BACnet™ - A standard communication infrastructure for intelligent buildings

By: Steven T. Bushby,

National Institute of Standards and Technology, USA

Published in *Automation in Construction*, Vol. 6 No. 5-6, 1997, p. 529-540

Abstract

Intelligent buildings require integration of a variety of computer-based building automation and control system products that are usually made by different manufacturers. The exchange of information among these devices is critical to the successful operation of the building systems. Proprietary approaches to providing this communication has created great challenges for system integrators and hampered the development of intelligent building technology. Even though digital automation and control technology has been widely available for more than a decade and islands of automation are common, intelligent buildings with integrated building services are still more of a promise than a reality.

BACnet™ is a standard communication protocol for ***B****uilding* ***A****utomation and* ***C****ontrol* ***Net****works* developed by the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) [1]. BACnet provides the communication infrastructure needed to integrate products made by different vendors and to integrate building services that are now independent. This paper describes the main features of the BACnet protocol and early experience implementing it.

Key words: communication protocol, building automation and control network, intelligent buildings, energy management and control systems, direct digital control, BACnet.

Introduction

In the early 1980s, microprocessor-based direct digital control (DDC) systems began to appear in the market place. One of the earliest and most successful applications of the technology in the building industry was heating, ventilation, and air-conditioning control systems (HVAC). DDC systems provided more precise control, greater operational flexibility, improved occupant comfort, and reduced energy costs in comparison to electronic and pneumatic control systems. The great potential of this technology was recognized by many people and the concept of an "intelligent building" with distributed control systems that provide HVAC control, lighting control, security, fire detection and suppression, and vertical transport in an integrated and coordinated fashion was born. In the past fifteen years there have been significant advances in the technology and reductions in its cost. In spite of these advances, "intelligent buildings" are still more of a promise than a reality. One of the main stumbling blocks has been a lack of communication protocol standards.

Historically, competitive pressures and a lack of standards forced manufacturers of building automation equipment to develop unique, proprietary communication protocols. The result for building owners has been great difficulty integrating products made by different manufacturers. Stand-alone energy management systems, lighting control systems, and fire detection and suppression systems are now common, but integration of these systems is rare.

Even within a single building automation function, e.g., HVAC control, there has been difficulty. If there is a need to expand or upgrade the control system, a building owner has been forced to either return to the same vendor who installed the existing system, replace it in its entirety, or install a separate independent system because the communication protocols for other products were incompatible. In some cases new products have been incompatible with older products from the same vendor.

As the capability of microprocessors has grown and the prices have dropped, control intelligence has become more and more distributed. Manufacturers of large HVAC equipment, e.g. chillers, now build controllers designed specifically for their equipment and sell the equipment and the controller as a packaged unit. These companies also developed their own proprietary communication protocols. It has become necessary to integrate these packaged systems into the larger control systems from traditional control vendors. This forced traditional control vendors to share proprietary protocols at least on a limited basis and a new trend toward custom integrators of various kinds emerged.

Manufacturers must expend considerable resources developing gateways or translators between the various products they make and those of other companies for which they have made agreements to exchange proprietary protocols. Each of these gateways must be re-engineered if either party changes their own protocol in any way. Many products cannot be integrated at all because access to one of the proprietary protocols has not been granted. Gateways have provided some relief to the integration problem but they are not a satisfactory long term solution. The only credible long-term solution is an industry standard communication protocol.

The History of BACnet™ Development

The many problems caused by incompatible proprietary communication protocols motivated the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE), in January, 1987, to begin developing an industry standard communication protocol for building automation and control systems. Standard Project Committee 135P (SPC 135P) was formed to accomplish this task. The membership of SPC 135P was selected to provide a broad and balanced representation of the building control industry. The individuals came from manufacturers, consulting engineering firms, universities, and governmental agencies from Canada and the United States.

The first meeting of SPC 135P occurred in June of 1987. In August of 1991 the first public review draft of the proposed BACnet standard was published for comment. That review generated 507 comments from 6 different countries. Both the unusually high number of comments and the international response were a reflection of the high level of interest in the development of this standard throughout the building industry. The draft standard was revised to accommodate the concerns raised during the first public review.

A revised version of the draft standard was published for a second public review in March of 1994. This second review generated 228 comments from 12 countries. Once again modifications were made to resolve the comments. A third, and final, public review version was published for comment in March of 1995. The third review generated 6 comments. All of the third public review comments were resolved without making a substantive change to the draft standard. The final draft version was approved for publication as an ASHRAE standard in June of 1995, eight and a half years after the formal standardization process was begun. BACnet was approved by the American National Standards Institute (ANSI) as a national standard in December, 1995.

While BACnet was being developed ASHRAE established a liaison with Europeans working on a similar task under the auspices of the European Committee for Standardisation (CEN) Technical Committee 247. This committee has not completed its work but a decision has been made to adopt multiple protocols as European Community Pre-standards. BACnet has been chosen as one of those protocols.

