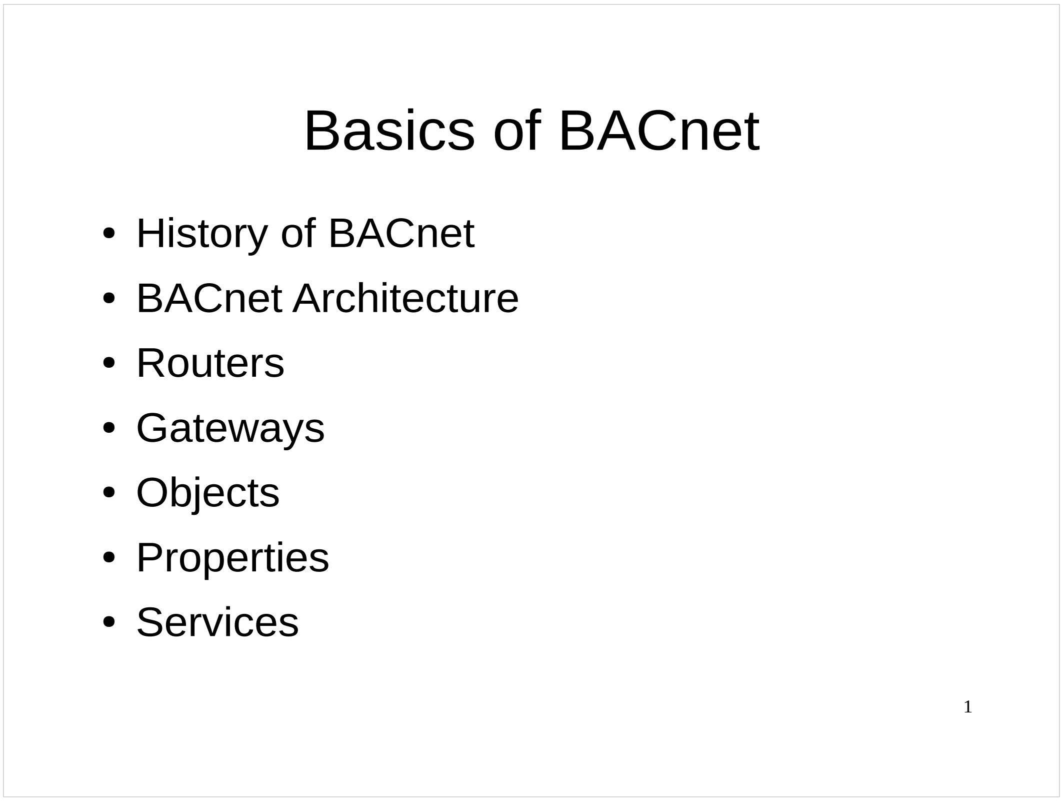
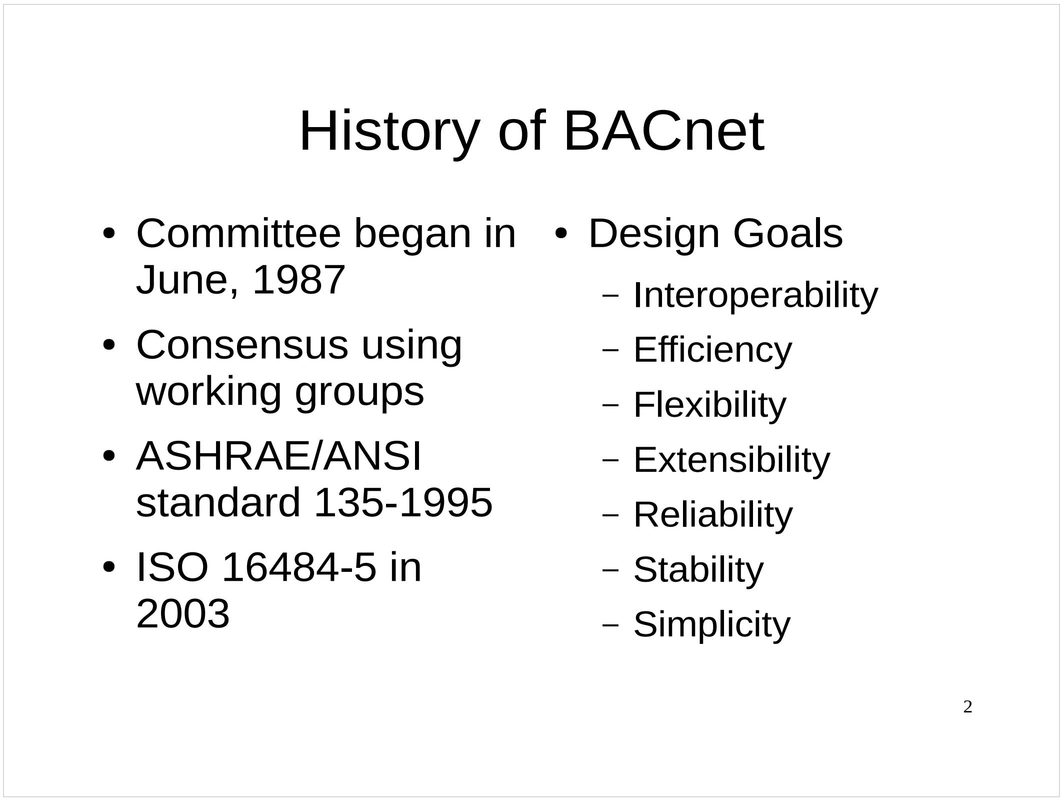


# Basics of BACnet

- History of BACnet
- BACnet Architecture
- Routers
- Gateways
- Objects
- Properties
- Services

This presentation provides an overview of the history, scope, and features of the BACnet protocol.

