that the application is providing a URL for locating the destination of a message. An example URL for the WebSocket transport port could look like:



From that URL, the scheme portion selects the transport port and IP application protocol being used, secure WebSockets in the example, the authority portion identifies the destination IP host which hosts the destination BACnet device, and the resource path portion locates the communication endpoint for the BACnet device on that IP host.

5.1.3. BACnet Services

BACnet/IT enables the communication of confirmed and unconfirmed BACnet services, as defined in the BACnet standard's Clauses 13 to 17. For this, the transport service access point provides sending and receiving complete request and response messages, releasing the application layer from performing segmentation. For more details on exchanging BACnet application layer service messages, see new Clause ZZ.2.4 of the BACnet/IT addendum.

For confirmed services, the application layer performs the transaction state machines as defined in Clause 5 of the BACnet standard. Contrary to when using the network layer, these transaction state machines are simplified in that only the states and transitions are taken for unsegmented requests and unsegmented responses.

Unicast unconfirmed service request messages are simply transmitted to the peer device. No transaction state machines are required.

Unconfirmed service request messages to groups of devices are supported by the device group coordination functionality. The destination of the initial message is a BACnet device group. On behalf of the device group, the device group coordination function for the device group receives the request and distributes it to the group members. See section 7 Device Group Coordination.

5.1.4. BACnet NPDUs

As an additional feature and to support certain connectivity and migration scenarios, the transport service access point also supports exchanging BACnet Network Layer PDUs. See below in section 10 BACnet Virtual Router Link (BVRL) which is defined to use this feature for connecting BACnet/NL stack based BACnet half-routers.

5.2 Transport Ports

A transport port implements a transport binding, i.e. the defined application of an IP application protocol for BACnet/IT. For the user of the BACnet/IT layer, an abstract interface is implemented that supports connection management and BACnet message transport. For specification details see Clause ZZ.6 in the BACnet/IT addendum.

5.2.1. Abstract Interface T-SAP

The abstract interface of the transport port is expressed in so called service primitives. The set of service primitives makes up the transport service access point (T-SAP). For connection management, the following primitives are defined to be supported by the transport port.

Table 2. T-SAP Connection Primitives

Primitive	Purpose
T-CONNECT.request	Request the establishment of an outbound connection.
T-CONNECT.confirm	Indicate the establishment of an outbound connection.
T-CONNECT.indication	Indicate the establishment of an inbound connection.
T-DISCONNECT.request	Request the disconnection from a peer.
T-DISCONNECT.confirm	Indicate that the request to disconnect has been completed
T-DISCONNECT.indication	Indicate the connection has been terminated.

For the transmission of messages, the following primitives are defined to be supported by a transport port:

Table 3. T-SAP Transmission Primitives

Primitive	Purpose
T-UNITDATA.request	Request to send a BACnet Transport PDU.
T-UNITDATA.indication	Indication of a BACnet Transport PDU received.
T-REPORT.indication	Indication of a failure to transmit a data unit.
T-RELEASE.request	Signal the transport port to release resources.

The messages transferred with the T-UNITDATA primitives are formatted as BACnet Transport Protocol Data Units (TPDU) and binary encoded. For the transport port they are handled as opaque octet strings.

5.2.2. BACnet Transport PDU (TPDU)

Any BACnet message, being a service request, a response, or an NPDU, that is to be sent through a transport port, or being received from the transport port, is formatted as a BACnet Transport PDU, or TPDU.

The TPDU contains a header with required and optional parameters, and a body which is a choice of payload options. The header of the message is a sequence of context tagged parameters. Some parameters are always present (R), others are optional (O) and may be required to be present depending on the TPDU body option.

Table 4. TPDU Header Parameters

TPDU Header Parameters		
Version	R	Definition version of the BACnet Transport PDU.
Priority	O	The network priority of the message in the body.
Destination EID	R	Destination EID of the message.
Original Destination EID	O	The original destination device group EID of the message.
Source EID	R	BACnet EID of the original sending BACnet device.
Invoke ID	O	Identifier for the confirmed service transaction.
Sequence Number	O	Source sequence number of the TPDU.
Forwards	O	Remaining forwards credits available for the TPDU.
Security	O	Security parameters for the security wrapper of Clause 24.

The TPDU body is a choice of one of the following payloads. Every option has a specific context tag and the option present in the PDU is identified by its respective context tag.