A Technical Overview of BACnet

BACnet provides a way to convey data including, but not limited to: hardware binary input and output values; hardware analog input and output values; software binary and analog values; schedule information; alarm and event information; files; and control logic. BACnet does not define the internal configuration, data structures, or control logic of the controllers. The information that needs to be visible over the communication network is abstracted from the implementation details through the use of standard objects. The mapping between standard objects and the underlying data and processes is left to the vendor. A rich and comprehensive set of services is defined by the standard but classes of conformance are defined so that the standard can be implemented by devices with a wide range of capabilities.

BACnet has a layered protocol architecture based on a collapsed version of the Open Systems Interconnection (OSI) – Basic Reference Model [2]. Layers 1, 2, 3, and 7 of the OSI Model are used as shown in Figure 1. Common application layer and network layer protocols are used with any of four options for local area networking (LAN) technologies or a point-to-point (PTP) protocol suitable for dial-up telephone communications. The network layer provides a way to interconnect dissimilar LANs to form an internetwork. Each of these protocol layers will be described in more detail in following sections.

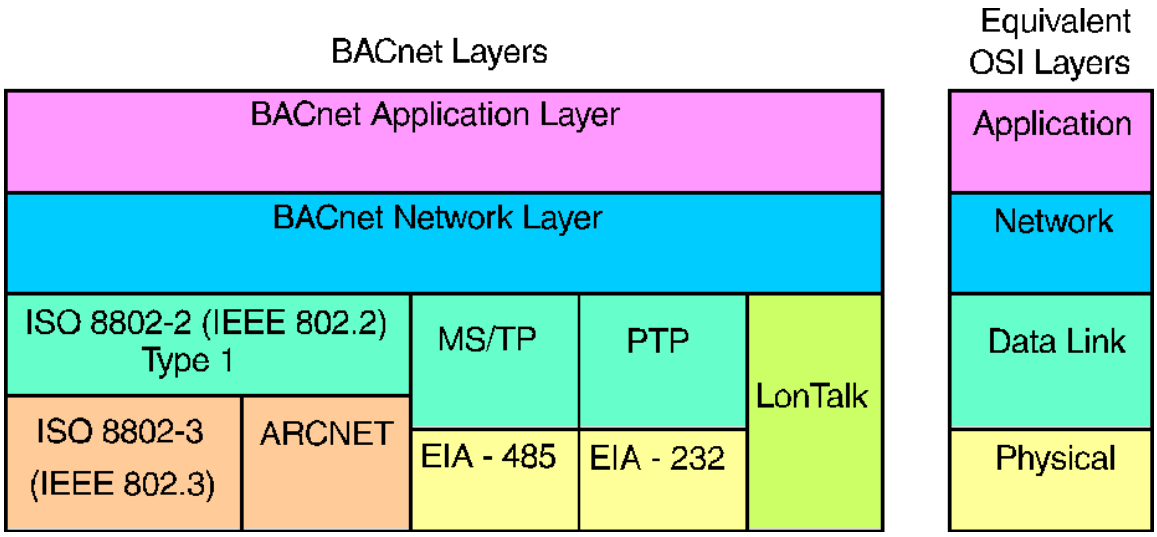

Figure 1. BACnet collapsed architecture

BACnet Application Layer

The key to understanding the BACnet Application Layer is to think of it as two separate but closely related parts: a model of the information contained in a building automation device; and a group of functions or "services" used to exchange that information.

The internal design and configuration of a BACnet device is proprietary in nature and different for each vendor. BACnet overcomes this obstacle by defining a collection of abstract data structures called "objects", the properties of which represent the various aspects of the hardware, software, and operation of the device. BACnet objects provide a means of identifying and accessing information without requiring knowledge of the details of a device's internal design. The communication software in the device can interpret requests for information about these abstract objects and translate those requests to obtain the information from the real data structures inside the device. Collectively, these objects provide a "network visible" representation of the BACnet device. BACnet defines 18 standard object-types as shown in Table 1.

Table 1. BACnet Standard Object Types

Analog Input*	Event Enrollment
Analog Output*	File
Analog Value*	Group
Binary Input*	Loop*
Binary Output*	Multi-state Input*
Binary Value*	Multi-state Output*

Calendar	Notification Class
Command	Program
Device	Schedule

*Optionally supports change of value (COV) reporting

The purpose of most BACnet objects is clear from the name of the object but some require explanation and are described as follows:

- ***Calendar*** represents a list of dates that have special meaning when scheduling the operation of mechanical equipment. A list of holidays would be one example.
- ***Command*** represents a multi-action command procedure, such as a sequenced startup of several devices.
- ***Device*** contains general information about a particular device, such as vendor name, model name, location, protocol version supported, object-types supported, etc.
- ***Event Enrollment*** provides one way to define alarms or other types of events and to indicate who should be notified when they occur. Some objects (Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, Binary Value, and Loop) contain optional properties to support intrinsic event reporting capability and do not need to use Event Enrollment objects.
- ***Group*** provides a shorthand way to read several values in one request. For example, it might be used to simultaneously update several fields on an operator graphic display.
- ***Loop*** can be used to represent any feedback control loop, which is some combination of proportional, integral, or derivative control.
- ***Notification Class*** provides a way to manage the distribution of alarm or event notifications that are to be sent to multiple devices.

