BSR/ASHRAE Addendum ai to
ANSI/ASHRAE Standard 135-2010

Public Review Draft

Proposed Addendum ai to Standard
135-2010, BACnet® - A Data
Communication Protocol for Building
Automation and Control Networks

Second Public Review (March 2012)
(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.

© 2012 ASHRAE. This draft is covered under ASHRAE copyright. Permission to reproduce or redistribute all or any part of this document must be obtained from the ASHRAE Manager of Standards, 1791 Tullie Circle, NE, Atlanta, GA 30329. Phone: 404-636-8400, Ext. 1125. Fax: 404-321-5478. E-mail: standards.section@ashrae.org.
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.

135-2010ai-1, Add Network Port Object Type, p. 2

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2010 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 addendum is provided for context only and is not open for public review comment except as it relates to the proposed changes.
135-2010ai-1 Add Network Port Object Type

Rationale

There is currently no BACnet-visible mechanism for viewing and/or configuring a device’s network settings. While this exists to some degree via some properties in the Device object or via BVLL messages, there needs to be a way for BACnet client devices to easily and consistently access and manipulate this information.

[Add new entries to Clause 3.3, p. 6, in alphabetical order]

...  
**DHCP**  Dynamic Host Configuration Protocol  
...
**DNS**  Domain Name Service  
...
**NAT**  Network Address Translation or Port Address Translation RFC 2663

[Add new **Clause 12.X**, p. 410]

12.X Network Port Object

The Network Port object provides access to the configuration and properties of network ports of a device. All BACnet devices shall contain one Network Port object per port.

Verification and validation of property values within a Network Port object is a local matter.

Property values which are required to maintain proper operation of the network shall be retained across a device reset.

As specified in Table 12-X and the text below, some properties of the Network Port Object are required if the object is used to represent a network of a given type. For example, a Network Port Object whose Network_Type is MSTP must include the Max_Master property, and a Network Port Object whose Network_Type is BACnet_IPV4 must include the BACnet_IP_Subnet_Mask property. Aside from the properties so required, it is a local matter whether a Network Port Object contains properties that do not apply to its Network_Type. For example, a Network Port Object whose Network_Type is MSTP may include the BACnet_IP_Subnet_Mask property, although the value of this property would not be used by the network. Some vendors may find it convenient to have all of their Network Port Objects support the same list of properties regardless of Network_Type. This is permitted, but not required.
Table 12-X. Properties of the Network Port Object Type

<table>
<thead>
<tr>
<th>Property Identifier</th>
<th>Property Datatype</th>
<th>Conformance Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object_Identifier</td>
<td>BACnetObjectIdentifier</td>
<td>R</td>
</tr>
<tr>
<td>Object_Name</td>
<td>CharacterString</td>
<td>R</td>
</tr>
<tr>
<td>Object_Type</td>
<td>BACnetObjectType</td>
<td>R</td>
</tr>
<tr>
<td>Description</td>
<td>CharacterString</td>
<td>O</td>
</tr>
<tr>
<td>Status_Flags</td>
<td>BACnetStatusFlags</td>
<td>R</td>
</tr>
<tr>
<td>Reliability</td>
<td>BACnetReliability</td>
<td>R</td>
</tr>
<tr>
<td>Out_Of_Service</td>
<td>BOOLEAN</td>
<td>R</td>
</tr>
<tr>
<td>Network_Type</td>
<td>BACnetNetworkType</td>
<td>R</td>
</tr>
<tr>
<td>Network_Number</td>
<td>Unsigned16</td>
<td>R</td>
</tr>
<tr>
<td>Network_Number_Quality</td>
<td>BACnetNetworkNumberQuality</td>
<td>R</td>
</tr>
<tr>
<td>Changes_Pending</td>
<td>BOOLEAN</td>
<td>R</td>
</tr>
<tr>
<td>Command</td>
<td>BACnetNetworkPortCommand</td>
<td>W</td>
</tr>
<tr>
<td>MAC_Address</td>
<td>OCTET STRING</td>
<td>O</td>
</tr>
<tr>
<td>APDU_Length</td>
<td>Unsigned</td>
<td>R</td>
</tr>
<tr>
<td>Link_Speed</td>
<td>REAL</td>
<td>R</td>
</tr>
<tr>
<td>Link_Speeds</td>
<td>BACnetARRAY[N] of REAL</td>
<td>O</td>
</tr>
<tr>
<td>Link_Speed_Autonegotiate</td>
<td>BOOLEAN</td>
<td>O</td>
</tr>
<tr>
<td>BACnet_IP_Mode</td>
<td>BACnetIPMode</td>
<td>O</td>
</tr>
<tr>
<td>BACnet_IP_Subnet_Mask</td>
<td>OCTET STRING</td>
<td>O</td>
</tr>
<tr>
<td>BACnet_IP_Default_Gateway</td>
<td>OCTET STRING</td>
<td>O</td>
</tr>
<tr>
<td>BACnet_IP_Multicast_Address</td>
<td>OCTET STRING</td>
<td>O</td>
</tr>
<tr>
<td>BACnet_IP_DHCP_Enable</td>
<td>BOOLEAN</td>
<td>O</td>
</tr>
<tr>
<td>BACnet_IP_DHCP_Lease_Time</td>
<td>Unsigned</td>
<td>O</td>
</tr>
<tr>
<td>BACnet_IP_DHCP_Lease_Time_Remaining</td>
<td>Unsigned</td>
<td>O</td>
</tr>
<tr>
<td>BACnet_IP_DHCP_Server</td>
<td>OCTET STRING</td>
<td>O</td>
</tr>
<tr>
<td>BBMD_NAT_Traversal</td>
<td>BOOLEAN</td>
<td>O</td>
</tr>
<tr>
<td>BBMD_Global_IP_Address</td>
<td>OCTET STRING</td>
<td>O</td>
</tr>
<tr>
<td>BBMD_Broadcast_Distribution_Table</td>
<td>List of BACnetBDTEntry</td>
<td>O</td>
</tr>
<tr>
<td>BBMD_Accept_FD_Registrations</td>
<td>BOOLEAN</td>
<td>O</td>
</tr>
<tr>
<td>BBMD_Foreign_Device_Table</td>
<td>List of BACnetFDTEntry</td>
<td>O</td>
</tr>
<tr>
<td>FD_BBMD_IP_Address</td>
<td>BACnetHostNPort</td>
<td>O</td>
</tr>
<tr>
<td>FD_Subscription_Lifetime</td>
<td>Unsigned16</td>
<td>O</td>
</tr>
<tr>
<td>Max_Master</td>
<td>Unsigned</td>
<td>O</td>
</tr>
<tr>
<td>Max_Info_Frames</td>
<td>Unsigned</td>
<td>O</td>
</tr>
<tr>
<td>Slave_Proxy_Enable</td>
<td>BOOLEAN</td>
<td>O</td>
</tr>
<tr>
<td>Manual_Slave_Address_Binding</td>
<td>List of BACnetAddressBinding</td>
<td>O</td>
</tr>
<tr>
<td>Auto_Slave_Discovery</td>
<td>BOOLEAN</td>
<td>O</td>
</tr>
<tr>
<td>Slave_Address_Binding</td>
<td>List of BACnetAddressBinding</td>
<td>O</td>
</tr>
<tr>
<td>Virtual_MAC_Address_Table</td>
<td>List of BACnetVMACEntry</td>
<td>O</td>
</tr>
<tr>
<td>Routing_Table</td>
<td>List of BACnetRouterEntry</td>
<td>O</td>
</tr>
<tr>
<td>Profile_Name</td>
<td>CharacterString</td>
<td>O</td>
</tr>
</tbody>
</table>

