

# Network Communications for Buildings



**CONTEMPORARY** CONTROLS®

## About Contemporary Controls

Contemporary Controls serves the building automation industry with products based upon open standards such as BACnet, Modbus and Ethernet. Our customers are systems integrators, contractors, mechanical and controls OEMs seeking simple and reliable networking and control products from a dependable source. BASautomation® – Building on BACnet provides routing, gateway and control solutions compatible with an internationally recognized building automation standard. CTRLink® – Ethernet Built for Buildings consists of unmanaged and managed switches, media converters, and wired and wireless IP routers. These products are designed for unattended operation in environments not conducive to office grade equipment. With headquarters based in the US, we have operations in the UK, Germany and China with self-manufacturing in the US and China.

## Celebrating 40 Years!

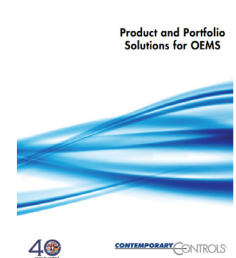
George Thomas established Contemporary Controls in 1975 as a system integration and consulting business focusing on microcomputer and PLC applications. As microprocessor bus-boards became more popular, in 1983 the company changed direction to become a hardware manufacturer by developing a series of STD-BUS microcomputer modules and became active in the STD-BUS Manufacturers' Group (STDMG). One of those modules developed was an ARCNET® adapter. As ARCNET acceptance increased, Contemporary Controls was the catalyst for creating the ARCNET Trade Association (ATA) in 1987 to better publicize this technology to the automation world. With most of the major building automation companies using ARCNET, Contemporary Controls became a supplier to all the OEMs that incorporated ARCNET in their systems.

With the emergence of fieldbuses such as Controller Area Network (CAN), CANopen, and DeviceNet, Contemporary Controls developed bus-board adapters and network analyzers in support of these technologies. In 1998, the EXTEND-A-BUS® was introduced as a convenient way of extending CAN networks in the field.

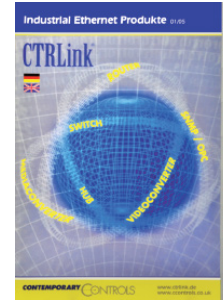
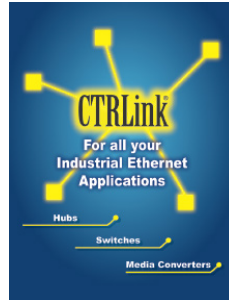
The rapid acceptance of Industrial Ethernet as a fieldbus replacement created the need for the CTRLink® family of Industrial Ethernet products intended for rugged applications. The CTRLink product line won the 2002 Editor's Choice award from Control Engineering. In 2003, Contemporary Controls created the Industrial Ethernet University as a way to educate the public on this technology.

In 2008, the company introduced the BAS Router – a BACnet/IP to BACnet MS/TP router and became active in the BACnet community. In 2012, Contemporary Controls received registered trademark status for its BASautomation line of building automation controllers, routers and gateways. The tagline for BASautomation is **Building on BACnet**.

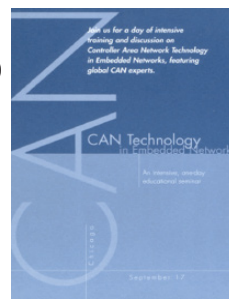
2015



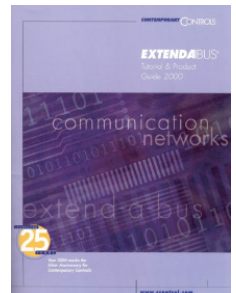
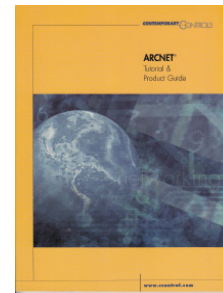
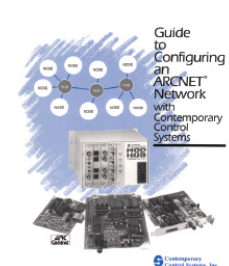
2010



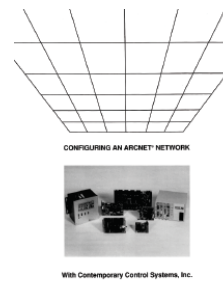
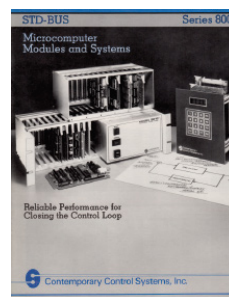
2005



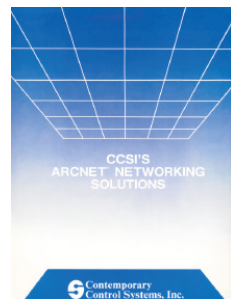
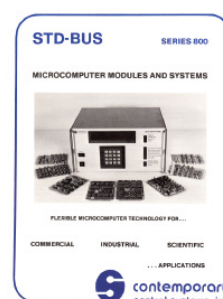
2000



1995



1990



1985

1980

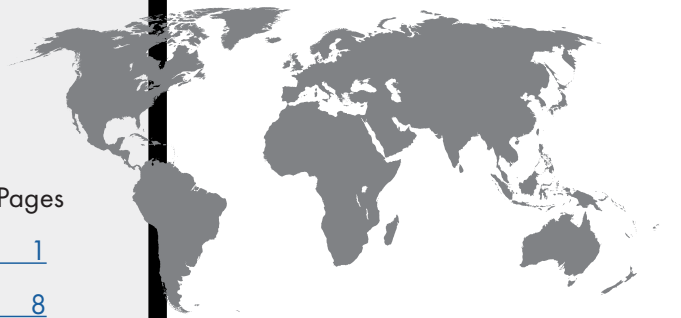
1975

# Network Communications for Buildings

[Welcome](#)

[Acknowledgments](#)

Technology	Pages
<a href="#">EIA-485 Physical Layer</a>	1
<a href="#">Shared Ethernet</a>	8
<a href="#">Switched Ethernet</a>	16
<a href="#">Fast and Faster Ethernet</a>	23
<a href="#">Ethernet with Fiber Optic Cabling</a>	31
<a href="#">Power over Ethernet (PoE)</a>	40
<a href="#">Internet Protocol (IP)</a>	46
<a href="#">Transmission Control Protocol (TCP)</a>	53
<a href="#">Subnetting IP Networks</a>	58
<a href="#">Simple Network Management Protocol (SNMP)</a>	63
<a href="#">Virtual Local Area Network (VLAN)</a>	69
<a href="#">Spanning Tree Protocol (STP)</a>	76
<a href="#">Modbus Protocol</a>	82
<a href="#">Modbus Serial and Modbus TCP</a>	88
<a href="#">Modeling a BACnet Physical Device</a>	93
<a href="#">Achieving BACnet Compliance</a>	99
<a href="#">BACnet Routing</a>	104
<b>Applications</b>	
<a href="#">Using Managed Switches</a>	110
<a href="#">Using IP Routers</a>	113
<b>Troubleshooting</b>	
<a href="#">Using Wireshark for Network Troubleshooting</a>	121
<a href="#">Using the BACnet Discovery Tool (BDT)</a>	127
<b>Reference</b>	
<a href="#">Glossary of Ethernet Terms</a>	131





# Introduction

## Welcome

The year 2015 marks our 40th anniversary as a company and as we reflect on this milestone, we cannot help but marvel at the technological changes that have impacted both the automation systems we produce as well as how companies are now doing business. We have witnessed the migration from standalone controllers to fieldbus-networked controllers to IP-based network controllers. The open-systems movement has helped smaller companies participate in markets once dominated by larger companies with their proprietary systems. The building automation market is a great example with technologies such as BACnet, Modbus and Ethernet for all to use.

As a company, we have always tried to educate the industry on best practices by publishing application notes, PowerPoint presentations, tutorials and videos. We believe all companies benefit if we effectively educate the industry on the best use of open-system technologies. That is why we put together this book – Network Communications for Buildings – from a collection of prior articles from our Extension and Essentials that were supplements to our print newsletters beginning in 1999. All the articles have been updated to the latest practice so now the best material on network communications can be found in one concise book. The intention of the book is to present the material in a commercial-free format in keeping with the mission of the open-systems movement. If you would like to learn more about topics in the book, please view the available material under the Support tab of our website at [www.ccontrols.com](http://www.ccontrols.com).

George Thomas  
President

## Acknowledgments

I would like to first acknowledge the technical and marketing staff at Contemporary Controls who contributed to this book.

Bennet Levine  
Mingshu Wang  
Harpartap Parmar  
Rhiannon LaPointe  
George Karones  
Joe Stasiek  
Kathleen Thomas  
Bill Greer  
Kirk Clousson

In addition, I would like to thank my colleagues on the BACnet International Education Committee and all others in the BACnet community who devote their time and effort to making BACnet successful.



## Ten Recommendations When Implementing EIA-485

1. Recognize that EIA-485 is only a physical layer standard and connectivity between two machines each with an EIA-485 interface is not ensured.
2. Since EIA-485 is basically a specification for the driver, receiver and transceiver chips, the manufacturer of the equipment needs to specify cabling, grounding, termination, fail-safe bias and connectors.
3. Data rate and segment lengths are not addressed in the standard and therefore, must be specified manufacturer. A high-speed EIA-485 design could be quite different from a low speed design.
4. EIA-485 is intended to be cabled as a linear bus with daisy-chain connections. Stubs may or may not be allowed. Do not cable a star topology.
5. Carefully review grounding practice. EIA-485 transceiver damage is usually due to excessive common mode voltage caused by unequal ground potentials at the various devices. Sometimes a third-wire ground connection must be carried to all nodes to ensure that the common mode voltage remain within limits.
6. Device protection circuitry can certainly minimize device failures, however, verify that high data rates can be maintained with protection applied.
7. Although optically isolated EIA-485 will not protect the transceivers themselves, it will provide a level of protection to the attached equipment. Be sure to run a common wire between all optically-isolated transceivers.
8. The design of the EIA-485 repeaters is tricky. Use only those repeaters recommended by the manufacturer.
9. Be careful when applying termination and fail-safe bias so as to not introduce excessive loading. Termination is only applied at each end of the network while bias is applied per the manufacturer's recommendation.
10. EIA-485 can be quite an effective network as long as it is applied properly.

### Our recommendations.

- If the devices are in the same control panel, use DC coupled EIA-485 and individual earth (chassis) connections for ground reference.
- If the devices are in the same control panels within the same building use optically-coupled EIA-485 and a separate reference ground wire.
- If the devices are in separate panels in different buildings, use fiber optics if at all possible.

## Introduction

One of the more popular technologies for interconnecting devices on a network is EIA-485. Known throughout industry as RS-485, the proper title for the standard is TIA/EIA-485-A "Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems." The EIA-485 standard is misunderstood to mean more than what it defines. According to the standard, it specifies the characteristics of the generators and receivers used in a digital multipoint system. It does not specify other characteristics such as signal quality, timing, protocol, pin assignments, power supply voltages or operating temperature range. A multipoint system consists of two or more generators and one or more receivers. A generator is the same as a transmitter and since two or more transmitters can exist on the same electrical bus, EIA-485 is suitable for multimaster systems.

The standard itself is very short consisting of only 17 pages. Actually more guidance is available from its sister publication TSB89 "Application Guidelines for TIA/EIA-485-A." An EIA-485 bus usually consists of two or more communication controllers each powered by a separate power source. At a minimum, a single shielded or unshielded twisted-pair cable interconnects the various controllers in a daisy-chain fashion. In some instances, a short stub is allowed; however, higher speed networks usually do not allow stubs. A star topology is definitely not recommended. Termination is usually applied to the ends of the network.

EIA-485 is basically a specification for the drivers, receivers and transceivers attached to the network. Therefore, parameters such as unit loads, output drive, short circuit current and common mode voltage are specified. Basically a driver must be able to source at least 1.5 volts differentially into 60 ohms (two 120 ohm terminators in parallel along with 32 unit loads) under a common mode voltage range of  $-7$  to  $+12$  Vdc. Data rates are not specified and there are a wide range of devices that conform to the standard but are intended either for high speed (up to 50 Mbps) or low speed (skew rate limited). So do not assume that all driver, receiver and transceiver chips are all the same. Some receivers and transceivers have higher input impedance thereby representing less than one unit load to the driver.

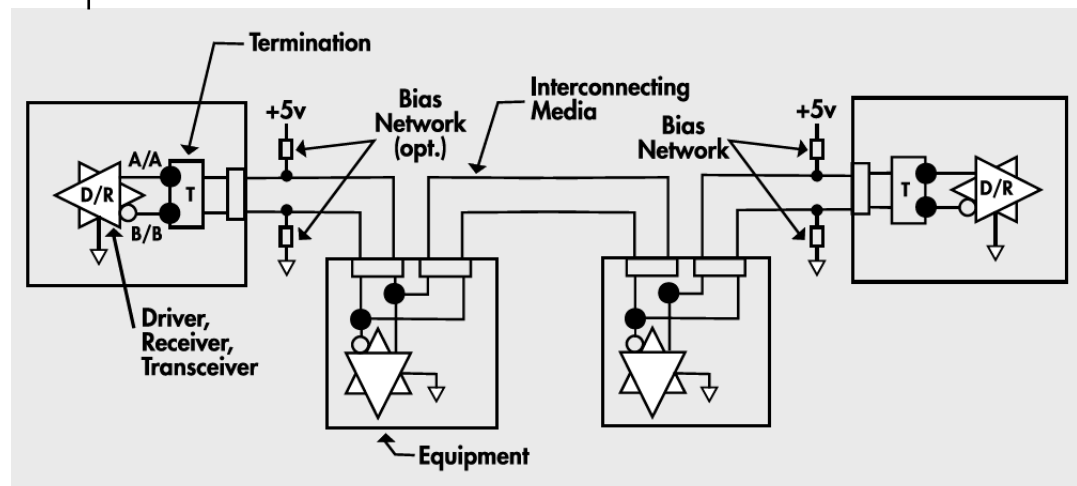


Figure 1—EIA-485 networks are usually configured in a daisy-chain fashion with termination at each end.

APPLICATION
PRESENTATION
SESSION
TRANSPORT
NETWORK
DATA LINK
PHYSICAL

**Figure 2—In terms of the OSI model, EIA-485 only addresses the lowest layer.**

### Physical Layer Standard

In terms of the Open Systems Interconnection Reference Model (OSI), EIA-485 only defines the lowest layer— the physical layer. It is used by Allen Bradley's DH-485, Profibus, BACnet's Master/Slave Token Passing option and ARCNET as well. Each of these implementations is different demonstrating that EIA-485 is not an all-encompassing standard. There are several key topics that must be considered when deploying EIA-485 networks such as termination, fail-safe bias, connectors, grounding, cabling and repeaters.

### Termination

Terminating a data cable with a value equal to its characteristic impedance reduces reflections that could cause data errors. However, if the data rate is low or the cables are short, termination may be unnecessary. As data rates increase, termination becomes important. Since any device on the bus can transmit, it is probable that a node within the middle of the bus will transmit requiring that termination be applied to both ends of the bus segment. National Semiconductor offers a highly in-depth discussion on termination in application note AN-903 and offers several alternatives. The most popular approach is DC termination although this approach results in higher power dissipation.

Resistive terminators typically have values of 120 to 130 ohms although twisted-pair cable impedances can be as low as 100 ohms. An 100 ohm terminating resistor is too low for the EIA-485 drivers. A value closely matching the cable impedance must be applied at some convenient location closest to the ends of the cable segment as possible. One possibility is to provide the resistor within a node with a jumper to disable this option if termination is not required. The problem with this approach is that each node will be configured differently since only two nodes should have terminators. Care must be exercised to ensure that only the proper modules have termination invoked in order not to cause excessive bus loading.

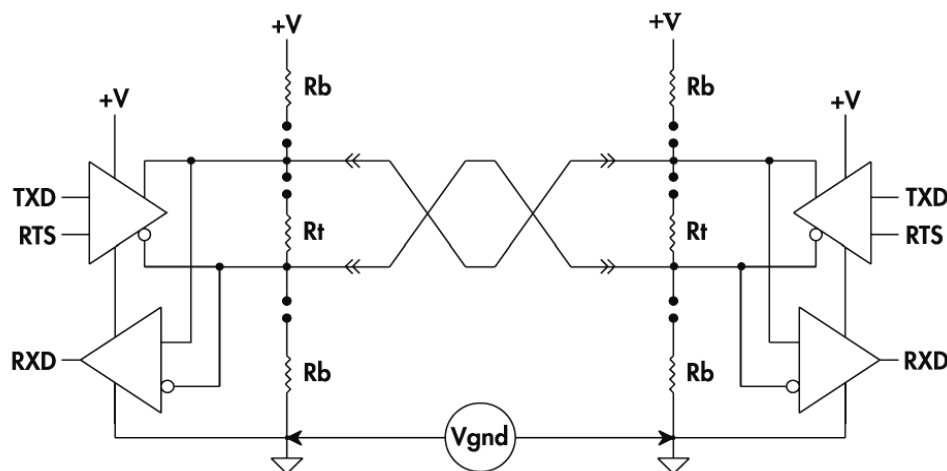
Another approach is to use external termination outside the node. Profibus uses this approach. Both terminating and bias resistors are located in the shell of a DB9 connector. DIP switches within the shell are used to disable this feature. The advantage of this approach is that all nodes on the network are the same while connectors are used to properly configure the network.

Allen-Bradley's DH-485 uses a slightly different approach. Although terminating resistors are located within all nodes, an external jumper applied at the connector invokes termination. Again, this keeps the configuration in the connector and not within the node.

### Fail-Safe Bias

EIA-485 is a multipoint standard where individual devices transmit and receive data sharing a common two-wire medium. The opportunities for collisions (two transmitters on at the same time) are immense and a method of medium access control (MAC) is required. The 485 standard does not provide a bus arbitration scheme since this is not a requirement of the physical layer but is a requirement of the data link layer. With a master/slave protocol such as Profibus DP, bus arbitration with a single master and multiple slaves may not be a problem since the master directs all the traffic.

Slaves are always listening and only respond to the master's request thereby avoiding collisions. During this time, the bus will "float" enabling noise to falsely trigger one of the bus receivers. This can occur because the receiver's output is undefined when the receiver's input voltage is less than 200 mv which could happen when the bus floats. To ensure that the bus assumes a defined state when all transmitters are off, fail-safe bias must be applied.



**Figure 3 —Disconnects are provided for fail-safe bias and termination.**

National Semiconductor's AN-847 application note fully discusses the need for fail-safe bias and recommends the proper biasing resistors needed to ensure that the bus differential voltage will not dip below 200 mv when idle. This note recommends a pull-up resistor to +5 volts attached to one signal line and a pull-down resistor to ground attached to the other. In conjunction with an end-of-line terminator, a voltage divider is created which impresses a bias across the line that exceeds 200 mv. Therefore, the receivers are biased in the mark (off,

logic 1) state when the network is idle or when the transmitter sends a logic 1. When a transmitter sends a logic 0, the line will revert to "space" (on, logic 0).

Bias can be applied at any point on the bus segment but it is not necessary to lump the bias at only one point. The bias can be distributed throughout the segment with each node providing a portion of the bias. The advantage of this approach is that there is no need to provide an external bias network and power supplies. The problem with this approach is that the amount of bias developed depends upon the number of nodes on the bus. If too few nodes are connected, insufficient bias may result. Too much bias can result if too many nodes are connected causing excessive loading. If it is desired to supply lumped bias, a source for +5 volt power needs to be found which may be awkward to arrange. The other approach is to provide the complete bias requirement within each node while providing disconnecting jumpers on the node. In this way, only one node needs to be strapped for bias so record keeping must be good to ensure that the location of this node is known when a replacement is necessary.

Profibus nodes source +5 volt power to the connector allowing for the bias resistors in the shell of the mating connectors. DIP switches within the shell disable the bias. Profibus uses 390 ohms for the pull-up and pull-down resistors and 150 ohms for termination. This provides about 800 mv of bias and a Thevenin equivalent termination resistance of 125 ohms. In TSB89 the resistance values are 620 and 130 ohms respectively which yields 475 mv of bias and a Thevenin equivalent termination resistance of 118 ohms. Either approach is adequate in terminating 120 ohm cable.



However, it must be remembered that a termination resistor exists at the other end of the cable. The distant terminator will load the bias network thereby reducing the bias voltage by a factor of two. So instead of having 475 mv of bias, the resulting bias will be only 240 mv which is still above the 200 mv limit. This analysis assumes there is no resistance in the cable.

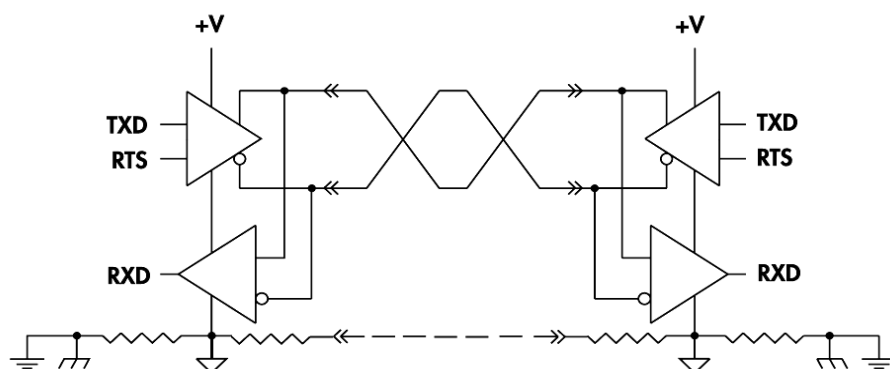
If the required bias is applied at two points, then cabling rules are simplified. Apply bias and termination only at the end two nodes by using a combination bias/termination resistor network. There is another benefit to applying bias at end nodes. If a lumped bias was applied to one end of a long cable with termination at each end, a voltage divider is formed with the DC resistance of the cable. For 24 AWG cable, the resistance is 24 ohms/1000 ft. Since there are two wires in the cable, the cable resistance is effectively 48 ohms/1000 ft. A 2500 foot cable would have the same DC resistance as the end terminator thereby reducing the effective bias at the end node by a factor of two. This could put the distant nodes in an unreliable state compared to devices closer to the source of bias. A way of correcting this is to increase the cable wire gauge to reduce resistance or apply an equal amount of bias at the distant end of the cable effectively eliminating the voltage drop due to cable resistance.

### Connectors

With coaxial and fiber optic cabling, specifying connectors is easy since there is common practice in the industry. However, with twisted-pair cabling there are many options. Since the EIA-485 standard does not address connectors, trade associations or manufacturers must do so. From practice there seems to be three popular approaches. The traditional approach is to use a four-pin, six position RJ-11 or eight position RJ-45 providing plenty of pins for signal and ground reference. Some RJ style connectors are shielded.

Another approach to connectorization is to use removable open style screw connectors. DH-485 uses a six-position connector providing all the necessary connections including termination.

A popular connector is the DB9 connector that is used with the Profibus standard. With nine pins, it is easy to accommodate signals, logic ground, shield connection and power pins. DB9 housings are also available with metal shrouds for better EMC performance and the housing has built-in bias and termination provisions. This connector, however, tends to be pricey.



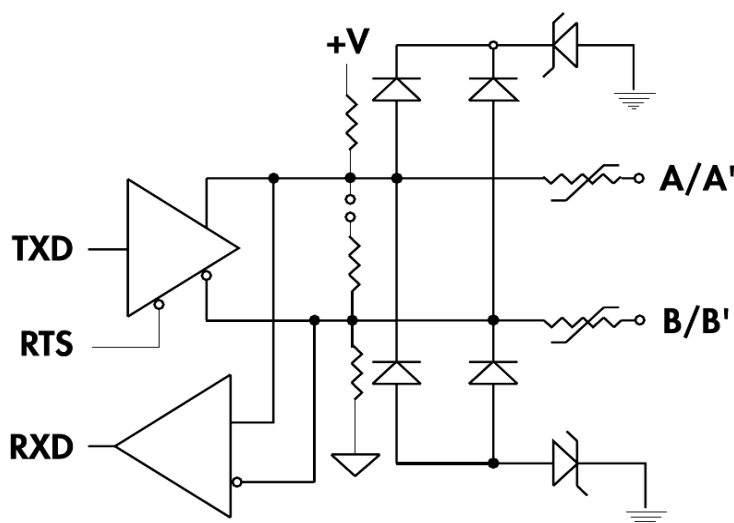
**Figure 4** —If a third wire connection is used, resistors must be used to limit circulating ground current.

## Grounding

Is EIA-485 a two wire or a three-wire system? It is most definitely a three-wire system. The standard clearly states that generators and receivers require a return path between circuit grounds at each end of a connection. This return path could be an actual wire in the cable connecting each of the logic grounds together or earth can provide the return path by having each logic ground returned to earth. Using the latter approach, a single pair twisted cable can be used. If the third wire is to be used, the standard states that the connection between logic ground and the third wire contain some resistance to limit circulating currents when other ground connections are provided for safety. This resistor could be between logic ground and frame (frame is tied to earth) or it can be between the logic ground and third connection. The standard uses 100 ohms as an example for both situations.

There is much confusion and misunderstanding of the third wire requirement and difficulty in even finding a third wire. If the logic grounds of the transceivers are tied to earth and a third wire is used, there is almost a guarantee of a ground loop current, which may or may not induce excessive noise that could disrupt data transmissions. The third wire will also be the path for fault currents, which could be significant when the two ground potentials are different due to a significant electrical event. Still the third wire helps to ensure that the common mode requirements (-7 to +12 volts) of the transceivers are maintained. Excessive common mode voltage is the most common reason for transceiver failure.

## Protection Circuitry



**Figure 5 —Protection circuitry is usually referenced to earth.**

To protect EIA-485 transceivers from excessive common mode voltages, diode protection circuits are used which are referenced to earth or logic ground. Usually protection is provided from each data line to earth and it is necessary to protect against either a positive or negative occurrence which doubles the protection circuitry. The more robust the protection, the more the capacitance which limits the data rate.

It is quite possible that systems will refuse to work at the desired data rate due to the increased capacitance. Some protection is afforded when the protection circuit consists of a bulky transient voltage suppressor in series with a diode. The capacitive divider created by the diode and suppressor in series yields a capacitance that is less than the diode itself thereby lessening the impact of

protection on data rate. Of course, protection is possibly required at each node increasing the likelihood that either data rate or distance will be compromised by adding protection circuitry.

### Optical Isolation

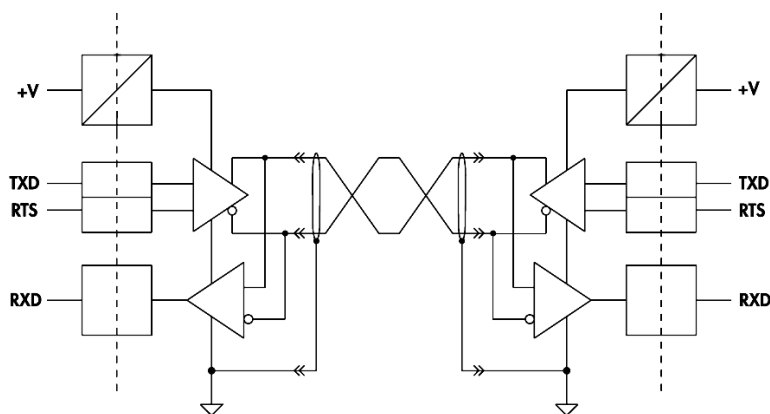
Optically isolated transceivers can be treated like DC coupled transceivers. The termination and fail-safe bias issues are the same, so what is isolated? What is isolated are signals TXD, RTS and RXD. Therefore, three opto-isolators are required. The two used for data should be high speed while the transceiver enable isolator can be slower. A DC-converter needs to be provided and its breakdown voltage will probably be the limiting factor in terms of isolation. The optically isolated transceiver design is the most expensive approach and it does not guarantee that the EIA-485 transceivers, which are connected directly to the cable, will survive abuse from severe electrical transients. Chances are, however, that the damage will stop at the isolators and not involve the equipment attached to the node. Optically-isolated EIA-485 forces a third wire connection since the transceivers must have a return path. However, this time there is no ground loop since logic ground of the transceivers is not connected to the earth. Where do we get the third wire? Many times the shield is used and not everyone is in agreement on the wisdom of this approach. Since such a small current is going to flow, it seems a reasonable approach. DH-485 uses a two pair cable with one wire of one pair dedicated as the common ground. A shield covers the two pairs and is only grounded at one point.

