ASHRAE Journal, Vol. 36, No. 1, January 1994

BACnet: A Technical Update

Substantive changes have been made to the proposed BACnet standard, which will undergo a second public review in 1994

By Steven T. Bushby and *H. Michael Newman*
Associate Member ASHRAE Member ASHRAE

BACnet is a communication protocol for **B**uilding **A**utomation and **C**ontrol **net**works that has been developed by ASHRAE Standards Project Committee (SPC) 135P. The committee has been working on the standard since the middle of 1987 and published a draft version for public review in August, 1991.

As a result of the public review (which generated 507 formal comments), numerous substantive changes have been made to the proposed standard. Included are changes in object type definitions, application layer services, and the network layer, MS/TP and Point-to-Point protocols. Before describing these changes and why they were made, a brief overview of BACnet would be helpful.

BACnet overview

BACnet aims to provide mechanisms by which computerized equipment of arbitrary function can exchange information, regardless of the particular building service it performs. To develop such mechanisms, SPC 135P decided that there were four key components to the development process that would have to be tackled.

The first component was to decide how to represent the internal functioning of any vendor's equipment in a common, network-visible way, recognizing both the proprietary nature of each vendor's internal design and the diversity of functionality involved.

The adopted solution was to model each BACnet device as a collection of data structures called objects, each of which is characterized by a set of attributes or properties. The 18 currently defined standard BACnet object types include analog and binary input, output and value, and various object types to represent control loops, alarm and event functions, and scheduling operations.[1,2]

The second component was to agree on a set of common commands or services that could be used between devices to get them to carry out the functions of distributed monitoring and control. In other words, what kinds of messages should the protocol provide to facilitate communications about common building automation functions and what rules should govern their exchange?

The standard currently provides 30 services in five areas: alarm and event services; file access services; object access services; remote device management services; and virtual terminal services.

The third component was to agree on how to encode the messages defined above in a standard way. How should the mes-

Continued on page S74

About the authors

Steven T. Bushby is an engineer in the Mechanical Systems and Controls Group, National Institute of Standards and Technology, Gaithersburg, Maryland. He received a bachelor's degree in chemical engineering from Northwestern University, a master's degree in chemical engineering from Colorado State University, and a master's degree in computer science from the University of Maryland. Bushby is the vice-chairman of ASHRAE TC 1.4 (Control Theory and Application) and secretary of SPC-135P as well as the manager of the NIST BACnet Interoperability Testing Consortium.

H. Michael Newman is the manager of the Facilities Engineering Computer Section, which oversees the extensive multi-vendor EMCS at Cornell University, Ithaca, New York. He received his bachelor's and master's degrees in engineering physics from Cornell and did post-graduate work in astrophysics in the NASA Center for Radiophysics and Space Research. Newman is the chairman of SPC-135P and has been a presenter of the ASHRAE Professional Development Seminar on DDC. He is also the author of a new book entitled *Direct Digital Control of Building Systems.*

BACnet: A Technical Update

Continued from page S72

sages be represented as binary zeros and ones on the communications media?

This problem was addressed by adopting the *Abstract Syntax Notation One* (ASN.1)[3] from the International Organization for Standardization (ISO), along with an encoding technique specifically designed to produce as little overhead as possible while still maintaining the generality and extensibility of ASN.1.

Finally, what network technologies should be used to actually get the BACnet messages from one device to another? To address this problem, SPC-135P decided to specify options that span the range of speed and throughput requirements appropriate to the variety of building automation and control equipment in use today and the foreseeable future.

This meant taking advantage of existing technologies wherever possible (such as Ethernet and ARCNET), but also developing standards for low-speed networking over twisted pairs using EIA-485 signaling, as well as providing for dial-up or point-to-point access. It was also necessary to provide for the case where two or more BACnet local area networks might need to be interconnected. To this end, BACnet provides its own routing standard as well as indicating how BACnet messages might be routed over existing networks using other routing technologies.

The framework for all of these developments was the ISO's *Open Systems Interconnection (OSI) Basic Reference Model.* This model divides communication and networking requirements into seven categories or layers.[4] The functions of the layers range from how the communication services interact with applications (the application layer) to how the zeros and ones are electrically generated and conveyed (the physical layer).

In terms of the OSI model, the SPC's development methodology was a top-down approach because its initial focus was on establishing the functioning of the application layer. As a result, discussions on the lower layers were deferred until the message structure, content, exchange rules and encoding of the application layer had, for all practical purposes, been decided.

BACnet is actually a "collapsed" architecture that specifies protocols for only four of the seven layers: application, network, data link and physical. The functions normally provided by the other layers were deemed unnecessary for building automation communication.

In essence, the BACnet application layer protocol specifies the services mentioned earlier and the rules for their use. The network layer provides the routing mechanisms necessary to interconnect multiple BACnet networks. Finally, the data link and physical layer specifications provide the details of how the BACnet networks function.

Object definition changes

The concept of representing various aspects of the hardware, software and operation of control devices as abstract objects is one of the key underpinnings of the BACnet application layer. It is not surprising that the details of how to accomplish this were very carefully scrutinized during the first public review.

Several significant changes were made to this portion of the standard as a result of this review. The most sweeping change was a new approach to identifying individual instances of objects.

New object identifier format. In the first public review draft of BACnet, each object had a property called Object Identifier which must be unique within the device that contains it. When combined with the address of the device, it provided a unique reference to all objects in the BACnet network.

There has been no change in this general approach. What has changed is the datatype of the Object Identifier property. The old concept was to permit a choice from among three possible data types: a character string, an octet string or an integer.