Table 5. TPDU Body Options

TPDU Body Options	
Transport Error	Error information for transport level errors
NPDU	Entire encoded NPDU as defined in Clause 6.
Confirmed Request	Complete and unsegmented confirmed BACnet request.
Unconfirmed Request	Unconfirmed BACnet request.
Complex ACK	Complete and unsegmented BACnet response.
Simple ACK	Simple BACnet response.
Error	Complete BACnet error response.
Abort	Abort message in a confirmed service transaction.
Reject	Reject message in a confirmed service transaction.

The "Transport Error" option is used for error situations that occur in local and remote transport ports and layers below.

The "NPDU" option is used for conveying NPDUs as defined in Clause 6 of the standard. This option is used only by the BACnet Virtual Router Link (BVRL). See section 10 BACnet Virtual Router Link (BVRL).

All remaining body options are used for confirmed and unconfirmed application layer service message transfers respectively. The service choice is indicated by a regular context tag within the respective body option. Note that there is no Segment-ACK option. This message is not required in BACnet/IT, since application layer segmentation is expected to be performed by the IP application protocol being used.

The TPDU is formally defined in ASN.1. See section 15 Important ASN.1 Definitions for BACnet/IT. Before being sent, the entire TPDU is completely and regularly ASN.1 encoded as defined in Clause 20.2 of the BACnet standard. Transport ports handle the encoded TPDU as an opaque octet string.

5.3 WebSocket Transport Port

The WebSocket protocol (RFC 6455) was chosen to be the first protocol for implementing transport ports for BACnet/IT. The WebSocket protocol enables bidirectional packet transfers across a TCP or TLS connection which is established in an HTTP and web context. See new Clause ZZ.6.2 in the BACnet/IT addendum.

The WebSocket sub-protocol identifier for BACnet/IT is proposed to be "asn.bacnet.org", and will be registered at IANA.

The following packet frame is defined for transfer across a WebSocket established for BACnet/IT. This frame conveys a TPDU, or a control message for operation of the transport port.

Table 6. WebSocket Packet Format for BACnet/IT

WebSocket Packet Format for BACnet/IT	
Version	Definition version of this format.
Control	Network priority and payload type.
Payload	The payload is one of the following options:
A) TPDU	A fully encoded TPDU as an octet string.
B) Control Message	A control message for the transport port.

A number of control messages enable optimized usage of the established connections. Control messages are defined for:

- WebSocket endpoint information, allows minimizing number of connections.
- Notification of BACnet devices and device groups reachable through the WebSocket connection.
- Notification of BACnet devices and device groups no longer reachable.

5.3.1. Configuration

A WebSocket transport port is represented by a BACnet application level Network Port object. This object allows configuring the support and use of the WebSocket protocol. In particular, it can be configured if the WebSocket port accepts inbound connections, initiates outbound connections, or both. A larger part of configuration parameters relate to keys and certificates for establishing secure WebSocket connections over TLS.

5.3.2. Security

The support of secure WebSockets is required. Secure WebSockets are based on TLS. The TLS connection is established already when the HTTP handshake is completed, and before the upgrade to the WebSocket connection takes place.

For the TLS connection, support of mutual authentication is required. The authentication is based on x.509 certificates. Trust is established by having an accepted authority certificate for the certificate presented by the peer.

The minimum cipher suite to be supported is TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8, as defined in RFC 7251. Support of additional cipher suites is optional.

For bootstrapping security, the following modes are supported:

Table 7. Security Modes for Security Bootstrapping

WebSocket Transport Port Security Modes	
Plain Factory Default	This mode allows initial configuration of security using plain communication. It is intended to be used in physically separated networks only.
Secure Factory Default	This mode allows configuring security using secure communication. Communication is possible only if the default certificates establish trust.
Normal Mode	All security is configured and secure communication is established.

The public certificate, the private key, and the trusted authority certificates can be configured through the Network Port object.

In addition, there is support for devices that generate their own key set and the private key never leaves the device. These devices are expected to also be able to generate their public certificate which can be taken by some agent to be signed by some certificate authority. This signed certificate is then configured in the WebSocket port Network Port object, and will match the device's hidden private key.

5.4 Directory Ports

A Directory Port implements the directory protocol binding definition for use of an IP application protocol for BACnet directory functionality. See new Clause ZZ.7 in the BACnet/IT addendum.