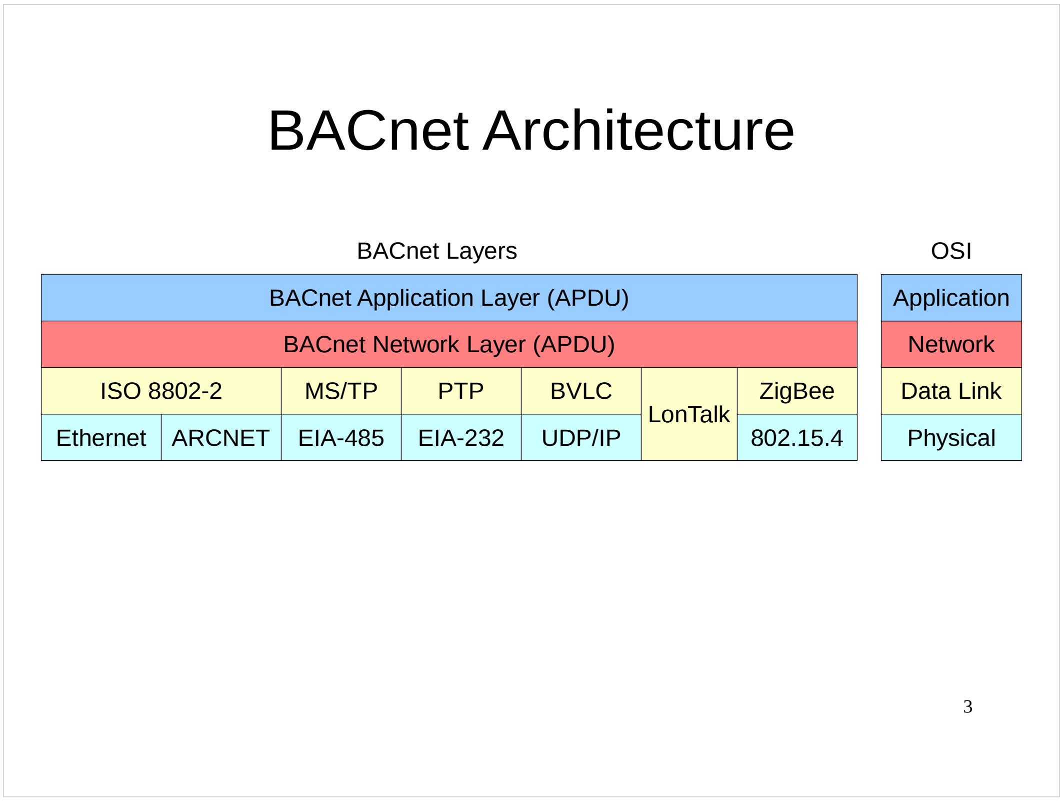


# History of BACnet

- Committee began in June, 1987
- Consensus using working groups
- ASHRAE/ANSI standard 135-1995
- ISO 16484-5 in 2003

- Design Goals
  - Interoperability
  - Efficiency
  - Flexibility
  - Extensibility
  - Reliability
  - Stability
  - Simplicity

The development of the BACnet protocol began in June, 1987, in Nashville, Tennessee, at the inaugural meeting of the Standard Project Committee (SPC). H. Michael Newman, the first chairman of the committee, presided over the meeting. The first meeting produced a list of desirable attributes of a good protocol, and what the BACnet protocol eventually became: Interoperability, Efficiency, Low Overhead, Highest Common Multiplier, Compatibility with other applications and networks, Layered OSI model Network, Flexibility, Extensibility, Cost Effective, Transmission Reliability, Apply to real-time processes, Maximum Simplicity, Allow priority schemes, Medium access fairness, and Stability under realistic loads.



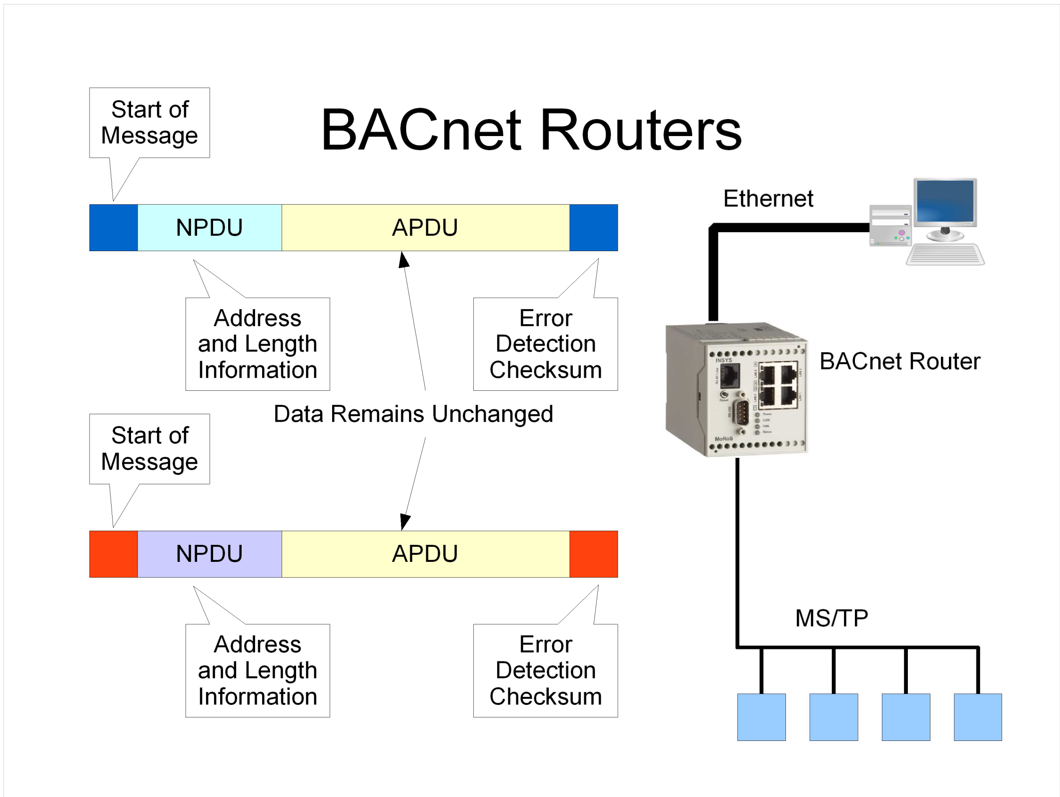
# BACnet Architecture

| BACnet Layers | | | | | | | | OSI |
|---|---|---|---|---|---|---|---|---|
| BACnet Application Layer (APDU) | | | | | | | | Application |
| BACnet Network Layer (APDU) | | | | | | | | Network |
| ISO 8802-2 | | MS/TP | PTP | BVLC | LonTalk | ZigBee | | Data Link |
| Ethernet | ARCNET | EIA-485 | EIA-232 | UDP/IP | | 802.15.4 | | Physical |

### The BACnet Collapsed Architecture

BACnet is based on a four-layer collapsed architecture that corresponds to the physical, data link, network, and application layers of the OSI model as shown in Figure 4-2. The application layer and a simple network layer are defined in the BACnet standard. BACnet provides five options that correspond to the OSI data link and physical layers. Option 1 is the logical link control (LLC) protocol defined by ISO 8802-2 Type 1, combined with the ISO 8802-3 medium access control (MAC) and physical layer protocol. ISO 8802-2 Type 1 provides unacknowledged connectionless service only. ISO 8802-3 is the international standard version of the well-known "Ethernet" protocol. Option 2 is the ISO 8802-2 Type 1 protocol combined with ARCNET (ATA/ANSI 878.1). Option 3 is a Master-Slave/Token-Passing (MS/TP) protocol designed specifically for building automation and control devices as part of the BACnet standard. The MS/TP protocol provides an interface to the network layer that looks like the ISO 8802-2 Type 1 protocol and controls access to an EIA-485 physical layer. Option 4, the Point-To-Point protocol, provides mechanisms for hardwired or dial-up serial, asynchronous communication. Option 5 is the LonTalk protocol. Collectively these options provide a master/slave MAC, deterministic token-passing MAC, high-speed contention MAC, dial-up access, star and bus topologies, and a choice of twisted-pair, coax, or fiber optic media. The details of these options are described in Clauses 7 through 11.