The reason for the character string choice was that character strings are easy to remember and convenient for people to use. This is an advantage when writing programs or troubleshooting problems. However, the problem with character strings is that they are potentially very long and string matching is complicated. This is too much for simple devices to handle.

The octet string and integer choices were provided to meet the needs of devices that could not support the overhead of character strings. After reflection, some prototype implementation and receiving public review comments, SPC-135P decided that having a choice was a very bad idea.

How do you handle the situation where different devices use different choices? Even if a device only supports integer identifiers internally, it still must be able to communicate with devices that use the other choices.

Another problem is that object identifiers are used frequently, so inefficient encoding uses up a lot of bandwidth. This is especially a problem for low-speed Master-Slave/Token-Passing (MS/TP) networks and Point-to-Point (PTP) connections.

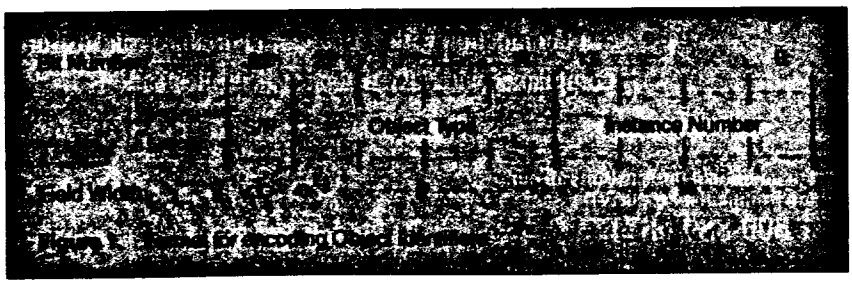
For these reasons, the choices were eliminated and a single, compact format for object identifiers was created. Object identifiers are now represented as a data structure consisting of three fields.

The first field indicates whether the object type is defined in the BACnet standard or if it is a proprietary extension. The second field indicates which object type it is. The last field indicates which particular instance of the object it is. All of this information is encoded in three octets (8-bit bytes) as shown in *Figure 1.*

To retain the convenience of character string names, a new required property called Object_Name was added to all objects. The Object_Name has a datatype of character string. The new *Who-Has* and *I-Have* services described below can be used to dynamically create a mapping between a convenient character string name and the compact object identifier representation. This character string can be as small as one character to avoid resource limitations in simple devices.

This approach eliminates the problems caused by having a choice for object

Continued on page S76



Continued from page S74

identifier datatypes and it reduces the overhead for transmitting object identifiers. However, the utility of character strings is retained.

BACnetNumeric datatype becomes real. Many standard object types have one or more properties that had a datatype of *BACnetNumeric*. A *BACnetNumeric* was a choice between an integer or a floating point number.

Once again when implementations began to be developed, it became clear that having this choice was a source of trouble from the standpoint of interoperability. When a device initiates a read request for one of these properties, it must be prepared to get the answer back in either format. The device must also accept an attempt to write the property in either format. It was decided that this was an unnecessary complication.

To solve the problem, the datatype for these properties was changed to floating point. This eliminates the complications of having a choice. This approach forces devices that are capable of doing only integer arithmetic to accept and handle data that is a real number.

Techniques for carrying out floating point operations using integer arithmetic are well known and commonly used in building controllers today. This was considered to be less of a burden than dealing with the more complicated logic of accepting any valid choice for the datatype.

Some objects were renamed or eliminated. Three object types were eliminated as a result of the review process and one new object type was created. The objects that were eliminated were the Directory object, the Mailbox object and the Device Table object.

The Directory object was replaced by adding new services to perform this function. These new services are described below. The Mailbox object was determined, upon reflection, to be a poor substitute for electronic mail. The choices were to develop a better approach to electronic mail or eliminate it. In the committee's opinion, this kind of mail service is not essential to the success of the protocol and so it was eliminated.

The Device Table object was eliminated because changes in the way alarm and event processing are handled, combined with the creation of a new Class object, made it unnecessary. The new Class object is used to manage information about who is to be notified when an event occurs. The changes to alarm and event processing are discussed in more detail below.

Application service changes

The semantics of all BACnet application services are defined in the standard through the use of ASN.1. BACnet also defines a set of rules that are applied to encode the services for transmission.

Some changes were made to these rules to make it easier to write software for encoding and decoding BACnet messages. The changes pertain to the way tags are used to delimit the substituent parts of the message. The details of these changes are beyond the scope of this article but are very important to anyone who wishes to build a BACnet device.

In the BACnet public review draft, all character strings were assumed to consist of characters defined by the American Standard Code for Information Interchange (ASCII). It was acknowledged that there may be a need for other character sets, particularly character sets used in foreign languages.

This problem was left as a "local matter." In other words, it was left for implementors to work out for themselves. Non-English language systems are used in the United States and many US companies sell products in foreign countries. A clear message was received that BACnet needed to explicitly address this problem.

The solution adopted was to officially recognize three character sets: ASCII, the IBM/Microsoft Double Byte Character Set, and JIS C 6226. These character sets can accommodate most languages currently in use in the world today.

The character set being used is specified when the string is encoded. The first octet of the encoded string is an integer enumeration that denotes the character set used. It is also possible to extend the enumeration to new values and thus add new character sets.

In addition to these changes which are very broad in their impact, there were changes that apply only to specific services. New directory services have been added to replace the functionality of the Directory object, new security services have been added, there have been changes in the way alarms and events are handled, and procedures for making proprietary extensions to BACnet have been clarified. Each of these changes and additions is discussed below.

New directory services. A Directory object type was defined in the BACnet public review draft. Instances of this object offered what amounted to a static table indicating where network resources could be found.