1. Required to be writable in routers, secure devices, and any other device that requires knowledge of the network number for proper operation.
2. Required if the port is not a PTP link.
3. This property shall be present if Link_Speed is writable.
4. Required to be writable if the port is a BACnet/IP port.
5. Required if the port is a BACnet/IP port.
6. Required to be present if the port is a BACnet/IP port and supports multicast.
7. Required if the port can be configured by DHCP.
8. Required if the port is a BACnet/IP port and the device is capable of functioning as a BBMD in a NAT environment.
9. Required if the port is a BACnet/IP port and the device is capable of functioning as a BBMD in a NAT environment.
10. Required to be writable if the port is a BACnet/IP port and the device is capable of functioning as a BBMD.
11. Required if the port is a BACnet/IP port and the device is capable of functioning as a BBMD.
12 Required to be writable if the port is a BACnet/IP port and BACnet_IP_Mode is set to FOREIGN.
13 Required to be present and writable if the port is an MS/TP port.
14 Required if the port is an MS/TP port.
15 This property shall be present and writable if the port is an MS/TP port, and the device is capable of being a Slave-Proxy device.
16 This property shall be present if the port is an MS/TP port, and the device is capable of being a Slave-Proxy device that implements automatic discovery of slaves.
17 This property shall be present if the port is an MS/TP port, and the device is capable of being a Slave-Proxy device.
18 This property shall be present if the network represented by this object requires VMAC addressing.

12.X.1 Object_Identifier
This property, of type BACnetObjectIdentifier, is a numeric code that is used to identify the object. It shall be unique within the BACnet Device that maintains it. The instance number (see Clause 20.2.14) shall correspond to the Port ID of the associated network as described in Clause 6.

12.X.2 Object_Name
This property, of type CharacterString, shall represent a name for the object that is unique within the BACnet Device that maintains it. The minimum length of the string shall be one character. The set of characters used in the Object_Name shall be restricted to printable characters.

12.X.3 Object_Type
This property, of type BACnetObjectType, indicates membership in a particular object type class. The value of this property shall be NETWORK_PORT.

12.X.4 Description
This property, of type CharacterString, is a string of printable characters whose content is not restricted.

12.X.5 Status_Flags
This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of the Network. The four flags are:

\{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE\}

where:

IN_ALARM Always logical FALSE (0).

FAULT Logical TRUE (1) if the Reliability property does not have a value of NO_FAULT_DETECTED, otherwise logical FALSE (0).

OVERRIDDEN Always logical FALSE (0).

OUT_OF_SERVICE Logical TRUE (1) if the Out_Of_Service property has a value of TRUE, otherwise logical FALSE (0).

12.X.6 Reliability
This property, of type BACnetReliability, provides an indication of whether the network in question is "reliable" as far as the BACnet Device or operator can determine and, if not, why.

12.X.7 Out_Of_Service
The Out_Of_Service property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the network port is out of service.
When a network port is Out_Of_Service, all communication through that port shall be disabled, and writing to the Command property value shall result in an error response with an ‘Error Class’ of PROPERTY and ‘Error Code’ of VALUE_OUT_OF_RANGE.

**12.X.8 Network_Type**

This read only property, of type BACnetNetworkType, represents the type of network this Network Port object is representing.

This property may have any of the following values:

- **ETHERNET** ISO 8802-3 (“Ethernet”), as defined in Clause 7
- **ARCNET** ARCNET, as defined in Clause 8
- **MSTP** MS/TP, as defined in Clause 9
- **PTP** Point-To-Point, as defined in Clause 10
- **LONTALK** LonTalk, as defined in Clause 11
- **BACNET_IPV4** BACnet/IP, as defined in Annex J
- **ZIGBEE** ZigBee, as defined in Annex O
- **VIRTUAL** Indicates that this port represents the configuration and properties of a virtual network as described in Annex H.2.
- **<Proprietary Enum Values>** A vendor may use other proprietary enumeration values to indicate that this port represents the use of message structures, procedures, and medium access control techniques other than those contained in this standard. For proprietary extensions of this enumeration, see Clause 23.1 of this standard.