A four-layer collapsed architecture was chosen after careful consideration of the particular features and requirements of BAC networks, including a constraint that protocol overhead needed to be as small as possible. The reasoning behind the selection of the physical, data link, network, and application layers for inclusion in the BACnet architecture is outlined in this subclause.



BACnet Routers

Start of Message

NPDU

APDU

Address and Length Information

Error Detection Checksum

Data Remains Unchanged

Start of Message

NPDU

APDU

Address and Length Information

Error Detection Checksum

Ethernet

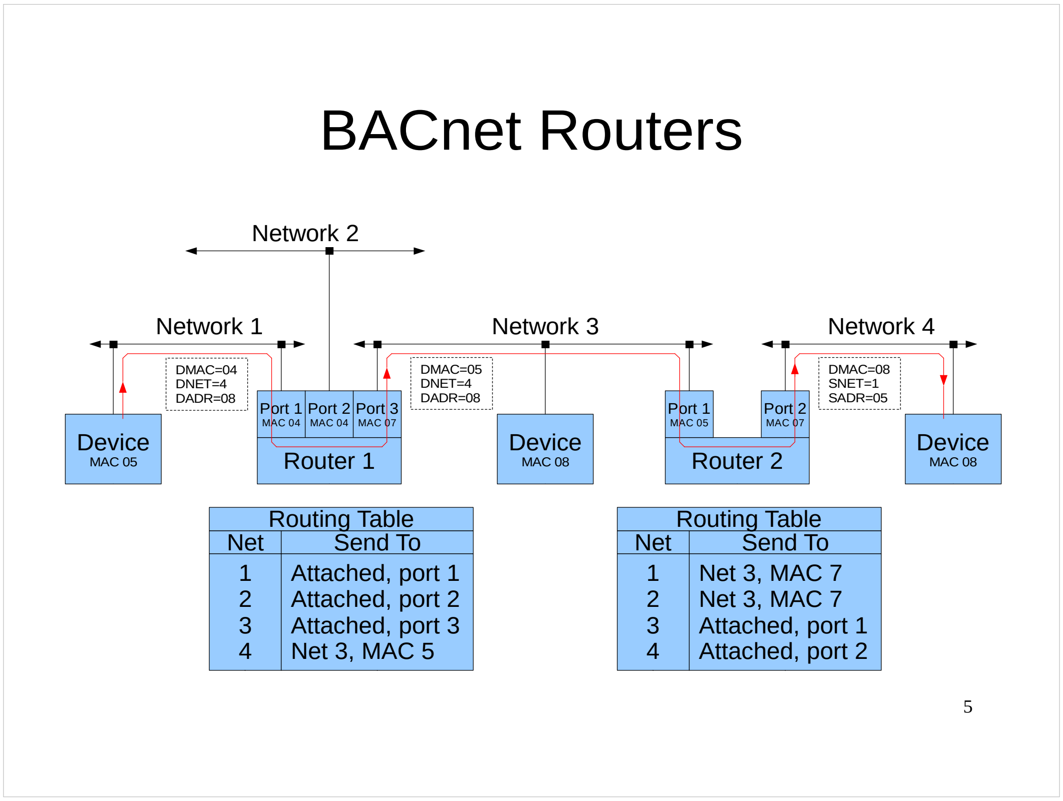
BACnet Router

MS/TP

THE NETWORK LAYER

The purpose of the BACnet network layer is to provide the means by which messages can be relayed from one BACnet network to another, regardless of the BACnet data link technology in use on that network. Whereas the data link layer provides the capability to address messages to a single device or broadcast them to all devices on the local network, the network layer allows messages to be directed to a single remote device, broadcast on a remote network, or broadcast globally to all devices on all networks. A BACnet Device is uniquely located by a network number and a MAC address.

Devices that interconnect two disparate BACnet LANs, e.g., ISO 8802-3 and ARCNET, and provide the relay function described in this clause are called "BACnet routers." Devices that interconnect two disparate BACnet networks through a point-to-point (PTP) connection (see Clause 10) are also BACnet routers. BACnet routers build and maintain their routing tables automatically using the network layer protocol messages defined in this clause. Network layer protocol messages facilitate both the auto-configuration of routers and the flow of messages to, and between, routers. BACnet routing capability may be implemented in stand-alone devices or, alternatively, in devices that carry out other building automation and control functions.



# BACnet Routers

| Routing Table | |
|---|---|
| Net | Send To |
| 1 | Attached, port 1 |
| 2 | Attached, port 2 |
| 3 | Attached, port 3 |
| 4 | Net 3, MAC 5 |

| Routing Table | |
|---|---|
| Net | Send To |
| 1 | Net 3, MAC 7 |
| 2 | Net 3, MAC 7 |
| 3 | Attached, port 1 |
| 4 | Attached, port 2 |

5

## Network Layer Protocol Messages

Clause 6.4 describes the format and purpose of the ten BACnet network layer protocol messages. These messages provide the basis for router auto-configuration, router table maintenance, and network layer congestion control.

### Who-Is-Router-To-Network

This message is indicated by a Message Type of X'00' optionally followed by a 2-octet network number. Who-Is-Router-To-Network is used by both routing and non-routing nodes to ascertain the next router to a specific destination network or, in the case of routers, as an aid in building an up-to-date routing table. See Figure 6-5.

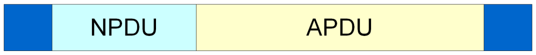
### I-Am-Router-To-Network

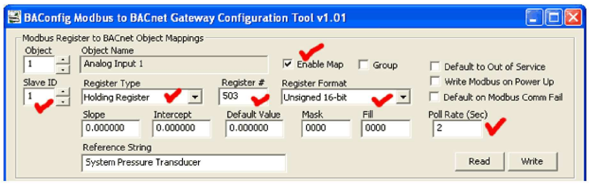
This message is indicated by a Message Type of X'01' followed by one or more 2-octet network numbers. It is used to indicate the network numbers of the networks accessible through the router generating the message. It shall always be transmitted with a broadcast MAC address.