Although the abstract D-SAP interface to be implemented by a directory port is not specified in every detail or by primitives, this interface is required to support the following functionality used by the directory function:

- Resolve an EID to the respective URL, for both devices and device groups.
- Discover and enumerate BACnet devices, supporting a range filter for the devices to be discovered.
- Discover and enumerate BACnet objects, by object name or object identifier, supporting a range filter for the devices to respond.

The D-SAP interface essentially supports the functionality formerly achieved by Who-Is/I-Am and Who-Has/I-Have services.

A directory port, if its protocol includes server functionality, generally serves the directory information of the registry in response to remote queries received. For a centralized directory, the server function may also be implemented on some external server for the protocol. For example, the DNS server for centralized DNS-SD queries may be external to the IP host of the BACnet device that contains the central directory and controls the BACnet directory information on this central DNS server.

5.5 DNS-SD Directory Port

DNS-based service discovery, as defined by RFC 6763, is the first IP application protocol that is defined for a BACnet/IT directory port. See new Clause ZZ.7.2 in the BACnet/IT addendum.

5.5.1. DNS-SD Service Types

BACnet/IT defines the following DNS-SD service sub-types for BACnet. The basic BACnet service type is identified as "_bacnet._tcp" for TCP based protocols, and "_bacnet._udp" for all protocols using a transport layer other than TCP.

For differentiation of BACnet devices, device groups, objects, and the central directory, the following sub-types have been defined.

Table 8. DNS-SD Service Sub-Types for BACnet/IT

DNS-SD Service Sub-Types	
dev._sub._bacnet._tcp	This service subtype is used for BACnet devices and device groups.
obj._sub._bacnet._tcp	This service subtype is used for BACnet objects, by their identifier or object name.
bds._sub._bacnet._tcp	This service subtype is used for the central directory, known as the BACnet Directory Server (BDS).

5.5.2. DNS Protocol Support

The DNS.SD directory port supports both regular DNS for a centralized directory, and Multicast DNS (mDNS; RFC 6762) for the decentralized directory approach.

If a centralized BACnet directory of a BACnet Directory Server (BDS) and its DNS server are in place that serves the BACnet service information, then the DNS-SD port uses this DNS server for resolution and discovery.

If no BDS is in place, then mDNS is used for resolution and discovery. The mDNS protocol allows to multicast DNS requests to all BACnet/IT devices. DNS responses are returned by any of the devices that have matching DNS records representing the local devices and objects.

6. BACnet Directory

Today, directory information is collected by a BACnet device's application layer through broadcast Who-Is and Who-Has requests. The use of these services is a substantial part of the broadcast traffic generated by BACnet devices. In IT environments, this may not be acceptable. BACnet/IT therefore includes the concept of a centralized directory, which can be contacted using unicast services. See new Clause ZZ.3 in the BACnet/IT addendum.

6.1 Local Directory Function

The BACnet local directory functionality for the local devices, device groups, and objects is a required part for BACnet/IT, and expected to be implemented in the application layer. This functionality includes:

- A Resolver for the resolution of an EID to a URL
- A Registrar for registering local entities in the central directory
- A Local Registry for the local entities, queried by remote devices.

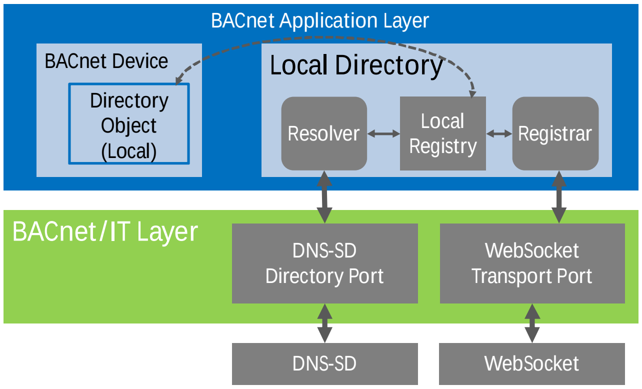


Figure 9. Local Directory

The Resolver is used to resolve, discover, and enumerate BACnet devices, device groups, and objects through use of information in the local registry, and by using all directory ports available for remote queries.

The Local Registry includes information on all BACnet devices, device groups, and objects local to the IP host. This information is represented by the local scope Directory object in all devices present on the IP host. A directory port uses this information for supporting directory queries addressed to the IP host, such as with mDNS.