**12.X.9 Network_Number**

This property, of type Unsigned16, represents the BACnet network number associated with this network.

The range for this property shall be 0..65534. A Network_Number of 0 indicates that the Network_Number is not known or cannot be determined. Writing 0 to Network_Number shall force the value of the Network_Number_Quality property to UNKNOWN and allows the device to attempt to learn the network number, if possible. Writing a value other than 0 shall force Network_Number_Quality to CONFIGURED.

If the Network_Type is PTP, then this property shall be read only and contain a value of 0.

This property shall be writable in routers, secure devices, and any other device that requires knowledge of the network number for proper operation. Routers are permitted to refuse a value of 0. In that case, the write request shall result in an error response with ‘Error Class’ of PROPERTY and an ‘Error Code’ of VALUE_OUT_OF_RANGE.

If this property is writable, then writing to this property shall set the Changes_Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

**12.X.10 Network_Number_Quality**

This read only property, of type BACnetNetworkNumberQuality, represents the current quality of the Network_Number property. If the Network_Type is PTP, the Network_Number_Quality shall be CONFIGURED.

This property shall have one of the following values:
UNKNOWN  None of the below meanings.

LEARNED  The Network_Number was learned via receipt of a Network-Number-Is message with a flag value of 0 (learned).

LEARNED_CONFIGURED  The Network_Number was learned via receipt of a Network-Number-Is message with a flag value of 1 (configured).

CONFIGURED  The Network_Number is configured for this port.

12.X.11 Changes_Pending
This property, of type BOOLEAN, indicates whether the configuration settings in the Network Port object map to the current configuration settings. A value of FALSE indicates that the configuration settings reflect the current port configuration information. A value of TRUE indicates the configuration settings have been modified but have not been activated on the port.

12.X.12 Command
This property, of type BACnetNetworkPortCommand, is used to command this Network Port object to enter various states.

Any of the following commands may be written to this property:

**ACTIVATE**
The device shall activate the currently visible configuration settings of this object and set the Changes_Pending property to FALSE. Once all pending changes have been made, the value of this property shall revert to IDLE. Writing a value of ACTIVATE to this property does not imply that configured values can be rolled back to their previous settings. If the device cannot perform the activation without a re-initialization, then it shall reinitialize upon execution of the ACTIVATE command. Any failure to activate the settings shall result in the value of the Reliability property being set to ACTIVATE_FAILURE and the value of this property being set to IDLE.

**ROLLBACK**
If rolling back of changes is supported, this object shall attempt to revert to the set of property values contained in this object until Changes_Pending was first set to TRUE. If rolling back of changes is not supported, writing this value shall result in an error response with ‘Error Class’ of PROPERTY and an ‘Error Code’ of OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED. Any failure to roll back the values shall result in the value of the Reliability property being set to ROLLBACK_FAILURE and the value of this property being set to IDLE.

**RENEW_FD_REGISTRATION**
This port shall attempt to renew its foreign device registration with the BBMD indicated in FD_BBMD_IP_Address. Upon successful registration, the value of this property shall revert to IDLE. If the value of Network_Type is not BACNET_IPV4, writing this value shall result in an error response with ‘Error Class’ of PROPERTY and an ‘Error Code’ of VALUE_OUT_OF_RANGE. If the attempt to renew the foreign device registration fails, then the value of the Reliability property shall be RENEW_FD_REGISTRATION_FAILURE, and the value of this property shall be IDLE.

**RESTART_SLAVE_DISCOVERY**
The device shall restart the slave detection algorithm as described in 12.X.39. If the value of Network_Type is not MSTP, writing this value
shall result in an error response with ‘Error Class’ of PROPERTY and an ‘Error Code’ of VALUE_OUT_OF_RANGE. If the value of Network_Type is MSTP but the device does not support MS/TP Slave Proxy functionality, writing this value shall result in an error response with an ‘Error Class’ of PROPERTY and an ‘Error Code’ of OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED. If the attempt to restart the slave detection algorithm fails, then the value of the Reliability property shall be RESTART_SLAVE_DISCOVERY_FAILURE, and the value of this property shall be IDLE.

RENEW_DHCP
If DHCP is supported, then this device shall attempt to renew the DHCP lease. If the value of Network_Type is BACNET_IPV4 but this port does not support DHCP or is incapable of renewing the DHCP lease, writing this value shall result in an error response with an ‘Error Class’ of PROPERTY and an ‘Error Code’ of OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED. If the value of Network_Type is not BACNET_IPV4, writing this value shall result in an error response with an ‘Error Class’ of PROPERTY and an ‘Error Code’ of VALUE_OUT_OF_RANGE. If the attempt to renew the DHCP address fails, then the value of the Reliability property shall be RENEW_DHCP_FAILURE, and the value of this property shall be IDLE.

RESTART_AUTONEgotIATION
This port shall restart the link speed auto-negotiation algorithm. If this port does not support auto-negotiation, writing this value shall result in an error response with an ‘Error Class’ of PROPERTY and an ‘Error Code’ of OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED. If the auto-negotiation algorithm fails, the value of the Reliability property shall be RESTART_AUTONEGOTIATION_FAILURE, and the value of this property shall be IDLE.

DISCONNECT
This port shall terminate the network connection. If the value of Network_Type is not PTP, writing this value shall result in an error response with an ‘Error Class’ of PROPERTY and an ‘Error Code’ of VALUE_OUT_OF_RANGE.

RESTART_PORT
This port shall restart and reconnect to the network as if the device were reinitialized. All data that was learned, cached, or otherwise automatically determined for the port’s operation shall be cleared. All initialization, negotiation, and registration functions the port is expected to be performed upon device initialization are performed again. If the restart fails, the value of the Reliability property shall be RESTART_FAILURE, and the value of this property shall be IDLE.