To make effective use of this table, the *ReadPropertyConditional* service would have had to be used. Maintaining accurate tables is difficult from an administrative point of view. The *ReadPropertyConditional* service is also by far the most complex service to implement in BACnet and would be beyond the capabilities of many devices. For these reasons, a new approach was developed to locate resources on the network.

The first public review draft of BACnet had a pair of unconfirmed services, *Who-Is* and *I-Am*. These services can be used to locate a particular device on the network. A similar pair of services, *Who-Has* and *I-Have*, was created to perform the function of locating particular objects.

Consider the case where a device wants to find out how to access outdoor air temperature. It would issue a *Who-Has* service request conveying the appropriate parameters. Any device that has an object instance that matches the parameters in the *Who-Has* service request would respond by issuing an *I-Have* service request.

The original device would get an answer from every device that supported an object matching the supplied description. If the object is unique, only one reply would be received. If it is not unique, then some criteria must be applied to select the one to use.

The *Who-Has* and *I-Have* services are much easier to implement than *ReadPropertyConditional*. Thus, this important directory service capability can be made available to even simple BACnet devices. As a bonus, the complexities of maintaining up-to-date tables have also been eliminated.

New security enhancements. The BACnet public review draft provided minimal security features. The assumption was that local security measures at the operator interface to the network would be sufficient. Several review comments were received challenging this assumption.

Furthermore, one goal of BACnet is to provide a mechanism for integrating various kinds of building services. One of these building services may well be security

Continued on page S78

BACnet: A Technical Update

Continued from page S76

devices that restrict access to buildings or parts of buildings. To do its job, this kind of system must have additional security features that are not present in the public review draft.

To address the concerns raised about security, a new clause has been added to the standard that provides for an optional security infrastructure. The new security features are based on the widely known Data Encryption Standard (DES).[5]

BACnet applies the techniques of this standard to provide peer entity authentication, data origin authentication, operator authentication and data encryption. Cryptographic keys are used to accomplish these tasks. The details of how this works are beyond the scope of this article.

To make use of these security features in BACnet, 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 a *Request Key* service and an *Authenticate* service. It is possible for the BACnet network to have a mixture of devices, some of which support security features and some of which do not.

It is a local decision whether security is important for 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.

Changes in alarm and event processing. One of the most important functions of building control devices is to detect the occurrence of specified events or alarm conditions and to take appropriate action when they occur. This is also one of the most complicated tasks they perform.

The techniques used to carry out these functions and their underpinning philosophies vary widely in the industry. It was probably harder to develop consensus on alarm and event handling than any other task SPC-135P had to face.

The approach taken in the BACnet public review draft was to define a general purpose Event Enrollment (EE) object and a set of commonly used algorithms for defining alarms and events.

The EE object is a kind of general template. By filling in the template with appropriate values, it is possible to define almost any kind of condition as an event or alarm, apply it to any property of any object with the appropriate datatype, and construct lists of recipients for both confirmed and unconfirmed notifications that the event had occurred.

The mechanism also included the ability to acknowledge alarms and report changes of value. This approach is very general and perhaps even elegant in some ways. It is also complicated.

This complexity became a problem. The project committee became convinced that this approach is too great a burden for many simple devices that must have at least a limited ability to detect and announce alarms or events. The general purpose mechanism is desirable, but there is also a need for a simpler approach that can accommodate the mundane tasks that make up the majority of alarms and events in real systems, things like out-of-limit detection and change-of-value (COV) reporting.

Accordingly, modifications have been made to the protocol to provide a simple, but inflexible, mechanism for several routine situations and yet retain the more general capabilities for situations that require them.

The key to simplification was the definition of "intrinsic events" and COV subscription. An intrinsic event is a pre-defined algorithm that is applied to a specific property of an object. Any parameters that must be adjusted in the algorithm appear as new properties of the object.

One property of the object indicates who is to be notified when the event occurs. No EE object is required and there is no flexibility in either the algorithm or the property being monitored. An example should help make the concept clear.

The Analog Input object now has an intrinsic event that provides limit checking on the Present_Value property. This would be used, for example, to set high and low temperature limits. The new properties used to implement the limit checking are:

Class	Deadband
Time_Delay	Limit_Enable
High_Limit	Alarm_Enable
Low_Limit	Acked_States

The Class property is a pointer to a Class object that contains all of the details about who is to be notified and whether the notification is confirmed or unconfirmed. It permits time-dependent lists of recipients so messages can be routed to different places at different times. The Limit and Deadband properties have conventional meanings.

The Time_Delay property specifies how long the condition should persist before the event is considered to have occurred. Alarm_Enable is a property consisting of three independent flags that indicate if notifications are to be sent for off-normal, fault or return events. The Acked_States property also consists of three flags that indicate whether an acknowledgement has been received for the most recent off-normal, fault or return event notification.

The implementation of intrinsic events is optional. Similar intrinsic events are defined for binary input and multi-state input objects.

COV reporting was simplified by defining three new services: *SubscribeCOV; ConfirmedCOVNotification;* and *UnconfirmedCOVNotification.* The standard specifies which objects and properties support this COV service. *Table 1* summarizes this information.