### Cabling

One of the more critical decisions to make is the selection of cable. There are many choices of cable and people incorrectly assume that any 24 AWG telephone cable will do. Cable selection depends on several factors including data rate, signal encoding and distance desired. Cables attenuate the transmitted signal and introduce distortion of the signal waveform itself. Additional distortion occurs by the way receivers are biased. Jitter can occur when the receiver attempts to recover the distorted data. Intersymbol interference results when a new signal arrives at the receiver before the last signal reached its final value.

Therefore, the two successive symbols interfere with one another resulting in a time shift in the data recovery which is called jitter. National Semiconductor discusses this phenomenon in AN-808. Some jitter is usually acceptable, however, if it is excessive, the only solution is to obtain better cable, reduce the modulation rate or reduce the distance.

Other cabling issues include the wire size of the conductors, the need for shielding, the presence of a third wire ground and the type of insulation. It is best to only use the manufacturer's recommended cable and not substitute without consulting with the manufacturer first.



**Figure 6** — Sometimes the shield is used as the third wire with an optically-isolated interface.



## Repeaters

EIA-485 segments can be extended using active hubs or repeaters; however, care needs to be exercised in the selection of repeaters. Since only two wires are used, the direction of signal flow through the repeater must change dynamically. Usually a direction control line is provided to the repeater to control flow or the repeater automatically senses traffic and adjusts accordingly. Do not assume any low-cost EIA-485 repeater will work at all speeds. Repeaters that sense line activity could be fooled in believing data flow from one direction has stopped when in fact it was a series of logic "1"s or "0"s without state transitions. This is especially true of RZ or NRZ encoded data. Repeaters that operate with a knowledge of the data link protocol are much more reliable than off-the-shelf solutions.

## Summary

With some attention to detail, EIA-485 can be an effective physical layer technology.

## References

*Telecommunications Industry Association*, Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems, TIA/EIA-485-A, March 1998.

*Telecommunications Industry Association*, Application Guidelines for TIA/EIA-485-A, TSB89, June 1998.

*Rockwell Software*, *SLC500 Modular Hardware Style Installation and Operation Manual*, SLC500CD, July 1998.

*Profibus Trade Organization*, Profibus Standard-Part 1, DIN 19245, 1993.

*ASHRAE*, *A Data Communications Protocol for Building Automation and Control Networks*, ANSI/ASHRAE 135-1995, Dec. 1995.

*FieldComms USA*, ARCNET's Already Flexible Physical Layer Enhanced with Several EIA-485 Variants, George Thomas, June 1997.

*National Semiconductor*, *A Comparison of Differential Termination Techniques*, AN-903, 1993.

*National Semiconductor*, *Fail-safe Biasing of Differential Buses*, AN-847, July 1992.

*National Semiconductor*, *Long Transmission Lines and Data Signal Quality*, AN-808, March 1992.

## Introduction

There has been much discussion regarding the applicability of using Ethernet at various levels of the control hierarchy. Since Ethernet is so prevalent in the office and frequently used as the enterprise network for high-end controllers, it would seem to be a natural to use Ethernet at the control level or even at the device level as proposed by some in our industry. The arguments for its use include low cost, good connectivity and simple migration to higher speed networks. Over the years Ethernet has enjoyed a remarkable evolution in terms of speed, performance and network size but it all began with a basic technology called Shared Ethernet which continues to exist today.

## What is Shared Ethernet?

Shared Ethernet describes a network technology where all stations reside within one collision domain and communicate to one another over a half-duplex path. It is based on a 2.94 Mbps version that came out of Xerox's Palo Alto Research Center (PARC) in the early 70s. In 1980, Digital Equipment Corporation (DEC), Intel and Xerox published the DIX V1.0 standard which boosted the speed of Ethernet to 10 Mbps while maintaining Ethernet's thick trunk cabling scheme. In 1982 the DIX V2.0 standard was released which is now commonly referred to as Ethernet II. Xerox then relinquished its trademark.

At the time of the first DIX standard, the Institute of Electrical and Electronic Engineers (IEEE) was attempting to develop open network standards through the 802 committee. In 1985, the IEEE 802.3 committee published "IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications." This technology is called 802.3 CSMA/CD and not Ethernet; however, it is frequently referred to as Ethernet even though the frame definition differs from DIX V2.0. Although 802.3 and DIX frames can coexist on the same cable, interoperability is not assured. Therefore, when discussing "Ethernet," it is necessary to clarify 802.3 frames or DIX V2.0 frames.

To further confuse issues, standard Ethernet sometimes means an attached protocol mainly TCP/IP. Ethernet only defines the data link and physical layers of the Open Systems Interconnect (OSI) Reference Model whereas TCP/IP defines the transport and network layers respectively of the same model. Therefore, when the suggestion is made to use standard Ethernet for control does this mean TCP/IP connectivity as well?

APPLICATION
PRESENTATION
SESSION
TRANSPORT
NETWORK
DATA LINK
PHYSICAL

**Figure 1—Ethernet defines the lower two layer of the OSI Reference Model.**

## Ethernet Frames

The two types of Ethernet frames used in industry are similar. The DIX V2.0 frame, frequently referred to as the Ethernet II frame, consists of an eight-byte preamble, six-byte source and destination addresses, a two-byte type field used to identify higher layer protocols, a variable data byte field followed by a four-byte frame check sequence (FCS) field. The IEEE 802.3 frame divides the preamble into a seven-byte preamble followed by a single byte start of frame delimiter (SFD). The two-byte type field now becomes a two-byte length field. The data field now includes an 802.2 logical link control (LLC) field that precedes the actual data. The FCS remains the same.

### Preamble

The DIX preamble consists of 64 bits of alternating "1s" and "0s" but ending with two "1s" to indicate that a valid frame is to begin. This creates a 10 Mhz signal that synchronizes the receivers on the network before actual data arrives. Ethernet uses Manchester encoding.

The IEEE redefined the preamble to be seven bytes of preamble, the same as the DIX preamble, followed by a one-byte start of frame delimiter (SFD) which looks like the last byte of the DIX preamble. There is no change in operation between the DIX preamble and the IEEE preamble and SFD byte. Both preambles are not considered part of the frame when calculating the size of the overall frame.

### Destination Address

In the DIX standard the first bit of the 48-bit destination address indicates if the address is a multicast address or a physical address. A "0" indicates a unicast transmission to the indicated destination while a "1" indicates a multicast or group address.

Ethernet II DIX Frame						
64 bits	48 bits		48 bits	16 bits	368 to 12000 bits (46 to 1500 bytes)	32 bits
Preamble	Individual/ Group Address Bit	Destination Address	Source Address	Type	Data	Frame Check Sequence

IEEE 802.3 Frame								
56 bits	8 bits	48 bits			48 bits	16 bits	368 to 12000 bits (46 to 1500 bytes)	32 bits
Preamble	SFD	Individual/ Group Address Bit	Globally/ Locally Administered Address Bit	Destination Address	Source Address	Length	LLC/Data	Frame Check Sequence

The IEEE standard further defines the second bit of the 48-bit destination to indicate if the address is locally administered or globally administered. This bit is a "0" when the address is globally administered; that is, assigned by the Ethernet interface manufacturer.

A 48-bit address of all "1s" is a broadcast address in both DIX and IEEE formats indicating that the

**Figure 2—Two types of Ethernet frames are used in the industry.**

transmission is directed to all devices on the network.

### Source Address

The 48-bit source address is appended to the transmission as an aid to the higher layer protocols. It is not used for medium access control. To avoid duplicate node IDs for global addresses, the Ethernet adapter manufacturer obtains an Organizationally Unique Identifier (OUI) from the IEEE (for an administration fee). The OUI is 24-bits long and is used as the most significant portion of the 48-bit address. The manufacturer, using good record keeping, will assign sequential numbers to each adapter card he makes thereby creating a worldwide unique address. With 24-bits to work with, a lot of adapters can be produced from a single manufacturer. A list of OUI assignments can be found on the Internet.



### ***Type and Length Field***

The original intention of Ethernet was never to use its data link layer as the means for providing guaranteed delivery of data. It was always the intent that a higher layer protocol would do that service. Therefore it was only necessary to identify by number which higher layer protocol was being used through the two-byte field in the DIX frame. Originally, Xerox maintained the assignments and now IEEE provides the administration.

The 802.3 standard does not include the type field but instead defines it as a length field. Per the 802.3 standard, a value in this field of 1518 or less indicates the length of the data field, while values above this may be ignored, discarded or used in a private manner. These out of bound values could then be used to identify higher layer protocols just like DIX frames.

What is important here is that since DIX and IEEE frames are identical in terms of the number of bits and length of fields, both frames can coexist on the same network but may not be able to communicate to one another. Much of the existing TCP/IP software that binds to Ethernet uses DIX frames and not 802.3 frames, so care must be exercised when selecting or developing software or claiming interoperability.

### ***Data Field***

A raw Ethernet frame (no encapsulated protocol or LLC) can be up to 1500 bytes long but no less than 46 bytes. This is the DIX frame.

Although the total available length of the IEEE data field is the same as the DIX frame, the LLC header reduces the amount of field available for actual data or payload as it is sometimes referred to. If the LLC header and actual payload are less than 46 bytes, the data field must be padded to 46 bytes to ensure that the transmission is not interpreted as a runt packet or packet fragment.

### ***Frame Check Sequence***

Both the DIX and IEEE standard use four bytes to hold the CRC-32 check on the complete frame from destination address all the way to the end of the data field. The receiving station calculates its own CRC-32, checks on the received data and compares the results with the transmitted CRC-32 value for a match indicating a successful reception. Note that there is no inherent mechanism in the Ethernet data link layer protocol to inform the source node that a reception was accepted or rejected due to a failed CRC-32 check. That task is left to the higher layer protocol.

## Ethernet Physical Layers

Although Ethernet was originally designed as a coaxial bus system, alternate physical layers have evolved since the early 80s. The IEEE 802 committee has defined several physical layers and that is why it is important to specify the correct option when selecting Ethernet.

### 10BASE5

The original Ethernet was configured as a bus system with a thick coaxial cable as the medium. That is what was specified in the 1980 DIX standard. An external transceiver called a medium attachment unit (MAU) clamps at particular points on the cable marked by stripes every 2.5 meters. From the transceiver, an attachment unit interface (AUI) cable connects to an AUI port on the actual Ethernet adapter that fits into the computer. The AUI port is a DB-15 connector. A coaxial segment can be up to 500 meters long and AUI cables are each restricted to 50 meters in length. A total of 100 transceivers can occupy one trunk segment. Individual trunk segments can be cascaded using repeaters up to 2000 meters. In 1985 the IEEE standardized this configuration as 10BASE5 to signify 10 Mbps baseband signaling up to 500 meters in length.

Thick coaxial cable is indeed bulky and its topology is not always convenient to wire in a plant. Troubleshooting a 100-station segment could be a nightmare, so you do not see new 10BASE5 installations. There is no support for this cable with Fast Ethernet technology.

### 10BASE2

The answer to the bulkiness of 10BASE5 along with its expense was Thinnet or Cheapernet standardized in 1985 as 10BASE2. Thinnet again was a bus topology but with internal transceivers. A thin RG-58/u coaxial cable interconnects up to 30 stations to a maximum length of 185 meters. Segments can be repeated up to 740 meters. BNC style connectors, terminators and taps are used to cable the system. Although easier to install than 10BASE5, the focus on new installations is towards twisted-pair cabling. This cable is likewise not supported by Fast Ethernet.

### 10BASE-T

In 1990 the IEEE published 10BASE-T after pioneering work was done to introduce twisted-pair cabling and star topology to Ethernet installations. The 10BASE-T Ethernet adapters have internal transceivers and RJ-45 connectors. Usually two-pair unshielded cabling is attached to a hub in a point-to-point fashion. Bus connections are not allowed. The connection between an adapter and hub cannot exceed 100 meters in length. Hub-to-hub connection length can vary depending upon the medium used. If another twisted-pair connection is used, the maximum length is again 100 meters. With Thinnet it is 185 meters and with thick coaxial cable 500 meters.

The star topology is much easier to troubleshoot than a bus system; however, the reliability of the hub now must be considered in the overall reliability of the system. Another reason for the focus on twisted-pair is that development of Fast Ethernet is based on twisted-pair and not coaxial cable providing no migration path for installed coaxial cable.

**10BASE-F**

The 10BASE-F standard is actually a series of fiber optic standards. Fiber optics provides long distance, higher-speed migration, noise immunity and electrical isolation. There are three media standards:

**10BASE-FL** This fiber link standard replaces older FOIRL standard.

**10BASE-FB** This backbone standard is not very popular.

**10BASE-FP** This passive hub technology is also not popular.

The 10BASE-FL standard requires a duplex 62.5/125 $\mu$ m fiber optic cable for each link. Transmission distances of up to 2km are possible as well as full-duplex operation.

**Medium Access Control**

What follows is a discussion of the medium access control protocol for a 10 Mbps half-duplex Ethernet network operating with several nodes.

When a station wants to transmit, it first waits for an absence of a carrier, which would indicate that some other station is transmitting. As soon as silence is detected, the station waiting to transmit continues to defer until the Interframe Gap (IFG) time has expired which is a minimum of 96-bit times (9.6 $\mu$ s). If a carrier still appears to be absent, the station begins to transmit while observing its collision sense circuitry. If no collision is detected, the transmitting station assumes the transmission was sent successfully. If the transmitter detects an early collision, one which occurred during the preamble, the station continues to send the preamble plus 32 bits of data called a jam signal. This ensures that other stations will note the collision as well. After the collision, the transmitting station will backoff from retransmitting based upon a backoff algorithm. If no collisions are detected after 512-bit times (not counting the preamble), the station is assumed to have acquired the channel and no late collisions should occur on a properly working network. The collision counter is cleared. This 512-bit time (51.2 $\mu$ s) is called the slot time and is critical in the way Ethernet arbitrates access to the cable.

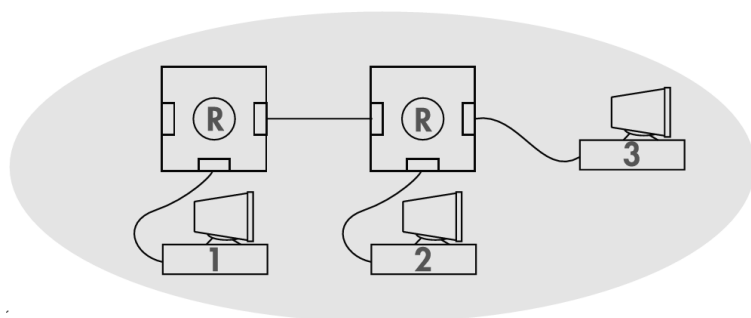
**Collision Domain**

This slot time defines the upper bound limit of the total propagation delay of a transmitted symbol from one end of the network to the farthest end and back. This includes the time it takes the symbol to travel through cables, repeaters and MAUs and varies with devices used. However, regardless of the path, the resulting propagation delay must be less than the slot time. Therefore the slot time defines Ethernet's maximum network diameter which limits its collision domain. A collision domain that exceeds the maximum network diameter violates Ethernet's medium access control mechanism resulting in unreliable operation.

Collisions can generate runt packets that are less than 512 bits in length. These can be detected by the receiving nodes and discarded accordingly. That is why it is important that a minimum valid Ethernet frame always be sent to distinguish valid packets from packet fragments. A minimum of 46 bytes in the data field ensures that a valid Ethernet frame is 512-bits long. Control messages are typically short so it should be remembered that the shortest Ethernet frame is 64 bytes in length.



If the network diameter is small, collision detection is faster and the resulting collision fragments are smaller. As the network diameter increases more time is lost detecting collisions and the collision fragments get larger. Increased network diameter aggravates the collision problem. Silence on the line does not necessarily mean a distant transmitter has not already sent a packet down the cable, which will eventually result in a collision.



**Figure 3—For proper operation, a collision domain must be within the maximum network diameter.**

### Collision Detection

A collision is defined as two stations attempting to transmit at the same time. On coaxial cable transceivers, there is circuitry to detect the DC level of the signal on the cable. This is the indicator of a collision. On fiber optic and twisted-pair interfaces with separate receive and transmit circuitry, a collision is detected by the simultaneous receiving and transmitting of data. Remember that we are discussing half-duplex Ethernet that allows either transmitting or receiving but not at the same time. Only transmitters look for collisions and it is their responsibility to reinforce a collision with a jam signal. Receivers only look for valid packets and automatically discard runt packets that are caused by collisions. Once a collision is detected by simultaneous transmitters, these transmitters will follow a backoff algorithm.

### Backoff Algorithm

When a collision occurs on the network, the colliding transmitters will backoff from retransmitting for a time determined by a backoff algorithm. This algorithm requires each transmitter to wait an integral number of slot times (51.2μs) before attempting a new transmission sequence. The integer is determined by the equation:

$$0 < r < 2k \text{ where } k = \min(n, 10)$$

The variable  $k$  is actually the number of collisions capped at a maximum of 10. Therefore,  $r$  can range from 0 to 1023 when  $k = 10$ . The actual value for  $r$  is determined by a random process within each Ethernet node. As the number of consecutive collisions increases, the range of possible backoff times increases exponentially. The number of possible retries is also capped but at 16.

For example, assume two stations A and B on the network wanting to transmit. They both wait for an absence of carrier and then wait for the IFG time to expire before initiating a transmission. It does not matter if they are 10 meters or 2500 meters apart. They could both be sensing silence and simultaneously begin to transmit causing a collision at some point. They each sense the collision and back off for either 0 or 1 slot time. The odds are 50-50 they will pick the same value and collide again. If they do, they will now back off for either 0, 1, 2 or 3 slot times. The probability of collision is now 25%. Eventually, one will win in which case its collision timer is cleared to zero while the other collision timer continues to increment until a successful transmission.

BACKOFF RANGE AS A FUNCTION OF COLLISIONS			
Collision on Attempt Number	Estimate of Number of Other Stations	Range of Random Numbers	Range of Backoff Times ( $\mu$ s)
1	1	0.....1	0.....51.2
2	3	0.....3	0.....153.6
3	7	0.....7	0.....358.4
4	15	0.....15	0.....768.0
5	31	0.....31	0.....1587.2
6	63	0.....63	0.....3225.6
7	127	0.....127	0.....6502.4
8	255	0.....255	0...13056.0
9	511	0.....511	0...26163.2
10	1023	0...1023	0...52377.6
11	1023	0...1023	0...52377.6
12	1023	0...1023	0...52377.6
13	1032	0...1023	0...52377.6
14	1023	0...1023	0...52377.6
15	1023	0...1023	0...52377.6
16	Too High	N/A	Discard Frame

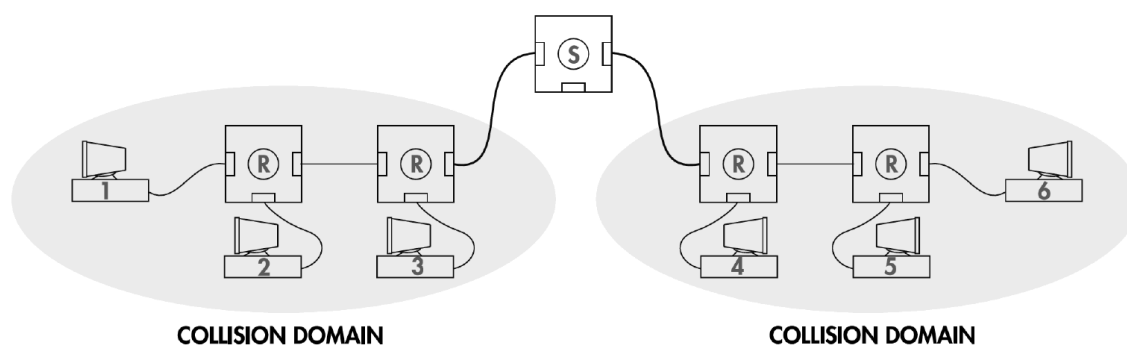
**Table 1—Backoff range increases exponentially with the number of collisions.**

A high number of retries indicates a busy network with more stations wanting to transmit than originally assumed. That is why the backoff time range is increased exponentially to provide more possible slot times for the additional stations. At ten retries, it is assumed that 1024 simultaneous transmitters exist. This becomes the upper bound limit of stations that can coexist on one Ethernet network. Actually this is the logical limit. Physically it may be impossible to have that many stations on one collision domain without violating cabling rules.

### Channel Capture

As shown above, the Ethernet backoff algorithm provides a means for peer stations to each gain access to the network. Access is provided to all but in an unpredictable fashion. The question is if access is fair?

Assume the same two stations A and B as before. This time, however, they both have high amounts of data to send and they attempt to send at the same time and collide on the first attempt. They both back off but this time A was successful. A's collision counter is cleared but B's does not clear. If station A has more data to send and it is quick to assemble another packet to send, it might collide with B again. This time B could be selecting higher and higher backoff times as its collision counter continues to increment. However, station A feels it has only experienced the first collision and will probably select a much lower timeout allowing it to transmit and assemble another packet and could beat station B again in the backoff contest. This phenomenon of channel capture is real and demonstrates that access to the network is neither fair nor predictable. The next time around station B could get the upper hand and limit A's access. If another station C decides to transmit as well, it could beat out station A due to the state of A's collision counter. In actuality a station that was last to arrive could transmit first.



**Figure 4—A switching hub, bridge or router is required to interconnect two or more collision domains.**

### Expanding an Ethernet Network

Expanding an Ethernet network is possible by the use of repeaters while maintaining one collision domain. If expansion is required beyond a collision domain, this can only be accomplished by the use

of bridges, switches or routers. To maintain one collision domain, a symbol sent from the extreme end of the network must be able to make a complete round trip within the slot time of 512-bits (51.2 $\mu$ s at 10 Mbps). Calculating the complete propagation delay through adapters, AUI cables, transceivers, trunk cables and repeaters is possible but is also a challenge. Table 2 at right provides information on the maximum number of MAUs per segment and the maximum segment length. The maximum allowable segment length, as well as the repeaters themselves, has been assigned delay values by the 802.3 specification.

The 802.3 specification discusses ways to interconnect cable segments with repeater sets without exceeding the collision domain. A repeater set is defined as repeater electronics and two or more attached MAUs—one for each segment to be connected. The system designer can use either transmission system model 1 or transmission system model 2. Approach 2 is the detailed approach where exact delay calculations and Interframe Gap shrinkage calculations are made. Approach 1 is the simplified approach, which is not as exacting as approach 2. Approach 1 has been further simplified by creating the 5-4-3 rule.

### 5-4-3 Rule

The 5-4-3 rule states that a system can have up to five segments in series, with up to four repeaters and no more than three mixing segments. The remaining two segments must be link segments. A mixing segment is defined as a segment that may be connected to more than two transceivers — in other words, a bus segment. Only coaxial cable can be used for a bus segment (we are ignoring 10BASE-FP) while fiber optic and twisted-pair cable can be used as link segments. A link segment can only have two transceivers and it must support full-duplex operation (separate transmit and receive channels) to speed up collision detection. This simplified rule does not address all the possible combinations but it does yield some gross network diameters. For example, all five segments cannot be 10BASE5 or 10BASE2. If all five were 10BASE-T then the diameter would be 500 meters. With fiber optics it is different. You cannot use the maximum segment length for all five segments.

In the case of 10BASE-F the maximum diameter is 2500 meters. You need to read the standard to understand this restriction.

The 5-4-3 rule does not address the three repeater configuration which yields four segments. In this case, all segments can be mixing providing a network diameter of 2000 meters for 10BASE5 and 740 meters for 10BASE2. For other configurations you need to refer to approach 2.

ETHERNET MAXIMUM MEDIA SEGMENT LENGTH		
Media type	Maximum number of MAUs per segment	Maximum segment length (m)
<i>Mixing segment</i>		
10BASE5	100	500 (trunk) 50 (AUI)
10BASE2	30	185
<i>Link segment</i>		
FOIRL	2	1000
10BASE-T	2	100
10BASE-FL	2	2000

**Table 2—Expansion rules require that segments be identified as being either mixing or link.**

### Summary

What has been discussed is the operation of shared Ethernet's physical and data link layers. Ethernet has evolved beyond this 10 Mbps half-duplex technology with its confusing expansion rules to much higher speeds, improved determinism and larger network diameters. The newer terms will be fast Ethernet and switched Ethernet but every network adapter or interconnection device must still obey the rules of shared Ethernet in order to be Ethernet compliant.

### References

*Ethernet: The Definitive Guide*, Charles E. Spurgeon, 2000, O'Reilly & Associates, Inc.

*Switched and Fast Ethernet*, Second Edition, Robert Breyer and Sean Riley, 1996, Macmillan Computer Publishing USA.

*International Standard ISO/IEC 8802-3 ANSI/IEEE Std 802.3*, 1996, The Institute of Electrical and Electronic Engineers, Inc.

## Introduction

With Shared Ethernet, repeaters are used to increase network diameter. The network diameter of an Ethernet network can be increased using repeaters as long as the network diameter does not exceed the collision domain of Ethernet. All Ethernet nodes must be able to recognize the occurrence of a collision regardless of the physical location of the nodes since the detection of collisions is fundamental in the manner Ethernet arbitrates media access. In this section, the concept of switching will be introduced as an alternative to the deployment of repeaters. Switches can not only increase the overall network diameter, but will improve the performance of Ethernet networks as well.

## Classifying Devices

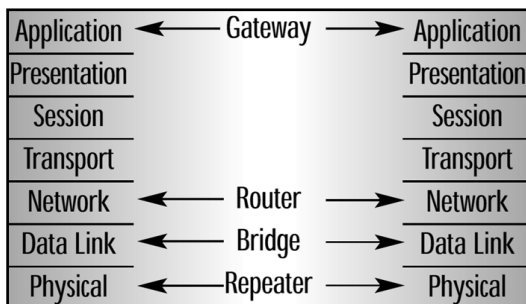
Although from the outside a switching hub looks very much like a repeating hub, they are from different classes of equipment. If you study the OSI Communications Model, you will notice seven distinct layers corresponding to different communication services.

At the lowest layer you have the physical layer which is concerned with the actual signals on the medium that represent data. These signals are called symbols and repeaters or repeating hubs receive these symbols and recondition them when extending networks. References such as 10BASE5 and 10BASE-T are physical layer standards.

Above the physical layer is the data link layer which handles the actual transmission and reception of frames sent and received over the physical layer. Issues such as station addressing (MAC or medium access control), framing of the data, and error detection are handled by the data link layer. The IEEE 802.3 standard is basically a data link standard although references to physical layer standards are included as well. Bridges operate at the data link layer. A bridge and a switch are one in the same.