<Proprietary Enum Values>
A vendor may use other proprietary enumeration values to allow command values other than those defined by the standard. For proprietary extensions of this enumeration, see Clause 23.1 of this standard. A proprietary command failure shall result in the value of the Reliability property being set to PROPRIETARY_COMMAND_FAILURE and the value of this property being set to IDLE.

If the value of the Changes_Pending property is TRUE, writing a value other than ACTIVATE or ROLLBACK shall result in an error response with an ‘Error Class’ of PROPERTY and an ‘Error Code’ of INVALID_VALUE_IN_THIS_STATE.

When this property has a value of IDLE, then any pending commands have been applied. Writing a value of IDLE shall result in an error response with an ‘Error Class’ of PROPERTY and an ‘Error Code’ of INVALID_VALUE_IN_THIS_STATE.
When this property has a value other than IDLE, writing to this property shall result in an error response with an 'Error Class' of PROPERTY and an 'Error Code' of INVALID_VALUE_IN_THIS_STATE.

When changing property values that require activation, the value of the Changes_Pending property will automatically be set to TRUE, indicating that the current property values are not the values actively in use. It is necessary for the client to write a value of ACTIVATE to the Command property in order to activate the currently visible configuration settings. This interaction is necessary to support atomic updating of multiple properties when modifying a network port configuration.

It is a local matter as to whether or not resetting the device discards pending change, activates pending change, or leaves the changes pending.

It is a local matter whether property values will be validated upon writing a value of ACTIVATE to this property. If a device performs validation of property values and the validation fails, then the Reliability property shall indicate CONFIGURATION_ERROR, and this write request shall result in an error response with an ‘Error Class’ of PROPERTY and an ‘Error Code’ of INVALID_CONFIGURATION_DATA. Failed validation shall have no effect on the Changes_Pending property.

This enumerated value is extensible. Writing an unknown value to this property shall result in an error response with an ‘Error Class’ of PROPERTY and an ‘Error Code’ of VALUE_OUT_OF_RANGE.

12.X.13 MAC_Address

This property, of type OCTET STRING, contains the BACnet MAC address used on this network. The value of this property shall be conveyed with the most significant octet first. For example, if Network_Type is BACNET_IP, then the value of this property contains the six octet combination of the BACnet/IP address and UDP port.

If this property is writable, then writing to this property shall set the Changes_Pending property to TRUE. The value of this property becomes effective when a value of ACTIVATE is written to the Command property.

12.X.14 APDU_Length

This read only property, of type Unsigned, is the maximum number of octets that may be contained in a single indivisible application protocol data unit sent or received on this port. The value of this property shall be greater than or equal to 50. This property also indicates the maximum number of octets that may be contained in a single individual network service data unit sent or received on this port.

12.X.15 Link_Speed

This property, of type REAL, represents the network communication rate as the number of bits per second. A value of 0 indicates an unknown communication rate.

If the value of the Link_Speed_Autonegotiate property is TRUE, then this property shall be read only.

If this property is writable, then writing to this property shall set the Changes_Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

12.X.16 Link_Speeds

This read only property, of type BACnetARRAY of REAL, is an array of link speeds supported by this network port when the Link_Speed property is writable.

12.X.17 Link_Speed_Autonegotiate

This property, of type BOOLEAN, represents the auto negotiation setting of the port.

A value of TRUE indicates that the device automatically determines the speed of this network port. In this case, Link_Speed shall be read only and indicate the determined speed, if available. A value of FALSE indicates that the link speed is indicated by the value of the Link_Speed property.
If this property is writable, writing to this property shall set the Changes Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

12.X.18 BACnet_IP_Mode

This property, of type BACnetIPMode, indicates the IP mode of the device.

This property may have any of the following values:

- NORMAL: A normal BACnet/IP device. If the device is capable of performing BBMD functionality, then BBMD functionality shall be disabled.
- FOREIGN: The device is operating as a foreign device.
- BBMD: The device is capable of being a BBMD, and BBMD functionality is enabled.

Writing to this property shall set the Changes Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

12.X.19 BACnet_IP_Subnet_Mask

This property, of type OCTET STRING, indicates the current subnet mask for this network. This property shall be conveyed with the most significant octet first. Writing to this property causes the subnet mask of the corresponding IP network to be changed. If the BACnet_IP_DHCP_Enable property is TRUE, this property shall be read only.

If this property is writable, then writing to this property shall set the Changes Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

12.X.20 BACnet_IP_Default_Gateway

This property, of type OCTET STRING, indicates the IP address of the current default gateway for this network. This property shall be conveyed with the most significant octet first. Writing to this property causes the default gateway of the corresponding IP network to be changed. If the BACnet_IP_DHCP_Enable property is TRUE, this property shall be read only.

If this property is writable, then writing to this property shall set the Changes Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

12.X.21 BACnet_IP_Multicast_Address

This property, of type OCTET STRING, contains the BACnet/IP multicast group address to be used by BACnet/IP for this network. The value of this property shall be conveyed with the most significant octet first.

If present, this property shall be writable. Writing to this property shall set the Changes Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

12.X.22 BACnet_IP_DHCP_Enable

This property, of type BOOLEAN, indicates whether or not the network is configured via DHCP. A value of TRUE indicates that DHCP configuration is enabled, FALSE indicates it is not.

This property is required if DHCP configuration is supported by this network port.

If this property is writable, then writing to this property shall set the Changes Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

12.X.23 BACnet_IP_DHCP_Lease_Time

This read only property, of type Unsigned, indicates the lease time of the last DHCP lease obtained for the port. If BACnet_IP_DHCP_Enable is FALSE, or no lease has been acquired, this property shall be 0.

12.X.24 BACnet_IP_DHCP_Lease_Time_Remaining
This read only property, of type Unsigned, indicates the lease time remaining of the last DHCP lease obtained for the port. If BACnet_IP_DHCP_Enable is FALSE, or no lease has been acquired, this property shall be 0.

12.X.25 BACnet_IP_DHCP_Server

This read only property, of type OCTET STRING, indicates the address of the DHCP server from which the last DHCP lease was obtained for the port.