The Registrar is responsible for registering and maintaining the local entities in a central directory if available. For this, the registrar uses BACnet services to populate the Directory object of the central directory, and therefore a transport port is used for this.

6.2 Central Directory Function

The BACnet central directory function is an optional functionality of the application layer and provides directory information for all BACnet devices, device groups and objects at a central place, allowing unicast directory queries.

A device supporting a central directory is referred to as a BACnet Directory Server (BDS). The directory structure is similar to the local directory, except that the registry and the Directory object are centralized and have a system scope and therefore contain all directory information of the system. A registrar is not needed in this case, since the local entities are included in the registry and the Directory object. The Resolver is still needed to provide this function to the local entities, but is not shown in the following figure.

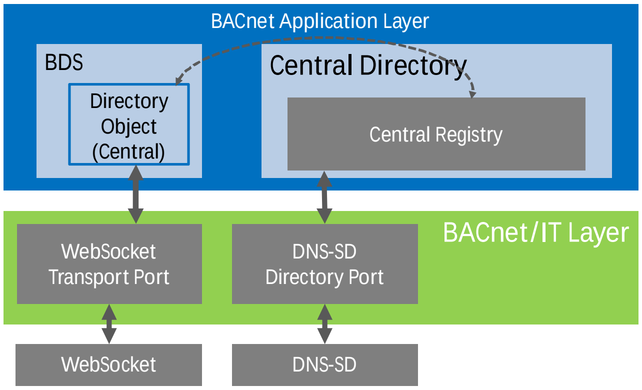


Figure 10. Central Directory in a BDS

The Central Registry includes information on all BACnet devices, device groups, and objects of the system. This information is represented by the central scope Directory object. The Directory object is required to accept registration and de-registration requests received through a transport port. AddListElement and RemoveListElement requests are executed for this. A directory port uses this information for supporting directory queries addressed to the IP host or respective external server, such as a DNS server.

For DNS-SD, it is expected that the BDS is maintaining a DNS server to contain the BACnet directory information. This DNS server is not required to be on the same IP host as the BDS. The BDS is only required to be able to update that DNS server with BACnet directory information. The protocol used for this is out of scope of BACnet/IT.

Since a central function always begs the question on how to deal with a single point of failure. For the BDS, the concept of a primary BDS and one or multiple secondary BDS is introduced. In addition, it is expected that devices cache at least resolution information for some time so that short outages of a BDS can be bridged without immediately stopping communication if the BDS is not readily available.

7. Device Group Coordination

Device Group Coordination is an optional functionality of the application layer. See new Clause ZZ.4 of the BACnet/IT addendum. BACnet Device Groups in BACnet/IT require some active entity that is receiving and distributing unconfirmed request messages to multiple devices, for the device group.

The device group coordination function for a device group acts as the receiver for request to the group, and uses the available transport ports for sending the request to the device group members.

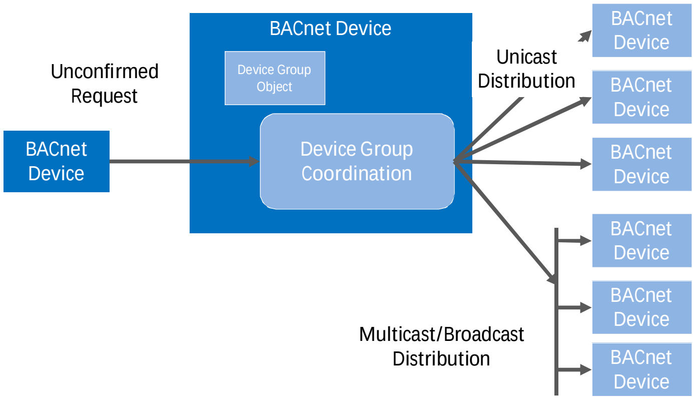


Figure 11. BACnet Device Group Coordination

Depending on the transport port used for distribution to group members, the distribution of the request may be by a series of unicast messages, but also a multicast or broadcast to the members, or any mix of these.

The configuration and other settings for the device group coordination function for a device group is represented by the Device Group object. This object also manages the configured and registered members of the device group.

Device group membership can be established through configuration of the device group object, being registered dynamically in the Device Group object, or simply by listening to a multicast or broadcast network address.