Above the data link layer is the network layer which addresses the issues of transferring data, not over just one data link, but over multiple data links. This is classified as internetworking with the Internet Protocol (IP) being the most popular internetworking protocol. Routers are used to direct traffic between multiple data links and the transmission units are called packets. Switches do not commonly operate at this layer but there is such a thing as a layer 3 switch. This is actually a router with some switching functionality that can improve the performance of Ethernet networks as well.

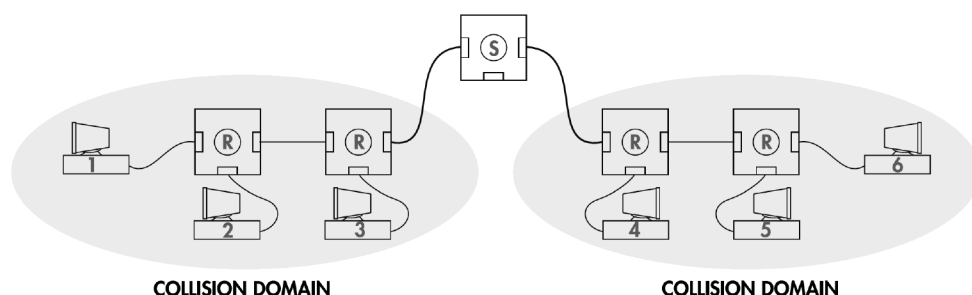
In terms of hardware, the next layer of interest is the application layer seven. This is where the gateways reside when it is necessary to interconnect dissimilar networks and dissimilar protocols. Gateways are aware of the actual application being run while all other devices such as repeaters, bridges and routers are not. We will concentrate only on one class of device called the bridge.



**Figure 1—Repeaters, bridges, routers and gateways operate at different layers of the OSI communications model.**

## Bridge (Switch) Construction

A switch is a bridge and the terms will be used interchangeably. The original bridges were two port devices interconnecting two similar data links to form one larger data link. If this can be accomplished without disruption, the bridge is considered a transparent bridge since communication within a data link or between data links appears the same.



**Figure 2—Collision domains terminate at the switch port.**

duplicated among the various data links. Unlike the traditional bridge with two ports, the switch has several ports and is usually referred to as a switching hub or just a switch.

Unlike a repeating hub, a switch has basically the same Ethernet interface on each of its ports as found on an Ethernet host adapter. That is because each port must function just like another Ethernet device. It must be able to receive and decode Ethernet frames and test for frame integrity as well as assemble and transmit Ethernet frames. However, each port does not necessarily require its own MAC address as would be required by an Ethernet host adapter. Each switch port functions in promiscuous mode by receiving all frames on its port independent of destination MAC address.

During transmissions, the Ethernet port masquerades as the originating device by assuming its source address. Therefore, each port on the switching hub does not require its own MAC address unless bridge addressing is required (the spanning tree protocol requires bridge addressing).

By having an Ethernet interface on each port, the Ethernet collision domain terminates at the switch port. With a repeating hub, the complete hub is part of the collision domain. By having a switch, the effective network diameter can double with the addition of one switch. This is because the network can be broken into two distinct data links. This is one benefit of switches. The effective network diameter can be increased with the addition of switches. This is especially important at 100Mbps since the collision domain is only 205 meters wide for copper-based systems.

Another difference between a repeating hub and a switch is that the repeating hub must operate at only one speed— either 10Mbps or 100Mbps. A switching hub can have multi-speed ports which can adjust to the capabilities of the device attached to its port. This is called auto-negotiation and different speeds on different ports are allowed. Some switches have fixed low speed ports (10Mbps) and one or more high-speed ports (100Mbps) for connection to servers where most of the traffic will be experienced.

You may think that we are describing a router but we are not. A router would consider each data link as an actual network with a corresponding network address. A bridge considers each individual data link as part of one larger data link or one network. The concept of network addressing is not used and individual station addresses (MAC addresses) are not



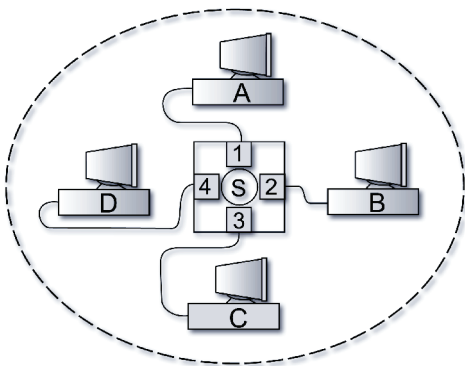
By terminating collision domains at each of its ports, a switch effectively segments the network into separate collision domains. If only one device is attached to a switch port (an Ethernet host adapter or another switch port), this is called microsegmentation. Under these circumstances full-duplex operation is possible yielding no collisions. However, if a shared Ethernet collision domain is present on a switch port (multiple host adapters and a repeating hub), only half-duplex operation is allowed and the switch port must conform to Ethernet's medium arbitration rules.

### Switch Operation

To understand the operation of a switch, we will assume that there are no provisions for programming the switch. The switch we will discuss will only modify its operation through a learning process.

Assume our switch has four identical ports. When power is applied to the switch it will behave just like a repeating hub. A data stream received on any one of its ports will be replicated, without modification, onto all other ports except for the arrival port. There will be no subsequent transmission on the port from which the data was received. In this situation the switch is functioning just like a repeating hub. There is, however, one difference. Switches operate at the data link layer and act upon frames. In the general case, a complete Ethernet frame, regardless of length, is received before being transferred to the switch's output ports. Therefore, data latency is introduced that varies with the length of the received frame. A repeating hub operates at the physical layer and acts upon symbols. A received symbol is transferred to the repeating hub's output ports usually within a few bit times. The data latency through a repeating hub is short and independent of the length of the incoming frame.

Let's assume that station A on port 1 is attempting a unicast (one to one) message to station B located at port 2. With a repeating hub or an unlearned switch, all stations on all ports are going to hear this transmission to station B even though they are not part of the conversation. This creates unnecessary traffic on the network that prevents other stations from initiating transmissions since they must defer to this traffic. Only when silence is sensed on the network will a deferring station initiate a transmission. Unlike a repeating hub, a learning switch will note the source address of the transmission from station A on port 1 and will enter into its table the fact that station A resides on port 1. However, at this time the switch does not know where station B resides and, therefore, must send the transmission to all other ports. This is called flooding. Not until station B initiates a transmission will the switch learn that station B resides on port 2. Once station A and B's port assignments are entered into the switch's table, all subsequent unicast transmissions between these two stations will only appear on ports 1 and 2. All other ports will not know a transmission is occurring allowing other stations, not located on ports 1 and 2, to initiate a simultaneous transmission. This is why switches offer improved throughput over repeating hubs.



**Figure 3—Switches direct traffic only to necessary ports**

What happens if station B is physically moved to port 3? If station A again initiates a transmission to station B, the transmission will fail since it will only be delivered to port 2 where the switch thinks station B resides. In order for station B to be found again, it must initiate a transmission. If it does, the switch will note a change in port assignment for station B and change its table accordingly. But what if station B never reports in? Perhaps this station speaks only when spoken to. There is no way for the switch to learn the new location of station B. That is why the switch's table must be aged.

Aging is the process of unlearning. Periodically the switch checks to see if all stations in its table have initiated a transmission within the aging limit. If a particular station has not, it will be removed from the table. In the example above, station B's entry would be erased. Therefore, when station A initiates a transmission to station B, the switch, finding no entry in its table for station B, will flood all ports allowing station B to hear station A. If station B responds to station A by initiating a transmission, the switch will learn station B's new port assignment and note it in its table. This aging process usually occurs every four to five minutes, so it may take a while to learn locations of devices that are quiet.

### Switch Fabric

The beauty of a switch is that, unlike a repeating hub, a switch allows simultaneous transmissions on its ports thereby increasing throughput. This is only true if the switching mechanism within the switch is fast enough to handle simultaneous transmissions on all its ports. If the switch can do this, it is said to be operating at wire speed and it is called a non-blocking switch. If it cannot keep up with the traffic, it may need to queue frames or lose frames. This is called a blocking switch. This switch mechanism within the hub is called its switch fabric and it must be extremely fast for the switch to be effective.

A switch's primary mission is to reliably transfer frames from one port to another. Its secondary mission is to note in its table the port location of various source addresses it learns. Its final mission is to age the table so that stations can be relocated to other ports and still be found by the switch. Depending upon the amount of traffic being handled by the switch, the switch may not be able to do all these tasks with each frame. It is possible that a switch may forgo updating its table when transferring multiple frames meaning that some source addresses will not be noted the first time they appear. The aging process is generally a background process anyway and aging time may vary with traffic.

IEEE 802.3 Frame								
56 bits	8 bits	48 bits			48 bits	16 bits	368 to 12000 bits (46 to 1500 bytes)	32 bits
Preamble	SF	Individual/ Group Address Bit	Globally/ Locally Administered Address Bit	Destination Address	Source Address	Length	LLC/Data	Frame Check Sequence

Figure 4—A store-and-forward switch must read in the complete Ethernet frame before forwarding.

### Data Latency

A repeating hub operates upon symbols while a switch operates upon frames. A switch must receive the complete frame from one of its input ports, observe the destination address, look up the port assignment, note the source address, verify that the frame is not in error and then forward the frame to the indicated port number. This is called store-and-forwarding. At 10Mbps, the longest allowable Ethernet frame will take over 1.2 ms to transfer through the switch. The shortest allowable frame would still take over 500 $\mu$ s to send. The store-and-forward nature of the switch introduces significant data latency. Compare this latency to that of a repeating hub which introduces a delay less than a microsecond. To reduce this latency, the concept of cut-through switches was introduced.

Since the destination address follows the preamble in an Ethernet frame, it only takes about 11 $\mu$ s for the switch to know to which port the frame must be transferred to. The switch could immediately begin transferring the frame to the required port. This, of course, assumes the output port is available. If the port is available, data latency can be reduced significantly. There are, however, problems with this approach.

If the output port is unavailable, the switch would need to queue the frame just like a store-and-forward switch. If the frame was corrupted, as evidenced by a failed FCS test, the switch would have forwarded a defective frame. Defective frames should be discarded by switches and not propagated through the network. However, with a highly reliable local area network, the chance of a failed FCS is rare so this may not be a significant issue. What is significant though, is that runt frames may exist that are the result of collisions. These runt frames are less than 576 bits in length but could be more than the 112 bits of preamble and destination address. Therefore, the switch could be guilty of propagating an error frame by initiating the forwarding of the frame before determining that it is actually a runt. The solution to this problem is the modified cut-through approach where forwarding does not commence until at least 576 bits of frame are received. Only at this time should the forwarding of the frame begin.

Sometimes cut-through operation is not possible anyway. For example, if a switch receives a broadcast, multicast or unknown destination address, it must flood all ports. The probability that all port output queues are simultaneously available for immediate transmission is remote. In this case, the complete frame must be received and sent to the port output queues for eventual transmission.

The significance of data latency can be debated. If each transmitted packet must be acknowledged by the receiving station, then data latency can be important since throughput is impacted by the delay in sending packets and receiving acknowledgments. However, if it is possible to stream the data, the delay in transmission is insignificant since the delay of store-and-forwarding is not accumulative. The delay in sending one frame versus many frames in a row is the same. Streaming of data using the TCP/IP protocol is possible. Knowledge of the transport layer protocol is important when determining if switch data latency is going to be an issue.

## Flow Control

With a high-speed switch fabric, there appears to be no bounds to the amount of simultaneous traffic that can be processed by a switch. However, traffic patterns may not be so evenly dispersed. Typically, you will have one port, possibly the port connected to a server or master controller, processing most of the traffic that originate from the other ports. If the switch has no flow control mechanism to limit the traffic being received on input ports and the congested output port has no more buffer available, frames will be simply dropped without any notification. To minimize this possibility, two methods of flow control were developed for switches—backpressure and PAUSE.

Backpressure is used on switch ports connected to half-duplex or shared Ethernet data links. The switch port simply uses the built-in collision detection and backoff algorithm of Ethernet to force collisions on its segment thereby requiring the attached devices to resend their data. When the switch is able to recover the backpressure is removed.

For full-duplex links there are no collisions, so backpressure will not work. There is instead a PAUSE function developed solely for full-duplex links. A PAUSE frame initiated by a switch port tells the sourcing device to stop sending traffic for a defined amount of time. This scheme only works if the attached device can invoke full-duplex operation and can interpret a PAUSE frame.

## IEEE 802.1D

There is a standard for bridges that is available from the IEEE as standard 802.1D. This standard is entitled, "Information technology— Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 3: Media Access Control (MAC) Bridges." This standard addresses the uses of bridges and, therefore, switches. There are some interesting parameters in the standard that impact the operation of automation networks.

### *Aging*

Aging is the amount of time the bridge waits until it removes a source address from the table due to the fact that the source address has not initiated a transmission within the aging time. The standard allows for an extremely wide range of values from 10 seconds to 1,000,000 seconds. The default, however, is 300 seconds or five minutes, which is what most bridges use.

### *Bridge Transit Delay*

The maximum amount of data latency introduced by a switch is specified. Although the recommended maximum is one second, up to four seconds is allowed. This amount of time seems long. Couple this time with the maximum allowable number of switches that can be cascaded (seven), the theoretical delay could be as much as 28 seconds! This is an eternity for an automation system. Fortunately, modern switches operate much faster than the standard requires.

**FCS checking**

A switch is required to perform a frame check sequence test on incoming frames and discard defective ones. To do this, the switch must receive the complete frame before forwarding. This means that the standard does not allow cut-through or modified cut-through operation.

**Bridge addressing**

The standard requires that not only must the bridge have a MAC address, each port must have a MAC address. This is unnecessary for normal switch operation. Many commercial switches do not support this requirement.

**Summary**

Switches are classified as bridges and operate at the data link layer. They can create a much larger network diameter by segmenting the network into separate collision domains. Switches can learn from their environment and then restrict traffic only to necessary ports. This frees up other ports to initiate their own independent transmissions thereby increasing the performance over a shared Ethernet network. Repeating hubs have their place but depending upon the application, switches could provide a better solution.

**References**

*The Switch Book*, Rich Seifert, 2000, Wiley Computer Publishing  
*Ethernet: The Definitive Guide*, Charles E. Spurgeon, 2000, O'Reilly & Associates, Inc.  
*International Standard ISO/IEC 8802-3 ANSI/IEEE Std 802.3*, 1996, The Institute of Electrical and Electronic Engineers, Inc.  
*ANSI/IEEE Std 802.1D*, 1998, Information technology- Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 3: Media Access Control (MAC) Bridges.



### Introduction

Since its introduction in the early 70's, Ethernet has continued to evolve evidenced by the fact the IEEE 802.3-2012 standard, which describes the technology, is now over 3700 pages! Ethernet was created with a multi-drop physical layer using a thick backbone eventually operating at 10 Mbps. A thin coaxial bus topology was introduced to reduce cabling expense, but it was the twisted-pair version that received the greatest acceptance. Throughout these changes, Ethernet maintained its 10 Mbps data rate. With demand for higher speeds, there was much discussion on how this could be achieved.

Fiber Distributed Data Interface (FDDI) operating at 100 Mbps did exist but was considered expensive. The question was, "Could Ethernet operate at 100 Mbps?" The answer was "yes," but there were competing approaches to Fast Ethernet. One approach was to not make any changes at all, just scale the protocol to 100 Mbps. This would provide backward compatibility and was favored by most vendors. The second approach favored a redesign of the medium access control (MAC) in order to gain a feature called demand-priority at the sake of backward compatibility. In the end the "technically elegant" demand-priority solution, which utilized a token-passing protocol, lost out both in the IEEE 802.3 committee, and in the marketplace.

Introduced as 100VG-AnyLAN, the technology never gained widespread support outside its two proponents, Hewlett Packard and AT&T. Although the words "Fast Ethernet" are not used, the IEEE 802.3u was adopted as the Fast Ethernet standard in 1995.

### How fast is fast?

It is generally accepted that Fast Ethernet implies Ethernet at 100 Mbps. Compared to 10 Mbps, 100 Mbps seems quite fast. In fact, to most microcontrollers it is exceptionally fast and a difficult speed to maintain. However, there are still much faster Ethernet technologies – 1000 Mbps and 10000 Mbps. Still many attributes of 10 Mbps Ethernet are retained in the 1000BASE-T standard. This technology is called Gigabit Ethernet and its successor operates at 10 Gigabits/second. Although 100 Mbps is not the fastest, we will still concentrate on this technology since it has the most application with automation controllers. However, Gigabit Ethernet will be addressed as well.

### Scaling Ethernet

In Table 1 you will see a summary of attributes for both Ethernet and Fast Ethernet. You will notice values of the various Ethernet attributes, in terms of bits or bit-times, are maintained even as Ethernet is scaled from 10 Mbps to 100 Mbps. However, one bit-time at 10 Mbps is 100ns and at 100 Mbps it is 10ns. Look at Table 2 where the actual times replace the bit and bit-time figures. The good news is that an equivalent amount of data sent at 10 Mbps will be received in a tenth of the time at 100 Mbps. Also notice that the minimum frame size and slot time have been reduced by a factor of ten as well.

	<b>Ethernet/802.3</b>	<b>Fast Ethernet 802.3u</b>
<b>Slot Time</b>	512 bit-times	512 bit-times
<b>Interframe Gap</b>	96 bit-times	96 bit-times
<b>Attempt Limit</b>	16 tries	16 tries
<b>Backoff Limit</b>	10 (exponent)	10 (exponent)
<b>Jam Size</b>	32 bits	32 bits
<b>Max Frame Size</b>	1518 bytes	1518 bytes
<b>Min Frame Size</b>	64 bytes	64 bytes
<b>Address Size</b>	48 bits	48 bits

**Table 1—Scaling Ethernet by 10 fold yields Fast Ethernet without any change in attribute parameters.**

These attributes are directly linked to the collision domain of Ethernet and determine the maximum network diameter of the network based upon the maximum round-trip delay of a signal propagating between the two furthest points on the network. While the Ethernet protocol scales nicely from 10 Mbps to 100 Mbps, the actual time it takes for signals to pass down wires does not. The result is that the maximum network diameter must be reduced from about 2800m for 10 Mbps Ethernet to a low of 205m for Fast Ethernet. The only way to increase distance while retaining 100 Mbps speed is to use full-duplex links which eliminate collisions altogether; therefore, the link segments are not limited by timing.

### Half-Duplex or Full-Duplex

Full-duplex links are the key to extending the maximum network diameter of Fast Ethernet. Full-duplex requires separate receive and transmit paths and link segments consisting of no more than two devices. These devices can be Ethernet adapters or switching hub ports. Notice that there is no mention of repeating hub ports. A repeating hub is part of the collision domain and reinforces collisions received on any of its other ports. A repeating hub is not capable of full-duplex operation. Although it is possible to have just two Ethernet adapters configured for full-duplex, expansion beyond two adapters requires a switching hub capable of supporting full-duplex operation.

	<b>Ethernet/802.3</b>	<b>Fast Ethernet 802.3u</b>
<b>Slot Time</b>	51.2 $\mu$ s	5.12 $\mu$ s
<b>Interframe Gap</b>	9.6 $\mu$ s	0.96 $\mu$ s
<b>Attempt Limit</b>	16 tries	16 tries
<b>Backoff Limit</b>	10 (exponent)	10 (exponent)
<b>Jam Size</b>	3.2 $\mu$ s	0.32 $\mu$ s
<b>Max Frame Size</b>	1214.4 $\mu$ s	121.44 $\mu$ s
<b>Min Frame Size</b>	6.4 $\mu$ s	0.64 $\mu$ s
<b>Address Size</b>	4.8 $\mu$ s	0.48 $\mu$ s

**Table 2—In terms of actual time, there are significant differences between Ethernet and Fast Ethernet.**

Half-duplex means transmitting and receiving over the same medium but not at the same time. Full-duplex allows for simultaneous sending and receiving. Coaxial-based transceivers such as 10BASE5 and 10BASE2 were never able to invoke full-duplex since they do not have separate receive and transmit paths. However, 10BASE-T and 10BASE-FL do have separate receive or transmit paths and are capable of full-duplex operation depending upon the sophistication of the Ethernet adapter or switching hub. If these interfaces are configured for half-duplex, then the simultaneous detection of receive and transmit activity will trigger collision detection. These same interfaces, configured for full-duplex, would disable this collision detection logic since full-duplex does not follow the CSMA/CD rules of shared Ethernet.

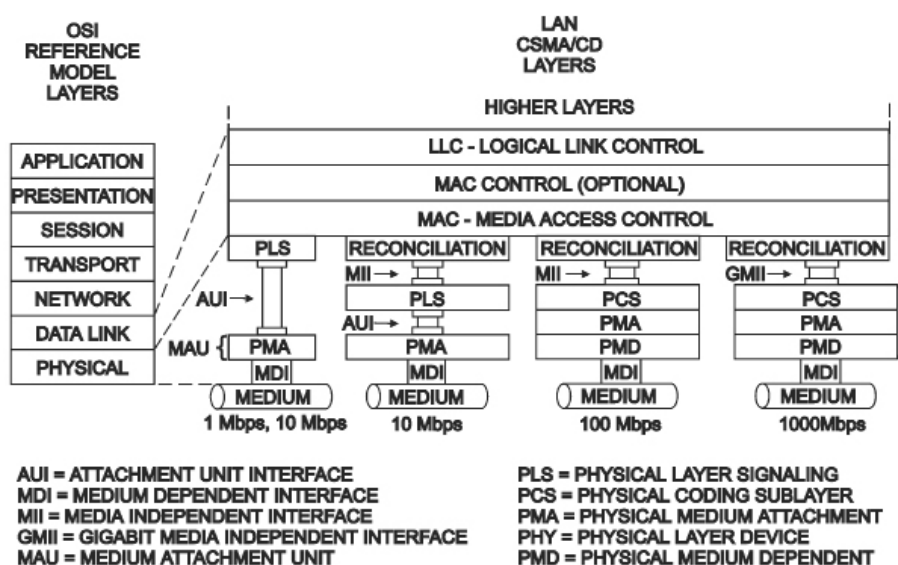
It is very important that a full-duplex link be configured properly. A station or switching hub port will send out frames at will, ignoring the CSMA/CD protocol of shared Ethernet if it is configured for full-duplex. If the other end is configured for half-duplex, it will incorrectly detect collisions and take actions that could cause late collisions (which are not automatically resent) and CRC errors. The result is a general slowdown of the network negating the benefits of migrating to Fast Ethernet.

## 200 Mbps?

One virtue of full-duplex is the claim that throughput is doubled to 200 Mbps by simply invoking full-duplex over half-duplex. Although mathematically correct that two simultaneous data streams can travel on the medium—one going 100 Mbps in one direction and another one going 100 Mbps in the other direction—this occurrence is not common when using automation protocols. Most automation protocols operate as either master/slave or command/response. With these types of protocols, a station initiates a command to another station, which must decode and execute the command before providing the response. This type of communication could function quite well with a half-duplex link since only a command stream or a response stream is active at one time. In backbone applications, full-duplex operation can be helpful, but do not assume that a system will double in performance by simply invoking full-duplex. The benefits of full-duplex are mostly to gain a larger network diameter and eliminate the complexity and uncertainty of the CSMA/CD protocol.

## Ethernet According to 802.3

If you look at Figure 1, you will see a very complex diagram of the various flavors of Ethernet using the definitions of 802.3. Notice that the possible data rates can range from 1 Mbps to 1000 Mbps. The 10000 Mbps rate is not shown in this figure and the 1 Mbps rate was not commercialized. We will concentrate on the 10 Mbps, 100 Mbps, and 1000 options. In terms of the OSI Reference Model, all the data rate variations are considered physical layer attributes, while the data link layer is common to all data rate variations. The 802.3 model does not even address the upper layers of the OSI model.



**Figure 1—The complex 802.3 model only addresses two layers of the OSI Reference Model.**

Of course, the 10 Mbps versions are the simplest. However, notice that two are shown in the figure. The one on the left is the traditional model showing the Attachment Unit Interface (AUI). At this point any physical medium can be attached by way of the DB-15 AUI connector. This was the connector originally used to attach to a 10BASE5 (thick trunk) transceiver which was usually mounted in the ceiling. In 802.3 terms, this transceiver would be called a Physical Medium Attachment (PMA) or simply a Medium Attachment Unit (MAU). In order to get down to the Physical Layer Signaling (PLS) of the Ethernet interface, an AUI cable, or drop cable, is used acting like an extension cord. On one end is a male DB-15 and the other end is a female DB-15. The AUI port is medium independent. If a fiber optic medium is desired, the appropriate MAU would be connected to the AUI port. On the fiber optic side, the fiber optic cables would be connected to the Medium Dependent Interface (MDI) which would include ST-style fiber optic connectors. This completes the connection to the medium.

With the introduction of Fast Ethernet, the model changed somewhat. A new connection was defined and was called the Media Independent Interface (MII). It provided more bandwidth at 100 Mbps than could be provided by the AUI interface. Notice that at 10 Mbps, the MII is inserted between the PLS and the Reconciliation interface. This is shown to the immediate right of the traditional model. Further to the right is the 100 Mbps variation, which introduces additional terms. The Physical Medium Dependent now attaches to the MDI and the Physical Coding Sublayer (PCS) attaches to the MII. In the middle is the PMA. Collectively, the PCS, PMA and PMD are termed the PHY for Physical Layer Device. The MII connector is defined as 50 mm wide dual-row, 40 pin. Like the AUI, it is only an option. Most Ethernet interfaces only provide the necessary MDI connector and nothing else. To understand the standard these terms and connections must be defined. You might ask why a MII interface was defined for 10 Mbps Ethernet when an AUI seems to have served the industry quite well? The MII interface supports both 10 Mbps and 100 Mbps data rates which is key to providing dual-speed support between legacy 10 Mbps and newer 100 Mbps systems. On the other hand, the AUI only supports 1 Mbps and 10 Mbps while the new Gigabit Medium Independent Interface (GMII) only supports 1000 Mbps.

### **100BASE-T**

Fast Ethernet was introduced with the 100BASE-T standard. In this standard, new concepts such as Reconciliation Sublayer (RS), Media Independent Interface, Auto-Negotiation and Management are introduced along with several different physical layers. Therefore, unlike 10BASE-T, specifying 100BASE-T is not sufficient when designating a physical layer since three are mentioned in the standard—100BASE-T4, 100BASE-TX and 100BASE-FX. What 100BASE-T basically describes is the new MII interface which is much more sophisticated than the AUI. Instead of a serial bit stream for data, four-bit nibbles are transferred between the PHY and RS over the MII for both reception and transmission. For 10 Mbps operation, clocking is reduced from that of 100 Mbps otherwise the interface is the same. It is not a requirement that both speeds be supported.

### **Management Interface**

Another unique feature of Fast Ethernet is the ability to manage the PHY itself through a two-wire interface in the MII. As a minimum, one control and one status register must be provided. The PHY must report its abilities and respond to commands. The PHY reports its type of physical layer, its ability to auto-negotiate and if it is capable of handling full-duplex operation. The station management might command the PHY to a fixed speed or to auto-negotiate. The PHY reporting its abilities ensures that it will not be commanded to perform what it is incapable of doing.

### **100BASE-T2, 100BASE-T4**

Two Fast Ethernet physical layers that are mentioned are 100BASE-T4 and 100BASE-T2. Both were developed to utilize lower speed cabling that may exist in an installation. 100BASE-T4 requires four pairs and is incapable of full-duplex operation. 100BASE-T2 only needs two pairs but uses a sophisticated signaling scheme that is difficult and expensive to implement. The industry seems to have rewired for the higher speed Category 5 cable so these two interfaces are not popular.