12.X.26 BBMD_NAT_Traversal

This property, of type BOOLEAN, indicates whether (TRUE) or not (FALSE) this port is configured to operate in a NAT environment, as described in Annex J.7.8 and the global IP address is indicated by the value of the BBMD_Global_IP_Address property.

If present, this property shall be writable. Writing to this property shall set the Changes_Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

12.X.27 BBMD_Global_IP_Address

This property, of type BACnetHostAddress, indicates the public address and UDP port on which the network port can be reached from the public address space. The value of this property shall be conveyed with the most significant octet first. How the public IP address and UDP port are determined is a local matter. When the BBMD_Global_IP_Address is not known, this property shall contain a NULL value.

If this device does not automatically determine the public address, then this property shall be writable.

If this property is writable, then writing to this property shall set the Changes_Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

12.X.28 BBMD_Broadcast_Distribution_Table

This property, of type List of BACnetBDTEntry, is required to be present and writable if BACnet_IP_Mode is BBMD.

The value of this property maps to the BDT (as specified in Annex J) for this port as follows:
(a) The current value of the BDT may be read at any time with the Read-Broadcast-Distribution-Table BVLL message.
(b) The current value of the BDT may be set at any time with the Write-Broadcast-Distribution-Table BVLL message.
(c) If this property has no pending changes, reading this property shall return the current value of the BDT.
(d) If this property has pending changes, reading this property shall return the last value written to the property, and not the current value of the BDT.

Writing to this property shall set the Changes_Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

12.X.29 BBMD_Accept_FD_Registrations

This property, of type BOOLEAN, indicates whether (TRUE) or not (FALSE) this device shall accept foreign device registrations. This property is required to be present and writable if BACnet_IP_Mode is BBMD.

Writing to this property shall set the Changes_Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

12.X.30 BBMD_Foreign_Device_Table

This read only property, of type List of BACnetFDTEntry, is required to be present if BBMD_Accept_FD_Registrations is TRUE.
The value of this property reflects the current value of the foreign device table.

**12.X.31 FD_BBMD_IP_Address**

This property, of type BACnetHostNPort, indicates the BBMD with which the local device is to register as a foreign device when BACnet_IP_Mode is FOREIGN. This property shall be present and writable if BACnet_IP_Mode is FOREIGN.

If this property is writable, then writing to this property shall set the Changes_Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

**12.X.32 FD_Subscription_Lifetime**

This property, of type Unsigned16, indicates the Time_To_Live value, in seconds, to be used in the Register-Foreign-Device BVLL message. This property shall be present and writable if BACnet_IP_Mode is FOREIGN.

If this property is writable, then writing to this property shall set the Changes_Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

**12.X.33 Max_Master**

This property, of type Unsigned, shall be present and writable if the device is a master node on the MS/TP network connected to this port. The valid range for the value of this property is 0 to 127. See Clause 9.5.3.

Writing to this property shall set the Changes_Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

**12.X.34 Max_Info_Frames**

This property, of type Unsigned, shall be present if the Network_Type is MSTP. The value of Max_Info_Frames specifies the maximum number of information frames the node may send on this port before it must pass the token and shall have a minimum value of 1 and a maximum value of 255. If Max_Info_Frames is not writable or otherwise user configurable, its value shall be 1. See Clause 9.5.3.

Writing to this property shall set the Changes_Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

**12.X.35 Slave_Proxy_Enable**

This property, of type BOOLEAN, is an indication of whether (TRUE) or not (FALSE) the device will perform Slave-Proxy functions for this port. This property shall be present and writable if the device is capable of performing the functions of a Slave-Proxy device on this port.

**12.X.36 Manual_Slave_Address_Binding**

This property, of type List of BACnetAddressBinding, describes the manually configured set of slave devices for which this device is acting as a Slave Proxy as described in 16.10.2. This property shall be present if the device is capable of performing the functions of a Slave-Proxy device. If present, and Network_Type is MSTP, then this property shall be writable.

This property is used to manually configure a set of slave devices connected to this port for which this device will be a proxy. This property allows a Slave-Proxy that does not support automatic slave discovery to be configured with a set of slaves for which this device will be a proxy. It also allows a Slave-Proxy device to be a proxy for Slave devices that do not support the special object instance of 4194303 as described in Clause 12. When enabled, the Slave-Proxy device shall periodically check each device that is in this list, and not in the Slave_Address_Binding list, by reading the device's Protocol_Services_Supported property from the device's Device object using the ReadProperty service. If the device responds and indicates that it does not execute the Who-Is service, it shall be added to the Slave_Address_Binding property. The period at which the devices are checked is a local matter.

**12.X.37 Auto_Slave_Discovery**

This property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the device will perform automatic slave detection functions for this port. This property shall be present if the device is capable of performing the functions of a Slave-Proxy device on this port.
Slave detection shall be accomplished by the proxy device using ReadProperty services to read, at a minimum, the Device object's Protocol_Services_Supported property for each MAC address on the network connected to this port. The ReadProperty service shall use the special object instance of 4194303 as described in Clause 12. If the device is found to support execution of the Who-Is service, it is ignored; otherwise, the device shall be added to the Slave_Address_Binding property. The slave detection algorithm shall be repeated periodically. The period at which it is repeated is a local matter.

12.X.38 Slave_Address_Binding

This property, of type List of BACnetAddressBinding, describes the set of slave devices for which this device is acting as a Slave-Proxy on this port as described in 16.10.2. This property shall be present if the device is capable of performing the functions of a Slave-Proxy device on this port.

The set of devices described by the Slave_Address_Binding property consists of those devices described in the Manual_Slave_Address_Binding and those devices that are automatically discovered. When enabled, the Slave-Proxy device shall periodically check each device in this list by reading the device's Protocol_Services_Supported property from the device's Device object using the ReadProperty service. If the device fails to respond or indicates that it executes the Who-Is service, it shall be removed from the list. The period at which the devices are checked is a local matter.