Table 9. Device Group Member Types

Device Group Member Types	
Configured	The member of the device group is configured to be a member of the device group at the Device Group object. The member device is not required to know about this.
Registered	The member of the device group itself is configured to be a member in the device group. The member device is required to register itself at the respective Device Group object.
Listening	The member is configured to listen at a multicast or broadcast address. The Device Group object must have a member entry that resolves to the multicast or broadcast address so as requests are distributed to the multicast or broadcast address. The group member is required to use its respective network interface to listen at the multicast or broadcast address.

8. Device and Device Group Proxying

BACnet/IT defines a device and device group proxying feature. See Clause ZZ.5 in the BACnet/IT addendum. This functionality may optionally be implemented in the application layer. In general, it allows communicating with devices over a protocol that the original peer device, or original device group coordination for a device group, does not support.

The device proxy functionality can be used for structuring the network connections needed for a particular protocol. For example, a single device proxy function on some host can provide access to an entire site, using a single WebSocket connection only. All devices of the site would be reachable through that single connection.

Once additional transport bindings and respective transport ports are available, the device proxy function will be the bridge between BACnet/IT devices that do not support the same transport binding. The device proxy represents the device or device group as if reachable through the protocol and at URL as advertised by the device proxy for that side.

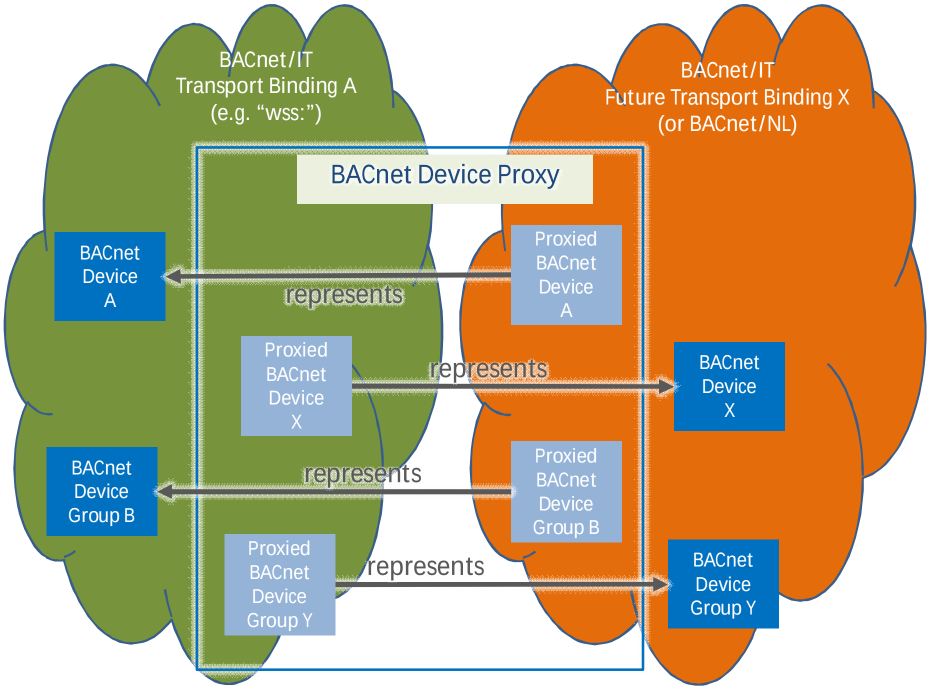


Figure 12. BACnet Device Proxy

The device proxy functionality is also defined for bridging between BACnet/IT devices and devices groups and BACnet/NL devices and networks.

9. Device Proxying for BACnet/NL Devices

Connecting BACnet/NL devices with BACnet/IT devices and for broadcast access to BACnet/NL networks, the BACnet/NL Device Proxy function is defined. See new Clause ZZ.5.2 in the BACnet/IT addendum.

9.1 Device and Device Group Representation

BACnet/IT devices are represented as BACnet/NL devices on a virtual network and the device instance is used as VMAC. In turn, BACnet/NL devices are represented as BACnet/IT devices to be addressed by the device ID, and accessible through the BACnet/IT layer and a transport port.