A device wishing to receive COV notifications subscribes via the *SubscribeCOV* service. The subscription lasts for a

Continued on page S80

Table 1. COV Reporting Conditions

Object Types	Criteria for Reporting
Analog Input Analog Output Analog Value	If Present_Value changes by COV_Increment or Status_Flags change at all, then report Present_Value and Status_Flags
Binary Input Binary Output Binary Value Multi-state Input Multi-state Output	If either Present_Value or Status_Flags change at all, then report Present_Value and Status_Flags
[illegible]	If Present_Value changes by COV_Increment or Status_Flags change at all, then report Present_Value, Status_Flags, Setpoint, and [illegible]

Continued from page S78

specified amount of time. If the subscription is accepted, a COV notification is immediately sent indicating the current value of the property. After that, changes in value of the monitored property cause a notification to be generated. This process continues until the subscribing device cancels the subscription using the *SubscribeCOV* service or until the specified time elapses.

Proprietary extensions to BACnet. BACnet was designed to permit proprietary extensions. It is possible to define new object types, new properties for objects and new application services. This capability has always been part of BACnet. It is widely believed that this is a very important strength, a means to ensure that there is an open path for development of new and innovative technology and products.

In the BACnet public review draft, this was basically an unmanaged capability. It was fine to make these extensions, but there was no mechanism to manage them to prevent one vendor's extension from conflicting with another's. Thus, there was a need to find a way to manage the extensions and prevent conflicts.

This goal was accomplished by making a few minor changes. The key idea was to assign a unique ID to all vendors that make BACnet products. ASHRAE would maintain a register of vendor IDs and assign new ones as needed.

This vendor ID was added as a property to the Device object and is mandatory in all BACnet devices. Thus, by reading this property, it can be determined who the vendor is. Any extensions encountered are extensions made by that vendor.

Note that it is not necessary to include this information in every message. Once you know who the vendor is, it does not change. By making it a property of the Device object, this information is available to anyone who needs it, but it does not add overhead to any messages.

The other part of the solution to this problem was specifying more precisely how to make extensions. The datatype for all properties and all service parameters defined in BACnet are constructed from a set of 13 primitive datatypes. All proprietary properties or service parameters must also be a datatype constructed from this set. The consequence of this restriction is that all possible extensions can be encoded (and decoded) using the rules already contained in the standard.

Extensions to object types can now be accommodated by extending enumerations already in the standard and by defining the datatype for the extension. For each enumeration that can be extended, a range of enumeration values is explicitly set aside for vendors to use for this purpose. The combination of the vendor ID and the extended enumeration value uniquely identifies each extension. Examples of the kinds of extensions possible via this mechanism are: new object types; new properties of objects; new error codes; and new alarm types.

Extending application services is handled somewhat differently. The first public review draft had an application service called *BlockTransfer*. This was a confirmed service used to transmit an arbitrary block of data that is not defined in the standard.

This service has been modified and now provides the mechanism for adding a proprietary application service. An unconfirmed version of the service was also added.

The names were also changed to better reflect the new purpose. These services are now called *ConfirmedPrivateTransfer* and *UnconfirmedPrivateTransfer*. These services convey both the vendor ID and a service number that are used to uniquely identify the proprietary service being invoked. The private transfer services are the only permissible way to add proprietary application services.

Network layer protocol changes

The purpose of the network layer protocol in BACnet, as in other OSI protocol stacks, is to facilitate the routing of messages from one network to another in the internetwork. An internetwork is a set of two or more networks, possibly of different LAN technologies, joined together by devices (routers) that implement the network layer routing protocol.

Thus, BACnet routers may interconnect ARCNET, Ethernet and MS/TP

Continued on page S82

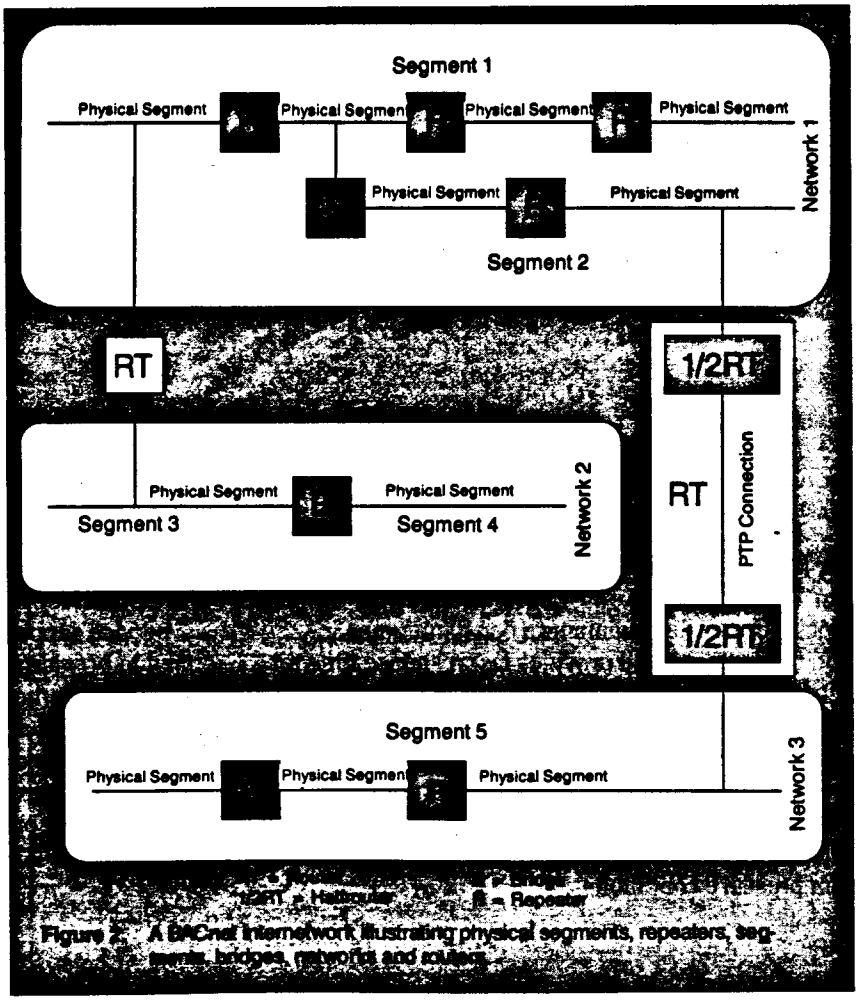


Figure 2: A BACnet internetwork illustrating physical segments, repeaters, segments, bridges, networks and routers.