12.X.39 Virtual_MAC_Address_Table

This property, of type List of BACnetVMACEntry, is the list of VMAC entries as described in Annex H.7.

If this property is writable, writing to this property will set the Changes_Pending property to TRUE. A value written to this property becomes effective when a value of ACTIVATE is written to the Command property.

12.X.40 Routing_Table

This read only property, of type List of BACnetRouterEntry, contains the table of first hop routers to remote networks reachable through this port.

Router table entries shall contain the following information:

- **network-number**: The network number reachable through the router specified by mac-address.
- **mac-address**: The MAC address of the router on the network connected to this port that leads directly or indirectly to that network number.
- **status**: Conveys whether the associated network is able to receive traffic.
- **performance-index**: This optional field is used to convey the performance index as conveyed in an I-Am-Router-To-Network network layer message. See Clause 6.4.3.

12.X.41 Profile_Name

This property, of type CharacterString, is the name of an object profile to which this object conforms. To ensure uniqueness, a profile name must begin with a vendor identifier code (see Clause 23) in base-10 integer format, followed by a dash. All subsequent characters are administered by the organization registered with that vendor identifier code. The vendor identifier code that prefixes the profile name shall indicate the organization that publishes and maintains the profile document named by the remainder of the profile name. This vendor identifier need not have any relationship to the vendor identifier of the device within which the object resides.

A profile defines a set of additional properties, behavior, and/or requirements for this object beyond those specified here. This standard defines only the format of the names of profiles. The definition of the profiles themselves is outside the scope of this standard.
BSR/ASHRAE Addendum ai to ANSI/ASHRAE Standard 135-2010, BACnet® - A Data Communication Protocol for Building Automation and Control Networks
Second Public Review

[Change Clause 6.6, p. 65]

BACnet routers are devices that interconnect two or more BACnet networks to form a BACnet internetwork. A router may, or may not, provide BACnet application layer functionality. BACnet routers shall provide BACnet application layer functionality. BACnet routers shall be capable of routing messages of the maximum NPDU length of the data link layer types supported by the router as defined in Table 6-1. BACnet routers make use of BACnet network layer protocol messages to maintain their routing tables. Routers perform the routing tasks described in 6.5. See Figure 6-12 for a flow chart of router operation.

[Change Clause 9.5.2, p. 96]

TS
"This Station," the MAC address of this node. TS is generally read from a hardware DIP switch, or from nonvolatile memory. This variable represents the value of the MAC Address property of the node’s Network Port object which represents this MS/TP port. Valid values for TS are 0 to 254. The value 255 is used to denote broadcast when used as a destination address but is not allowed as a value for TS.

[Change Clause 9.5.3, p 96]

N_max_master
This parameter represents the value of the Max_Master property of the node's Device object Network Port object which represents this MS/TP port. The value of Max_Master specifies the highest allowable address for master nodes. The valid range of this parameter is 0 to 127. The value of Max_Master shall be less than or equal to 127. If Max_Master is not writable in a node, its value shall be 127.

N_max_info_frames
This parameter represents the value of the Max_Info_Frames property of the node's Device object Network Port object which represents this MS/TP port. The value of Max_Info_Frames specifies the maximum number of information frames the node may send before it must pass the token. Max_Info_Frames may have different values on different nodes but shall have a minimum value of 1 and a maximum value of 255. This may be used to allocate more or less of the available link bandwidth to particular nodes. If Max_Info_Frames is not writable in a node, its value shall be 1.

[Change Table 12-13, p 196]

<table>
<thead>
<tr>
<th>Property Identifier</th>
<th>Property Datatype</th>
<th>Conformance Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>O6, X</td>
</tr>
<tr>
<td>Max_Master</td>
<td>... Unsigned(1..127)</td>
<td></td>
</tr>
<tr>
<td>Max_Info_Frames</td>
<td>Unsigned</td>
<td>O6, X</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

* If the node supports multiple MS/TP networks, then these property values shall reflect the corresponding values from the Network Port object whose Network_Type is MSTP and has the lowest object instance.

[Change Clause 12.11.18 p. 199]

12.11.18 Max_APDU_Length_Accepted

This property, of type Unsigned, is the maximum number of octets that may be contained in a single, indivisible application layer protocol data unit. The value of this property shall be greater than or equal to 50. The value of this property is also constrained by the underlying data link technology and shall be less than or equal to the largest APDU_Length of the enabled Network Port objects used to represent the underlying data links. See Clauses 6 through 11.

If the value of this property is not encodable in the 'Max APDU Length Accepted' parameter of a ConfirmedRequest-PDU, then the value encoded shall be the highest encodable value less than the value of this property. In such cases, a responding device may ignore the encoded value in favor of the value of this property, if it is known.
12.11.32 Max_Master

The Max_Master property, of type Unsigned, shall be present and writable if the device is a master node on an MS/TP network. The value of Max_Master specifies the highest possible address for master nodes and shall be less than or equal to 127. The valid range for the value of this property is 0 to 127. If the Max_Master property is not writeable via BACnet services, its value shall be 127. See Clause 9.5.3.

If the device supports multiple MS/TP networks, then this property shall reflect the value of the Max_Master property of the enabled Network Port object with the lowest object instance whose Network_Type is MSTP.

12.11.33 Max_Info_Frames

The Max_Info_Frames property, of type Unsigned, shall be present if the device is a node on an MS/TP network. The value of Max_Info_Frames specifies the maximum number of information frames the node may send before it must pass the token. If Max_Info_Frames is not writable or otherwise user configurable, its value shall be one. See Clause 9.5.3.

If the device supports multiple MS/TP networks, then this property shall reflect the value of the Max_Info_Frames property of the enabled Network Port object with the lowest object instance whose Network_Type is MSTP.

Several object types defined in this clause have a property called "Reliability." This property is an enumerated datatype that may have different possible enumerations for different object types. The values defined below are a superset of all possible values of the Reliability property for all object types. The range of possible values returned for each specific object is defined in the appropriate object type definition.