BACnet/IT device groups as such are not yet foreseen to be accessible from the BACnet/NL side. BACnet/NL networks are represented as BACnet/IT device groups in BACnet/IT. Through this, BACnet/IT devices proxied as BACnet/NL devices on a virtual network belong to the device group formed by that virtual network, and can be reached by BACnet broadcasts from the BACnet/NL side.

Automatic BACnet/NL device proxies are expected to represent as proxies:

- All BACnet/NL devices it is connected to as BACnet/IT devices
- All BACnet/NL networks it is connected to as BACnet/IT device groups
- All BACnet/IT devices to be BACnet/NL devices on one virtual BACnet/NL network
- This virtual BACnet/NL network as a BACnet/IT device group as well.

For configured proxies, the Device Group object representing the virtual network on which BACnet/IT devices appear to reside accepts member configuration and registration. BACnet/IT devices that are member of this device group are then proxied to be BACnet/NL devices on the virtual BACnet/NL network the Device Group object represents.

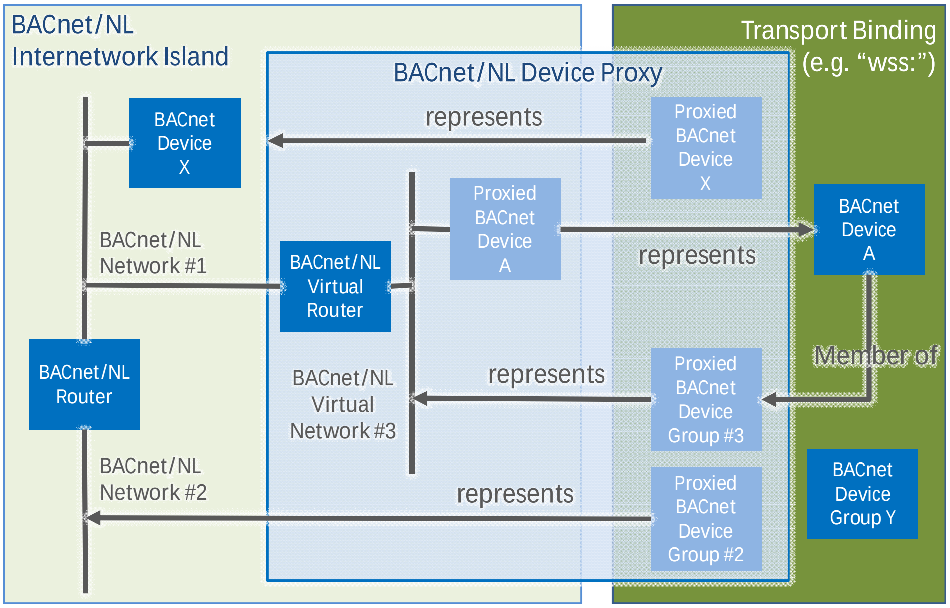


Figure 13. BACnet/NL Device Proxy

In BACnet/IT, the range of device group IDs 1 to 65534 is reserved for representing the BACnet/NL networks with the respective network number.

9.2 Message Exchange

On the BACnet/IT side of the BACnet/NL device proxy, BACnet TPDUs are exchanged as defined for BACnet/IT. On the BACnet/NL side, NPDUs, eventually conveying APDU segments are exchanged.

For unconfirmed or unsegmented messages, the BACnet/NL device proxy simply reframes the messages and forwards them to the other side.

To forward large requests or response messages received from BACnet/IT in a single TPDU to BACnet/NL, the device proxy must apply segmentation as of the BACnet application layer segmentation. When receiving a segmented request or response message from the BACnet/NL side for a BACnet/IT device, all segments must be received first as defined by the application layer, before the entire message is transferred in a single TPDU on BACnet/IT.

Since only the transfer of segments for a single message is performed, the BACnet/NL device proxy is not required to maintain full transaction state machines. The overall transaction state machines for response timeouts and retries are maintained by the ultimate peer devices, one BACnet/NL device and a BACnet/IT device in this case.

The BACnet/IT device, when communicating with a BACnet/NL device, must account for the time required for transferring segments on the BACnet/NL side in its transaction state machines. Response timeouts must be extended to allow the segments being transferred including segment timeouts and segment retries. This is supported by the so called entity class that is part of the directory information. The entity class indicates if the ultimate peer device is a BACnet/IT or BACnet/NL device.