### 100BASE-X

Like 100BASE-T, 100BASE-X is not a unique physical layer but details the encoding for the two most popular physical layers—100BASE-TX and 100BASE-FX. One physical layer is for copper and the other for fiber optics, yet the standard applies to both. Much of the 100BASE-X standard comes from the FDDI standard including the 4B/5B encoding.

#### 4B/5B

Data transfers over the MII are done with 4-bit nibbles that represent actual data. With 10BASE-T, Manchester encoding is used which guarantees a transition within every bit cell regardless of logic state. This effectively creates a 20 Mbaud signal for a 10 Mbps data rate. If the same encoding were used for Fast Ethernet, a 200 Mbaud signal would result making it difficult to maintain the same 100m maximum segment length due to high frequency attenuation. If we could use a code such as non-return to zero (NRZ), we could match the data rate with the baud rate. The problem with NRZ is that it has a DC component, which transformers do not like, and it provides little information about clocking. A compromise is the 4B/5B code where the 4-bit nibbles being transferred over the MII are actually encoded as five-bit symbols sent over the medium. The encoding efficiency is 80% and the baud rate increases to 125 Mbaud. This is still fast but not as fast as 200 Mbaud. The actual codes used are chosen so that sufficient transitions occur in the resulting bit pattern so as not to lose clocking. With twice as many codes as necessary, there are many that are left unused and some that are defined for control purposes. The 4B/5B scheme is used for both the 100BASE-TX and 100BASE-FX physical layers.

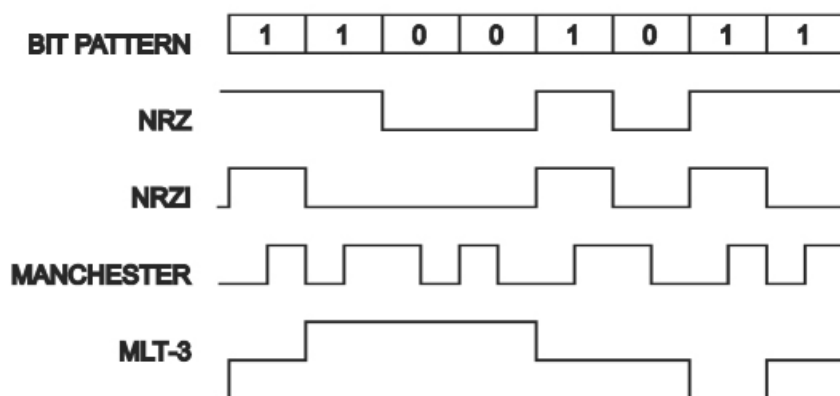
### 100BASE-TX

The 100BASE-TX twisted-pair physical layer retains the same MDI (RJ-45) connector and pinout as 10BASE-T so that auto-negotiation is possible. Two twisted-pairs with separate receive and transmit paths allow for full-duplex operation. Cabling requires a higher performance Category 5. This cable type is an unshielded twisted-pair characterized as 100 ohms. The standard mentions a 150 ohm shielded twisted-pair (STP) option with a DB-9 connector that can be used as well. A unique feature of 100BASE-X encoding is that the link is always active even without data since a unique symbol called IDLE is sent continuously during inactivity. Reception of IDLE serves as a link integrity function. Maximum segment length for 100BASE-TX is 100m just like 10BASE-T. Signaling on the twisted-pair incorporates a three level multi-level technique called MLT-3. The benefit of this code is to further reduce the electromagnetic emissions (EMI) and the bandwidth requirements of the medium.

**Table 3—Ethernet physical layers differ in terms of data rate, encoding and signaling.**

	10BASE-T	100BASE-TX	100BASE-FX
<b>Data Rate</b>	10 Mbps	100 Mbps	100 Mbps
<b>Encoding</b>	Manchester	4B/5B	4B/5B
<b>Signaling</b>	5V Differential	MLT-3	NRZI
<b>Wires</b>	4	4	2 (Fibers)
<b>Cable</b>	Cat. 3 UTP	Cat. 5 UTP	62.5/125 $\mu$ m
<b>Connector</b>	RJ-45	RJ-45	SC or ST
<b>Max Segment</b>	100m	100m	2km
<b>Max Transceivers</b>	2	2	2





*Figure 2—Various encodings for the same bit pattern. Notice that MLT-3 approximates a sine wave with a much lower fundamental frequency than the data rate.*

to zero inverted) since there is no concern for EMI on fiber optic links. With NRZI, the state of signaling inverts on logic 1s, thereby providing some clocking information unlike NRZ. Table 3 summarizes three most important Ethernet physical layers and Figure 2 shows the various encoding schemes.

### Gigabit Ethernet

If you thought Fast Ethernet is confusing, Gigabit Ethernet is worse. It is a marvel that it actually works with its increased sophistication. As with Fast Ethernet, Gigabit borrows heavily from earlier Ethernet versions as well as other technologies. Although Figure 1 shows a GMII connection, it is not practical to bring out due to the high-speed nature of Gigabit Ethernet. Instead, Gigabit transceivers are built-in to the network adapters and switch ports. The user will only see the MDI connector and in the case of twisted-pair this would be a common RJ-45 connector with all pins used for data transfer. The GMII is only used for Gigabit Ethernet. Gigabit Ethernet chips support both the MII and GMII interfaces so if 10 or 100 Mbps operation is indicated by the Auto-negotiation protocol (to be introduced later), the MII is used for this purpose.

### 1000BASE-T

This is the twisted-pair standard of interest and it relies upon Fast Ethernet twisted-pair technologies that were not commercial successes. However, the Fast Ethernet encoding differs from that of Gigabit Ethernet which uses a block encoding scheme called 4D-PAM5 which means four pairs of Pulse Amplitude Modulation using five levels. In addition, the four pairs allow for simultaneous receiving and transmitting. Eight-bits of data are sent every signal transition – two bits per pair – which allows a 125 Mbaud transition rate at 1000 Mbps. That is the same baud rate as 100BASE-T. This high efficiency encoding with five-level modulation versus three certainly makes 1000BASE-T more complex than 100BASE-TX.

### 100BASE-FX

The 100BASE-FX fiber optic physical layer is very similar in performance to 10BASE-FL. Maximum segment length 2km for both technologies; however, for 100BASE-FX this is only achieved on full-duplex links. On half-duplex links the segment length cannot exceed 412m. Either SC or ST fiber optic connectors can be used, but SC is recommended. Multi-mode fiber optic cable (62.5/125μm) is what is normally used; however, it is possible to use single mode fiber optics for greater distances on full-duplex links. The signaling on fiber optics is NRZI (non-return

Gigabit twisted-pair cabling requires at a minimum a Category 5 rating with Category 5e and 6 commonly found in the field. RJ-45 connectors are retained but their signal definitions change. Unshielded twisted-pair cable can be used with a maximum segment length of 100 meters. Only link segments are supported (two transceivers per segment) and the crossover function is accomplished internally in the companion Gigabit switch port.

### Auto-Negotiation

With the proliferation of Fast Ethernet and the similarity of the cabling components to conventional Ethernet, a means was proposed in IEEE 802.3u to automatically configure Fast Ethernet ports to work with either legacy Ethernet ports or other Fast Ethernet ports. This configuration protocol was based upon National Semiconductor's NWay standard. There is a way for twisted-pair links to automatically configure compatible formats in order for links to begin communicating. This scheme is only suitable for twisted-pair links and not coaxial or fiber optic links. Coaxial cable is a legacy 10 Mbps standard that is not in the plans for evolving Ethernet. Fiber optics is a different story. Although fiber optics is very much in the plans for evolving Ethernet, there is no simple way for two fiber optic devices to auto-negotiate data rates since a 10BASE-FL device operates at 850nm while a 100BASE-FX device operates at 1300nm. These devices will not interoperate. The Auto-Negotiation protocol is better intended for twisted-pair links where there are only two devices on a segment. Notice that twisted-pair bus segments, which are favored in automation systems, are nowhere supported by Ethernet standards. Automation customers must be content wiring in a star topology and using either repeating hubs or switching hubs.

The benefit of auto-negotiation is to provide hands-free configuration of the two devices attached to the link segment. At connection time, each of the two devices will advertise all their technical abilities. These abilities have been ranked by the standard as shown in Table 4. The lowest possible ranking is 10BASE-T which assumes half-duplex or shared Ethernet operation. The very next ranking is 10BASE-T full-duplex indicating that full-duplex has higher performance than half-duplex. Notice that there is no ranking for 100BASE-T4 full-duplex. This is because this technology is not capable of full-duplex operation. Finally, the highest ranking is 1000BASE-T full-duplex. This ranking scheme has been provided for completeness. It is not assumed that a particular adapter can handle all technologies. In fact, some of these technologies may not have been commercialized. However, they are all listed consistent with the IEEE 802.3 standard.

Each device examines each other's technical abilities and determines the lowest common denominator. For example, if an Ethernet adapter can only handle 10BASE-T while a switch port can handle either 10BASE-T or 100BASE-TX, 10BASE-T will be chosen by both. If two Ethernet adapters, one only advertising 10BASE-T and the other only advertising 100BASE-TX connect, there will be no subsequent communication since no compatibility exists.

1000BASE-T full-duplex  
1000BASE-T  
100BASE-T2 full-duplex  
100BASE-TX full-duplex  
100BASE-T2  
100BASE-T4  
100BASE-TX  
10BASE-T full-duplex  
10BASE-T

**Table 4—Auto-Negotiation assumes a ranking of priorities. 10BASE-T is at the bottom.**

### FLP and NLP

With legacy 10BASE-T systems, a Normal Link Pulse (NLP) is sent between two devices in order to demonstrate that the link is functional in both directions. With Fast Ethernet and Gigabit Ethernet, the NLP is replaced with a Fast Link Pulse (FLP) that carries the Auto-Negotiation information. The FLP is interpreted by legacy 10BASE-T devices as a NLP so as to provide backward compatibility. If a connection is broken, as evidenced by a lack of link pulses, a series of FLPs will be observed once the link is reestablished. After auto-negotiation is completed, link pulses or link activity are monitored as with legacy 10BASE-T networks.

### Summary

Fast Ethernet provides the path to higher performance Ethernet systems with the ability to interoperate with legacy 10 Mbps systems. Added features such as full-duplex and auto-negotiation also make this migration path attractive. Fast Ethernet originally meant 100 Mbps operation but Gigabit Ethernet is now commonly deployed.

### References

*Ethernet—The Definitive Guide*, Charles E. Spurgeon, 2000, O'Reilly & Associates, Inc.  
*Switched and Fast Ethernet, Second Edition*, Robert Breyer and Sean Riley, 1996, Macmillan Computer Publishing USA.  
*International Standard ISO/IEC 8802-3 ANSI/IEEE STD. 802.3, 2012 Edition*, The Institute of Electrical and Electronic Engineers, Inc.

## Ethernet with Fiber

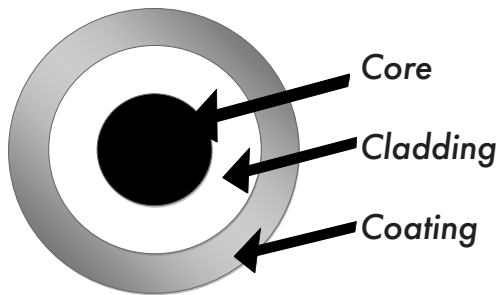
### Optic Cabling

#### Introduction

The use of fiber optics in local area networks (LANs), such as Ethernet, has increased due to the inherent advantages of using fiber. High data rates can be maintained without electromagnetic or radio frequency interference (EMI/RFI). Longer distances can be achieved over that of copper wiring. For the industrial/commercial user, fiber offers high-voltage isolation, intrinsic safety and elimination of ground loops in geographically large installations. Ethernet will function with no difficulty over fiber optics as long as some simple rules are followed.

#### Cabling Basics

Optical fiber consists of three basic elements: core, cladding and the coating. The core constructed of either glass or plastic provides the basic means for transmitting the light energy down the cable. The cladding prevents the light from exiting the core and being absorbed by the cable itself. The coating provides protection to the fiber core while providing strength. Final protection is provided by an overall jacket that may consist of other strength and protective elements.



**Figure 1—A single fiber consists of three basic elements.**

#### Fiber Size

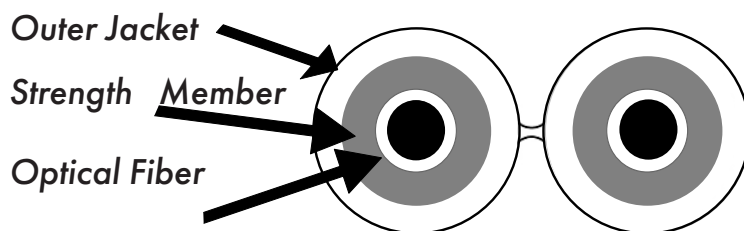
Optical fibers are classified by their diameter in microns (1 micron = one-millionth of a meter). Frequently the core, cladding and coating are specified using slashes (/) to separate the values. For example, 50/125/250 means the core is 50  $\mu\text{m}$ , the cladding is 125  $\mu\text{m}$  and the coating is 250  $\mu\text{m}$ . These dimensions all pertain to the concentric diameters of the various elements. A short form way of specifying the fiber is to only list the core and cladding sizes. In the above example, this fiber would be classified as 50/125. Core sizes range from as small as 5  $\mu\text{m}$  to as high as 1000  $\mu\text{m}$ . Depending upon the core size, either one or two modes of light transmission will be experienced. The two modes are called single-mode and multimode.

#### Single-Mode Operation

With very small diameter fibers in the range of 5 to 10  $\mu\text{m}$ , all light rays have a tendency to propagate along the axis of the fiber. Since there is only one path for the light to take, the light is termed to be experiencing a single-mode of operation. As the core diameter increases, the light rays have the option of traveling at an angle to the core axis while attempting to exit through the cladding. This second effect is called multimode operation.

#### Multimode Operation

With fiber core sizes of 50, 62.5 and above, multimode operation will be experienced. Not only will the light transfer down the axis of the fiber, but it will also travel away from the axis and toward the cladding. The cladding helps reflect the light rays back toward the fiber axis. The cladding provides this effect because it has a lower index of refraction than the core.



**Figure 2—Fiber optic cable is available as paired cable (duplex cable) with an appearance similar to “zip cord.”**

### Index of Refraction

The index of refraction of a material ( $n$ ) is defined as the ratio of the speed of light in a vacuum compared to the speed of light in the material. When light passes from one material to another with a different density, part of the light will be reflected and the remainder refracted. The angle of the refracted ray will be different from the incident wave and will obey Snell's Law:

$$n_1 \sin \theta_1 = n_2 \sin \theta_2$$

Where  $n_1$  and  $n_2$  are the corresponding indexes of refraction and the two angles are measured relative to a perpendicular axis to the boundary of the two materials. At some angle called the critical angle,  $\theta_2$  becomes  $90^\circ$ . For all values of  $\theta_1$  greater than the critical angle, total internal reflection will occur. This is the fundamental principle of fiber optic communications. The light energy is constrained to the inner core. The cladding with its lower index of refraction provides the total internal reflection necessary for proper operation.

### Multimode Signal Distortion

In multimode operation, light waves travel down the axis of the fiber as well as a zigzag course bouncing off the cladding. Since some of the light rays take a longer trip when they exit the far end of the core (due to its zigzag course), distortion of the original signal will occur as it recombines with the light ray that took the shorter path down the axis of the core. This results in pulse broadening at the receiver end. This distortion is called modal dispersion because the paths of the light rays are at different lengths. To counteract this multimode phenomenon, graded-index fiber was developed.

With graded-index fiber, the index of refraction is highest along the center axis of the fiber and gradually decreases from the axis to the circumference. Light travels slower with a higher index of refraction and faster with a lower index of refraction. With this approach, the light that travels down the center axis is deliberately slowed to match the time required for light to travel a zigzag course nearer the circumference. The result is less distortion and higher bandwidth.

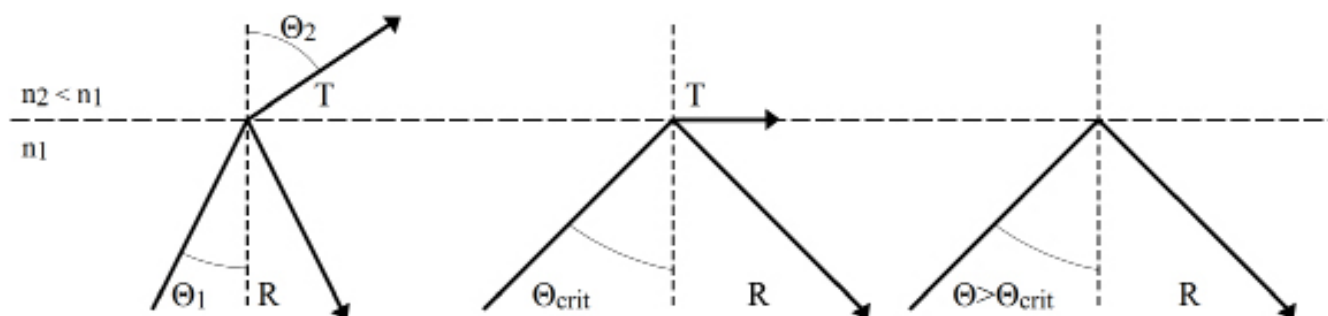


Figure 3—Total internal reflection occurs when the incident angle exceeds the critical angle.

Bandwidth requirements are generally not an issue with Ethernet. Multimode fibers have bandwidth specifications in frequency-distance units (Mhz-km) that depend upon the operating wavelength. Doubling the distance halves the signaling rate; however, even at minimal bandwidth specifications (160 Mhz-km or so), the attenuation limitations of increased fiber length will be met before the bandwidth limitations – except when approaching gigabit speeds.

Lower bandwidth fiber exists with a 200  $\mu\text{m}$  core diameter. This is step-index fiber meaning that only one index of refraction exists in the core and another in the cladding. This fiber is intended for shorter runs and is easier to connect and is more resilient to physical abuse due to its larger core size. This fiber is found in plant floor applications but is not recommended for Ethernet.

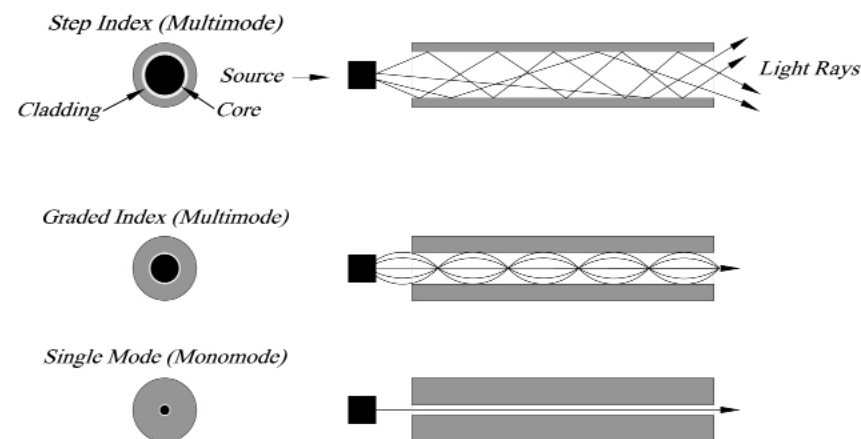
### Operating Wavelengths

Fiber optic transmitters and receivers are generally classified to operate in either of three frequencies. These frequencies have been found to have the lowest attenuation across a band of frequencies. The regions of lowest attenuation are called windows. The particular frequencies the industry uses are 850 nm, 1300 nm and 1550 nm. The two shorter wavelengths offer cost/performance trade offs that are of interest in Ethernet applications. The 850 nm technology is readily available at the lowest cost. However, fiber optic cable attenuation is higher in the 850 nm band than in the 1300 nm band and the bandwidth is less. This attenuation is what mostly limits the fiber optic segment lengths when using Ethernet. The 1300 nm receivers and transmitters are more costly but are recommended when long distances are to be encountered or 100 or 1000 Mbps operation is required. The 850 nm technology is generally used with multimode

applications, while the 1300 nm technology is used with either single-mode or multimode operation. Because of cost, the 1550 nm technology is not as popular with Ethernet.

### Fiber Optic Transmitters

Both 850 nm and 1300 nm fiber optic transmitters can be found in hubs, switches and network interface modules (NIMs), and the two technologies cannot be mixed. These transmitters are available with ST, SC or MIC connectors. The ST “straight tip” connector operates similar to a small coaxial BNC connector. It prevents over-tightening and provides repeatable insertion loss. The SC connector is a low-cost, snap-in connector while the similar style MIC connector



**Figure 4—Step-index fiber has the lowest bandwidth, while single-mode fiber offers the highest.**

was originally intended for Fiber Distributed Data Interface (FDDI) applications. In fiber optic implementations, a separate transmitter and receiver are usually used instead of a transceiver.



Fiber optic links use a duplex cable for NIM-to-hub and hub-to-hub connections. A transmitter at point A connects to a receiver at point B. Point B's transmitter attaches to point A's receiver. Therefore, a crossover function must be accomplished in the cabling. Transmitters and receivers may or may not be color-coded so care must be exercised to pair a transmitter to a receiver.

#### ***Transmitter Power***

Transmitters are rated in dBm with 0 dBm corresponding to 1 milliwatt of power. Transmitter output can vary from device to device, so it is important to 100% test transmitters to ensure that none are shipped below the minimum specified in the Ethernet standard. Testing is usually accomplished by applying a square wave signal and measuring the average power with an optical power meter. Transmitter output also depends upon the fiber size. More energy is launched into larger fiber sizes; therefore, a power rating shown in a specification is based upon a particular core size.

#### ***Receiver Sensitivity***

Receiver sensitivity is also rated in dBm and is based upon receiving the same square wave signal generated by the transmitter. Typically only a maximum sensitivity rating is given which represents the weakest signal discernible by the receiving electronics. Separate receivers are required for 850 nm and 1300 nm operation. Receiver sensitivity is typically the same over a batch of receivers and does not exhibit the same variability as transmitters.

#### ***Optical Power Budget***

When specifying a fiber optic installation, attention must be paid to the available optical power budget. The power budget is the difference between the light source strength minus receiver sensitivity expressed in dB. This value must be compared to the link loss that is the total attenuation due to optical cable and optical connectors. The link loss must be less than the power budget. The difference is called the power margin that provides an indication of system robustness. Optical power budget allowances vary with the media type selected and the Ethernet data rates.

#### ***Link Loss***

To determine the link loss, all losses due to fiber length and cable connections must be summed. Fiber optic cable attenuation is usually specified by the cable manufacturer. Use this figure to determine the attenuation for a particular length of fiber cable. It is also necessary to include losses due to cable terminations. Connectors usually create a loss of from 0.5 to 1 dB for each connection.

For example, assume a 1500-meter run of 62.5  $\mu\text{m}$  cable which the cable manufacturer specifies as having a cable attenuation of 3.5 dB per 1000 meters. The cable loss would therefore be 5.25 dB. Assume there are two connector losses of 0.5 dB each for a total of 1 dB. The link loss would therefore total 6.25 dB. If the light source produced  $-20$  dBm and the receiver sensitivity is  $-30.4$  dBm, then the power budget would be 10.4 dB which is greater than the link loss by 4.15 dB. This difference would represent a high degree of margin since a 3 dB margin is what is typically recommended to account for aging. Recommendations on acceptable attenuation values can be found in TIA/EIA-568-A Commercial Building Telecommunications Cabling Standard.

### **Overdrive**

Overdrive occurs when too little fiber optic cable is used resulting in insufficient attenuation; thereby, saturating the receiver. To correct this condition, a longer length of fiber optic cable must be installed between the transmitter and receiver. This is potentially a problem with larger core cable. Another solution is to have a receiver with a wide dynamic range. Over this range, the receiver will accept varying levels of signal without overload. Frequently you will find a minimum cable length for Ethernet.

### **Delay Budget**

People frequently assume that with fiber optics, signals propagate at the speed of light. This is not true. In fact, the propagation factor is  $0.67c$  or 5 ns/m which is slower than an electrical signal over coaxial cable. The delay through cables and hubs is an issue for shared Ethernet systems that operate over half-duplex links and must obey the rules for collision detection. It is not an issue for full-duplex links which avoid collisions altogether.

### **Ethernet Standards**

Ethernet standards are published in ISO/IEC 8802-3:2012 which is also known as IEEE Std 802.3, 2012 Edition. This is an evolving standard with information on 10, 100, 1000 Mbps operation and beyond. This is a very complex standard and is over 3700 pages long in six parts. From the standard we will review those portions dealing with fiber optics.

### **FOIRL**

The Fiber Optic Inter-Repeater Link (FOIRL) was the original fiber optic specification. It was intended to link two repeaters together with a maximum of 1 km fiber optic cable while operating at 10 Mbps. This standard has been superseded by the 10BASE-FL specification.

### **10BASE-F**

The 10BASE-F standard is actually a collection of fiber optic standards for 10 Mbps operation. It consists of three separate standards—10BASE-FL, 10BASE-FB and 10BASE-FP. It is not sufficient to claim 10BASE-F compatibility because of these three specific implementations. The -FB and -FP standards are not popular and will not be discussed.

10BASE-FL

This standard is the most popular 10 Mbps fiber implementation. The standard calls for a maximum segment length of 2 km of multimode fiber optic cable and a minimum length of 0 km. This means that the transmitter cannot create an overdrive condition. A 10BASE-FL unit must be able to communicate with a FOIRL unit but be limited to 1 km. Connectors are the ST-style and a segment consists of a pair of cables; thereby, allowing for full-duplex communication. The operating wavelength of the receivers and transmitters are 850 nm allowing for the less expensive components. The minimum average transmit level is -20 dBm while the maximum is -12 dBm. The receiver must be able to distinguish a -32.5 dBm signal and not overload from a -12 dBm signal. That means that the receiver's dynamic range must be at least 20.5 dB and that the power budget must be 12.5 dB. The intention is to use 62.5/125 fiber optic cable. If a larger core is used, more energy will be launched which could cause overdrive on short runs. Manchester encoding is used just like 10BASE-T.

100BASE-X

Like 10BASE-F, 100BASE-X is not a unique physical layer, but details the encoding for the two most popular Fast Ethernet physical layers—100BASE-TX and 100BASE-FX. One physical layer is for copper and the other for fiber optics, yet the standard applies to both. Much of the 100BASE-X standard comes from the FDDI standard including the 4B/5B encoding.

4B/5B

Data transfers over the Medium Independent Interface (MII), defined for Fast Ethernet, are done with 4-bit nibbles that represent actual data. With 10BASE-FL, Manchester encoding is used which guarantees a transition within every bit cell regardless of logic state. This effectively creates a 20 Mbaud signal for a 10 Mbps data rate. If the same encoding were used for Fast Ethernet, a 200 Mbaud signal would result making it difficult to maintain the same 2 km maximum segment length due to bandwidth restrictions. A solution is the 4B/5B code where the 4-bit nibbles being transferred over the MII are actually encoded as five-bit symbols sent over the medium. The encoding efficiency is 80% and the baud rate increases to 125 Mbaud. This is still fast but not as fast as 200 Mbaud. The 4B/5B scheme is used for both the 100BASE-TX and 100BASE-FX physical layers.

100BASE-FX

The actual governing specification for 100BASE-FX is ISO/IEC 9314-3 which describes FDDI's Physical Layer Medium Dependent (PMD). The 100BASE-FX fiber optic physical layer is very similar in performance to 10BASE-FL. Maximum segment length is 2 km for both technologies; however, for 100BASE-FX this is only achieved on full-duplex links. On half-duplex links the segment length cannot exceed 412 m. Either SC, MIC or ST fiber optic connectors can be used, but SC is recommended. Multimode fiber optic cable (62.5/125) is what is normally used; however, larger cores can be substituted. Minimum transmitter power is -20 dBm and maximum receiver sensitivity is -31 dBm. The signaling on fiber optics is NRZI (non-return to zero inverted) since there is no concern for EMI on fiber optic links.