Any BACnet device may have zero, one, or many objects of each type except for the Device object. All BACnet devices have exactly one Device object. Any object is accessed by means of a property called "Object_Identifier" that uniquely identifies each object within a single device. The Object_Identifier can be thought of as the "name" of the object. Device objects have a special requirement that their Object_Identifiers be unique throughout the entire BACnet internetwork. Thus, any BACnet object can be uniquely identified by the combination of its Object_Identifier and the Object_Identifier for the device in which it resides.

Object_Identifiers are represented by a data structure that consists of four octets (8-bit bytes) divided into two fields as shown in Figure 2. The Object Type field conveys an enumerated value that corresponds to a particular object type. The Instance Number field identifies which particular object of the specified type is being referenced. A portion of the Object Type enumerations have been reserved for use by vendors as a means to extend the standard by defining additional object types. Vendor extensions to BACnet are discussed in more detail below.

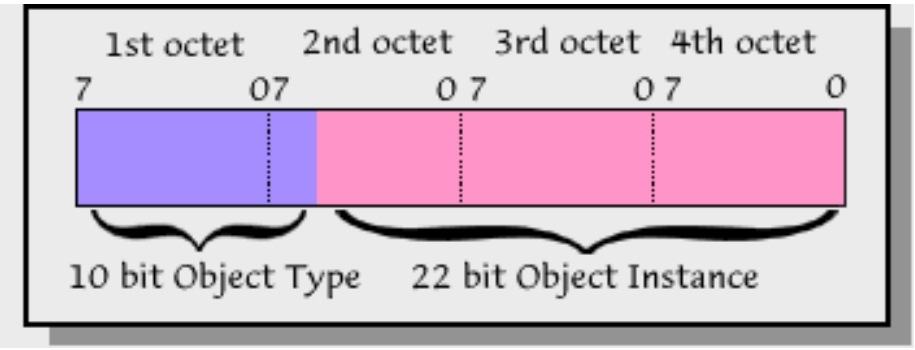

Figure 2. Object Identifier format

BACnet objects also have a property called Object_Name. This property is a character string that can be used to identify an object in a way that is meaningful to an operator. Meaningful names usually require more than four characters. Thus, most BACnet application layer services require the use of an Object_Identifier to specify a particular object because it is compact and has a fixed size. The Who-Has and I-Have services provide a way to find the Object_Identifier of an object that has a particular Object_Name.

Each property of a BACnet object has associated with it a "conformance code". This code specifies whether the property is optional (O), required to be present and readable using BACnet services (R), or required to be present, readable, and writable using BACnet

services (W). There is no requirement that all objects of a given type in a single device must support the same combination of optional properties.

In some cases there are restrictions on the way optional properties are used. For some objects support of a particular functionality implies a requirement to support one or more optional properties. For example, if an Analog Input object supports the ability to intrinsically detect out-of-range events, several optional properties, that collectively establish the lower and upper limits and the devices to be notified when an out-of-range event occurs, become required properties.

Support of one optional property sometimes implies a requirement to support one or more other optional properties because they form a matched set. For example, a Binary Output object may support the Elapsed_Active_Time property to accumulate the run time for a motor. Supporting this property implies a requirement to support the Time_Of_Active_Time_Reset property that indicates when the Elapsed_Active_Time was last reset to zero.

Application Services

Objects provide an abstract representation of the "network visible" portion of a building automation device. The "application services" provide "commands" for accessing and manipulating this information as well as providing some additional functions. BACnet defines 35 application services. Some application services are "confirmed services", meaning that an acknowledgment of some kind is expected. Other services are "unconfirmed". The standard groups the application services into six categories. Table 2 lists all of the services and shows how they are grouped.

Table 2. Application Layer Services

Alarm and Event Services	**Object Access Services**
AcknowledgeAlarm	*AddListElement*
ConfirmedCOVNotification	*RemoveListElement*
ConfirmedEventNotification	*CreateObject*
GetAlarmSummary	*DeleteObject*
GetEnrollmentSummary	*ReadProperty*
SubscribeCOV	*ReadPropertyConditional*
UnconfirmedCOVNotification	*ReadPropertyMultiple*
UnconfirmedEventNotification	*WriteProperty*
	WritePropertyMultiple
Remote Device Management Services	**Virtual Terminal Services**
DeviceCommunicationControl	*VT-Open*
ConfirmedPrivateTransfer	*VT-Close*
UnconfirmedPrivateTransfer	*VT-Data*
ReinitializeDevice	
ConfirmedTextMessage	**Security Services**
UnconfirmedTextMessage	*Authenticate*

UnconfirmedTextMessage	*Authenticate*
TimeSynchronization	*RequestKey*
Who-Has	
I-Have	**File Access Services**
Who-Is	*AtomicReadFile*
I-Am	*AtomicWriteFile*

The Alarm and Event Services provide a way to subscribe to change of value notifications, request a status summary for alarms or events, notify devices that alarms or events have occurred, and acknowledge that an operator has seen an alarm notification. In BACnet, "events" are changes of value of certain properties of certain objects, or internal status changes, that meet predetermined criteria. BACnet does not specify which events should be considered to be an "alarm". Instead, a property called Notify_Type is defined that allows the user to indicate on a case-by-case basis which events are to be considered an alarm.