10. BACnet Virtual Router Link (BVRL)

The BACnet Virtual Router Link (BVRL), introduced in sections 2 and 3 of the BACnet/IT addendum, is a new option for the BACnet/NL stack and is a virtual half-router link established using the connection support of the BACnet/IT transport ports. The BVRL is used to connect BACnet half-routers to form a logically complete BACnet/NL router.

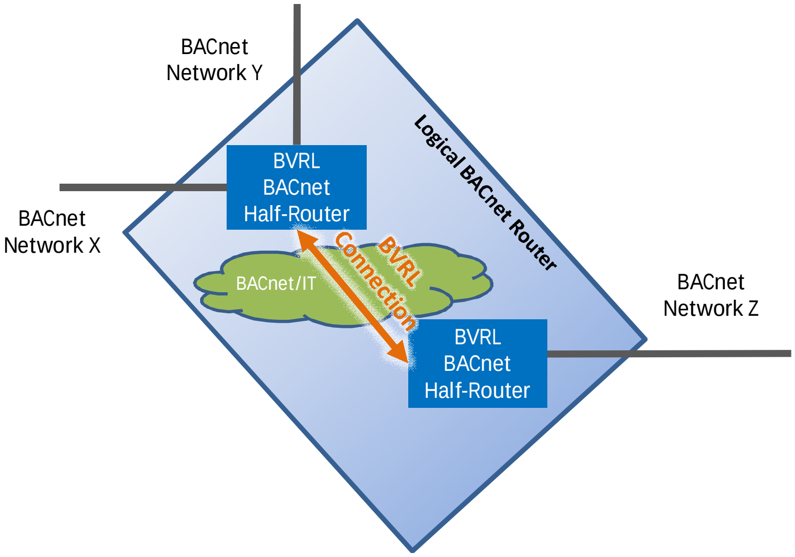


Figure 14. BACnet Virtual Router Link (BVRL) Half-Routers

For the connection between two BVRL half-routers, connections provided by the transport ports of the BACnet/IT layer are used. Therefore, the BACnet/IT and IP network infrastructure is used as a link layer only.

The expanded BACnet/NL stack including the BVRL on the BACnet/IT Layer:

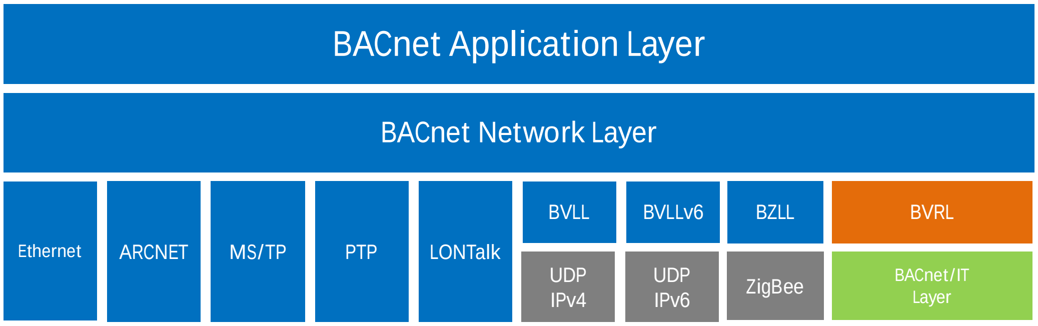


Figure 15. BACnet Virtual Router Link (BVRL) in BACnet/NL Stack

It is important to note that the BVRL does not support BACnet devices being connected to it directly. There is no concept of a BACnet MAC address on this link. The link is used for connecting BACnet half-routers exclusively.

10.1 Connection Management

Through the Network Port object, the BACnet/NL stack that implements the BVRL can be configured to initiate and/or accept a connection. BVRL connections are designed as permanent connections. Connection control by the BACnet network layer (I-Could-Be-Router-To, Establish-Connection-To-Network, Disconnect-Connection-To-Network) is not supported.

The peer BVRL half-router is identified by either the device ID of the peer half-router, or directly by a URL. If a device ID is configured, then this ID needs to be resolved to a URL by the resolver of the local directory. See section 6 BACnet Directory.

Once the URL is available to the initiating side of the connection, the respective transport port of the BACnet/IT layer is used to establish and maintain the connection to the peer. After the connection is established, exchange of routing information occurs between the BVRL ports to learn the connected and reachable BACnet networks. These procedures are equal to those in place for the PTP based half-router.