	10BASE-FL	100BASE-FX
Data Rates	10 Mbps	100 Mbps
Encoding	Manchester	4B/5B
Fibers	2	2
Cable	62.5/125 μm	62.5/125 μm
Frequency	850 nm	1300 nm
Propagation factor	0.67c	0.67c
Connectors	ST	ST, SC, MIC
Segment Length (max.)	2 km	412 m (half-duplex) 2 km (full-duplex)
Output power (average)	-20 dBm (min.) -12 dBm (max.)	-20 dBm (min.) -14 dBm (max.)
Sensitivity (average)	-12 dBm (min.) -32.5 dBm (max.)	-14 dBm (min.) -31 dBm (max.)

Table 1 —The two popular fiber physical layers are the 10BASE-FL and 100BASE-FX.

With 100BASE-FX, 1300 nm technology using LED transmitters is used and since communication between 850 nm devices does not exist, there is no support for the Fast Ethernet Auto-negotiation scheme. For 100 Mbps operation, the fiber optic cable must have a minimum bandwidth of 500 Mhz-km. This does not necessarily require a cable change since the same fiber optic cable used at 10 Mbps (160 Mhz-km at 850 nm) will have the necessary bandwidth at 1300 nm. Therefore, the 2 km maximum segment length can be maintained if operating as a full-duplex link.

It is interesting to note that both 10BASE-FL and 100BASE-FX only specify multimode cable. The use of single-mode cable is vendor specific. Therefore, it is best to match the same vendor equipment at each end of the single-mode link and observe maximum segment lengths. Distances of 15 km are common but full-duplex operation is a necessity.

### **100BASE-SX**

The 100BASE-SX standard was released as a low-cost upgrade in performance from 10BASE-FL systems. It is basically the 100BASE-TX standard, but utilizes 850 nm devices and ST connectors. Segment lengths are limited to 300m, but Auto-negotiation of data rates is possible with other 100BASE-SX compatible devices.

### **1000BASE-X**

This is an encoding standard for Gigabit Ethernet which is based upon the ANSI X3T11 Fiber Channel standard. Three media segments are introduced – two for fiber and one for copper. The 1000BASE-CX copper segment is not popular. The 1000BASE-LX media segment is for long-wave transmission and the 1000BASE-SX is for short-wave transmission. Instead of 4B/5B encoding found in the 100BASE-X standard, the 1000BASE-X standard introduces 8B/10B encoding where 8-bit bytes are turned into 10-bit code-groups but instead of using all possible 1,023 code-groups only 256 code-groups are used. One of the code groups is the IDLE signal which is continually sent when no other data is present. Needless to say, understanding the actual encoding scheme is quite complex. What is important to know is that to maintain a 1,000 Mbps data rate requires a 1,250 Mbaud signaling rate which is too fast for LED fiber transceivers. Instead, laser transmitters are used introducing a potential eye-safety issue for those handling fiber optic installations. For Gigabit Ethernet it is the data rate that mostly limits maximum link segment length and not just cable attenuation.

**1000BASE-LX**

This is the long wavelength (1300 nm) standard that can operate with either multimode (50 or 62.5  $\mu\text{m}$ ) or single-mode (10  $\mu\text{m}$ ) fiber optic cable. Link power budgets can be as low as 7.5 dB. Maximum cable distances are 550 m for multimode and 5,000 m for single-mode. Minimum cable distances are 2 m.

There is a 1000BASE-LX/LH long haul fiber variation solely based on single-mode fiber that has a 10,000 m operating distance. There are also some proprietary schemes which can go even further.

**1000BASE-SX**

This is the short wavelength (850 nm) standard that only operates with multimode (50 or 62.5  $\mu\text{m}$ ) cables. The link power budget is also 7.5 dB. Depending upon the MHz-km rating of the fiber, maximum distances can range from a low of 220 m to a high of 550 m. So it is possible to reuse fiber cabling from older 10 and 100 Mbps installations for Gigabit installations if the much lower distances can be tolerated.

**Gigabit Ethernet Transceivers**

When we discuss fiber optic Ethernet we usually refer to a transmitter and receiver interconnected with a single strand of fiber optic cable using SC connectors – a pair of which constitutes a link. That is still possible with Gigabit Ethernet but there is such a thing as a fiber optic transceiver – the most popular of which are Small Form Factor Pluggable (SFP) compliant. This compact form factor was an outgrowth of the Gigabit Interface Converter (GBIC) described in the earliest 802.3 Gigabit Ethernet standard. The SFP defines an electrical/mechanical socket that is mounted into equipment like an Ethernet switch. By agreement, different vendors will make their switch products with one or more SFP-compliant sockets. Fiber optic transceivers designed for a specific variant of Fast or Gigabit Ethernet are then plugged into one of these available sockets. A compliant SFP transceiver usually accommodates two of the smaller LC-style fiber optic connectors thereby providing a link connection between two pieces of equipment. The crossover function is accomplished by just flipping the connections within the pair. If the larger SC connectors are to be used, a patch cable to LC connectors is required although there is a multiplexing scheme provided by a particular transceiver where a single SC connected fiber can be used as a link segment. There is also a transceiver that can accommodate a single RJ-45 copper connection. SFP provides flexibility in a Gigabit switch when various fiber types, connectors and operation modes need to be provided.

**Summary**

Robust Ethernet networks can be designed using fiber optics supporting the popular data rates of 10, 100 and 1,000 Mbps. By utilizing full-duplex communications, high-speed reliable communication can occur over large distances in a LAN environment.

## References

*Ethernet: The Definitive Guide*, Charles E. Spurgeon, 2000, O'Reilly & Associates, Inc.  
*International Standard ISO/IEC 8802.-3 ANSI/IEEE STD. 802.3 2012 Edition*, The Institute of Electrical and Electronic Engineers, Inc.

*International Standard ISO/IEC 9314-3 Information Processing Systems—Fiber Distributed Data Interface (FDDI)—Part 3: Physical Layer Medium Dependent (PMD)*, 1990.

*Industrial Fiber Optic Networks*, John C. Huber, Instrument Society of America, 1995.



## Introduction

Ethernet, as a fieldbus replacement technology, lacks two attributes found in fieldbuses. The first is bus topology. The second is power sourced from the network cable to energize field devices. DeviceNet is a good example of a fieldbus that can accomplish both. DeviceNet can be wired in a bus topology while providing 24-volt power in the cable for powering field devices such as photo-eyes, push-button stations, and limit switches. Higher-powered actuators usually have their own power sources. Modern Ethernet only supports star topology and, until now, could not provide power over the cable without implementing a non-standard approach. With the approval of IEEE 802.3af in 2003 and its revision to IEEE 802.3at in 2009, the power-sourcing problem has been solved with the Power over Ethernet (PoE) standard. PoE not only provides a safe and effective way of applying power to field devices, but also utilizes its star topology to its advantage by controlling the amount of power each connected device receives while protecting non-powered devices from harm.

## Twisted-pair Cable Carries the Power and Data

PoE was not designed for the automation markets but for a much larger information technology market incorporating IP phones, IP cameras, and wireless access points. An IP phone, which utilizes Voice over IP (VoIP) standards, should look and feel like an ordinary telephone. A telephone is powered from its data connection and works when the power is lost because its 48-volt power source operates from batteries. These same attributes are attractive in automation systems when it is inconvenient or expensive to run higher voltage power in the field or when it is desirable to back up the entire automation system from one power source. Most automation systems operate from 24 volts and not 48 volts so the 802.3at standard must be examined for applicability to the automation industry.

Modern Ethernet cabling, complying with the 10 Mbps 10BASE-T and the 100 Mbps 100BASE-TX standards, consists of four twisted-pairs of which only two pairs are employed. The original 802.3af standard allowed for either unused pairs to carry 48 volt power or the two data-pairs. The Gigabit standard, 1000BASE-T, uses all four pairs so clearly, this would not work. The 802.3at revision provided a unified approach to powering devices up to four data-pairs while addressing the issue of protecting non-PoE compliant devices that are not expecting power from damage. The 802.3at version also significantly increased the amount of power that can be used to power field devices – from 15.4 to 30 watts.

## PSE and PD

The 802.3at standard identifies two types of devices. The Power Sourcing Equipment (PSE) provides the required power just like its name implies. The Powered Device (PD) is the receiver of the power. The two are connected via the Ethernet communications cable. Understanding Ethernet's star topology, the PSE would naturally be assumed to be a port on a hub or switch while the PD would be an end station or node on the network. This type of PSE is called an Endpoint PSE.

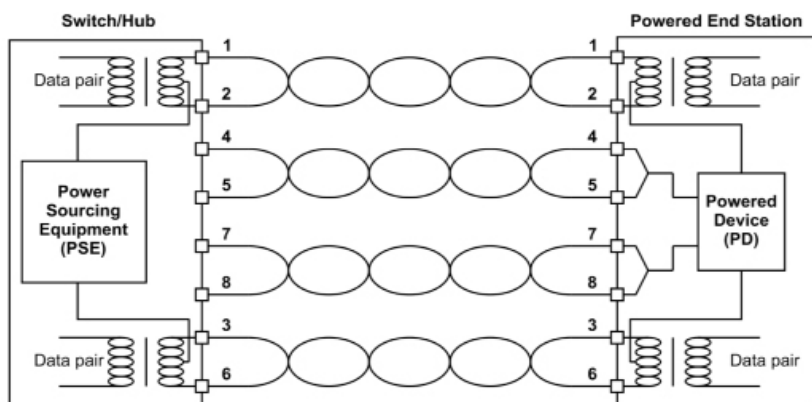


Figure 1—10BASE-T/100BASE-TX Endpoint PSE – Alternative A

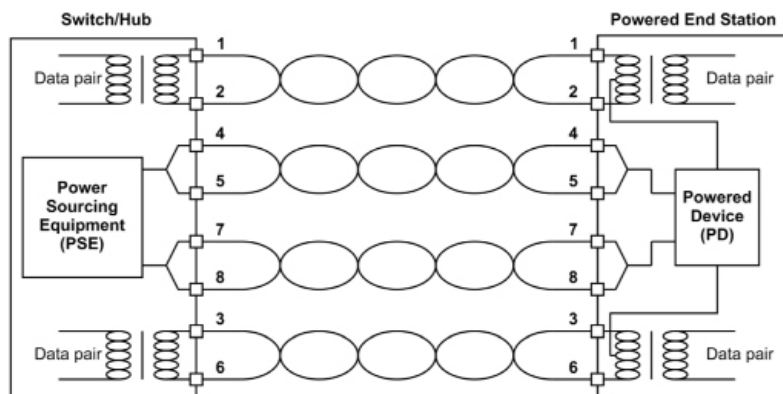


Figure 2—10BASE-T/100BASE-TX Endpoint PSE – Alternative B

With the 802.3at revision addressing 10, 100 and 1000 Mbps twisted-pairs, two definitions of PSEs were introduced. The 10BASE-T/100BASE-TX PSE supports the two data-pair technologies 10BASE-T and 100BASE-TX; while the 1000BASE-T PSE supports both two and four data-pair technologies – 10BASE-T, 100BASE-TX and 1000BASE-T. First we will talk about the two data-pair technologies.

### 10BASE-T/100BASE-TX Power Sourcing Equipment

With two data pairs there are two ways of connecting power as shown in Figures 1 and 2. The first method, Alternative A, is via the data lines while the second method, Alternative B, is via the unused pairs. Using both approaches simultaneously is not allowed. When using the unused pairs, the wires within the pair are connected together in order to increase the cable's current carrying capacity. The same trick can be used with the data lines, but in this case the power must be fed through a center-tapped transformer. If the DC-current flow in each wire of the pair is equal, there will be no DC bias established in the winding of the transformer so this method could be very effective for carrying power and data simultaneously. In fact, this method is the only one possible for using Gigabit Ethernet since no spare pairs exist. A PSE can be designed to source power either through the data lines or through the unused pairs. The PD must be able to do either, making the PD design a bit more complex.

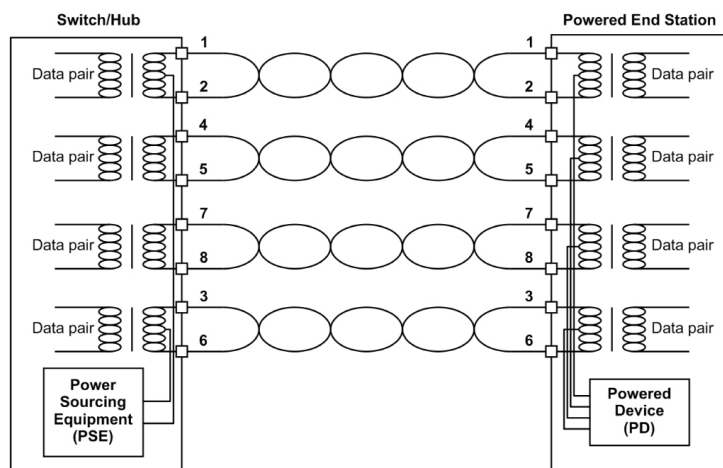


Figure 3—1000BASE-T Endpoint PSE – Alternative A

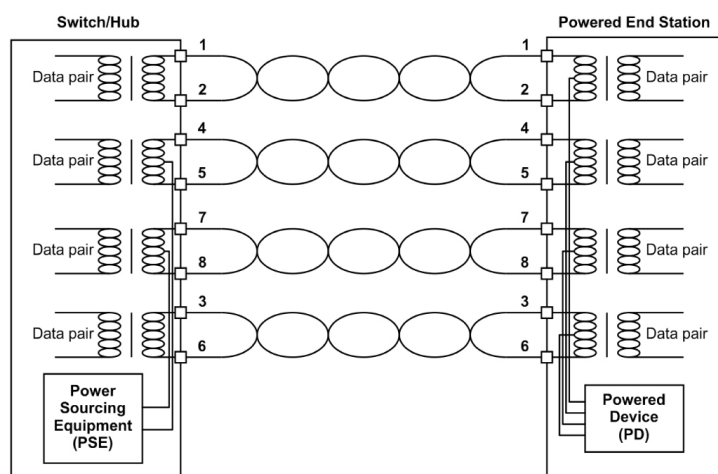


Figure 4—1000BASE-T Endpoint PSE – Alternative B

The power being supplied by the PSE is a nominal 48 volts (44 to 57 volts) although the 802.3at revision tightens the range to be between 50 and 57 volts for higher powered devices. When using the spare pairs for power, the voltage polarity is defined. When using the data pairs for power, the voltage polarity is undefined requiring the PD to be able to accept both polarities and still function. This is because the crossover cable could be between the PSE and the PD. Normally; a straight-through cable is used to connect a switch to an end device. The end device is wired as an MDI while the switch is wired as an MDIX. The "X" means that the switch implements an internal crossover function by having transmitters on the switch connect to receivers on the end device, and receivers on the switch connect to transmitters on the end device. If two MDI-compliant devices need to communicate to one another (an end device to another end device), then a crossover cable is required to make the equivalent connection. Modern switches have a feature called Auto-MDIX where the switch port will adjust to either a straight-through cable or a crossover cable and still function with an end device or another switch port. A crossover cable matches up data pairs by having pins 1 and 2 on one side connect to 3 and 6 on the other. Likewise, pins 3 and 6 are matched up with 1 and 2. Because of this crossover situation, the polarity of the voltage on the data pairs will reverse at the PD. To guard against this reverse voltage condition, an auto-polarity circuit should be used at the PD. Crossover cables do not affect the spare pairs. Therefore, the PD will experience no polarity change. To be safe, it is best to incorporate auto-polarity on the spare pairs as well.

### 1000BASE-T Power Sourcing Equipment

With Gigabit Ethernet, four data-pairs are required so there is no hope for using spare wires. Figures 3 and 4 show the possible connections. Notice that there are still two alternative options that are similar to that of the 10BASE-T/100BASE-TX PSE configurations except that we are dealing only with data-pairs. Alternative A and alternative B mimic the power pin connections of the 10BASE-T/100BASE-TX PSE.

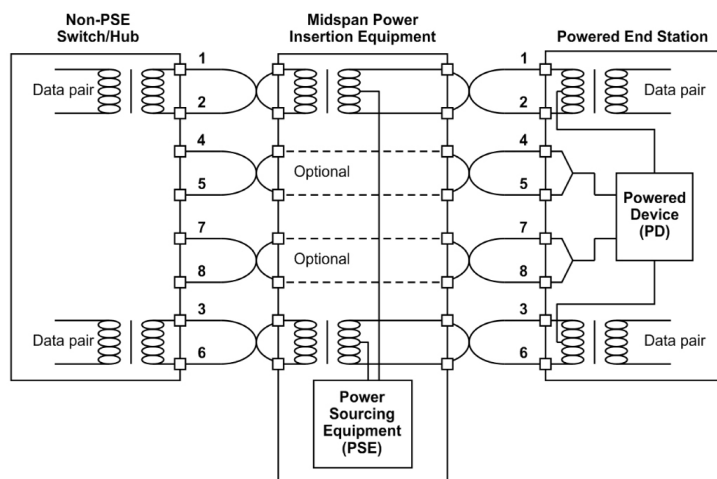


Figure 5—10BASE-T/100BASE-TX Midspan PSE – Alternative A

## Endpoint and Midspan PSEs

There are two kinds of PSEs. The first is an end-point PSE (Figures 1 through 4) where the power sourcing equipment is within the hub or switch. The Ethernet switch now becomes a Power Sourcing Ethernet switch adding complexity and expense. However, field cabling is not disturbed. The standard Ethernet switch is now replaced with a PoE Ethernet switch. In terms of data handling, the PoE switch operates identically to that of a non-PSE switch. Additional circuitry and an adequate power supply are necessary to serve the powered Ethernet ports, but not all ports need to be powered and frequently this is done to reduce power supply requirements. A switch-to-switch cascading connection (uplink) would not need powered ports so non-PoE ports would be present in PoE switches.

The second PSE is the midspan PSE (Figures 5 through 8) where the midspan device can reside anywhere within the 100-meter length of Ethernet cable attaching a conventional hub or switch port to an end station. The advantage of this device is that standard off-the-shelf Ethernet switches can be retained in PoE applications since the power comes from the midspan device and not the switch. In this application, the Ethernet switch is not disturbed, as is the field cabling. The midspan device is not a hub or a switch but mostly resembles that of a patch panel with a conventional Ethernet input port and a corresponding powered output port. The downside of it is that it doubles the connections and introduces another piece of equipment to mount and energize. Midspan devices are beneficial for testing the PoE concept or for operating with legacy equipment during a retrofit.

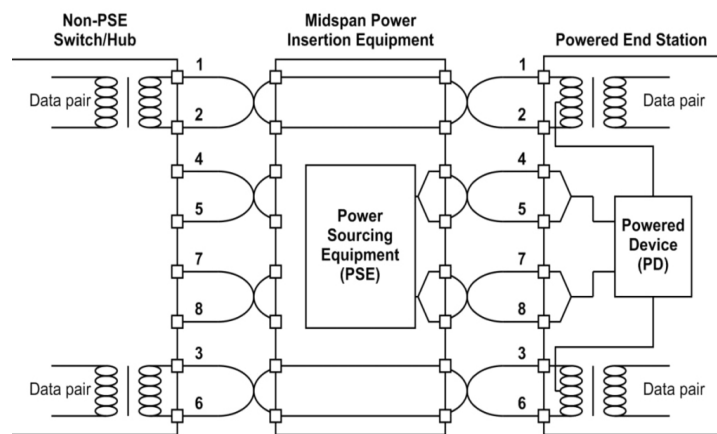


Figure 6—10BASE-T/100BASE-TX Midspan PSE – Alternative B

## Power Management

The original 802.3af standard required a PSE device to be capable of delivering 350 mA at 44 volts as a minimum. This yields 15.4 watts of power at the terminals of the PSE. However, the PD could draw as little as 12.95 watts due to voltage drops over a 100 m span of twisted-pair cable. For most automation applications, 13 watts of power is quite adequate.

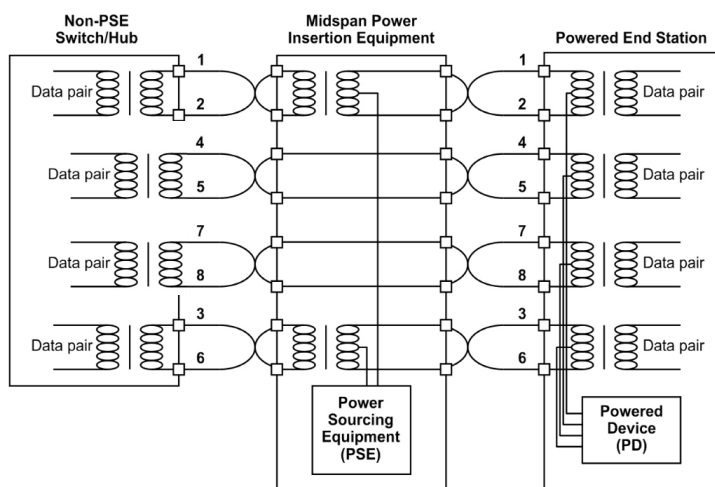


Figure 7 —1000BASE-T Midspan PSE – Alternative A

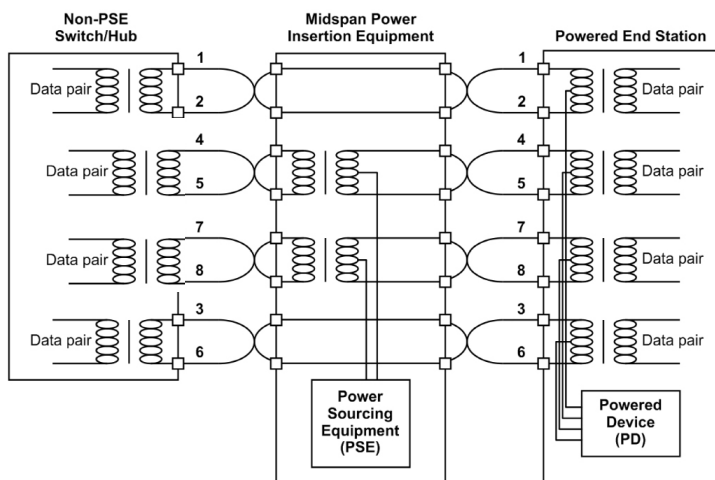


Figure 8—1000BASE-T Midspan PSE – Alternative B

With the 802.3at revision, allowances were made to increase the minimum power sourcing level to be 600 mA at 50 VDC. These 30 watts of power guarantees that the PD will receive a minimum of 25.5 watts at its terminals assuming Category 5 cable is used. Although each port on a multi-port PSE can deliver its 15.4 or 30 watts, it does not mean that the PSE in total needs to be rated for the maximum power at each port. This is where power management comes in. By having PDs report only the power necessary for each attached device, it is possible to undersize the total power supply requirement of the PSE.

### Device Identification

In order to protect non-PoE compliant devices from receiving unexpected power, an identification process is performed during connection time. Unattached PSE ports are un-powered in their dormant state. When a device is first attached to a PSE port, it must present itself as a nominal 25k ohm resistance (within the range of 15-33k ohms). This is called its Detection Signature, which the PSE tries to identify by applying up to 10 volts while measuring the current. If the attached device fails the identification process, no PoE power is applied. If the device is identified as a PD, a second process called classification may or may not occur that determines what power is reserved for the PD.

### Classification - Type 1 and Type 2 PSE and PD Devices

With the increased power allowed with the revised 802.3at standard comes a more complex method of communicating the ratings of attached equipment in order to implement power management. The resulting scheme allowed legacy 802.3af devices to co-exist with newer more sophisticated 802.3at devices. There are two types of PSE and PD devices. A Type 1 PSE complies with the original 802.3af standard that allows the sourcing of up to 15.4 watts of power based upon a Physical Layer Power Classification chart which defined classes 0-3. The Type 2 PSEs support the 802.3at revision that defined a class 4 device which was a reserved class in the original specification. See Table 1 for the classification chart.

Physical Layer Power Classification - P <sub>Class</sub>	
0	15.4 watts
1	4.0 watts
2	7.0 watts
3	15.4 watts
4	30.0 watts

**Table 1—The 802.3at revision defined classification level 4.**

Likewise, a Type 1 PD with associated cabling can consume up to 15.4 watts while a Type 2 PD with its cabling can consume up to 30 watts. To make things a bit more complex, the 802.3at standard allows for the mutual interrogation of a PSE and a PD using the Link Layer Discovery Protocol (LLDP). This is called *Data Link Classification* and it is used for dynamic power management after detection and initial classification. The 802.3af version only allowed the PSE to classify the PD at the physical layer or not bother at all. If a Type 1 PSE did not implement classification after a successful detection, it would be required to provide the full 15.4 watts of power to the PD. This was what simple PSEs would do that did not support power management and this is still allowed within 802.3at. This is what is called Class 0 operation – no power management. The 802.3at revision also introduces 1-Event and 2-Event physical layer classification schemes which relate to detecting high power devices.

Classes 1 through 4 offer a range of power management as shown in Table 1. The PSE will apply a higher voltage (15.5-20.5 volts) in order to determine the class of device by measuring the resulting current. This current signifies the Classification Signature identifying the P<sub>Class</sub>. If none of the anticipated currents are measured, the device is deemed to be a class 0 device and no power management is possible. In either case, the PSE output voltage is then increased to its required output range. For a Type 1 device this is between 44 and 57 volts and for a Type 2 device this is between 50-57 volts.

### Injectors and Splitters

Some PoE schemes were developed before the 802.3af standard was ratified and are still in use. These schemes may or may not be compliant to the standard. They utilized the spare pairs to power legacy devices, however, the concept is still applicable today. Usually an injector is used to apply power to an Ethernet segment, and a splitter is used to extract the power from the segment before it appears at a non-PoE end station. To follow the standard, the injector must apply a nominal 48 volts, however, products exist that provide non-compliant voltages (such as 24 volts). There are two types of splitters. The passive splitter simply removes the power before the end station and then feeds it directly to the power input of the legacy device. The regulated splitter will adjust the voltage on the cable to exactly match the requirements of the legacy device. This approach can be very effective with single station applications, however, to be truly compliant to the 802.3at standard, the tap must participate in the signature detection process and the injector must comply with all the requirements of a midspan PSE. This may not be the case.

### Summary

Power over Ethernet involves more issues than simply defining the cable connections, and the 802.3at effort provides valuable guidance on how it should be accomplished. Power over Ethernet makes a significant step towards making Ethernet a fieldbus replacement and with the addition of the higher powered 802.3at standard larger loads can now be supported.

### References

*IEEE Standard for Ethernet*, IEEE Std 802.3-2012, Institute of Electrical and Electronic Engineers, Inc.



# Internet Protocol (IP)

## Introduction

The push to incorporate Industrial Ethernet or even “plain vanilla” Ethernet into control networks implies that by making that choice completes the selection process. As mentioned in a previous article, Ethernet II and IEEE 802.3 are strictly data link layer technologies which do not guarantee the delivery of messages over a network or between networks. Protocol stacks such as TCP/IP or SPX/IPX provide that functionality and without them Ethernet would be useless. With the immense interest in the Internet and the potential of attaching control networks to the Internet, the protocol stack of choice is TCP/IP because it provides the foundation for the Internet. This article addresses issues related to the IP portion of the TCP/IP stack as it applies to control networks.