### I-Could-Be-Router-To-Network

This message is used to respond to a Who-Is-Router-To-Network message containing a specific 2-octet network number when the responding half-router has the capability of establishing a PTP connection that can be used to reach the desired network but this PTP connection is not currently established.

# BACnet Gateway




BACconfig Modbus to BACnet Gateway Configuration Tool v1.01

Modbus Register to BACnet Object Mappings

Object: 1

Object Name: Analog Input 1

☑ Enable Map

□ Group

□ Default to Out of Service

□ Write Modbus on Power Up

□ Default on Modbus Comm Fail

Slave ID: 1

Register Type: Holding Register

Register #: 503

Register Format: Unsigned 16-bit

Slope: 0.000000

Intercept: 0.000000

Default Value: 0.000000

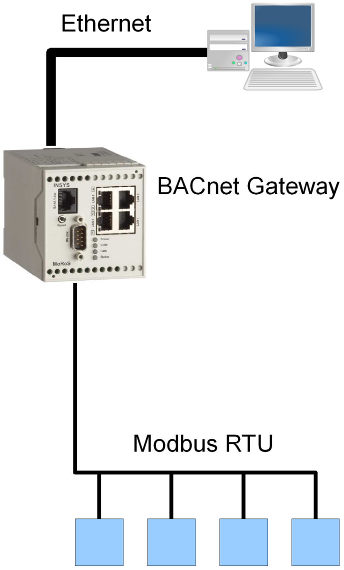
Mask: 0000

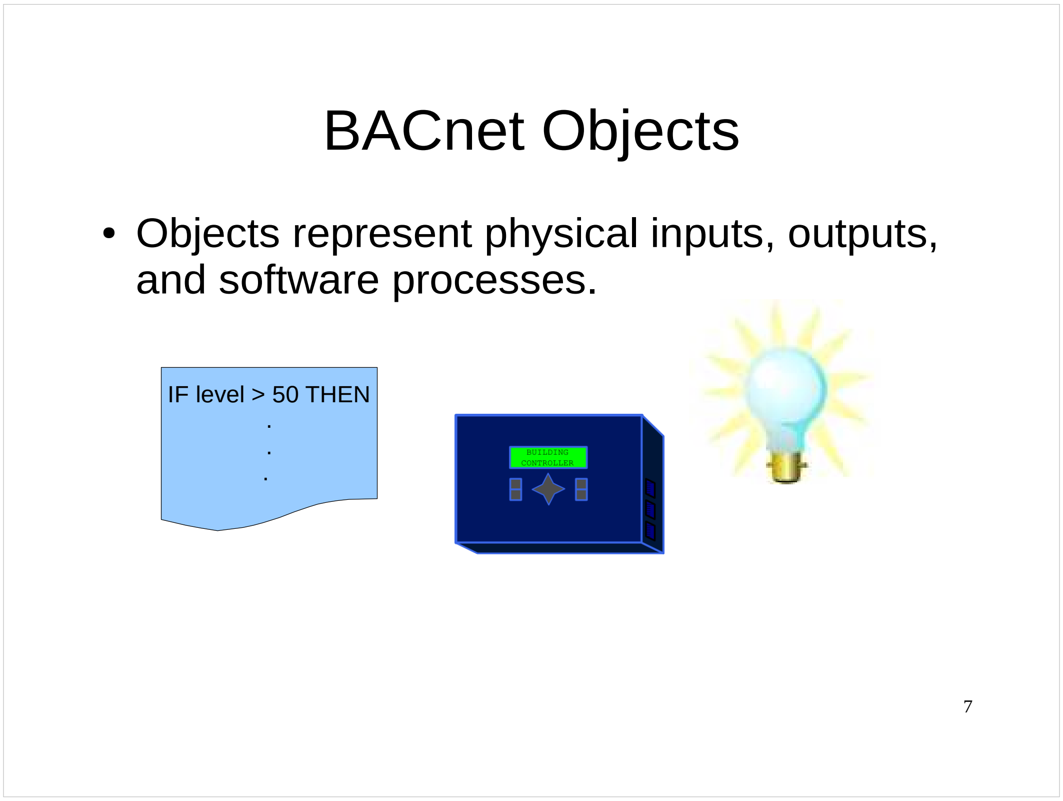
Fill: 0000

Poll Rate (Sec): 2

Reference String: System Pressure Transducer

Gateway Configuration tool from Control Solutions, Inc





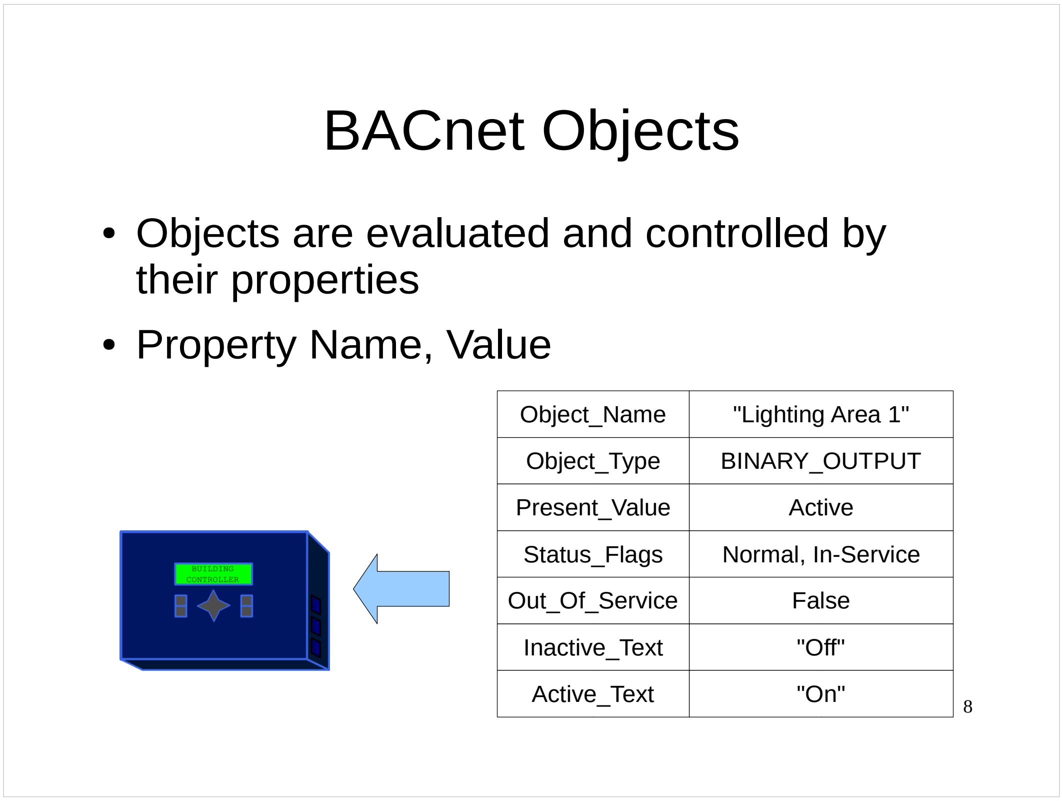
# BACnet Objects

- Objects represent physical inputs, outputs, and software processes.