There are three mechanisms provided in BACnet for managing events: change of value reporting (COV), intrinsic reporting, and algorithmic change reporting. COV reporting allows a client to subscribe on a permanent or temporary basis to receive reports of some changes of value of some referenced property based on fixed criteria. Optional properties are provided in several objects to support this functionality as shown in Table 1.

Intrinsic reporting is a mechanism used to define alarm or event conditions that are intrinsic to a particular object and represented solely by the properties of that object. Intrinsic events are based on an algorithm that is specific to the object type and linked to specific properties of that object. For example, Analog Input objects optionally support intrinsic reporting of out-of-range events. Optional properties define the acceptable range of values. The range checking can only be applied to the Present_Value property.

Algorithmic change reporting is a more general mechanism that can be applied to any property of any object. Six event algorithms are defined in BACnet as shown in Table 3. These algorithms are similar to the algorithms that are used in intrinsic event reporting. The principal difference is that they may be applied to any property of any object. Event Enrollment objects provide a network visible view of the event criteria and the linkages to the various objects involved.

Table 3. Algorithmic Change Event Algorithms

CHANGE_OF_BITSTRING
CHANGE_OF_STATE
CHANGE_OF_VALUE
COMMAND_FAILURE
FLOATING_LIMIT
OUT_OF_RANGE

File Access Services provide the means to read and write files, including the ability to upload and download control programs and databases. The services are called *AtomicReadFile* and *AtomicWriteFile* because during a read or write operation no other read or write operation is permitted for the same file. Thus, the service is carried out "atomically." The current version of BACnet does not define file formats. It does however provide access to files in either a record-structure format or as a continuous stream of octets.

Object Access Services provide a means to read the properties of objects, write to properties of objects and, in some cases, to create or delete an object.

Remote Device Management Services provide tools for troubleshooting and maintaining devices. The *Who-Is* and *I-Am* services may be used to dynamically locate peer devices. The *Who-Is* service request may be used to find the address of a device whose identifier is already known, or it may be used as a general solicitation to build a local table of all active devices on the BACnet internetwork. An *I-Am* service request is used to "answer" a *Who-Is* request. It may also be used to advertise that a device is present on the network. For

example, some BACnet devices initiate an *I-Am* service request on power-up. The *Who-Has* and *I-Have* services perform a similar function but they may be applied to any object, not just device objects.

Because the hardware and configuration details of building automation devices will continue to be proprietary and vendor-specific, there is a need to provide operators with tools that are vendor specific for configuring these devices. This is the role of the Virtual Terminal (VT) Services that provide a mechanism for bi-directional exchange of character-oriented data. This allows an operator console to interact with a BACnet device as if it were a directly-connected, dumb terminal. The protocol service merely conveys the character streams between the two devices. The meaning of the character streams is not defined in BACnet and must be interpreted in a proprietary manner. VT services can also be used as a security mechanism. If portions of the object database should not be changed by other devices on the network, these properties may be read-only. VT services provide a way to change these properties with vendor-specific security control.

BACnet provides other application services that are specifically oriented toward applications where security is an important concern. Peer entity authentication, data origin authentication, operator authentication, and data encryption are provided by the Security services. To make use of these features there must be one device on the network that plays the role of a key server. Each device that uses the security features must be assigned a private cryptographic key and must also support the *RequestKey* and *Authenticate* services. It is possible to have a mixture of devices, some of which support security features and some of which do not. It is a local decision whether or not security is important in a particular transaction. Thus, it is possible for devices that do not implement any security services to communicate with devices that do, as long as secure measures are not required for that particular kind of transaction.

Custom Extensions to BACnet

BACnet includes a formal description of all applications services written in Abstract Syntax Notation One (ASN.1), an international standard formal description language [3]. BACnet defines a set of encoding rules that can be applied to any valid ASN.1 production. This approach adds clarity to the informal description of the application services and provides a very clean mechanism for managing vendor proprietary extensions to BACnet and future additions to the standard.

There are four independent ways in which vendors may add custom extensions to the BACnet protocol: (1) extend values for enumerations that are designated as extensible; (2) invoke proprietary services using the *ConfirmedPrivateTransfer* or *UnconfirmedPrivateTransfer* services (see Table 2); (3) add proprietary properties to a standard object type; and (4) define new proprietary object types. BACnet clearly indicates which ASN.1 productions can be modified and the encoding rules determine how the modified messages and data are to be encoded. This makes it possible for vendor extensions to be completely interoperable.

As long as vendor extensions are publicly known, any device in the BACnet network can make use of the extended functionality. Conflicts between extensions made by different vendors are avoided by using a registered Vendor Identifier. This Vendor Identifier is a required property of the Device object. Any BACnet messages containing extended information implicitly reference this Vendor Identifier. Manufacturers of building automation devices can apply for a registered Vendor Identifier by contacting the ASHRAE Manager of Standards.