## The TCP/IP Stack

Actually TCP/IP is a set of protocols defined by a series of RFCs (request for comments) that have evolved over the years. In general the Internet Protocol (IP) is used to route messages between networks and, therefore, properly resides at the network layer of the OSI Reference Model. Transmission Control Protocol (TCP) sits on top of IP and is used to guarantee the delivery of messages. Above TCP is the application layer. The services of the presentation and session layers of the OSI Reference Model are incorporated into the application layer. Therefore, the reference model for TCP/IP-based systems actually consists of only five layers. Technologies such as Ethernet II and IEEE 802.3 reside at the lower data link and physical layers of the same model.

## Data Encapsulation

The data sent over wires is represented as frames. An Ethernet II frame consists of a preamble, source and destination addresses, type field, data field and a frame sequence check field. You can lump these fields into Ethernet header, data and trailer fields. The IP data sits above the data link layer and its data, called a datagram, is inserted into the data field of the Ethernet frame. The datagram has its own header and data fields but no trailer field. Above the IP layer is the transport layer where TCP and User Datagram Protocol (UDP) reside. Data from this layer is likewise applied to the data portion of the IP datagram. TCP applies segments while UDP applies datagrams. Both TCP and UDP have headers as well. Finally above the transport layer is the application layer which needs to insert its own data into the data portion of the transport layer as well as its own header.

This application data is simply referred to as data since there is no defined structure in terms of the TCP/IP stack. That is why if two application data structures are different, communication between these applications will not be effective even with strict adherence to TCP/IP standards.

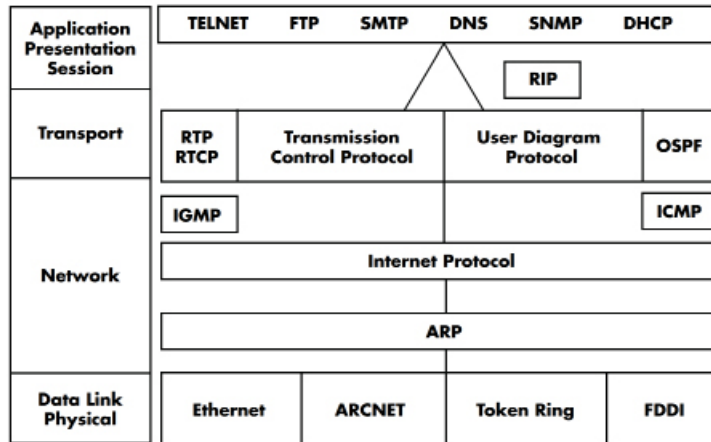


Figure 1—The TCP/IP stack is actually a set of protocols. IP resides at the network layer of the OSI Reference Model shown on the left.

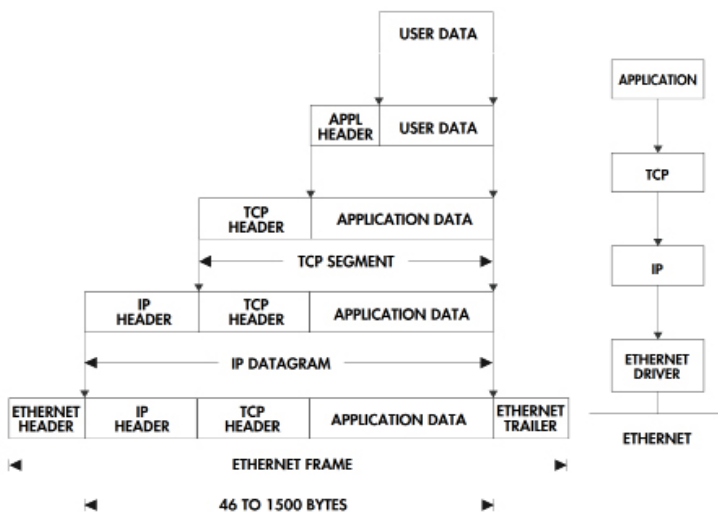


Figure 2—The wrapping of data into the data field of the next immediate lower layer is called encapsulation.

This wrapping of data within the data field of the next immediate lower layer of the protocol stack is called encapsulation while the unwrapping of the same data at the receiving side is called demultiplexing. In order to reduce confusion on what is the actual data we will say that frames are sent over the data link layer. The IP sends out datagrams to the data link layer in the form of packets. A packet can be a datagram or a fragment of a datagram. The TCP sends segments while the UDP sends datagrams. Finally, the application sends data. To further add to the confusion, the terms packet and frame are sometimes used interchangeably.

### The Internet Protocol

The Internet Protocol provides the basic unit of data transfer, provides addressing, routing and fragmentation. The Internet Protocol resides at the network layer and sends and receives blocks of data called datagrams received from upper layer software. IP feeds these datagrams to its attached data link layer which sends and receives these datagrams as a series of packets. A datagram is analogous to a first-class letter sent in the Post. In general, it will reach its destination but there is no formal acknowledgment that the letter was received like there would be with either registered or certified mail. IP utilizes a "best effort" or "connectionless" delivery service between source and destination addresses. It is connectionless because there was no formal session established between the source and destination before the data was sent. Packets can be lost as they traverse the network or networks thereby corrupting datagrams. It is not the responsibility of IP to guarantee the delivery of messages and, therefore, IP is frequently termed an unreliable delivery service. That may be a little harsh of a criticism of IP but it is the responsibility of the transport layer and not the network layer to guarantee end-to-end message delivery. IP is simply responsible for the addressing and routing of datagrams.

Address Identifier	Network Address	Host Address
Class A		
0	7 bits of network address	24 bits of host address
First byte		Last three bytes
Class B		
10	14 bits of network address	16 bits of host address
First two bytes		Last two bytes
Class C		
110	21 bits of network address	8 bits of host address
First three bytes		Last byte
Class D		
1110	Multicast address in the range of 224.0.0.0 - 239.255.255.255	
Class E		
11110	Class E - Reserved for future use	

**Figure 3—Address classes define the split between network and host IDs.**

### Routers and Hosts

Unlike repeaters that operate at the physical layer and bridges that operate at the data link layer, routers operate at the network layer. A router is used to interconnect two networks together to form an internet. An internet is a general term used to denote a collection of networks. It is not to be confused with the Internet which is the public network that requires strict addressing standards in order for different systems to communicate. With a control network, we may want to keep it completely private and not connect it to the Internet or the corporate internet (sometimes called an Intranet) but if we do we will need a router. This is being mentioned here because IP is a routable protocol and routers are used to implement the protocol.

The end-to-end devices on the internet are called hosts. If two hosts are on the same local network, then messages are routed directly involving no routers. If the two hosts are on different networks, a router must pass the message. This is called indirect routing.

### IP Addressing

The IP is responsible for source and destination addresses and its structure is defined in RFC 761. IPv4 is the most common version of addressing and it uses 32-bit addressing. The newer IPv6 calls for 128-bit addressing and was developed because the explosive growth of the Internet will soon deplete the inventory of possible 32-bit addresses. IPv6 will not be discussed here since there is ample confusion in simply discussing 32-bit IP addressing.

An IP address must not only address a particular host but a particular network as well. The IP address must not be confused with the Ethernet II address which is a 48-bit address sometimes called the MAC address. The MAC address is used to facilitate communication only at the data link layer. The IP address facilitates communication over networks and must be universally recognized, even if the host is an Ethernet II node attached to a local area network or a serial port attached to a modem.

The format of the address is <netid, hostid> but is shown as one 32-bit address split up as four bytes. However, each byte is shown as a decimal number from 0 to 255. Therefore, an IP address is usually represented as XXX.XXX.XXX.XXX. This address can be shown as a binary or hexadecimal number as well but the decimal-dot-decimal notation is the most popular. Therefore, the range of addresses is from 0.0.0.0 to 255.255.255.255. An example of an address would be 128.8.120.5 but looking at the address it is hard to tell which is the network address and which is the host address.

There are five classes of IP addresses: A, B, C, D, E. Class D is for multicasting, a message from one host to many hosts, and class E is reserved for experiments. That leaves classes A, B and C which are the most important. These three classes break up the 32-bit address field into defined address ranges for the netid and hostid. You need to examine the very first byte of the IP address to determine the class. If the first bit of this byte is a '0' then this is a class A address. In a class A address the first byte identifies the network and the remaining three bytes identifies the host. That means you can have 16,777,214 hosts for every network!

If the first two bits of the first byte are a "10," then this is a class B address. With class B addresses the first two bytes identify the network and the remaining two bytes identify the host. This provides a slightly more reasonable 65,534 host addresses.

If the first three bits of the first byte are a "110," then this is a class C address. With class C addresses the first three bytes identify the network and the remaining byte identifies the host. This provides a reasonable 254 hosts.

Class D and class E addresses can be identified in the same way. A class D address has a leading bit pattern of "1110" while a class E address has a leading bit pattern of "11110."

Class A :	1-127
Class B :	128-191
Class C :	192-223
Class D :	224-239
Class E :	240-254

**Figure 4—The class of an IP address can be quickly identified by observing only the first byte.**

There are also other reserved addresses. Regardless of class, a host address of all 1s is reserved for a broadcast message to all hosts on that network while a host address of all 0s is reserved to mean "this network." Network address 127 is also reserved and is used for loop-back testing. This effectively wastes 16 million possible host addresses. Network address 0 is reserved as well.

If the control network is to become part of the public Internet then strict adherence to the class addressing rules must be followed. Usually these addresses will be issued by the corporate network administrator or by an Internet Service Provider (ISP). But what if the control network is to become strictly a private network? Cannot any addressing scheme work? Yes, any address scheme could work but there is even an RFC guideline for this situation. According to RFC 1918, only non-routable IP addresses should be used.

These addresses, which a router will not pass, are as follows:

10.0.0.0 to 10.255.255.255

172.16.0.0 to 172.31.255.255

192.168.0.0 to 192.168.255.255

### IP Header

IP transmits and receives data-grams. Within the datagram is a header and the data portion of the datagram. The minimum size of the IP header is 20 bytes consisting of five 32-bit words. The first three words provide control information while the remaining two words provide address information. An optional field can follow the address information. The information in the header is as follows:

**Version:** A four-bit field identifies the IP version. A 4 identifies IPv4 while a 6 identifies IPv6.

**Header Length:** A four-bit field indicates how many four-byte words are in the header. The header length cannot exceed 60 bytes thereby allowing 40 bytes for options.

**Type of Service:** Of the eight-bit field only six bits are used. The Delay bit indicates the datagram should be processed with low delay by the router. The Throughput bit requests high throughput while the Reliability bit requests high reliability. There are three other bits to indicate precedence. These bits are set at higher layers of the protocol stack and are suggestions given the router. This looks like a nice feature for control networks since control networks require low delay and high reliability. However, it is not clear that routers even look at these bits. It appears that this was a feature with great promise but never really implemented. This is to be rectified in IPv6.

**Total Length:** The total length of the datagram including the header cannot exceed 65,535 bytes. This 16-bit field is for the datagram itself and not the packet length in the data link layer. If this datagram is larger than the maximum packet length that can be sent, the datagram will need to be fragmented into manageable successive packets. In this case the total length field will represent the length of the fragment sent and not the length of the original datagram.

**Datagram Identification:** A unique 16-bit identifier assigned by the host will accompany the datagram. This is necessary in order for the receiving host to reassemble fragmented datagrams. All fragments will contain the same datagram identifier.

**Flags:** Three bits are reserved for flags but only two are used. The Don't Fragment bit tells the router not to fragment the datagram. If this cannot be done an error message is returned. The More Fragments bit is used in the fragmentation process. A 1 means that the datagram being sent is actually a fragment of a larger datagram. A 0 means that either the datagram is not fragmented (first and only datagram) or it's the last fragment. Receiving hosts need this information in order to reassemble fragments.

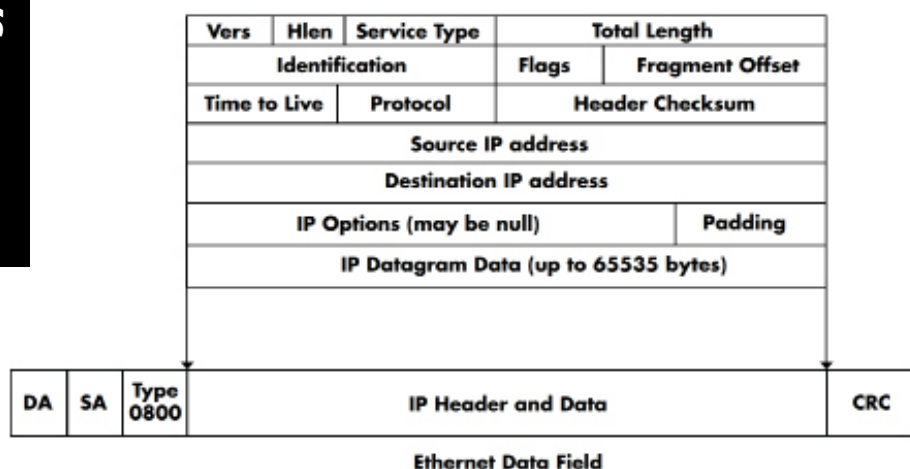


Figure 5—The IP datagram consisting of a header and data is inserted into the Ethernet data field.

**Fragment Offset:** Thirteen bits are used to indicate which fragment is being sent. Fragmentation is the process of breaking up large data-grams into manageable packets. Ideally you would like to restrict datagram size to packet size in order to avoid fragmentation. With Ethernet II the maximum packet size is 1500 bytes. This is called its Maximum Transmission Unit (MTU) and within a private or local network the MTU is known and can be adhered to. The problem occurs between networks. Intermediate networks may have a lesser MTU requiring the router to fragment the original message even though it was originally sent unfragmented. The router does the fragmentation on its own (as long as the datagram was not marked as "do not

fragment") and the fragments must be recombined at the destination host. Routers do not recombine fragments.

The default MTU is 576 bytes and all routers must be able to handle that size transmission. By restricting the datagram to 576 bytes, it will never need to be fragmented. Of course that puts an undue restriction on the Ethernet II network since packets can be as long as 1500 bytes. So for local networks set the maximum datagram size to the local network's MTU. If the datagram is to be sent beyond the local network set the maximum datagram size to 576 bytes. For control networks, fragmentation may never be an issue since control information packets are usually short not exceeding 256 or 512 bytes. Fragmentation should be avoided since it increases data latency and increases the chances of a corrupted datagram since multiple packets must be sent per datagram.

If fragments are to be sent it is necessary to load in the fragment offset. Notice that with every fragment the IP header is resent with just a slight modification. The fragment offset will change on every fragment and possibly along with one flag bit. Fragments must be sent in eight-byte multiples because there are only 13 bits available for identifying fragments and datagrams can be 64KB in length. For example, if the first fragment is 1024 bytes long, the fragment offset of the next fragment will indicate that the accompanying fragment begins the 1025th byte of the original datagram.



## What Defines TCP/IP?

The TCP/IP stack and its associated protocols are described in Request for Comments (RFCs). There are about 2700 RFCs in existence and unlike many industrial control standards these are free! You can simply download them from the Internet. One possible location is <http://www.ietf.org>. Which RFCs do you need? Matthew Naugle, in his book *Illustrated TCP/IP*, suggests as mandatory reading RFCs 1122, 1123 and 1812. These will give a good overview but you can always seek out others once you find an index. You can also author your own RFC by following the instructions and format in RFC 1543. Most of the RFCs originate from the working groups (WGs) of the Internet Engineering Task Force (IETF). Some RFCs obsolete prior RFCs. If your control strategy is based upon the TCP/IP protocol, it is recommended that you document which RFCs are important to your system. These documents might be the only documents available that define your system's compliance.

With knowledge of the datagram identifier, fragment offset, the source IP address and the fragments themselves, the complete datagram can be reassembled by the receiving host even if the fragments are received out of order. That is the true strength of the IP. Packets can take different routes to the intended destination and still be reassembled into the original datagram.

**Time to Live:** This eight-byte field is strictly used by the routers to prevent a datagram from a faulty transmission sequence to endlessly circulate around an internet. Originally the unit of measure was seconds because it was believed that it would take a router one or more seconds to process a datagram from its queue. Once the datagram was processed, the router would decrement this field by the amount of time that occurred. However, in practice modern routers are much faster than early routers and usually process the datagram within a second but only decrement the field by one (the minimum amount). Therefore, the field has come to be treated as a hop counter. A hop being an instance of a datagram being processed by a router. The originating host sets the Time to Live field and each router decreases it by one. If a router decrements the count to zero it will discard the datagram and inform the originating host that the datagram failed to reach its destination.

**Protocol:** The eight-bit protocol field informs the upper layer protocol that the received datagram is for its use. Usually the upper layer protocol is TCP or UDP but there are other protocols as well that could be sending and receiving data. The protocol field provides this distinction.

**Header Checksum:** The complete IP header is checked for integrity with the aid of the 16-bit header checksum. The originating host applies the checksum and all routers check the header for integrity and regenerate a new checksum when the datagram is resent. A new checksum is required since the Time to Live field would have been changed by the router. Finally, the checksum is again reconfirmed by the receiving host.

**Source/Destination Address:** The 32-bit source and destination addresses are included in the header. These are the IP addresses and not MAC addresses.

**IP Options:** There may be no options in which case this field is null or there can be options usually intended for router use only. The option fields must be at least 32-bits in length and must be padded to that amount if shorter.

## ARP

As mentioned before, the IP routes datagrams between source and destination addresses in the form of packets over a data link layer. The data link does not understand datagrams nor does it understand IP addresses. It does know, however, its own MAC address and knows how to communicate to other MAC addresses when told to do so. Somehow we need to inform each host what IP address its MAC address or physical address has been assigned and we need to inform the same host all the other physical address assignments on the local network in order to have communication.



Usually the host IP address-physical address assignment is stored in non-volatile memory or in a file. Using a 32-bit DIP switch for assignment is not practical. Sometimes a serial port on the device is used for programming the IP address but once programmed all other hosts on the local network must still need to learn the assignment. The Address Resolution Protocol (ARP) is used for learning physical address assignments. ARP has its own structure and does not use that of IP. ARP directly communicates to the data link layer and, therefore, must be aware of the various types of network adapters that are available.

When a host needs to send a datagram to another host on a local network, it first checks its ARP table to determine the physical address for that IP destination address. If one is found, the datagram transmission proceeds. If none is found, an ARP request is made. An ARP request consists of a broadcast message to all hosts on the local network. Within the ARP request is the originator's IP and physical addresses as well as the requested IP address. Since it is a broadcast message, all hosts have the opportunity to learn the IP address and physical address pairing of the requestor which can be appended to that host's ARP table. Only the host with the requested IP address responds to the ARP request by providing its IP address and physical address pairing. This message is sent as a unicast message back to the requestor. Once the physical address is known by the requestor, the datagram can be sent.

### Summary

The IP is responsible for the end-to-end delivery of datagrams over an internet. It also provides host and network addressing and the means for fragmenting datagrams into manageable packets. IP is a routable protocol and much of its complexity is due to its ability to route packets directly within a local network or indirectly through routers. Routers are not ideal for a control network since they reduce determinism and increase data latency. Still to accept TCP as a transport layer for an Ethernet control network requires acceptance of IP as well. By understanding the limitations of IP, a control network can still be designed using the TCP/IP family of protocols. This is especially true if the control network is restricted to that of a private or local network.

### References

- Illustrated TCP/IP*, Matthew Naugle, 1998, Wiley Computer Publishing.  
*Practical Networking With Ethernet*, Charles E. Spurgeon, 1997, International Thomson Computer Press.  
*International Standard ISO/IEC 8802-3 ANSI/IEEE std 802.3*, 1996, The Institute of Electrical and Electronic Engineers, Inc.  
*TCP/IP Clearly Explained*, Pete Loshin, 1997, Academic Press.  
*TCP/IP Illustrated*, Volume 1, The Protocols, W. Richard Stevens, 1994, Addison-Wesley Publishing Company.

# Transmission Control Protocol (TCP)

## Introduction

In the previous section we discussed the impact of the Internet Protocol (IP) on control networks. IP resides at the network layer of the OSI communications model and provides the basic unit of data transfer, which includes addressing, routing, and fragmentation. The transport layer of this same model resides above the network layer and provides station-to-station communication and a common interface to the application layer. This implies reliable communication, which is either accomplished at the transport layer or at the application layer. With control networks this is usually accomplished at the application layer since many control networks were designed before the popularity of TCP/IP took hold. Still there are some control network protocols, such as MODBUS TCP, which do rely upon the guaranteed delivery mechanism of TCP and there may be more in the future. Actually, at the transport layer of the TCP/IP stack there are two transport protocols, each of which find service in control networks. The User Datagram Protocol (UDP) and the Transmission Control Protocol (TCP) will both be discussed in this section.

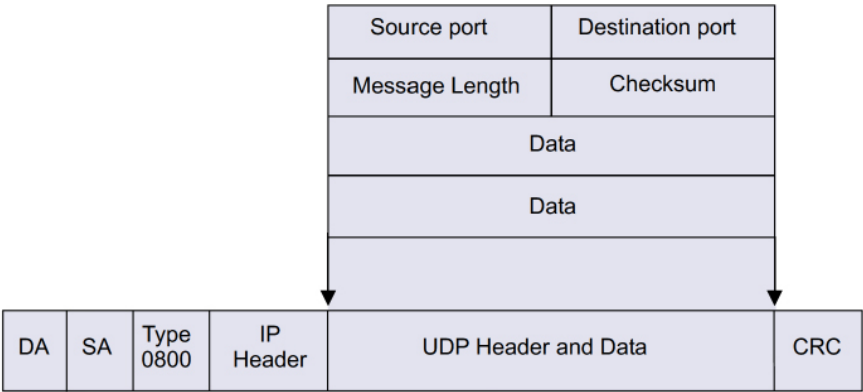
## User Datagram Protocol

UDP provides a connectionless and unreliable transport service since it does not issue acknowledgments to the sender upon receipt of data nor does it inform the sender that data was lost. Data integrity can suffer by dropped packets, mis-sequenced packets or by the receipt of duplicate packets. Any of these situations can occur without the knowledge of the sender. It appears that UDP is no better than the IP protocol but there is one big difference.

UDP introduces the concept of port numbers, which are used by the application layer that resides above UDP. Port numbers have significance in terms of actions requested by the application itself that require a particular response by the receiving station.

The UDP header is short and simple. Only eight bytes are required in the header. Source and destination ports are each 16 bits long and, therefore, require four bytes. The message length of 16 bits indicates the length of the header and attached data. A 16-bit checksum is used to check the validity of the header and data.

The UDP header and attached data, which comes from the application layer, are encapsulated into the IP data field. An IP header, which provides station addressing, precedes the UDP datagram and the complete IP datagram is encapsulated into the frame of the data link layer technology used, such as Ethernet, and sent to the desired station where the complete process is reversed. Notice that the only contribution UDP provided was the assignment of port numbers for use by the application layer. If UDP is to be used, the application layer must worry about acknowledging message receipt, correctly ordering received packets into meaningful messages, discarding duplicate packets and requesting retransmission of faulty packets since UDP does not provide this service.



**Figure 1—The UDP header is quite short and, along with its data, it is encapsulated into an IP datagram.**

## Port Number Assignments

Both UDP and TCP use port numbers and, if possible, the same assignment is used for both. The 16-bit numbers are classified as either assigned (well-known), registered or dynamic (private). Port number assignments are maintained by the Internet Assigned Numbers Authority (IANA) and the complete list can be found at <ftp://ftp.isi.edu/in-notes/iana/> under port numbers. Numbers in the range of zero to 1023 are classified as well-known and are used by common processes. However, individual organizations can register numbers in the range of 1024 to 49151 for proprietary purposes and will not be used by other organizations. Looking at the list, we find some companies in our industry. Opto22 has registered 22000 and 22001 for their SNAP I/O and Opto Control products. The BACnet building automation protocol has registered 47808. DeviceNet and ControlNet intend to use 44818. Interestingly, the MODBUS/TCP specification calls for port 502, which is in the well-known port group. Port numbers 49152 to 65535 are considered as either private or dynamic and can be used by anyone.

However, if the application layer was originally designed to provide this reliability of service there is no reason to have the transport layer duplicate this effort so UDP makes sense. UDP has low overhead and executes quickly making it attractive for control networks.

### Port Numbers

UDP introduces the port number concept. When a station receives a UDP datagram, it serves up the port number to the application layer, which then allocates a buffer area for the attached data. The port number has significance since it identifies a particular application. Since there are many port number possibilities, several different applications can be simultaneously supported on the same station.

Port numbers are 16 bits long and are classified as being assigned, registered or dynamic. Assigned port numbers in the range of zero to 1023 have been defined by the Internet Assigned Numbers Authority (IANA) for various applications that are considered part of the TCP/IP protocol suite. These applications include TELNET, FTP, and other popular Internet applications. These port numbers are termed "well known" and cannot be used by other applications. The remainder of the port assignments is classified as being either registered or dynamic. Registered means that an organization wants to define some level of functionality and have registered a unique port number with the IANA. Other organizations are to respect this assignment and not use either assigned or registered port numbers. Finally, the dynamic port numbers are used in a random manner by a requesting station to signify the source port of an application request.

For example, if station A requests a trivial file transfer protocol (TFTP) service (TFTP happens to use UDP) from station B, it would insert a 69 (a well-known port assignment indicating TFTP services) into its destination port. A dynamic (random but non-conflicting) number will be put in its source port and the request will be sent to station B. Station B would receive the request and recognize it is a TFTP request from station A since port number 69 was used. Station B would then begin executing the process. But it would insert a 69 in its source port, and the dynamic number that was generated by station A in its destination port, and sends the response to station A. Station B knows how to handle this particular application since it recognizes both the source and destination port numbers. The use of port numbers along with the IP address creates what is called a socket, which can be represented as <netid, hostid, portid>. As long as there is structure to the assignment of ports, the socket assignment becomes a unique representation of a particular application on the complete IP network.

## Transmission Control Protocol

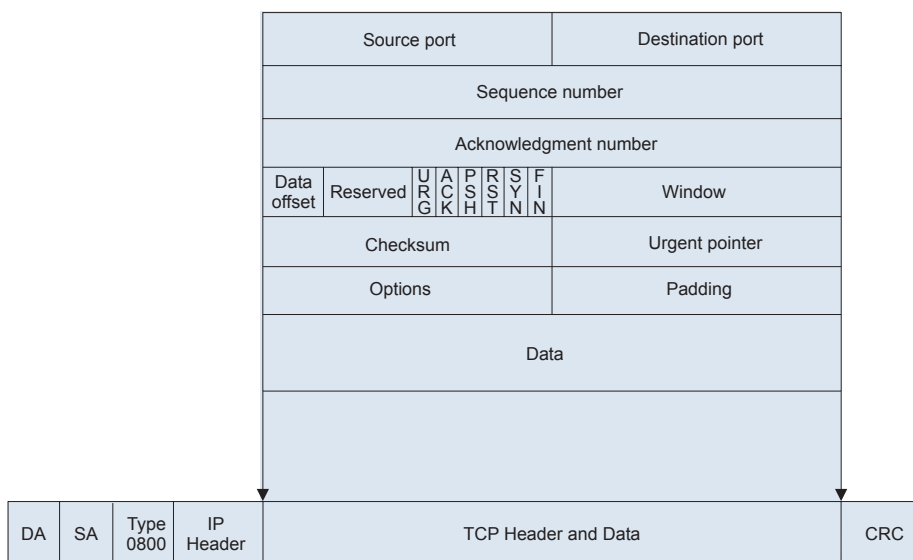
The second transport layer protocol is TCP which provides for a connection-based reliable message delivery service for processes. This relieves the application layer the responsibility of guaranteed message delivery. Besides reliable connections, TCP provides flow control to ensure stations are not flooded with data.