MODELING CONTROL DEVICES AS A COLLECTION OF OBJECTS

The data structures used in a device to store information are a local matter. In order to exchange that information with another device using this protocol, there must be a "network-visible" representation of the information that is standardized. An object-oriented approach has been adopted to provide this network-visible representation. This clause defines a set of standard object types. These object types define an abstract data structure that provides a framework for building the application layer services. The application layer services are designed, in part, to access and manipulate the properties of these standard object types. Mapping the effect of these services to the real data structures used in the device is a local matter. The number of instances of a particular object type that a device will support is also a local matter.

All objects are referenced by their Object_Identifier property. Each object within a single BACnet Device shall have a unique value for the Object_Identifier property. When combined with the system-wide unique Object_Identifier of the BACnet Device, this provides a mechanism for referencing every object in the control system network. No object shall have an Object_Identifier with an instance number of 4194303. Object properties that contain BACnetObjectIdentifiers may use 4194303 to indicate that the property is not initialized.



# BACnet Objects

- Objects are evaluated and controlled by their properties
- Property Name, Value

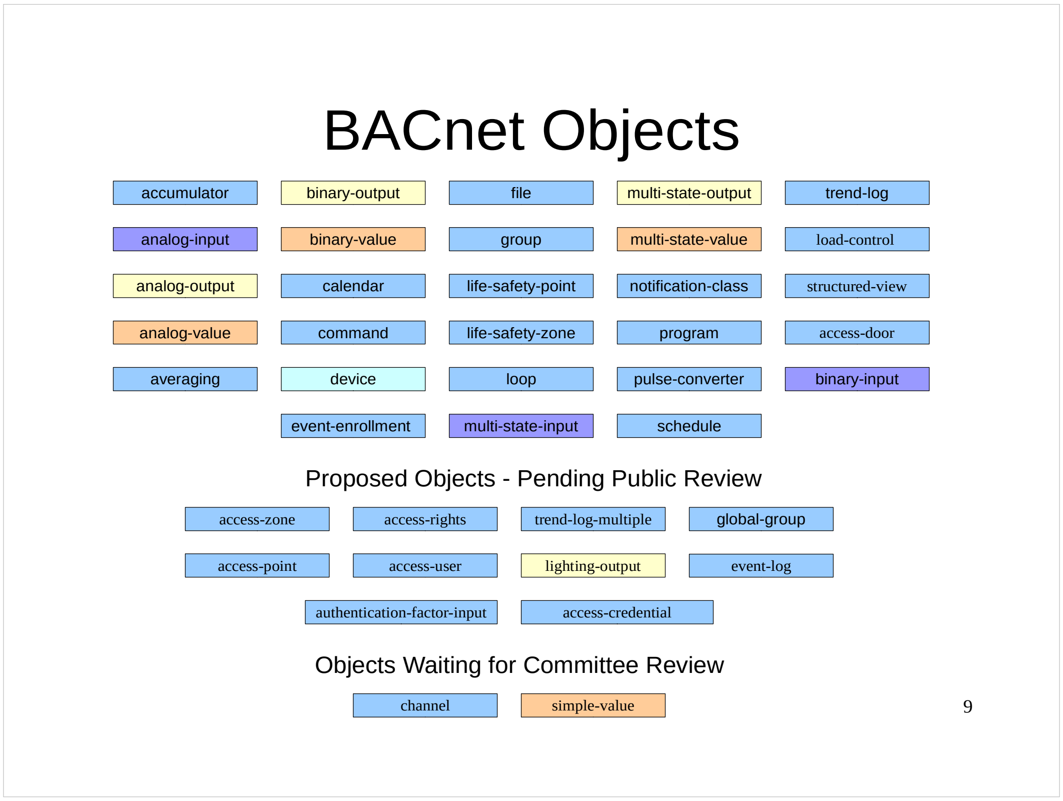
| | |
|---|---|
| Object_Name | "Lighting Area 1" |
| Object_Type | BINARY_OUTPUT |
| Present_Value | Active |
| Status_Flags | Normal, In-Service |
| Out_Of_Service | False |
| Inactive_Text | "Off" |
| Active_Text | "On" |

All objects in BACnet have properties that are used for information exchange.  Some properties are read-only.  Some properties are writable.

The graphic shows an example of a Binary Output object controlling a lighting load.  It shows a few properties and their values that might be in this object. Usually there are more properties in the object.

This object has a name ("Lighting Area 1") and an object type (BINARY_OUTPUT).  The Present_Value property indicates the the value at this moment.  The Status_Flags tell us if this object is functioning normally.  The Out_Of_Service indicates that the Present_Value is coupled to the physical output.

The Active_Text and Inactive_Text are values the help tell an operator the state of the present value.