10.2 NPDU Transmission

The BACnet network layer uses a BVRL port to send and receive entire and complete NPDUs to the peer half-router. These NPDUs include both the DNET/DLEN/DADR/HC and SNET/SLEN/SADR components in the network layer header.

Sending an NPDU across BACnet/IT, the BVRL creates a TPDU to convey the NPDU and sends it across the connection to the peer BVRL half-router.

On receiving an NPDU from the transport port, the BVRL unpacks the NPDU and hands it up to the BACnet network layer for processing.

11. Standardization Prospects

The BACnet/IT addendum is first going through an Advisory Public Review (APR), which is a rather informal public review for presenting the addendum to the public and gather comments on the approach. After that, the addendum needs to undergo at least one Publication Public Review (PPR). Once the addendum passes a PPR without requiring substantive changes, it will be ready to be published as an official standard addendum.

12. BACnet/IT PlugFests

The ASHRAE SSPC 135 BACnet committee together with BACnet International is working on a first online PlugFest on the base of the APR draft addendum. For this, functionality will be broken down and matched for reasonable interoperability testing sessions.
The committee sees this first PlugFest to take place in late spring 2017, for generating timely feedback for the addendum to proceed for a next review. More details will be broadly announced to the BACnet community early after the APR has started.
It is foreseen that these PlugFests will be repeated while the addendum is going through review cycles.

13. References

BACnet Standard	ANSI/ASHRAE Standard 135-2016
BACnet/IT Addendum	Draft addendum 135-2016bj to ANSI/ASHRAE Standard 135-2016
IETF RFC 6455 (2011)	The WebSocket Protocol, Internet Engineering Task Force
IETF RFC 6762 (2013)	Multicast DNS, Internet Engineering Task Force
IETF RFC 6763 (2013)	DNS-Based Service Discovery, Internet Engineering Task Force
IETF RFC 7251 (2014)	AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS, Internet Engineering Task Force

14. Abbreviations

Only some particular abbreviations are listed here. For other abbreviations used, please see the BACnet standard's list of abbreviations, or the Internet in general.

APDU	Application Protocol Data Unit
ASDU	Application Service Data Unit
BDS	BACnet directory server
BVRL	BACnet Virtual Router Link
DNS	Domain Name System
DNS-SD	DNS use for service discovery, RFC 6763
D-SAP	Directory Service Access Point, as defined in BACnet/IT Addendum
mDNS	Multicast DNS, RFC 6762
NPDU	Network Protocol Data Unit
T-SAP	Transport Service Access Point, as defined in BACnet/IT Addendum.
TPDU	Transport Protocol Data Unit, as defined in BACnet/IT Addendum

15. Important ASN.1 Definitions for BACnet/IT

The following listing shows important new ASN.1 definitions for BACnet/IT.

```
BACnetEID ::= CHOICE {
        not-assigned                [0] NULL,
        device-identifier           [1] Unsigned(0..4194302),
        device-group-identifier     [2] Unsigned(0..4194302)
        -- ASHRAE may add new options to this CHOICE anytime.
        }


BACnetTransportPDU ::= SEQUENCE {
        header              [0] SEQUENCE {
                version                         [0] Unsigned8,
                priority                        [1] BACnetNetworkPriority OPTIONAL,
                sequence-number                 [2] Unsigned (1..65535) OPTIONAL,
                source-eid                      [3] BACnetEID,
                destination-eid                 [4] BACnetEID,
                original-destination-eid        [5] BACnetEID OPTIONAL,
                invoke-id                       [6] Unsigned8 OPTIONAL,
                forwards                        [7] Unsigned8 OPTIONAL,
                security                        [8] BACnetSecurityParameters OPTIONAL
                },
        body                [1] CHOICE {
                transport-error                 [0] Transport-Error,
                npdu                            [1] OCTET STRING,
                confirmed-request               [2] BACnet-Confirmed-Service-Request,
                unconfirmed-request             [3] BACnet-Unconfirmed-Service-Request,
                complex-ack                     [4] BACnet-Confirmed-Service-ACK,
                simple-ack                      [5] BACnetConfirmedServiceChoice,
                error                           [6] BACnet-Error,
                abort                           [7] BACnetAbort,
                reject                          [8] BACnetRejectReason
                }
        }
        -- ASHRAE may add new options to the 'Header' and 'Body' anytime.
```