BACnet Network Layer

BACnet provides several options for networking technology that permit design flexibility based on cost and performance needs. A single networking technology can be used in a system or multiple options can be combined to form a BACnet internetwork. The purpose of the network layer protocol is to provide the means by which messages can be routed from one BACnet network to another, regardless of the BACnet data link technology in use on that network. The internetwork terminology used in BACnet is illustrated in Figure 3.

Some functions assigned to the network layer in the OSI Basic Reference Model are not required in BACnet. One example is selecting a communications path between source and destination devices. BACnet imposes a requirement that at most one active path can exist between two devices. Another network layer function that BACnet does not support is message segmentation and reassembly. BACnet imposes a limitation on the length of the messages that pass through a router. The maximum length shall not exceed the capability of any data link technology encountered along the path from source to destination. Messages longer than this can still be conveyed, but they must be segmented and reassembled at the application layer.

The network layer portion of a BACnet message contains a one-octet version number and a control octet that indicates the presence or absence of other network layer information. If the destination for the message is a device on the same network, no additional network layer information is needed. If the destination is on a remote network, the client device must include the destination network number and medium access control (MAC) address of the destination device. The router on the local network will insert addressing information about the local network so that a response can be returned. Thus, a device does not need to know its own network number. The only addressing information needed to send a message to a device on a remote network is its network number and its MAC address.

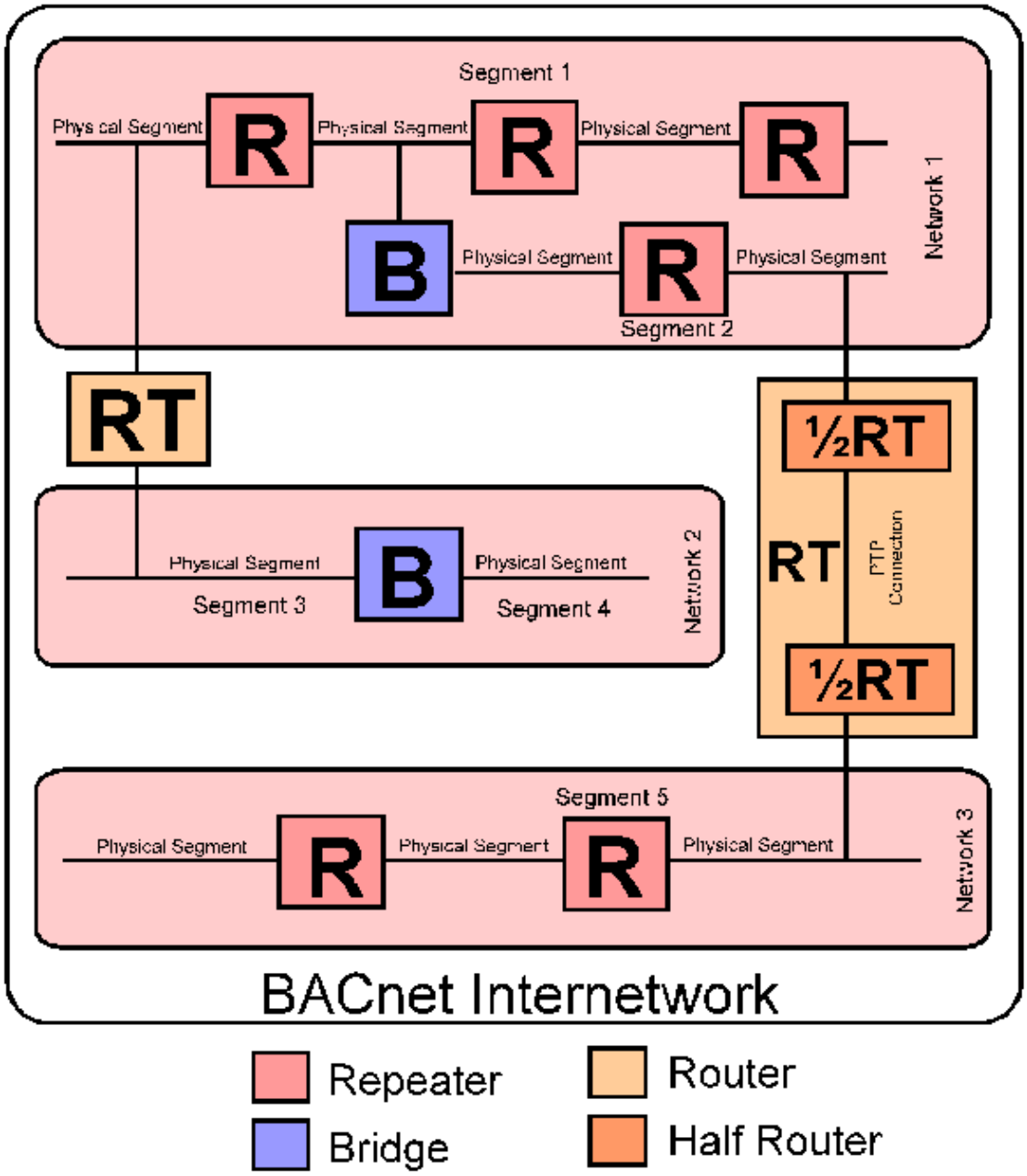

Figure 3. A BACnet Internetwork illustrating the concepts of Physical Segments, Repeaters, Segments, Bridges, Networks, Half Routers, and Routers.