Data transmitted between stations is sent as a series of packets. These packets are reassembled at the receiving end in the proper order to recreate the data stream that was transmitted. Along the way packets can be corrupted, lost, duplicated or received out of order. In order to make sense of all this, sequence numbers are applied to each packet transmission. A sequence number is assigned to the first packet of a message. It does not matter what is the initial value of the sequence number only that the second packet has been assigned the initial sequence number plus one. The rule is that successive packets of a data stream have ascending sequence numbers each incremented by one. After all the packets are received, sequence numbers are used to order the packets. Missing sequence numbers indicate the packet was lost. Duplicate packet numbers

indicate duplicate packets were received allowing them to be discarded. If all the packets are received in tact then the data stream was received properly and the receiving station could acknowledge receipt to the sender. If not, a request for retransmission of the missing packet can be made. There is no need to resend the entire data stream.

TCP provides a byte-oriented sequencing protocol that is more robust than the packet sequence scheme described above. Instead of sequencing each packet, TCP sequences each byte in the packet. Assigning a sequence number to indicate the first byte in a multi-byte packet does this. The second packet will have a sequence number equal

to the first sequence number plus the number of bytes in the first packet. The receiver expects the same. The receiver acknowledges receipt of good packets by transmitting an acknowledgment that includes the value of the next expected sequence number. In this case, it is the sequence number of the last byte in the last received packet plus one. Upon receipt of this acknowledgment, the sender can assume that previous bytes up until the indicated sequence number were received properly. Before a sender can discard the packets being sent it must wait until this acknowledgment has been received because it may be called upon to resend some of the data. Acknowledgments need not require a separate transmission and each packet need not be acknowledged. For efficiency, TCP allows an acknowledgment field to be sent along with data. For example, if station B is responding to station A's request for data, that data can be sent along with an acknowledgment of station A's requesting message. This speeds up processing.

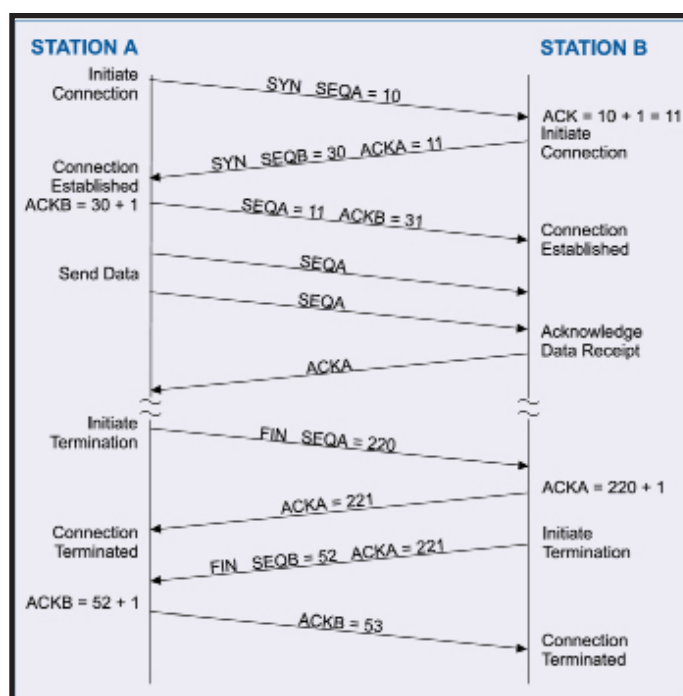


**Figure 2- The TCP header is much more complex than UDP and its length can vary.**

TCP's header is larger than that of UDP but uses the same port assignment scheme as UDP does. Unlike UDP, a 32-bit sequence number and acknowledgment number are included in the header as well as several flag bits. Only a few of the flag bits will be discussed here. Since the TCP header can be of varying length depending upon the content of the options field, a data offset field has been provided in order to determine the actual beginning of the data. The padding field has been provided so that the options field along with the padding field will end on a 32-bit boundary. This is typically done with all the headers within the TCP/IP stack. A window field in the header indicates to the sender how much data the receiver is willing to accept. This feature is used for flow control, which attempts to prevent buffer overflow in the receiver. Finally, data follows the TCP header. The header can be up to 40 bytes in length and the header with its appended data is called a segment. A checksum field ensures the integrity of the header and its associated data.

### Using Connections

When using the TCP protocol with processes a connection must be first established and maintained in order to provide for the flow of data. By establishing a connection between two stations, the proper buffer area is provided for the impending data. When station A wants to communicate to station B it must first establish a connection which allows for the synchronization of sequence numbers and acknowledgments.



**Figure 3—A full-duplex connection must first be established before data can be transferred using TCP.**

This process is shown in figure 3. One of the flags within the TCP header is the SYN bit, which is used to indicate the initial sequence number when a connection is established. This informs the receiver to synchronize its error checking means to this sequence number. Therefore, station A sends a TCP segment with SYN set and its sequence number, which in this case is 10, to station B. Station B responds by sending an acknowledgment with the value 11 to indicate that is the value of the sequence number it expects to receive next. Station B has its own sequence number that it sends to station A. In this case it is 30 which station A acknowledges by sending out a 31. This establishes two connections. Once the connections are established, data is sent from station A to station B with station A incrementing sequence numbers and station B acknowledging them. This is what is called a full duplex connection since two connections were established; one from A to B and another from B to A. These connections remain established until terminated.

Once the required data transfer has been completed, the two connections should be terminated in order to free up buffer space in the two stations. There is a flag called FIN, which is used to terminate the connection. Station A sends a TCP segment with FIN flag set along with a sequence number. Station B acknowledges the request and once the acknowledgment is received at Station A the connection from A to B is terminated. This does not mean that the connection from B to A is terminated. This must be done in a like fashion in order to terminate the full duplex connection.

## Flow Control

Flow control is the management of data transfer between two stations. Depending upon the role of the station, be it client or server, or its processing power, a station may not be able to keep up with the network traffic. In order to slow down events the TCP header has a field called window. The receiving station sets a value in the window field informing the sender how many bytes of data it will accept. The window is dynamic and the window can be increased as buffer space in the receiver becomes available. The window can also be zero halting transmission. If the sender still needs to communicate important information while in this condition it can send out a segment with the URG (urgent) flag set along with a sequence number in the urgent pointer field that indicates the first byte of data following the urgent data. The receiver should always allow room for urgent data.

## Summary

Although TCP is more reliable than UDP, UDP can be quite effective if the application layer can handle error checking and retransmission. For applications that require secure communication, TCP is the best choice. Both TCP and UDP use port numbers and when used with IP addresses create unique socket definitions across the network. These socket definitions facilitate the processes between the stations on the control network.

## References

*Illustrated TCP/IP*, Matthew Naugle, 1998, Wiley Computer Publishing.  
*TCP/IP Clearly Explained*, Pete Loshin, 1997, Academic Press.



<b>Class A</b>	<b>1-126</b>
<b>Class B</b>	<b>128-191</b>
<b>Class C</b>	<b>192-223</b>
<b>Class D</b>	<b>224-239</b>
<b>Class E</b>	<b>240-254</b>

*Figure 1—The class of an IP address can be quickly identified by observing only the first byte.*

## Introduction

In a previous section we discussed the Internet Protocol and the structure of IP addresses. An IP address identifies the source and destination of a directed or unicast message and is defined in RFC 761. IPv4 is the most common version of IP addressing requiring 32-bit addresses. Although IPv6, the 128-bit version, will be used in the future, this section will restrict the discussion to IPv4. IPv6 was developed because the explosive growth of the Internet will soon deplete the inventory of available addresses. At one time, 32-bit addresses seemed to provide more than enough addresses but there was much waste in initial assignments and the class structure of IP addresses was inefficient. In order to make more efficient usage of IP address, the concept of subnetting was introduced with RFC 950. This article introduces this concept.

## Networks and Hosts

When we talk about a network we usually envision a cluster of workstations with one or more servers connected to a local area network. Each server and workstation would have a unique address to distinguish it from the other computers. With IP addressing, servers and workstations are all termed hosts but each address not only identifies a host but the address of the network on which the host resides. This is because IP is an internetworking protocol that not only allows communication between hosts on the same network, but communication between hosts on different networks as well. The 32-bit IP address identifies a particular host along with the network on which the host resides. The structure of IP addressing is defined so that any host on the public Internet can be found by any other host.

The format of the 32-bit address is <netid, hostid> and it is usually shown as four bytes of data. Although each byte could be represented as a binary, decimal or hexadecimal number, the decimal-dot-decimal notation is the most popular. Therefore, the range of IP addresses can span 0.0.0.0 to 255.255.255.255. For example 193.5.8.254 is a valid IP address but it is difficult to determine which part of the address is the network ID and which part is the host ID. To understand the two you need to know about class addressing.

## Class Addressing

IPv4 is called a classful system under RFC 761 with IP addresses being defined as belonging to one of five classes A, B, C, D or E. Classes A, B and C define different possible combinations of network and host addresses. Class D is reserved for multicasting. Multicasting is the ability of one host to communicate with many other hosts with one transmission and is beyond the scope of this section. Class E is reserved for future use. The classes of interest to subnetting are A, B and C.

With class A addresses, the first byte of the address identifies the network address while the three remaining bytes identify the host. With class B addresses, the first two bytes identify the network address while the remaining two identify the host address. With class C addresses, the first three bytes identify the network address while the last byte identifies the host. That seems simple enough but how do you know you are looking at either an A, B, C, D or E address?

The four-byte IP address is viewed from left to right with the first byte on the left. This is the most significant byte. The first few bits (most significant) of that byte identify the class of address. For a class A address, the left most bit must be a zero. For a class B address, the first two bits must be a 10. For a class C address, the first three bits must be a 110. For a class D address, the first four bits must be a 1110. For a class E address, the first four bits must be a 1111. Therefore, it is only necessary to observe the first byte of the IP address to determine its class. Figure 1 shows the decimal value of the first byte for each class.

### **Reserved Addresses**

There are some reserved IP address besides those identified as classes D and E. For example, the class A network address 0.X.X.X cannot be used since it is used to indicate "this" network. Class A address 127.X.X.X is reserved for loop back testing. With the host portion of the address, you cannot have an all 0s host, which refers to the network address where the hosts reside. Likewise you cannot use the all 1s host address because that indicates a broadcast which is a message to all hosts on the network. Therefore, with any host addressing on either a class A, B or C network, you lose 2 host addresses. Still with 4 billion possible addresses from a 32-bit address space, you would think there are plenty of addresses even with reserved addresses. The problem is that there was much waste when addresses were originally assigned. For example, a class A address can handle 16 million hosts per one network ID. That is an enormous amount of hosts for just one network. Even a class B address can handle 65 thousand hosts per network ID. A class C address can handle only 254 hosts per network ID which may be too little for some networks. A scheme was needed to obtain a better balance between network and host assignments and that is called subnetting.

### **Subnetting**

Subnetting creates additional network IDs at the expense of host IDs and can be used with either A, B or C class addresses. If you look at figure 2, you will notice that a class B address uses 14 bits for network addressing and 16 bits for host addressing. By simply reassigning one of the host bits to a network bit, you would double the number of available network addresses but halve the number of host addresses. Carrying the argument further, move eight of the host bits (actually the complete third byte) to the network side. The result is 22 bits for network addressing and eight bits for host addressing which is quite similar to a class C address. These additional network addresses are called subnets and not networks because to the Internet, the original address is still a class B network address but locally the class B network address can be broken down to manageable subnets that function as actual network addresses. Why use subnets? Subnets are interconnected using routers, and routers improve network performance by reducing traffic and minimizing disruption due to broadcast messages. Large networks become more manageable when subnets are deployed.

## Masking

To create subnets you need a subnet mask that defines which bits will be used to create the new network address out of the 32-bit IP addresses. By “ANDing” the 32-bit IP address with a 32-bit mask, we create a 32-bit IP address that represents <netid, subnetid> becoming our new network address. What do these masks look like? If we start with a basic class A address and do not define any subnets, the mask would look like 255.0.0.0 which is called a natural or default mask. Only those bits that are set as a 1 will be considered when defining a network address. In this case, all the bits in the first byte of the IP address will be considered. The natural mask for a class B address is 255.255.0.0 and for a class C address it is 255.255.255.0. In order to create more network addresses (subnets) we need to move the mask bits to the right (changing 0 bits into 1s) in order to convert host bits into network bits. The best way to understand the concept is to use an example.

Assume we begin with IP address 165.10.0.0. From figure 1 we know that this is a class B address with a network address of 165.10 with the capability of assigning up to 65,534 hosts. We do not want 65,534 hosts on one network but would like to have up to 500 hosts on each subnet. In order to have 500 hosts on one subnet, we need to have 9 bits of host addressing. Currently, we have 16 bits of host addressing since we possess a class B address. That means that we can reassign 7 of those bits to signify subnet bits. Therefore, the subnet mask would be 255.255.254.0. In binary it would be:

```
11111111.11111111.11111110.00000000
```

The natural mask for a class B address is 255.255.0.0 so in order to create subnets we moved mask bits to the right in order to convert more host bits to network bits. It must be remembered that these mask bits must be contiguous from the left. For example, the above mask allows up to 510 host assignments. Remember that we cannot use either an all 0s or all 1s host address. The next jump would be to allow up to 1022 host addresses. What would be the subnet mask? It would be 255.255.252.0. The 1s are still contiguous from the left. This approach creates many subnets, but it is recommended that neither an all 0s nor all 1s subnet be used. This could cause a problem on some networks. How many mask bits can you have? You need to have some hosts on a network and two host addresses are unusable so the maximum number of mask bits is 30 leaving two valid host addresses.

## Notation

Using the last subnet mask in the above example, we have 1022 host addresses. What if our computer actually had host address 768 on subnet 4? What would be our actual IP address? We cannot say it is 165.10.4.768 since with decimal notation no byte can be more than 255. The actual IP address would be 165.10.7.0 so you do need to know the subnet mask before determining the actual subnet address and host address.

	Address Identifier	Network Address	Host Address
Class A	0	<b>7 bits of network address</b> <small>First byte</small>	<b>24 bits of host address</b> <small>Last three bytes</small>
Class B	10	<b>14 bits of network address</b> <small>First two bytes</small>	<b>16 bits of host address</b> <small>Last two bytes</small>
Class C	110	<b>21 bits of network address</b> <small>First three bytes</small>	<b>8 bits of host address</b> <small>Last byte</small>
Class D	1110	<b>Multicast address in the range of 224.0.0.0 – 239.255.255.255</b>	
Class E	1111	<b>Class E – Reserved for future use</b>	

Figure 2—Address classes define the split between network and host IDs.

There is a simpler way of representing the actual IP address and that is by using the Classless InterDomain Routing (CIDR) scheme. With this scheme the concept of A, B and C classes is eliminated, but the concept of subnetting is retained. In the above example, we use a total of 22 bits of contiguous 1s in our mask so we would display our IP address as 165.10.7.0/22. Although it is still not obvious that we are host 768 on subnet 4

of network 165.10, we can figure it out using this single notation which tells us exactly where the subnet mask separates the network and host addresses.

For example, in a previous article we mentioned that there were one A, 32 B and 256 C addresses that were strictly private and cannot be accessed through the Internet. These are as follows:

10.0.0.0 to 10.255.255.255

172.16.0.0 to 172.31.255.255

192.168.0.0 to 192.168.255.255

Notice that the first range is a single A address with 24 bits of host addressing, the second are B addresses with 16 bits of host addressing and the third are C addresses with 8 bits of host addressing. Using CIDR notation these same address ranges can be displayed as follows:

10.0.0.0/8

172.16.0.0/12

192.168.0.0/16

The natural mask for a class A address is 255.0.0.0 which means eight contiguous 1s from the left so 10.0.0.0/8 represents the natural mask for a class A address. This is what we would expect. A single class A network address with provisions for 24 bits of host addressing. The natural mask for a class B address is 255.255.0.0 which, with CIDR notation, would be /16 but the above class B addresses have only 12 mask bits of contiguous 1s. This seems to violate our rule for subnetting and it does. With subnetting you move the bits to the right of the natural mask thereby consuming host bits. Instead we are moving the mask to the left of the natural mask (changing 1 bits to 0s) consuming network bits. This is called supernetting which requires contiguous network addresses and will be discussed shortly. By moving the mask to the left by four bits from the natural mask, we can gain more host addresses at the expense of 16 contiguous network addresses. Therefore, the notation 172.16.0.0/12 is short for indicating a range of contiguous network addresses from 172.16.0.0 to 172.31.0.0. The same is true for the last example which are C class addresses. The natural mask for a C address is /24. Instead the CIDR notation is a /16 meaning eight less mask bits thereby yielding a range of network addresses from 192.168.0.0 to 192.168.255.0.

### SUBNETTING A CLASS C ADDRESS

Subnet mask	CIDR	# Subnets	# Host
11111111.11111111.11111111.00000000	/24	0	254
11111111.11111111.11111111.11000000	/26	2	62
11111111.11111111.11111111.11100000	/27	6	30
11111111.11111111.11111111.11110000	/28	14	14
11111111.11111111.11111111.11111000	/29	30	6
11111111.11111111.11111111.11111100	/30	62	2

The natural mask for a class C address is 255.255.255.000 which provides for up to 254 host addresses. By moving the mask bits to the right (replacing 0s for 1s), subnets are created at the expense of host bits. Not shown are masks /25 and /31 since they are not allowed. Similar charts can be made for class A and class B addressing. Class A subnetting begins at /10 and class B at /18. Both end at /30.

### Supernetting

The inverse of subnetting is supernetting. Instead of moving mask bits to the right of the natural mask for subnetting, we move mask bits to the left for supernetting. With subnetting we create more network addresses at the expense of host addresses. With supernetting we create more host addresses at the expense of network addresses. Supernetting is not for users since it would be difficult for users to be granted a range of contiguous network addresses. Supernetting is for Internet Service Providers (ISPs) who are attempting to obtain the most efficient allocation of IP addresses using the A, B, C class scheme.

### Summary

Although a 32-bit IP address offers an extremely large number of addresses, the A, B, C, class structure does not make efficient use of <netid, hostid> assignments. Subnetting improves the situation by allowing a finer split between network and host assignments while improving the performance and maintainability of large networks.

### References

- Illustrated TCP/IP*, Matthew Naugle, 1998, Wiley Computer Publishing.
- Practical Networking With Ethernet*, Charles E. Spurgeon, 1997, International Thomson Computer Press.
- International Standard ISO/IEC 8802-3 ANSI/IEEE std 802.3*, 1996, The Institute of Electrical and Electronic Engineers, Inc.
- TCP/IP Clearly Explained*, Pete Loshin, 1997, Academic Press.
- TCP/IP Illustrated, Volume 1*, The Protocols, W. Richard Stevens, 1994, Addison-Wesley Publishing Company.

# Simple Network Management Protocol (SNMP)

## Introduction

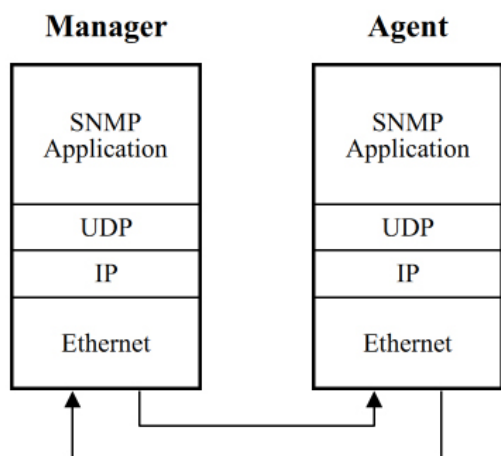
One of the numerous acronyms from the Internet world is SNMP which stands for Simple Network Management Protocol. Of course, anything termed "simple" is suspect. SNMP is an Internet protocol for managing devices on IP networks. Usually people think SNMP only applies to managed Ethernet switches, but it can be applied to any device that supports IP or TCP protocols. This includes printers, workstations, servers, modems and even automation I/O devices. SNMP introduces us to the concept of "managed" devices which offers numerous advantages over unmanaged devices and could prove beneficial in automation applications. As more and more devices embrace Ethernet, adding SNMP support can lead to greater advantages.

## SNMP Versions

When we say a device is managed, we mean the device supports the SNMP protocol in addition to its normal functions. The SNMP protocol, described in RFC 1157, was developed in the 80s as a simple means of accessing devices remotely. Originally intended to manage routers, SNMP can be used to manage any device including programmable logic controllers and remote I/O blocks. The example that is usually given refers to its use in monitoring the temperature inside a router. If this can be done, then there are a host of automation applications limited only by our imagination.

One would think there is only one version of SNMP since this acronym is frequently quoted as if it is understood by all. Actually, there are three. The first is SNMPv1 which remains the most popular version. SNMPv2 builds upon the commands of version 1. SNMPv3 addresses the biggest criticism of SNMP the commands are sent in clear-text and, therefore, insecure. SNMPv3 adds cryptography. Simply understanding SNMPv1 is enough to learn the concepts.

SNMP is an application layer protocol that sits above the TCP/IP stack. However, SNMP does not use TCP at all. It uses the UDP (datagram) protocol for communication, which provides no acknowledgment that a transmission was successful. This was done to minimize the software requirements in the "agent" which is the device being managed. The "manager" is the device requesting information from the agent and it is called a Network Management Station (NMS). The interaction between a manager and an agent is similar to the interaction between a master and a slave device. The manager can initiate a "poll" of the agent requesting information or directing an action. The agent, in turn, generates a response to the query from the manager. This is how a remote I/O protocol works. However, the manager can request that a "trap" be set by the agent. A trap is simply a report to be issued in the future which is triggered when a set of conditions are met, similar to an alarm. The trap is triggered upon an event and once it occurs, the agent immediately reports the occurrence without a poll from the manager. This is no different from having a remote I/O device report on a "change of state." The NMS that receives the trap can then take appropriate action such as notifying personnel of the event. In this situation, the NMS is acting as a server by gathering data from agents and providing information on the state of devices to clients.



**Figure 1—SNMP Communication occurs between a manager and agent by means of UDP datagrams.**

Let's consider a real-world example. We have a remote pumping station with a SCADA system attached to several devices. The SCADA system is powered from an uninterruptible power supply (UPS) that has an SNMP agent. An Ethernet fiber optic link is used for communication between the remote pumping station and the main control room. An Ethernet switch, located in the pump house, connects the UPS and the SCADA system to the Ethernet link. An SNMP manager application, running on a desktop workstation located in the main control room and functioning as a NMS, instructs the agent in the pump house UPS to set a trap that will be triggered if there's a loss of main power. If this condition occurs, the agent would send a trap message back to the NMS which, in turn, pages the maintenance shop. This is a simple case in point of how SNMP can aid applications in the automation industry.

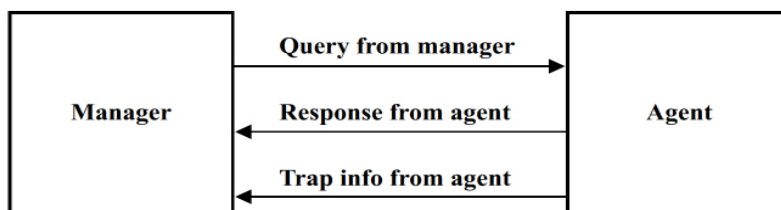
The beauty of SNMP is that it is indeed straightforward. There are only five commands with SNMPv1 and a total of nine for SNMPv2 and SNMPv3. The commands for SNMPv1 are listed below:

- get
- get-next
- set
- get-response
- trap

The additional commands for SNMPv2 and SNMPv3 are as follows:

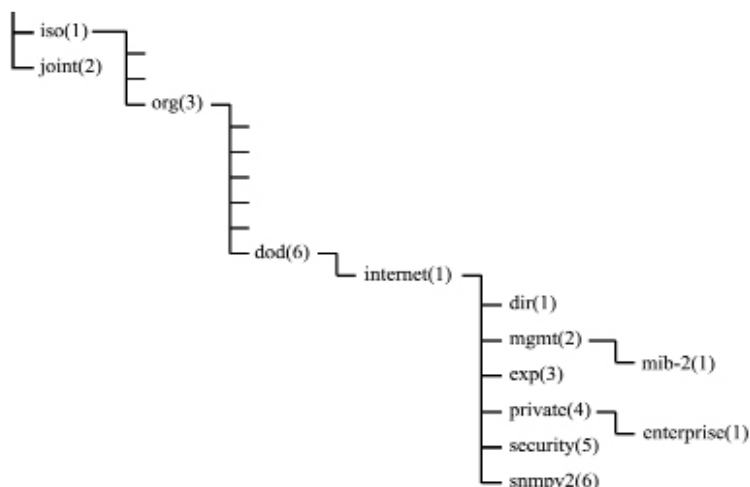
- get bulk
- notification
- inform
- report

To understand how the commands are applied, we need to introduce an integral component in the process: the managed objects that reside in the agent.



**Figure 2—A manager polls an agent in a similar fashion to a master/slave protocol.**





**Figure 3—The identification of objects follows a hierarchical structure.**

## Managed Objects

Each agent consists of a collection of managed objects that explain the capabilities and behavior of the agent in an abstract form. This is no different from the method by which a DeviceNet device is described by a collection of objects. The objects supported by a DeviceNet limit switch differ from that of a DeviceNet pneumatic manifold block; however, all DeviceNet devices support some common objects. This is the same situation with agents. All SNMP agents must support a common set of managed objects, called a Management Information Base (MIB). But an agent must support, at a minimum, what is defined in RFC 1213: MIB-2.

You might ask what happened to MIB-1? In the ever-changing Internet world, MIB-2 superseded MIB-1. Before we examine the details of MIB-2, we need to understand the structure and naming convention of MIBs. The Structure of Management Information (SMI) is described in RFC 1155. First, we will study the naming convention for managed objects and the MIBs themselves, which are simply a collection of managed objects. The term to identify an object is simply the Object ID (OID).

## Object ID

Managed objects within an agent are organized into a tree-like hierarchy similar to the way files and folders are used to represent the contents of a hard disk. In fact, some NMS software displays the management objects in a graphical fashion as if they were indeed files. However, the nomenclature is different. Managed objects are identified as a series of integers separated by dots representing the nodes on a tree. Naming begins at the root, followed by branches and ending in leaves. Let me give an example. In FIG. 3 you will see the tree structure for finding MIB-2. It begins at the root on the left. There are three branches, but we are interested only in iso(1). From iso(1) we have three more branches, but we are only interested in org(3). Next there are six more branches, but we follow dod(6). From this branch we go to internet(1). At this node we are at the base of all SNMP MIBs. The short form for representing where we are is 1.3.6.1 or we could say iso.org.dod.internet.

At this point we could follow either mgmt(2) or private(4) branches. If we follow the mgmt(2) branch, we will find standard MIBs. If we follow the private(4) branch, we will find vendor-specific MIBs. This is where a vendor can register unique products with corresponding unique management information. For example, a UPS would have much different information to share than an Ethernet switch. We will follow the mgmt branch and locate MIB-2 which is at 1.3.6.1.2.1 or you could simply say mgmt(1) which uniquely identifies its location.