# BACnet Objects

| | | | | |
|---|---|---|---|---|
| accumulator | binary-output | file | multi-state-output | trend-log |
| analog-input | binary-value | group | multi-state-value | load-control |
| analog-output | calendar | life-safety-point | notification-class | structured-view |
| analog-value | command | life-safety-zone | program | access-door |
| averaging | device | loop | pulse-converter | binary-input |
| | event-enrollment | multi-state-input | schedule | |

## Proposed Objects - Pending Public Review

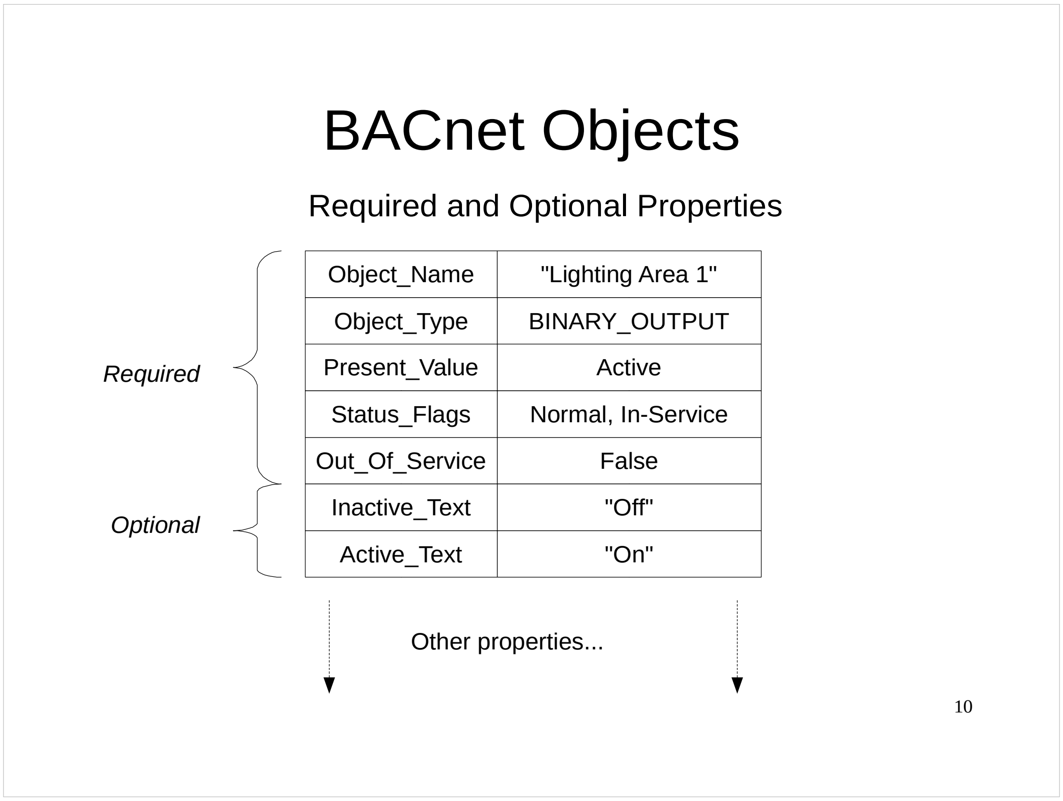
| | | | |
|---|---|---|---|
| access-zone | access-rights | trend-log-multiple | global-group |
| access-point | access-user | lighting-output | event-log |
| authentication-factor-input | access-credential | | |

## Objects Waiting for Committee Review

| | |
|---|---|
| channel | simple-value |

Although there are thousands of potentially useful object types in a building control system, BACnet defines a minimal number of Standard Object Types in detail. A BACnet standard object is an object whose behavior, in terms of the properties it provides and what they do, is defined in the BACnet standard.

BACnet devices are only required to implement a Device object. Other objects can be included as needed. BACnet has also defined a way for vendors to define their own non-standard objects.



# BACnet Objects

Required and Optional Properties

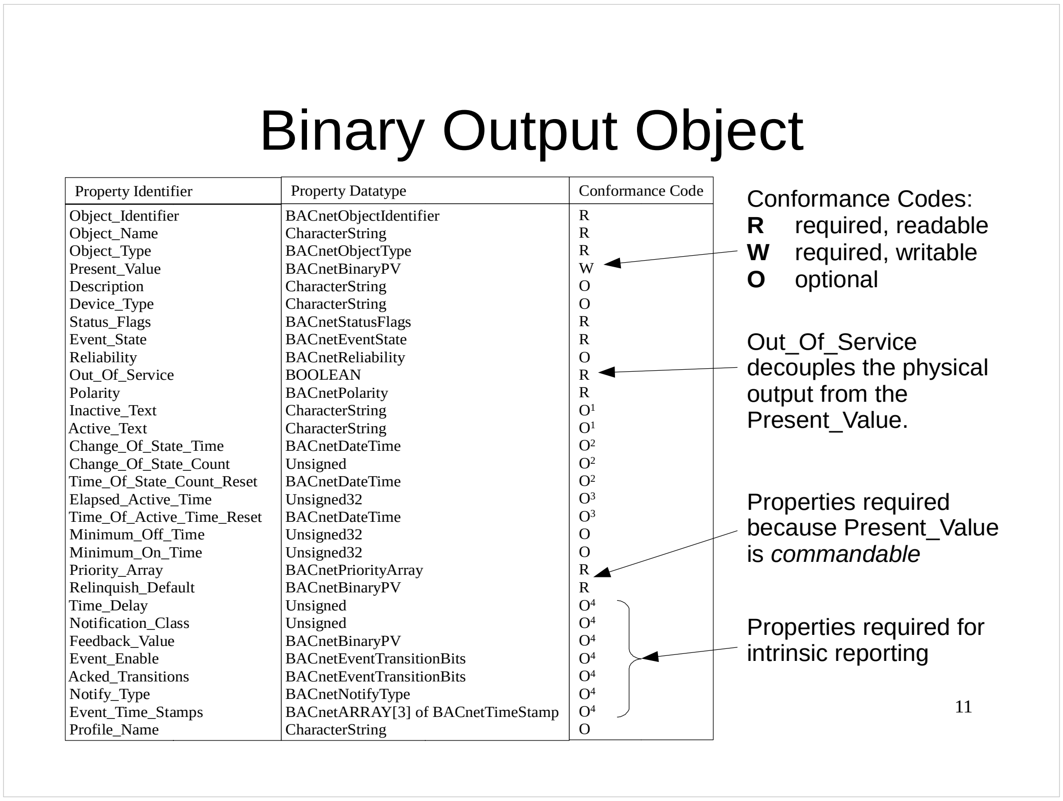
| | | |
|---|---|---|
| *Required* | Object_Name | "Lighting Area 1" |
| | Object_Type | BINARY_OUTPUT |
| | Present_Value | Active |
| | Status_Flags | Normal, In-Service |
| | Out_Of_Service | False |
| *Optional* | Inactive_Text | "Off" |
| | Active_Text | "On" |

Other properties...

Each standard object contains a set of *Required* properties, and a set of *Optional* properties. Required properties must be implemented.

The image above shows some required and optional properties of a Binary Output object.

BACnet has defined a way for vendors to define their own non-standard properties for use in standard objects.  Any number of non-standard properties can be defined with whatever behavior the vendor chooses.