BACnet: A Technical Update

Continued from page S80

LANs as well as provide point-to-point connections to other routers or BACnet devices (as shown in *Figure 2*).

Here are the principal changes made as a result of the first public review:

- Static routers (those with fixed routing tables) have been eliminated. All routers are now dynamic and must be able to build and maintain their routing tables based on the procedures described in the standard.
- Two new network layer messages have been added to permit the initialization and maintenance of the routing tables.
- Three new network layer messages have been added to facilitate the establishment and disconnection of point-to-point, possibly dial-up, connections.
- A decrementing hop-count field has been added to the network layer header to prevent a message from being indefinitely routed in a circle in the event that multiple paths have been established between interconnected networks contrary to the BACnet configuration rules.
- The description of how the network layer works with the MS/TP protocol has been improved.
- A means of identifying the version of the BACnet network layer protocol has been added to qualify BACnet for a standardized protocol identifier (issued by the Institute of Electrical and Electronics Engineers).
- Support has been added for vendor-proprietary network layer messages, which are potentially useful for interconnecting BACnet and existing non-BACnet networks.
- Finally, a procedure has been established whereby a message can be broadcast on a remote network without the originator of the message having to know the network technology in use on that network.

As a result of these refinements, the BACnet network layer protocol is more powerful than ever but still retains most of its previous virtues. Among these are: network layer overhead is low (only two octets) for messages that are intended to remain on the local network; messages may be easily broadcast on a single remote network or on all networks even without knowing what kind of network they are; and router operation remains simple because the BACnet configuration rules permit only a single path between any two devices on the internetwork.

The first public review revealed that there is considerable interest in routing messages through existing networks that may know nothing of the BACnet network layer. In recognition of this need, two annexes have been added to the standard to describe how BACnet messages may be conveyed by routers that use the US Department of Defense's Internet Protocol (IP) as well as Novell's IPX routing protocol. IP is found on many college and university campuses, while IPX is one of the most prevalent protocols in the commercial business environment.

Both protocols are capable of encapsulating/decapsulating BACnet messages and conveying them across an IP or IPX internetwork using a technique known as "tunneling." The new annexes describe the procedures in terms of devices called BACnet/IP (or IPX) Packet-Assembler-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 BACnet destination. This is illustrated in *Figure 3* for the case of an IP internet.

In addition, the annexes also describe how a device could be configured to combine the capabilities of a PAD and an IP(X) router in a single machine. Its operation is illustrated in *Figure 4*, again for the case of an IP internet.

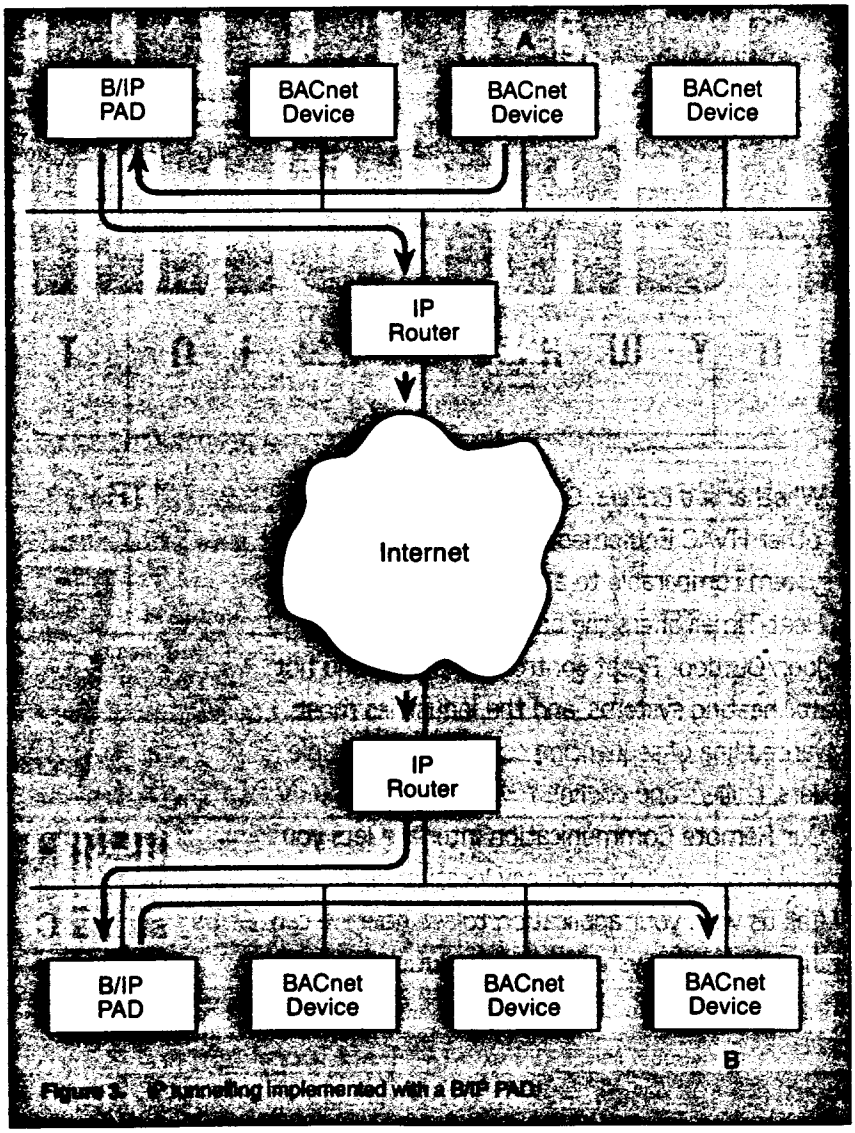


Figure 3. IP tunneling implemented with a B/IP PAD.