BACnet defines network layer protocol services that allow router tables to be configured, allow routers to search for the path to a destination network, and manage temporary connections to remote networks through a dial-up telephone connection. Routers can also indicate to client devices that a path to the destination device cannot be located.

BACnet also provides a way to route messages through existing Internet Protocol (IP) networks and Novell's Internetwork Datagram Protocol (IPX) networks. Both of these protocols are capable of encapsulating/decapsulating BACnet messages and conveying them using a technique known as "tunneling". The standard describes the procedures in terms of devices called BACnet/IP (or IPX) Packet-Asembler-Disassemblers (PADs).

These devices take a BACnet message intended for a device on a remote network, look up in a local table the address of a corresponding PAD on the distant network, encapsulate the BACnet message in an IP or IPX packet, and then send the packet to a standard IP or IPX router on the local network. The process is reversed at the remote PAD and the message is forwarded to its ultimate destination. This process is illustrated in Figure 4 for the case of an IP internet.

Data Link and Physical Layers

As shown in Figure 1, BACnet provides four LAN technology options and one point-to-point protocol. These particular options were selected for several reasons including: speed; availability of chips or boards that implement the protocol; familiarity with the LAN on the part of manufactures in the building automation industry; degree of compatibility with existing systems; and cost. Collectively, these options provide a range in performance capabilities and cost.

This flexibility permits systems designers to choose an option or options that are most appropriate for a particular application. Large building automation systems frequently have multiple networks arranged in a hierarchical structure. Application specific controllers reside on a low-cost, low-speed LAN and are supervised by more sophisticated controllers that are interconnected by a high-speed LAN. BACnet permits this kind of hierarchical structure but does not require it. The flexibility provided by the BACnet layered architecture will also permit the protocol to accommodate future changes in technology.

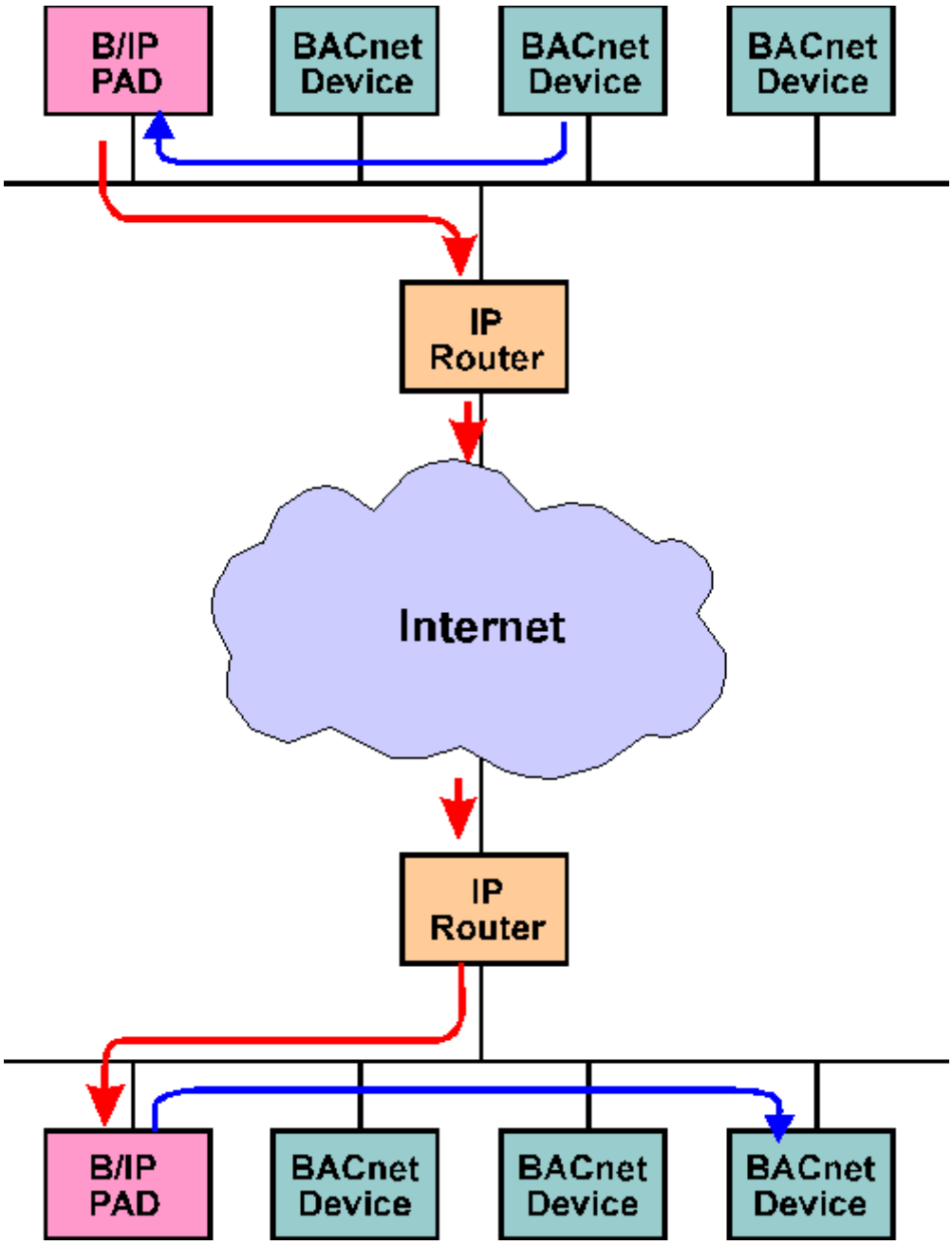

Figure 4. IP Tunneling with a B/IP PAD.