- **NO_FAULT_DETECTED**
  The present value is reliable; that is, no other fault (enumerated below) has been detected.

- **ACTIVATE_FAILURE**
  Activation of changes has failed.

- **ROLLBACK_FAILURE**
  Rolling back of changes has failed.

- **RENEW_FD_REGISTRATION_FAILURE**
  Renewing a foreign device registration with a BBMD has failed.

- **RESTART_SLAVE_DISCOVERY_FAILURE**
  Restarting of the slave discovery algorithm has failed.

- **RENEW_DHCP_FAILURE**
  The attempt to obtain an IP address from a DHCP server has failed.

- **RESTART_AUTONEgotIATION_FAILURE**
  The auto-negotiation algorithm has failed.

- **RESTART_FAILURE**
  The attempt to restart the port has failed.

- **PROPRIETARY_COMMAND_FAILURE**
  A proprietary command has failed.
[Change Clause 15.5.2, p. 476]

... When the object-type in the Object Identifier parameter contains the value 'Device Object' and the instance in the 'Object Identifier' parameter contains the value 4194303, the responding BACnet-user shall treat the Object Identifier as if it correctly matched the local Device object. This allows the device instance of a device that does not generate I-Am messages to be determined.

When the object-type in the Object Identifier parameter contains the value 'Network Port object' and the instance in the 'Object Identifier' parameter contains the value 4194303, the responding BACnet-user shall treat the Object Identifier as if it correctly matched the local Network Port object representing the network port through which the request was received. This allows the network port instance of the network port that was used to receive the request to be determined.
[Change Clause 18.3, p. 520]

18.3 Error Class - PROPERTY

... INVALID_DATATYPE - The datatype of a property value specified in a service parameter does not match the datatype of the property referenced by the specified Property_Identifier.

INVALID_VALUE_IN_THIS_STATE - The value specified in a service parameter is invalid in the current state of the property.

LOGGED_VALUE_PURGED - A previously logged value was purged due to a change to the list of logged properties.

... [Change ASN.1 productions in Clause 21, pp. 562]

Error ::= SEQUENCE {

-- NOTE: The valid combinations of error-class and error-code are defined in Clause 18.
error-class ENUMERATED ... ENUMERATED {

... invalid-time-stamp (14),
invalid-value-in-this-state (n),
key-update-in-progress (100),
...

-- numerical order reference
...
-- see abort-security-error (136),
-- see invalid-value-in-this-state (n),
...
}

-- 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.

}

[Add new productions to Clause 21 in appropriate alphabetical position, starting p. 585]

BACnetBDTEntry ::= SEQUENCE {

  bacnetip-address [0] BACnetHostNPort,
  subnetmask [1] OCTET STRING
}

...

BACnetFDTEntry ::= SEQUENCE {

  bacnetip-address [0] OCTET STRING,  -- the 6-octet B/IP address of the registrant
  time-to-live [1] Unsigned16  -- time to live in seconds
}

...
BACnetHostAddress ::= Choice {
    null-value [0] Null,
    ip-address [1] OCTET STRING, -- 4 octets
    hostname [2] CharacterString -- DNS hostname
}

BACnetHostNPort ::= Sequence {
    address [0] BACnetHostAddress,
    port [1] Unsigned16
}

BACnetIPMode ::= Enumerated {
    normal (0),
    foreign (1),
    bhmd (2),
    ...
}

BACnetNetworkPortCommand ::= Enumerated {
    idle (0),
    activate (1),
    rollback (2),
    renew-fd-registration (3),
    restart-slave-discovery (4),
    renew-dhcp (5),
    restart-autonegotiation (6),
    disconnect (7),
    restart-port (8),
    ...
}
-- Enumerated values 0-63 are reserved for definition by ASHRAE. Enumerated values
-- 64-255 may be used by others subject to the procedures and constraints described
-- in Clause 23.

BACnetNetworkNumberQuality ::= Enumerated {
    unknown (0),
    learned (1),
    learned-configured (2),
    configured (3)
}

BACnetNetworkReachability ::= Enumerated {
    available (0),
    busy (1),
    disconnected (2)
}

BACnetNetworkType ::= Enumerated {
    ethernet (0),
    arcnet (1),
    mstp (2),
    ...
}
ptp  (3),  
lon talk  (4),  
bacnet-ipv4  (5),  
z igbee  (6),  
virtual  (7),  
...

-- Enumerated values 0-63 are reserved for definition by ASHRAE. Enumerated values
-- 64-255 may be used by others subject to the procedures and constraints described
-- in Clause 23.

...

BACnetRouterEntry ::= SEQUENCE {
  network-number   [0] Unsigned16,
  mac-address   [1] OCTET STRING,
  reachability   [2] BACnetNetworkReachability,
  performance-index   [3] Unsigned8 OPTIONAL
}

...

BACnetVMACEntry ::= SEQUENCE {
  virtual-mac-address [0] OCTET STRING, -- maximum size 6 octets
  native-mac-address [1] OCTET STRING
}

[Change Clause 21, p. 609]

BACnetObjectType ::= ENUMERATED { -- see below for numerical order
...
  multi-state-value (19),
  network-port (n),
  network-security (38),
...
  -- see time-value (50),
  -- see network-port (n),
...
}

BACnetObjectTypesSupported ::= BIT STRING {
...
  time-value (50),
  network-port (n)
}

[Change Clause 21, p. 613]