MS/TP protocol changes

MS/TP is BACnet's Master-Slave/Token-Passing protocol for use in conjunction with EIA-485 signaling and twisted pair wiring. The development of MS/TP was motivated by the committee's desire to specify a tried-and-true technology with which virtually every vendor of building automation and control equipment has had extensive experience. It is intended to be a low cost alternative to the higher cost, higher performance ARCNET and Ethernet LANs.

Three major changes have been made to MS/TP as a result of the first public review. The first change has to do with how a node determines that an entire MS/TP message or frame has been received. In the first BACnet draft, this was done by listening for a specific amount of idle time on the network (a period of time in which no data was being transmitted by any node), subsequent to the start of a valid frame (indicated by the special sequence of bits X'55FF').

The problems with this approach were that it required highly accurate timing and it could not tolerate gaps in a transmission resulting from the normal timing fluctuations that occur in any computer system that uses its processor for more than a single task. Moreover, this timing constraint would have made it difficult to implement MS/TP on machines where direct software access to the communications hardware is cumbersome (such as PCs running standard *DOS* or *Windows*).

The new technique involves the use of the length field in the MS/TP message and an additional error checking octet in the header to guarantee the correct receipt of the length information. Instead of waiting for idle time, a receiver now simply counts message octets until the requisite number has been tallied. At this point, the receiver can check the validity of the entire message and process it as needed. This procedure can be implemented on virtually any computer with a software interface to the serial communication port's receive buffer and control registers.

The second significant change permits greater flexibility in assigning network addresses to master and slave nodes. In the first draft, addresses 0 to 31 were reserved for master nodes (nodes capable of participating in the token-passing, token maintenance, and message initiation activities of the network). Addresses 32 to 254 were reserved for slaves (devices that can only respond to inquiries from masters). In the new version of BACnet, addresses 0 to 127 can be used for either masters or slaves, while only slaves may have addresses greater than 127.

Besides greater flexibility in determining the mix of masters and slaves, the new scheme allows for greater efficiency in bringing new nodes into the logical token ring. This has been accomplished by adding a new configuration parameter to the Device object of all MS/TP nodes that indicates the highest address that a master node may have on the local network.

Thus, in the periodic searching for new masters that is part of all token-passing network implementations, only the addresses from 0 up to the maximum address contained in the device object must be explored for the presence of a node waiting to join the ring. While this is scarcely an earthshaking refinement, it will nonetheless allow MS/TP networks to be tuned to avoid unnecessary poll-for-master messages for addresses that will never be used for master nodes.

The third important change to MS/TP concerns what a master node is allowed to do during the time it possesses the token. In the first public review draft, a master node was only allowed to transmit a single message to a single recipient. If the recipient was a slave, the master was allowed to wait for a response for a specified maximum amount of time. This is still true.

However, if the recipient happened to be another master, the receiving master was