The highest performance LAN option is ISO 8802-3, better known as "Ethernet", the protocol developed by Digital Equipment Corporation, Intel, and Xerox, upon which the international standard is based [4]. It offers high speed (10 Mbps) and its use is extremely widespread. Several physical media are defined for ISO 8802-3, all of which are acceptable for use in BACnet. The BACnet standard states that ISO 8802-3, as amended and extended, is included in BACnet by reference. Thus, future enhancements to the Ethernet protocol automatically become part of BACnet.

Ethernet uses a carrier sense multiple access with collision detection (CSMA/CD) form of media access control. Ethernet makes very efficient use of the transmission medium until it becomes heavily loaded. At that point repeated collisions rapidly decrease message

throughput. For this reason it is not possible to predict, with absolute certainty, how long a device may have to wait before being able to transmit successfully. Nevertheless, properly engineered Ethernet networks will provide the highest speed performance and data throughput of any of the BACnet networking options.

The second alternative, ARCNET, was developed by Datapoint Corporation. It is now an American national standard (ATA/ANSI 878.1)[5]. ARCNET is a lower-cost alternative but, nonetheless, operates at the respectable speed of 2.5 Mbps. ARCNET claims to be the oldest commercially available LAN, with nearly 3 million nodes installed. The ARCNET standard, as amended and extended, is included in BACnet by reference. Thus, like Ethernet, any future enhancements automatically become part of BACnet. All of the media options defined in ATA/ANSI 878.1 are acceptable in BACnet. ARCNET is a token-passing protocol and thus is deterministic, meaning that it is possible to place a bound on the maximum time that a device could have to wait before having a chance to transmit a message. This could be a very important feature in some applications.

The third networking possibility in BACnet is based on the use of the EIA-485 (Electronic Industries Association) physical layer standard. The EIA-485 physical layer is one of the most commonly used physical layer technologies in building control systems, particularly for application-specific controller networks. Since the EIA-485 standard is a physical layer standard, it does not address the problem of regulating access to the transmission medium. BACnet defines a Master-Slave/Token-Passing (MS/TP) protocol to provide this data link layer function.

An MS/TP network has one or more master nodes that are peers on a logical token-passing ring. It may also have slave nodes that are unable to transmit messages until requested to do so by a master node. An MS/TP network can be made up entirely of master nodes forming a peer-to peer network. It can be a true master-slave network with a single master node and all other nodes as slaves. It can also be a combination of the two with multiple masters and slaves.

MS/TP has a one octet address space that is divided into three parts. The address X`FF' is reserved for broadcasts. The address range 128-254 is reserved for slave devices. The address range 0-127 may be used for either master nodes or slave nodes. This permits the address space to be configured to meet the needs of a particular application.

The final LAN protocol option in BACnet is LonTalk [6]. LonTalk is a patented, proprietary protocol developed by Echelon Corporation. LonTalk is a seven-layer protocol implemented on a single chip called a neuron. No physical medium is specified in the LonTalk protocol. Instead, a transceiver interface is defined. A wide variety of physical media can be used by selecting appropriate compatible transceivers. The LonTalk protocol is low-cost, low-speed, and has a very limited (128 octets) message size. Proprietary development tools from Echelon are required to implement the protocol.

Devices based on the LonTalk protocol are, in general, not interoperable because the protocol does not impose any particular application functionality. Interoperability can only be achieved by imposing external constraints. Echelon Corporation sponsors a LonMark program for the purpose of imposing such constraints for particular applications. BACnet is not part of the LonMark program and a LonMark certification does not indicate that a device is compatible with BACnet.

In BACnet the LonTalk protocol is used solely as a means to transfer data from one device to another just like the other BACnet LAN options. Interoperability is achieved by using the same application layer and network layer protocols as all other BACnet devices.

The final data link and physical layer option in BACnet is the Point-To-Point (PTP) protocol. The PTP protocol accesses the communication medium through an EIA-232 full duplex interface. A typical application would be to connect to a modem for dial-up access to a remote building automation system. The PTP protocol does not define how a physical connection is established. This permits the use of a custom connection sequence that can be used with a modem. The PTP protocol defines how BACnet communication is initiated, maintained, and terminated once a physical connection is established.

Early Implementation Experience

There are a number of field installations of BACnet devices in both the United States and Europe. These products, based on a draft version of the standard, have demonstrated the viability of BACnet in actual buildings. There is also a consortium of companies that has been formed to test the interoperability of their BACnet products.