# Binary Output Object

| Property Identifier | Property Datatype | Conformance Code |
|---|---|---|
| Object_Identifier | BACnetObjectIdentifier | R |
| Object_Name | CharacterString | R |
| Object_Type | BACnetObjectType | R |
| Present_Value | BACnetBinaryPV | W |
| Description | CharacterString | O |
| Device_Type | CharacterString | O |
| Status_Flags | BACnetStatusFlags | R |
| Event_State | BACnetEventState | R |
| Reliability | BACnetReliability | O |
| Out_Of_Service | BOOLEAN | R |
| Polarity | BACnetPolarity | R |
| Inactive_Text | CharacterString | O[1] |
| Active_Text | CharacterString | O[1] |
| Change_Of_State_Time | BACnetDateTime | O[2] |
| Change_Of_State_Count | Unsigned | O[2] |
| Time_Of_State_Count_Reset | BACnetDateTime | O[2] |
| Elapsed_Active_Time | Unsigned32 | O[3] |
| Time_Of_Active_Time_Reset | BACnetDateTime | O[3] |
| Minimum_Off_Time | Unsigned32 | O |
| Minimum_On_Time | Unsigned32 | O |
| Priority_Array | BACnetPriorityArray | R |
| Relinquish_Default | BACnetBinaryPV | R |
| Time_Delay | Unsigned | O[4] |
| Notification_Class | Unsigned | O[4] |
| Feedback_Value | BACnetBinaryPV | O[4] |
| Event_Enable | BACnetEventTransitionBits | O[4] |
| Acked_Transitions | BACnetEventTransitionBits | O[4] |
| Notify_Type | BACnetNotifyType | O[4] |
| Event_Time_Stamps | BACnetARRAY[3] of BACnetTimeStamp | O[4] |
| Profile_Name | CharacterString | O |

Conformance Codes:
**R** required, readable
**W** required, writable
**O** optional

Out_Of_Service decouples the physical output from the Present_Value.

Properties required because Present_Value is *commandable*

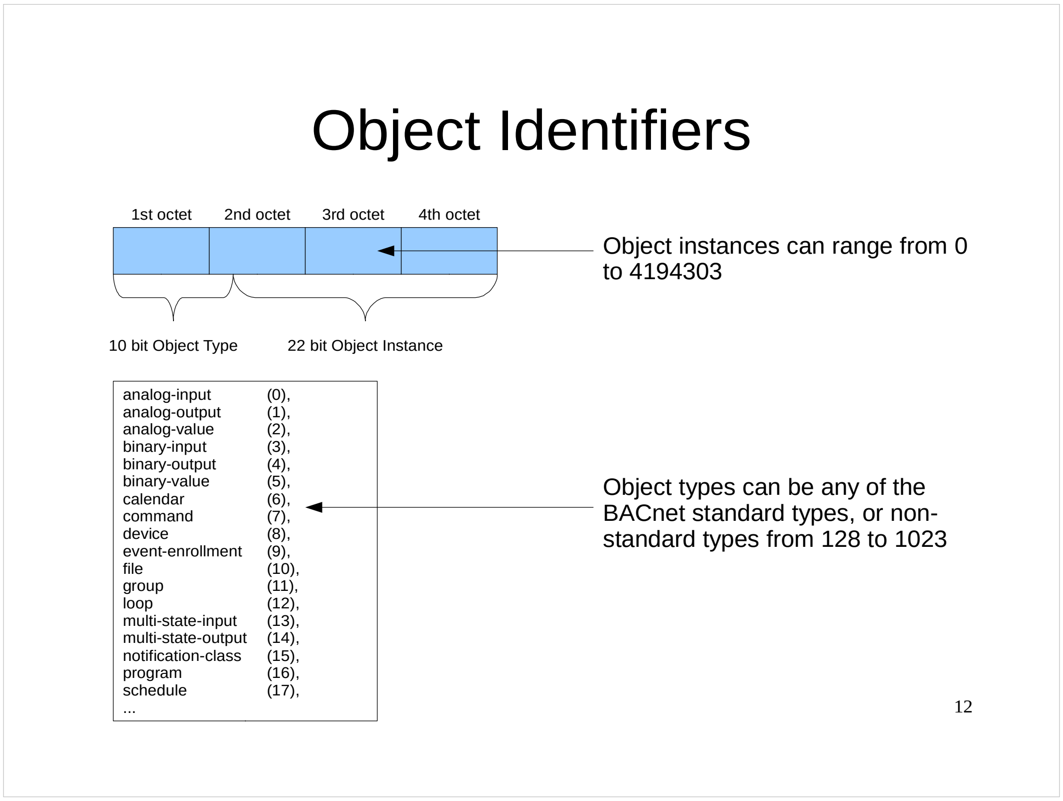
Properties required for intrinsic reporting

At the beginning of each standard object type specification that follows is a summary of the properties of the object type. The summary includes the property identifier, the datatype of the property, and one of the following : **O**, **R**, **W**

where **O** indicates that the property is optional,
**R** indicates that the property is required to be present and readable using BACnet services,
**W** indicates that the property is required to be present, readable,
and writable using BACnet services.

When a property is designated as required or **R**, this shall mean that the property is required to be present in all BACnet standard objects of that type. When a property is designated as optional or **O**, this shall mean that the property is not required to be present in all standard BACnet objects of that type. The value of **R** or **O** properties may be examined through the use of one or more of the ReadProperty services defined in this standard. Such **R** or **O** properties may also be writable at the implementor's option unless specifically prohibited in the text describing that particular standard object's property. When a property is designated as writable or **W**, this shall mean that the property is required to be present in all BACnet standard objects of that type and that the value of the property can be changed through the use of one or more of the WriteProperty services defined in this standard. The value of **W** properties may be examined through the use of one or more of the ReadProperty services defined in this standard. An **O** property, if present in a particular object, is not required to be writable unless specifically identified as such in the text describing that particular standard object's property.

In some devices, property values may be stored internally in a different form than indicated by the property datatype. For example, real numbers may be stored internally as integers. This may result in the situation where a property value is changed by one of the WriteProperty services but a subsequent read returns a slightly different value. This behavior is acceptable as long as a "best effort" is made to store the written value specified.



# Object Identifiers

1st octet 2nd octet 3rd octet 4th octet

Object instances can range from 0 to 4194303

10 bit Object Type

22 bit Object Instance

| | |
|---|---|
| analog-input | (0), |
| analog-output | (1), |
| analog-value | (2), |
| binary-input | (3), |
| binary-output | (4), |
| binary-value | (5), |
| calendar | (6), |
| command | (7), |
| device | (8), |
| event-enrollment | (9), |
| file | (10), |
| group | (11), |
| loop | (12), |
| multi-state-input | (13), |
| multi-state-output | (14), |
| notification-class | (15), |
| program | (16), |
| schedule | (17), |
| ... | |

Object types can be any of the BACnet standard types, or non-standard types from 128 to 1023

A BACnet Object Identifier value shall consist of two components:

(1) A 10-bit object type, representing the BACnetObjectType of the object, with bit 9 the most significant bit and bit 0 the least significant. For objects defined in this standard, the value for this field shall be determined by the BACnetObjectType enumeration in Clause 21.