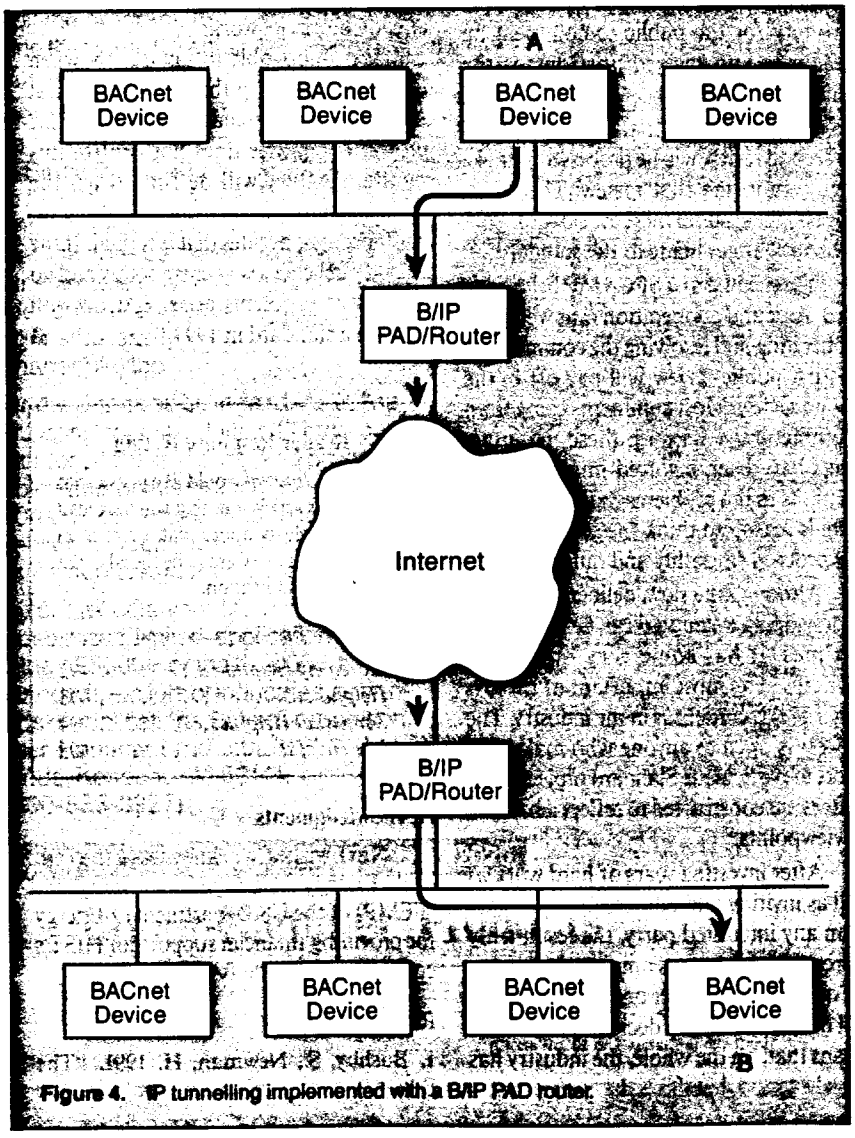


Figure 4. IP tunnelling implemented with a B/IP PAD router.

not allowed to respond until it was in possession of the token some time later. Now it may respond immediately.

Even more significant is that masters may now conduct more than one request/response transaction each time they possess the token. The number permitted is again a configuration parameter stored in the MS/TP node's Device object.

Thus, nodes that are expected to have an unusually high requirement for network access (such as operator workstations) can be granted preferential access to the network by increasing the maximum number of transactions they are allowed to carry out before they are required to pass on the token.

The effect of all these changes is to make MS/TP more efficient in terms of network utilization and the resulting throughput. This will be of particular benefit to the limited resource, low cost BACnet devices (such as unitary or application-specific controllers) that will most likely be found on MS/TP networks.

Point-to-Point protocol changes

The first public review draft contained a special protocol for providing dial-up telephone access to BACnet networks. This protocol has been significantly revised and renamed. It is now called the Point-to-Point (PTP) protocol.

The new name is intended to reflect the fact that it applies to a point-to-point connection that may or may not be achieved through the use of a telephone circuit. The protocol applies to any kind of point-to-point connection using serial asynchronous communications.

The state machine describing the PTP protocol has been completely rewritten. It now more closely matches the style of the state machines used to define the MS/TP protocol.

Two important enhancements have also been made to the protocol. The first is a character transparency mechanism that allows control characters to be sent to a modem. The second major change was the addition of a "heartbeat" message to provide a way to verify that the connection is still active during controller inactivity.

The road ahead

The building industry has been anxiously awaiting the completion of the BACnet protocol for a very long time. As mentioned, 507 comments were received during the first public review, which is the second highest number in the history of ASHRAE. Comments were received from people in six different countries. It took SPC-135P nearly two years to resolve the review comments.

This public interest clearly shows the importance of this effort. It also shows that expectations are very high. There are a number of inadequate solutions available today. What is needed is a standard protocol that is both comprehensive and robust. Doing the job right is more important than doing it fast.

The next step in the process is another public review. All of the substantive changes that have been made during the first review must now be published for everyone to see. This provides an opportunity for the public to comment on these new and revised portions of the standard.

Any comments that are received for the second review will be processed just like they were in the first review. The entire process repeats until there are no more substantive changes made to the standard.

The members of SPC-135P believe the hard work and cooperation that went into deliberating and resolving the comments in the first public review will pay off in the second review. Most comments came from manufacturers. Because these manufacturers have been involved in crafting the solutions to the problems that were raised, there is reason to think the second review will proceed smoothly and quickly.

Although the slow, deliberate process of developing a standard can be frustrating at times, it has some very important strengths. The most important of these is that it builds consensus in the industry. The process is open to anyone who makes the effort to participate. Standard project committees are constructed to reflect a balance of viewpoints.

After investing years of hard work as well as inviting and addressing comments from any interested party, the result truly represents a consensus opinion. In a sense, the question of whether or not the standard will be used is already decided. Consensus means that, on the whole, the industry has already agreed that this is the correct way to proceed.

However, having a final standard is not an end in itself. The real goal is interoperable products. To help achieve that goal, the US National Institute of Standards and Technology (NIST) has invited any manufacturer that wishes to build a BACnet implementation to join a consortium for testing the interoperability of the implementations. It is expected that this consortium will play a key role in assuring the correct implementation and interoperability of new BACnet products.

One of the goals is to pave the way for an industry-run testing and certification program for BACnet devices by developing the tools and techniques that will be needed to conduct the tests. As this time, 10 companies have already indicated an intention to join the consortium. By early 1994, it is expected that the consortium will have already begun its work.

It is impossible to say exactly when BACnet will be published as a final ASHRAE/ANSI standard. Ultimately, the project committee does not control this timing. BACnet will be final when the industry, through interaction in the standards process, decides that it is final. If the second public review goes smoothly and no significant problems arise, BACnet will become a standard in 1994. ■

Reader Response Rating

ASHRAE Journal would like to ask that you rate this article now that you have read it. Please circle the appropriate number on the Reader Service Card to be found at the back of the publication.