NIST BACnet Interoperability Testing Consortium

In 1993 the US National Institute of Standards and Technology (NIST) invited interested companies to join a consortium for testing the interoperability of BACnet products. The purpose of the consortium was to: (1) verify the technical soundness of the then draft BACnet protocol; (2) provide feedback to ASHRAE SPC 135P about portions of the draft standard that were ambiguous or needed modification before the standard was completed; (3) assist companies in developing and testing the interoperability of prototype BACnet products; and (4) develop testing tools and procedures for a future industry-run BACnet certification program.

At the present time the NIST BACnet consortium has 18 members including NIST, Cornell University, and 16 private companies. The member companies, including the three largest HVAC control companies in the world, collectively control the dominant share of the

building automation and control market in North America, Europe, and Asia. Most of the consortium members have already developed and tested one or more BACnet products or prototype products.

Thirteen of the BACnet Consortium members participated in a demonstration of interoperable BACnet products at the ASHRAE sponsored International Air-Conditioning, Heating, Refrigerating Exposition in February, 1996. The demonstration included a BACnet internetwork made up of all four BACnet LAN options (PTP communication was not demonstrated). The building control applications on display were a chiller system with thermal storage, a multi-zone variable air volume (VAV) system, a building lighting control system, and a laboratory fume hood control system. Four different BACnet workstations, made by different participants, illustrated the ability to access data from and control applications on any controller in the demonstration, without regard to the manufacturer of the controller. Peer-to-peer controller communication between products made by different vendors was also demonstrated.

Field Experience with BACnet

Two companies, both members of the NIST consortium, released products before the BACnet standard was finalized based on a draft version of the protocol. One of these companies is most well known for their chillers and BACnet provided a way to interface their packaged chiller controls with a building automation system or an industrial process control system. That company also makes and sells BACnet HVAC energy management systems. The other company is a traditional HVAC control vendor.

As of December, 1995 these two companies together have sold and installed approximately 200 BACnet systems in the United States and 50 systems in Germany and Switzerland combined. Approximately 30 of these systems are multi-vendor systems. The multi-vendor systems are with building automation system companies and industrial process control companies who built a custom BACnet interface to an installed product.

The results from this field experience has been positive and demonstrates both a strong customer demand and acceptance for BACnet products, and a firm commitment from manufacturers to supply the products.

The First Large Scale Multi-Vendor BACnet System

The largest landlord in the world is the United States General Services Administration (GSA). This agency owns and maintains most of the office space used by civilian US government agencies. Altogether GSA controls approximately 8% of all office space in the United States. GSA has decided to make conformance to BACnet a requirement for all building automation systems purchased for use in their buildings.

To demonstrate the capabilities of BACnet, GSA has begun a project to install a multi-vendor BACnet system in 130,000 m^2 building in San Francisco, California. This project includes a complete HVAC energy management system, a lighting control system, and a BACnet interface with the local utility for load management and real-time negotiation for electricity rates. This project will include BACnet devices from up to five separate vendors. Participating vendors must qualify by, among other things, providing products for BACnet conformance testing in NIST laboratories. The first stage of the procurement process has begun and installation is expected in late 1996.

Summary

The BACnet protocol standard was developed through a consensus process with input from a wide range of experts in building automation. It was subjected to three rigorous public reviews that involved comments from experts in twelve countries. It is now an ASHRAE standard and has been selected as a European Pre-standard by CEN. Many companies are developing BACnet products and some are marketing them today. Early field experience in the United States, Germany and Switzerland has been very successful. The BACnet protocol provides a comprehensive and flexible communications infrastructure for building automation systems that can provide the foundation for intelligent buildings.

Acknowledgments

The author wishes to acknowledge the Federal Energy Management Program (FEMP) and the Office of Energy Efficiency and Renewable Energy of the US Department of Energy, and the US General Services Administration for providing financial support for NIST's contribution to the development of the BACnet protocol and tools to test BACnet implementations.

Disclaimer

Certain trade names and company products are mentioned in the text or identified in an illustration in order to adequately specify their relationship to the BACnet standard. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purpose.

References

[1] ASHRAE, *Standard 135-1995: BACnet™ - A Data Communication Protocol for Building Automation and Control Networks*. American Society of Heating, Refrigerating, and Air-Conditioning Engineers. Atlanta, Georgia, USA. (1995).

[2] ISO 7498, *Information processing systems - Open Systems Interconnection - Basic Reference Model*. International Organization for Standardization (1984).

[3] ISO 8824, *Information technology - Open Systems interconnection- Specification of Abstract Syntax Notation One (ASN.1)*. International Organization for Standardization (1990).

[4] ISO 8802-3, *Information processing systems, Local area networks - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*. International Organization for Standardization (1993).

[5] ATA/ANSI 878.1, *ARCNET Local Area Network Standard*. American National Standards Institute, 11 W. 42nd St., 13th Floor, New York, NY 10036 (1992).

[6] *LonTalk® Protocol Specification Version 3.0*. Echelon Corporation, 4015 Miranda Ave., Palo Alto, CA 94304.