(2) A 22-bit object instance number, with bit 21 the most significant bit and bit 0 the least significant.

A "BACnetDevice" is any device, real or virtual, that supports digital communication using the BACnet protocol. Each BACnet Device contains exactly one Device object, as defined in 12.11. A BACnet Device is uniquely located by an NSAP, which consists of a network number and a MAC address.

Device object identifiers are unique "internetwork wide." If this is not the case, the Who-Is service will produce ambiguous results.

# Property Identifiers

| | |
|---|---|
| accepted-modes | (175), |
| acked-transitions | (0), |
| ack-required | (1), |
| action | (2), |
| action-text | (3), |
| active-text | (4), |
| active-vt-sessions | (5), |
| active-cov-subscriptions | (152), |
| adjust-value | (176), |
| alarm-value | (6), |
| alarm-values | (7), |
| all | (8), |
| all-writes-successful | (9), |
| apdu-segment-timeout | (10), |
| apdu-timeout | (11), |
| application-software-version | (12), |
| archive | (13), |
| attempted-samples | (124), |
| auto-slave-discovery | (169), |
| average-value | (125), |
| backup-failure-timeout | (153), |
| bias | (14), |
| buffer-size | (126), |
| change-of-state-count | (15), |
| change-of-state-time | (16), |
| client-cov-increment | (127), |
| configuration-files | (154), |
| controlled-variable-reference | (19), |
| controlled-variable-units | (20), |
| controlled-variable-value | (21), |
| count | (177), |
| count-before-change | (178), |
| count-change-time | (179), |
| cov-increment | (22), |
| cov-period | (180), |
| cov-resubscription-interval | (128), |
| database-revision | (155), |
| date-list | (23), |
| daylight-savings-status | (24), |
| deadband | (25), |
| derivative-constant | (26), |
| derivative-constant-units | (27), |
| description | (28), |
| description-of-halt | (29), |
| device-address-binding | (30), |
| device-type | (31), |
| direct-reading | (156), |
| effective-period | (32), |
| elapsed-active-time | (33), |
| error-limit | (34), |
| event-enable | (35), |
| event-state | (36), |
| ... | |

Property identifiers are specified for each standard object.

Non-standard property identifiers in the range of 512 to 4194303 allows each vendor to define their own properties.

# BACnet Services

- Service == Message
- Services allow Devices to do something
- Must implement at least ReadProperty

# BACnet Confirmed Services

- Confirmed == Unicast, Acknowledged

AcknowledgeAlarm

COV-Notification

EventNotification

GetAlarmSummary

GetEnrollmentSummary

SubscribeCOV

SubscribeCOVProperty

LifeSafetyOperation

AtomicReadFile

AtomicWriteFile

AddListElement

RemoveListElement

CreateObject

DeleteObject

ReadProperty

ReadPropertyMultiple

ReadRange

WriteProperty

WritePropertyMultiple

PrivateTransfer

TextMessage

ReinitializeDevice

VTOpen

VTClose

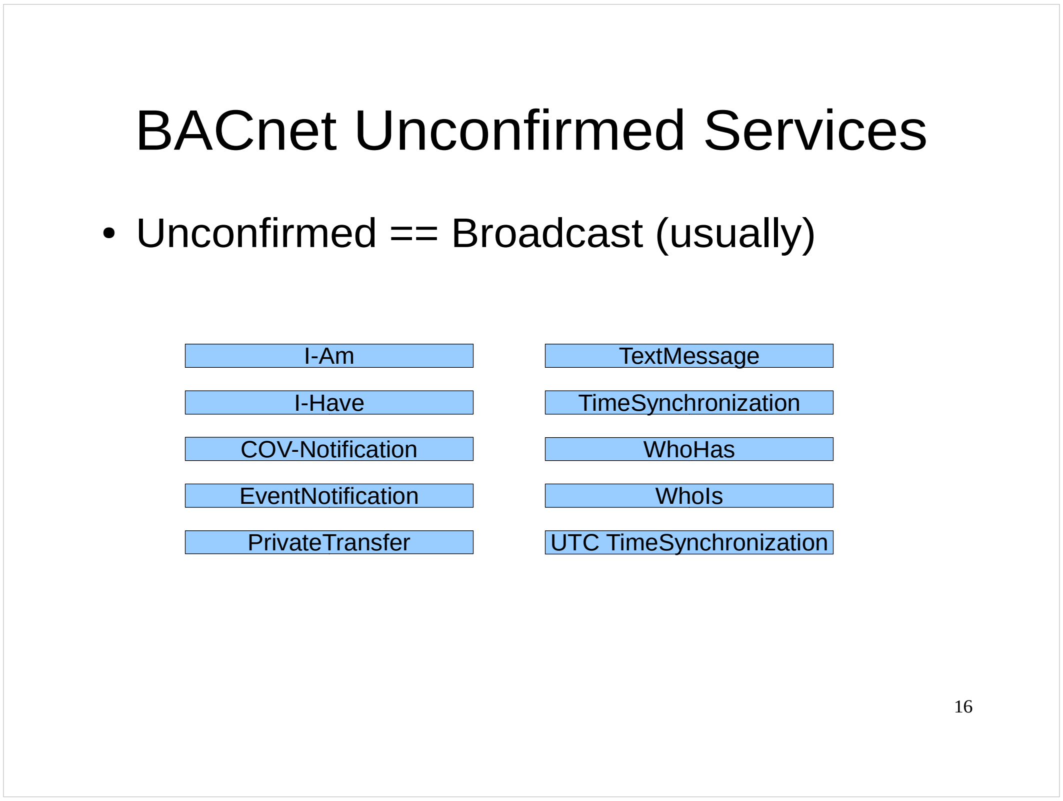
VTData

Authenticate

ReqestKey

ReadPropertyConditional

DeviceCommunicationControl



# BACnet Unconfirmed Services

- Unconfirmed == Broadcast (usually)

| | |
|---|---|
| I-Am | TextMessage |
| I-Have | TimeSynchronization |
| COV-Notification | WhoHas |
| EventNotification | WhoIs |
| PrivateTransfer | UTC TimeSynchronization |