Rating:

Extremely Helpful	492
Helpful	493
Somewhat Helpful	494
Not Helpful	495

Acknowledgments

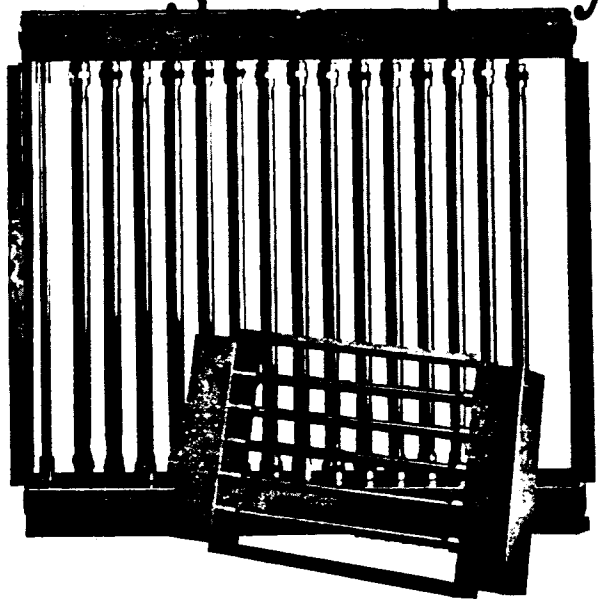
NIST wishes to acknowledge the Federal Energy Management Program (FEMP) of the US Department of Energy for providing financial support for NIST's contributions to the BACnet protocol.

References

1. Bushby, S., Newman, H. 1991. "The BACnet communication protocol for

Continued on page S86

We've gotten ourselves in the tightest spot yet.



Introducing ULTRA-SORB™

The greatest innovation in humidification in 20 years.

The leaders in humidification once again lead the way with virtually instantaneous absorption.

The need for rapid, drip-free steam absorption in "tight space" duct systems can now, for the first time ever, be truly satisfied with the ULTRA-SORB multiple dispersion tube panel from DRI-STEEM.

The ULTRA-SORB multiple tube panel provides virtually instantaneous absorption allowing it to be safely mounted within inches upstream of fans, coils, dampers, etc. with no fear of wetness.

Eliminates unwanted heating of duct air.

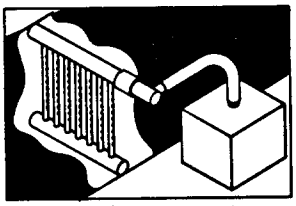
Because of the revolutionary design of the ULTRA-SORB panel, there is no need for steam jacketed dispersion tubes. The ULTRA-SORB dispersion tubes are hot only when actually humidifying, and that means energy cost savings for you.

Works on any steam pressure down to mere ounces.

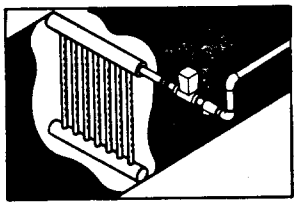
In ULTRA-SORB, steam-borne water droplets are removed by separation instead of re-evaporation, the method currently used in conventional steam jacketed humidifiers.

The separation of condensation in ULTRA-SORB occurs as the steam passes through the header/separator, down the closely spaced duct tubes and finally to the air stream through non-metallic orificed tubelets.

This design requires only ounces of pressure to function. Because of this, ULTRA-SORB can be used with boiler steam, as well as steam from any of our evaporative humidifiers such as STS, LTS or VAPORSTREAM.



Ultra-Sorb connected to VAPORSTREAM.



Ultra-Sorb connected to boiler steam.

Easy Installation.

Because the ULTRA-SORB panel comes pre-assembled within a mounting frame, it is easily installed or retrofitted into any duct system. Simply mount ULTRA-SORB, connect the steam and drain piping, and it's ready for use.

So call DRI-STEEM today and let us show you just how easy and dependable humidification can be...even in the tightest of spots.

SEE US AT BOOTH 3701

DRI-STEEM® HUMIDIFIER COMPANY

14949 Technology Drive, Eden Prairie, MN 55344, In MN: (612) 949-2415, Fax (612) 949-2933

© 1991, DRI-STEEM Humidifier Co. **Call toll-free: 1-800-397-8336**

(Circle No. 270 on Reader Service Card)

BACnet: A Technical Update

Continued from page 84

building automation systems." *ASHRAE Journal*. Atlanta, Georgia. Vol. 33, No. 4, pp. 14-21.

2. ASHRAE. 1993. *BACnet—A Data Communication Protocol for Building Automation and Control Networks, Advanced Working Draft 3*. ASHRAE SPC-135P-031. Atlanta, Georgia.

3. ISO. 1990. *International Standard 8824, Information Technology—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1)*. New York, New York: American National Standards Institute.

4. ISO. 1984. *International Standard 7498, Information Processing Systems—Open Systems Interconnection—Basic Reference Model*. New York, New York: American National Standards Institute.

5. FIPS. 1988. *Federal Information Processing Standard 46-1, Data Encryption Standard*. (Also known as ANSI X3.92-1987.) Gaithersburg, Maryland: National Institute of Standards and Technology.

Bibliography

ANSI. 1992. *American National Standard 878.1, ARCNET Local Area Network Standard*. New York, New York: National Institute of Standards and Technology.

EIA. 1983. *Electronic Industries Association Standard 485, Standard for Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems*. Washington, DC: Electronic Industries Association.

EIA. 1991. *Electronic Industries Association Standard 232-E, Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange*. Washington, DC: Electronic Industries Association.

ISO. 1989. *International Standard 8802-2, Information Processing Systems—Local Area Networks—Part 2: Logical Link Control*. New York, New York: American National Standards Institute.

ISO. 1990. *International Standard 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*. New York, New York: American National Standards Institute.

JIS. 1983. *Japanese National Standard JIS C 6226, Code of the Japanese Graphic Character Set for Information Interchange*. Japan Institute for Standardization.