BACnetPropertyIdentifier ::= ENUMERATED { -- see below for numerical order
  all-writes-successful (9),
  apdu-length (n),
  apdu-segment-timeout (10),
...
  backup-preparation-time (339),
  bacnet-ip-default-gateway (n+1),
  bacnet-ip-dhcp-enable (n+2),
  bacnet-ip-dhcp-lease-time (n+3),
bacnet-ip-dhcp-lease-time-remaining  \( (n+4) \),
 bacnet-ip-dhcp-server  \( (n+5) \),
 bacnet-ip-mode  \( (n+6) \),
 bacnet-ip-multicast-address  \( (n+7) \),
 bacnet-ip-subnet-mask  \( (n+8) \),
 base-device-security-policy  \( (327) \),
 bbmd-accept-fd-registrations  \( (n+9) \),
 bbmd-broadcast-distribution-table  \( (n+10) \),
 bbmd-foreign-device-table  \( (n+11) \),
 bbmd-nat-traversal  \( (n+12) \),
 belongs-to  \( (262) \),

... bit-text  \( (343) \),
 buffer-size  \( (126) \),

... change-of-state-time  \( (16) \),
 changes-pending  \( (n+13) \),
 client-cov-increment  \( (127) \),

... command  \( (n+14) \),

... fd-bbmd-ip-address  \( (n+15) \),
 fd-subscription-lifetime  \( (n+16) \),
 firmware-revision  \( (44) \),
 full-duty-baseline  \( (215) \),
 global-identifier  \( (323) \),
 global-ip-address  \( (n+17) \),
 group-members  \( (345) \),
 group-member-names  \( (346) \),
 high-limit  \( (45) \),
 inactive-text  \( (46) \),

... limit-monitoring-interval  \( (182) \),
 link-speed  \( (n+18) \),
 link-speeds  \( (n+19) \),
 link-speed-autonegotiate  \( (n+20) \),
 list-of-group-members  \( (53) \),

... low-limit  \( (59) \),
 mac-address  \( (n+21) \),
 maintenance-required  \( (158) \),

... network-access-security-policies  \( (332) \),
 network-number  \( (n+22) \),
 network-number-quality  \( (n+23) \),
 node-subtype  \( (207) \),
 network-type  \( (n+24) \),

... restore-preparation-time  \( (341) \),
 routing-table  \( (n+25) \),
 scale  \( (187) \),

... verification-time  \( (326) \),
 virtual-mac-address-table  \( (n+26) \),

... -- see event-message-texts  \( (351) \),
BACnetReliability ::= ENUMERATED {
    no-fault-detected (0),
    no-sensor (1),
    over-range (2),
    under-range (3),
    open-loop (4),
    shorted-loop (5),
    no-output (6),
    unreliable-other (7),
    process-error (8),
    multi-state-fault (9),
    configuration-error (10),
    -- enumeration value 11 is reserved for a future addendum
    communication-failure (12),
    member-fault (13),
    activation-failure (n),
    rollback-failure (n+1),
    renew-fd-registration-failure (n+2),
    restart-slave-discovery-failure (n+3),
    renew-dhcp-failure (n+4),
    restart-auto-negotiation-failure (n+5),
    restart-failure (n+6),
    proprietary-command-failure (n+7),
    ...
}

-- Enumerated values 0-63 are reserved for definition by ASHRAE. Enumerated values
22.1.5 Minimum Device Requirements

A device that conforms to the BACnet protocol and contains an application layer shall:

(a) contain exactly one Device object,
(b) execute the ReadProperty service,
(c) execute the Who-Has and Who-Is services (and thus initiate the I-Have and I-Am services) unless the device is an MS/TP slave device,
(d) execute the WriteProperty service if the device executes the WritePropertyMultiple, AddListElement or RemoveListElement services,
(e) allow the WriteProperty service to modify any properties that are modifiable by the AddListElement or RemoveListElement services,
(f) execute the WriteProperty service if the device contains any objects with properties that are required to be writable, and
(g) contain a Network Port object for each configured network port.

[Change Table 23-1, p. 636]
J.4.3 BBMD Concept

Each IP subnet that is part of a B/IP network comprised of two or more subnets shall have at least one BBMD. Each BBMD shall possess a table called a Broadcast Distribution Table (BDT). If there are two or more BBMDs on a single subnet, their BDTs shall not contain any common entries in order to avoid a broadcast forwarding loop. The BDT determines which remote IP subnets receive forwarded BACnet broadcasts. To reduce BACnet broadcast traffic, it is possible to configure the BDT to forward broadcasts only to IP subnets where they are required. If the BBMD has also been designated to register foreign devices as described below, it shall also possess a Foreign Device Table (FDT).

_B/IP devices which are unable to be configured as a BBMD shall be capable of being configured to register itself with a BBMD as a foreign device._

J.4.5 BBMD Operation - Broadcast Distribution

Upon receipt of a BVLL Write-Broadcast-Distribution-Table message, a BBMD shall attempt to create or replace its BDT, depending on whether or not a BDT has previously existed. If the creation or replacement of the BDT is successful, the BBMD shall return a BVLC-Result message to the originating device with a result code of X'0000'. Otherwise, the BBMD shall return a BVLC-Result message to the originating device with a result code of X'0010' indicating that the write attempt has failed. _A BBMD which supports a protocol revision greater than X shall return a BVLC-Result message to the originating device with a result code of X'0010' indicating that the Write-Broadcast-Distribution BVLL message is not supported._

J.5.2 BBMD Operation - Foreign Devices

In order for a foreign device to fully participate in the activities of a B/IP network, the device must register itself with a BBMD serving one of the IP subnets comprising that network. _If a B/IP device is unable to be configured as a BBMD, then it shall be capable of being configured to register itself with a BBMD as a foreign device._ "Full participation" implies the ability to send and receive both directed and broadcast messages. Registration consists of sending a BVLL Register-Foreign-Device message to an appropriate BBMD and receiving a BVLC-Result message containing a result code of X'0000' indicating the successful completion of the registration. Ascertaining the IP address of such a BBMD is a local matter but could involve the use of a domain nameserver or the distribution of a numeric IP address to authorized users. The UDP port X'BAC0' shall be considered the default, but the use of other port values is permitted if required by the local network architecture, e.g., where two B/IP networks share the same physical LAN.