### The 10 Managed Object Groups in MIB-2

mib-2	1	system	; General information about device for administrative purposes
mib-2	2	interfaces	; Keeps track of each interface on device
mib-2	3	at	; Address translation (only for backward compatibility)
mib-2	4	ip	; Tracks IP (Internet Protocol) aspects
mib-2	5	icmp	; Tracks ICMP (Internet Control Message Protocol) aspects
mib-2	6	tcp	; Tracks TCP (Transmission Control Protocol) aspects
mib-2	7	udp	; Tracks UDP (User Datagram Protocol) aspects
mib-2	8	egp	; Tracks EGP (Exterior Gateway Protocol) aspects
mib-2	9	(no longer used)	
mib-2	10	transmission	; Currently not used
mib-2	11	snmp	; Tracks SNMP (Simple Network Management Protocol) aspects

We have found MIB-2, but we do not know the location of the individual managed objects. It's best to remember that MIB-2 is a collection of objects and each object description is identified in RFC 1213. If we study RFC 1213, we will learn there are ten managed object groups in MIB-2

as explained on page 3. The first object group is system. The system group lets you enter the physical location of the device, the name of the device and who is responsible for the device. Therefore, if the device is queried by a management system, it could say it was tagged UPS-1, located in the pump house and if there is trouble to call Randy in the Instrument Shop. Another attribute of this object is up-time. It will continue to accumulate time until it is unpowered.

### Setting Traps

As mentioned before, a trap is an exception report similar to a change-of-state response from an I/O device. The manager establishes the trap in an agent. The agent monitors the situation and only reports to the manager if the trap is tripped. There are seven generic traps, but one is reserved for vendors for their specific application. The traps are as follows:

#### Generic Trap (Name, Number and Definition)

##### *coldStart* (0)

Indicates that the agent has rebooted. All management variables will be reset; specifically, Counters and Gauges will be reset to zero (0). When a device is powered on, it sends this trap to its trap destination.

##### *warmStart* (1)

Indicates that the agent has reinitialized itself. None of the management variables will be reset.

##### *linkDown* (2)

Sent when an interface on a device goes down and identifies which interface

##### *linkUp* (3)

Sent when an interface on a device comes back up and identifies which interface.

##### *authenticationFailure* (4)

Indicates that someone has tried to query the agent with an incorrect password.

##### *egpNeighborLoss* (5)

Indicates that an Exterior Gateway Protocol (EGP) neighbor has gone down.

##### *enterpriseSpecific* (6)

Indicates that the trap is vendor specific.

As seen from this list, a much simpler approach can be taken to monitoring a device in the field besides polling. For example, a coldstart could indicate some unauthorized activity in the field that triggered the trap. The use of traps is no different from having the benefit of a remote annunciator in the field but without the added expense. By studying the vendor specific traps that are available from a particular product, more ingenious reporting is possible.

### Configuration

Before commissioning a managed device in the field, its agent must be configured. This is not unlike the commissioning needed before installing a DeviceNet limit switch or photo-eye. With DeviceNet, you would use some tool or a program running on a laptop PC. Some devices will have a serial port that will support an ASCII terminal. If a terminal is unavailable, you could run a terminal emulation program on a PC. The advantage of this approach is that your network does not need to be up in order to commission the device. The second approach is to run a Telnet session over Ethernet. Of course, to do this the device must have its IP address already assigned. The screen on the PC will look the same but the network needs to be running. However, you could commission the device remotely from the control room with Telnet. In both of these cases, text screens are provided and the operator simply needs to fill in the blanks. The third approach is to use a web browser. This assumes that the managed device will serve up a web page for commissioning. With web technology, the screens are more colorful and data input is not restricted to simple command lines. Any of these approaches is possible but what data must be entered?

There are several parameters that must be set in the agent. The agent will consume an IP address for it to function as a management port. You might want to name the device, indicate its physical location and identify the person responsible for the device. You can even append a password to protect your settings. If traps are to be used, you need to identify the IP addresses of the managers that will receive the traps. There is usually space to list several IP addresses. What is significant here is that you need to know all this information before commissioning and to be careful not to reassign the master IP addresses, otherwise the traps will fail to find a manager. It would be a good idea to document all these parameters so a replacement device can be properly configured before putting the unit into service.

### Managers

Most of the discussion has been about agents and little about network management software. Command line programs can be used to poll agents and view responses, but the process is tedious since the operator needs to fully understand the structure of MIBs and each object's syntax. There are several commercial software packages and some freeware packages that will poll agents, set traps and receive and display trap responses while providing a more convenient user interface. Since SNMP was developed before the Worldwide Web protocols were developed, much of the data that is displayed is text-based. Later versions of network management software take advantage of Windows functionality and provide more versatility such as trending. It will take an operator some time to learn the intricacies of the program but from one workstation, an operator can view all SNMP compatible devices.

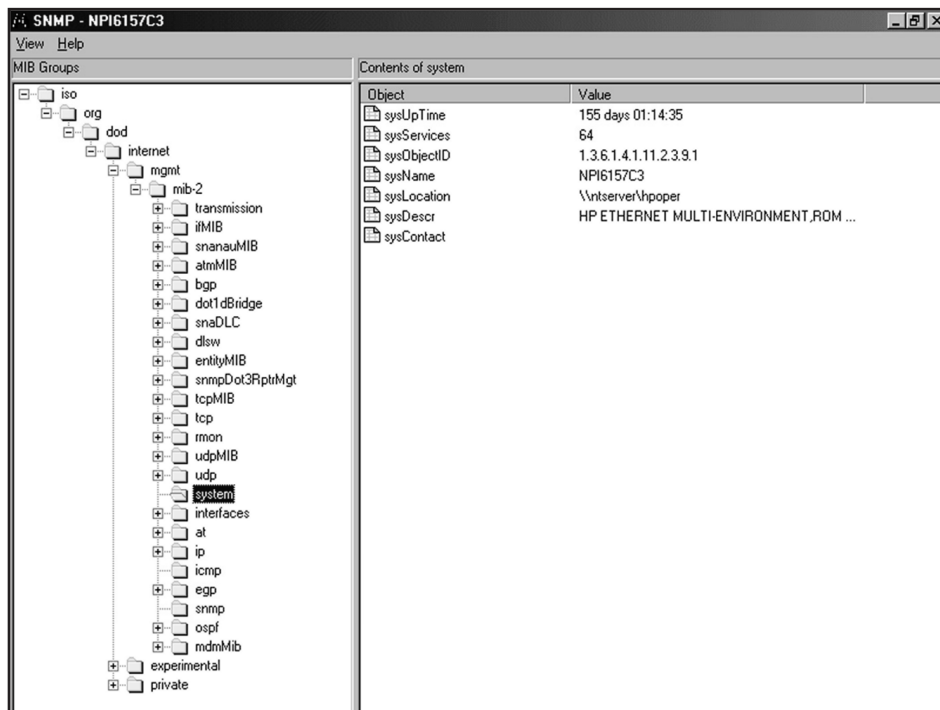


Figure 4—Typical manager screen.

With increasing interest in making a web browser the default operator interface for a system, can SNMP data be displayed on a browser screen? Some newer managed devices have built-in web servers that can serve up SNMP data. The advantage of a built-in web server is that it provides a convenient method of configuring the device and, an opportunity to verify that the device is functioning by being able to access it from the web. The other advantage is that the managed device with its internal web server can operate stand-alone without the need for any network management software. The trick comes in when several managed devices are to be viewed from one browser. There is no consistency of data presentation from the various vendors of

web-based managed devices. It is also inconvenient to remember all the various URLs that must be selected to view the individual managed devices.

For our industry, there is another approach. It is possible to have an OPC server running in the manager that understands the SNMP protocol and can query MIB data, but display the data in a format comfortable to the operator. If the operator is viewing a process automation screen to view instruments and controllers and alarms, the information from managed devices can be included within the same screen; thus, making for a neat uniform appearance. The operator does not need to run a totally different application program to monitor the health of the network. There are several vendors in our industry that provide such a product.

### Summary

With more and more devices embracing Ethernet and Internet protocols, the addition of SNMP protocol support adds benefits to the device. Managed devices support the SNMP protocol and are called agents. Agents consist of a collection of managed objects that can be queried by a manager to determine the health of the network or the status of particular devices. By displaying this data in an easily understood format, operators and maintenance personnel, located at a central site, can monitor the performance of the entire network by observing selected devices and pinpointing potential problems before they occur. The trend is to use more web-based tools. SNMP is not restricted to just the management of switches and routers. Any industrial device can have SNMP support and could provide much aid in industrial applications.

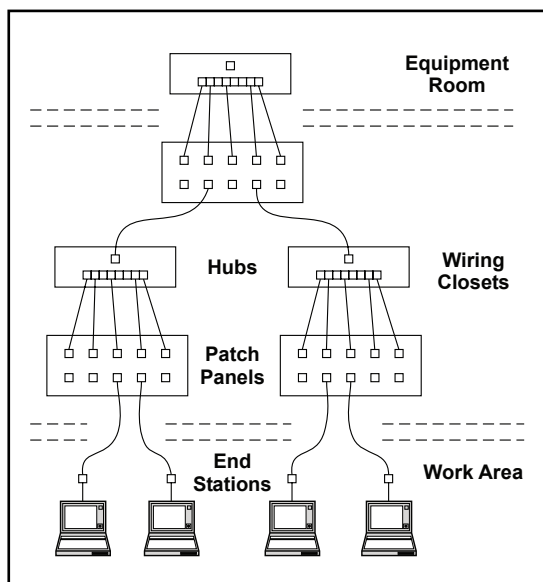
### References

- Essential SNMP*, Mauro, Douglas R. & Schmidt, Kevin J., O'Reilly & Associates, Inc., 2001.
- DeviceNet Specifications*, Open DeviceNet Vendors Association, Volume 1, Release 2.0, 1995.
- A Simple Network Management Protocol (SNMP)*, Internet Engineering Task Force, RFC 1157—1990.
- Management Information Base II*, Internet Engineering Task Force, RFC 1213—1990.

## Virtual LANS (VLANs)

### Introduction

A local area network (LAN) is a private network usually confined to one building. Virtual LANs (VLANs) allow a single physical LAN to be partitioned into several smaller logical LANs. VLANs limit the broadcast domain, improve security and performance and are ideal for separating automation systems from information technology systems.



**Figure 1—Structured wiring creates a hierarchy of hubs.**

### Structured Wiring

One of the advantages cited for migrating to Ethernet from fieldbus technology is found in the comment “our building is already wired for Ethernet. I do not need to run specialized wiring since twisted-pair wiring is already in place.” This could be true since Ethernet cabling installations typically follow structured wiring standards such as **TIA/EIA-568-A Commercial Building Telecommunications Cabling Standard**. Following the standard, end stations at each work area would be wired to patch panels in a wiring closet. These patch panels would also connect to repeating hubs or switching hubs mounted in the wiring closet (Figure. 1). The cross connection between end stations and hub ports are made with short patch cords. Out of each wiring closet is a single connection to a cascaded hub located in an equipment room. All wiring closet feeds go to the equipment room. It is possible to have more than one equipment room, but it is the intent of the standard to limit the number of levels of hierarchy. It is quite possible that the automation system is wired in a similar fashion and, in this way, all stations within the building share the same LAN.

Sharing the same LAN may not always be a good idea. LANs are typically maintained by the information technology (IT) department that has become increasingly more interested in a secure network than maximizing up-time. Disconnecting a user suspected of having a faulty station by removing a patch cord is typically done and is treated as an inconvenience to the user. However, the same action done to a device on a control system network could be disastrous. Therefore, it has been suggested to have two LANs—one for IT and one for automation systems. This would certainly remove the security concerns of the IT department, but segregating the physical wiring may not be possible nor convenient.

There is another reason to separate the information technology LAN and the automation system LANs. A LAN is considered a single broadcast domain. This means that broadcast messages (messages destined to all stations) will be sent to every station on the LAN. This is usually true for multicast messages (messages destined to many, but not all stations). If the exact location of stations that are to receive a multicast message is not known, then all stations will receive the message. Automation protocols frequently use the producer/consumer model in order to improve real-time response. In the producer/consumer model, one originating message that is produced by one station is consumed by several stations called consumers. With Ethernet, this generates many broadcast and multicast messages that can consume the total bandwidth of the LAN. Is there another way of retaining the same physical network, but allowing separate LAN functionality? Yes there is, and it is called virtual local area networks (VLANs).

## VLAN Structure

A LAN consists of stations, repeating hubs and switching hubs operating at the data link layer. LANs could be connected to other LANs if routers are used; thereby, creating an internetwork. Each LAN would then be given a network address. The best example of an internetwork is the Internet. Therefore, it is possible to have the automation system on one LAN and the information system on another LAN with the two linked by a router. However, the structured wiring within the plant may not support this wiring directly. Besides, configuring routers is more difficult than configuring VLANs. What is desired is to have the information system and industrial automation system on the same LAN, but logically separated into two LANs. That is what a VLAN can do.

Within a LAN that has all stations connected to repeating hubs, all stations hear all three types of transmissions—unicast, multicast and broadcast. In this situation, it is not possible to establish separate VLANs since there is no way of restricting traffic. A basic requirement of VLANs is the use of switching hubs. A switch learns the location of stations by observing the source MAC address present in a message received at an incoming port. The MAC address-port number association is so noted in its filtering database. All future transmissions destined to a MAC address that is stored in the switch's filtering database, will only be directed to the port associated with that MAC address unless the transmission originated on that port. If a MAC address is received with no association, the transmission is flooded to all ports (except for the received port) as if the switch were a repeating hub. The same is true for multicast and broadcast messages. Therefore, a switch provides an improvement in performance over repeating hubs by restricting unicast messages to only those stations involved, but it is this filtering capability that can be exploited for VLAN use. A single switching hub can be so configured and thus act as several independent switching hubs by creating VLAN associations to switch ports.

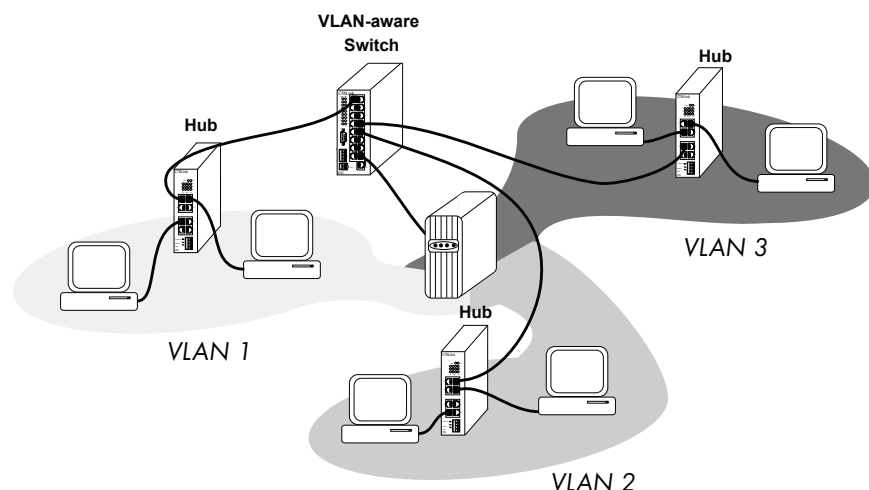
### Port VLAN

There are several ways of creating VLANs, but the easiest to understand is the Port VLAN. Switches create an association of MAC addresses and port numbers. What needs to be added is a VLAN association. This would have to be accomplished through some configuration of a switch that can support VLANs. VLAN support is not possible with a Plug and Play switch—one with no means of altering its personality through operator intervention. For example, within a sixteen-port switch we want to create three separate VLANs numbered one to three. During configuration, we associate each port on the switch to be a VLAN. From then on, traffic within a VLAN assignment will be restricted to only those ports associated with that VLAN assignment. Using our example of three VLANs, we established VLAN1 as associated with ports 1, 2, 3 and 4. A broadcast or multicast message on port 1 would be sent only to ports 2, 3 and 4 and no others.

The other VLANs would operate in a similar fashion. A unicast message would be forwarded as with any other switch. There would be a MAC address-port number association. However, added to this association would be the VLAN constraints. So if the MAC address-port number association is not present in memory for a destination address, flooding will only occur with the VLAN port group. What happens when a destination address is specified in a transmission received on a port from another VLAN group? The transmission should be discarded.



Figure 2 shows a Port VLAN application consisting of three VLANs, although more VLANs can be added. There is only one VLAN-aware switch located in the middle of the LAN. The other switches that are not VLAN-aware are considered part of the respective VLANs. Each port on the VLAN-aware switch has an association with a common port on the switch where a server resides. This overlapping of VLANs allows any workstation in a VLAN to access the server, but workstations in separate VLANs are not known to each other.



**Figure 2—In this Port VLAN application, the server in the middle is logically attached to all three VLANs.**

therefore, Port VLANs are best accomplished using a single VLAN-aware switch. Notice that there is no change in Ethernet frames with Port VLAN partitioning. End stations are unaware of the VLAN structure. More flexibility is gained if VLAN associations can be learned from the contents of the Ethernet frame. This is called implied tagging which allows VLANs to span multiple switches using the same cabling structure.

### Frame Encoded VLAN Schemes

With Port VLAN, there is no altering of Ethernet frames or any implicit tagging within Ethernet frames. Stations are unaware of the VLAN structure. There are alternate ways of establishing VLANs if the switches being used support the various schemes. You could simply associate particular MAC addresses to a VLAN. In this way the station assigned to the VLAN can be on any switch port and still be attached to a particular VLAN. Of course, if that station were ever replaced, all switches would need to be reconfigured for the new MAC address. Another approach to VLANs is to separate stations according to the network operating system being supported. By examining some protocol field, frames could be directed only to those stations supporting that operating system. This approach to VLANs was popular when there were several competing network operating systems with much different Ethernet frame definitions. The movement towards universal TCP/IP acceptance has now limited the frame structure choices. Another scheme is to define a proprietary protocol by coding the Ethernet frame with VLAN information. The problem with proprietary schemes is that they do not have wide industry support. To obtain wide industry support, you need an IEEE standard.

### Explicit VLAN Tagging

Ethernet has been around since the mid-70s, and the maximum length frame (less preamble) was always 1518 bytes. For automation systems, this frame size is quite large since most I/O messages are short. However, after all these years it appears that 1518 bytes are still not enough. The IEEE 802.1Q committee decided that four more bytes were needed in order to define a universally acceptable VLAN tag. There were concerns that stations and hubs could not handle an oversized frame and this new standard required a revision to IEEE 802.3. Everything we said about maximum frame size is now wrong. It is not 1518 bytes, but 1522 when VLAN tags are appended.

The IEEE 802.1Q VLAN tagging scheme is called an explicit VLAN scheme since something (VLAN tag itself) is appended to the frame versus being implied (implicit VLAN) by the contents of the frame. The four-byte tag is inserted immediately after the source address and before the Type/Length field (Fig. 3). The first two bytes are called the Tag Protocol Identifier and functions much like the Type/Length field. The contents of the two bytes are 0x8100, which is to be recognized as a VLAN tag. The following two bytes are the Tag Control Information. The remainder of the Ethernet frame stays the same except the Frame Check Sequence (FCS) must be recalculated because of the longer frame. Other than that restriction, a VLAN tag can be added or removed without affecting the contents or nature of the message.

The two-byte Tag Control Information consists of three bits for IEEE 802.1p priority levels (that has nothing to do with VLANs), one bit called the Canonical Format Indicator (CFI) and 12 bits for the VLAN identifier. With 12 bits of identifier, there could be up to 4096 VLANs. However, all ones are reserved and all zeros indicate no VLAN association, meaning that the tag is solely used to indicate priority level. All other identifiers can be used to indicate a particular VLAN along with the 802.1p priority level of the message.

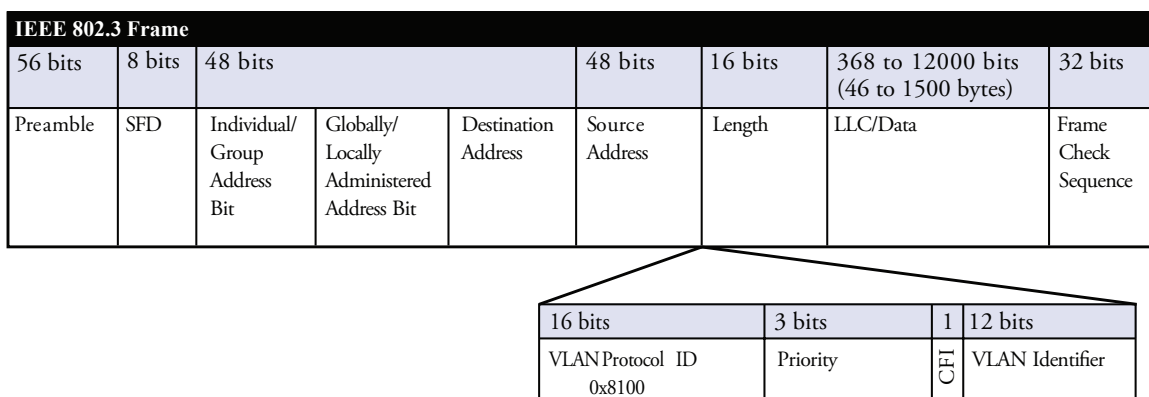


Figure 3—802.1Q VLAN standard inserts a four-byte tag into a standard Ethernet frame.

The CFI bit is used to indicate bit ordering within frames, which is an issue when communicating over non-Ethernet LANs. Since we are only interested in Ethernet LANs, the CFI bit is set to zero.

### ***VLAN-unaware End Stations and Switches***

Since 802.1Q arrived over 20 years after the invention of Ethernet, there are plenty of VLAN-unaware devices in the field. Although an end station will probably accept the elongated frame, will the software driver “choke” on receiving a 0x8100 Ethertype protocol identifier that it has never seen before? The best practice is for end stations never to see VLAN tags unless they are conditioned to do so. With the amount of legacy equipment in the field, it is a good bet the end stations are VLAN-unaware. A VLAN-aware end station is one that can receive and apply 802.1Q VLAN tags and, therefore, is termed tag-aware. However, the same is not true of switches. A VLAN-aware switch must be able to make VLAN-port associations but it may not understand 802.1Q tagging. A Port VLAN switch is a good example. A tag-aware switch understands 802.1Q tagging and can make VLAN-port associations as well.

### ***VLAN Edge Switches***

If a VLAN-aware station initiated a transmission received on a port of a tag-aware switch, it is a simple matter to read the value of the VLAN assignment and forward the frame intact to those ports in its filtering database for that particular VLAN assignment. However, if a transmission is instead received from a VLAN-unaware station, the tag-aware switch must append a VLAN tag equivalent to the VLAN association established previously for the received frame. This association could be based on the MAC address, protocol ID or port location as discussed earlier. Whatever the association rule was for the VLAN, the identifier for that VLAN must be the same as applied to the VLAN tag and the new frame forwarded in the output port or ports indicated in the switch’s filtering database.

In order to limit VLAN tags from being propagated to VLAN-unaware end stations, the tag-aware switch determines the appropriate forwarding action. Before it forwards the frame to one of its output ports, it looks in its table if the VLAN tag is to remain or be removed. If the message is going to VLAN-unaware stations, then the VLAN tag should be stripped. If it is going on to core VLAN switches, then it should be retained.

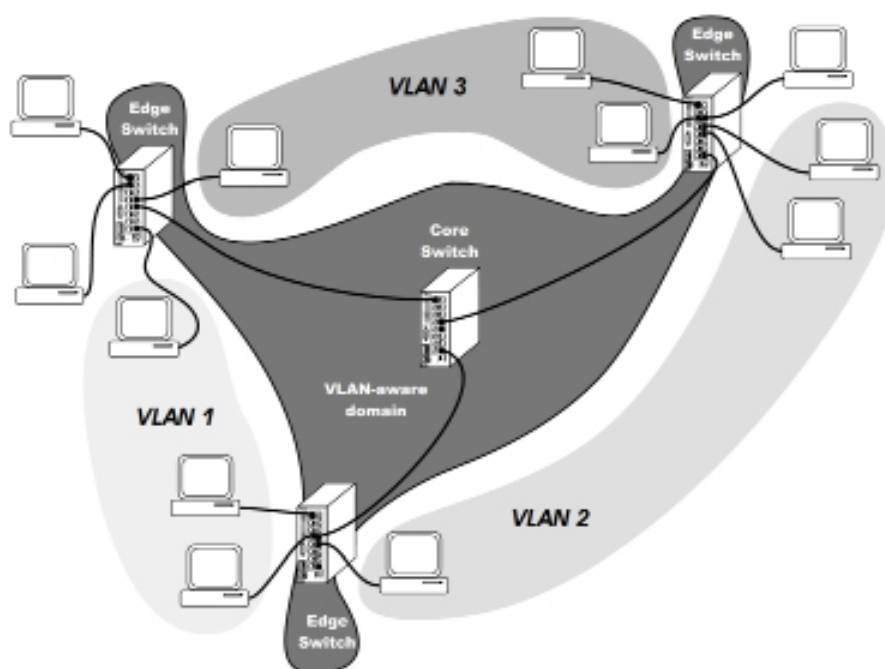
### ***VLAN Core Switches***

Core switches understand VLAN tags and reside in the backbone of the LAN and are usually only connected to edge switches. Therefore, their forwarding rules are much simpler and faster to implement. All incoming frames will have VLAN tags and all outbound frames will retain these tags. The filtering database could consist of only the 4094 possible VLANs and output port assignments. No source addressing would need to be learned. In actuality, an edge switch could be configured as a core switch, and since it would probably be too confusing to have two types of VLAN-aware switches in the building, restricting use to only edge switches could be the answer. Even though 4094 VLANs are possible according to the 802.1Q standard, not all switches can support that many VLANs simultaneously. Could you imagine the complexity of configuring and maintaining this many VLANs?

### Mobility

It would be convenient to be able to take your laptop and connect it to any available spare port on a switch within the LAN and examine the operation of an automation system on a particular VLAN. In order to effectively achieve this functionality, the laptop should be VLAN-aware and the attached switch must be programmed to allow access for that particular VLAN by having a valid VLAN-port association that would reach the VLAN desired. Using a VLAN-unaware laptop with implicit tagging would make the task even more difficult, but not impossible. Reconfiguration of the various switches in the path of the VLAN may be required in order to open up the port attached to the laptop. The use of Port VLANs would be impractical.

Figure 4 shows a typical LAN incorporating 802.1Q tagging with edge switches each connected to one core switch using a single cable. Within the VLAN-aware domain, edge switches must transmit VLAN-tagged frames to identify frame-VLAN associations. For any edge switch to have access to all possible VLANs (to ensure mobility), the port connected to the core switch must be associated with all possible VLANs.



**Figure 4—The most flexible VLAN arrangement can be achieved by the use of 802.1Q tags. Edge switches allow the use of both VLAN-aware and VLAN-unaware end stations.**

## Summary

VLANs are an effective means of portioning a larger LAN into manageable subsets. VLANs restrict the broadcast domain, improve performance and security, and they are ideal for isolating automation systems from IT systems while retaining the plant's structural wiring. The simplest of VLANs to implement are Port VLANs, but the most effective VLAN scheme is the IEEE 802.1Q VLAN tagging standard that improves mobility by allowing a user to potentially access any VLAN from any point on the LAN.

## References

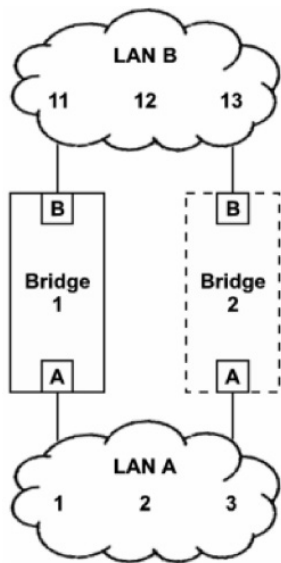
*The Switch Book*, Rich Seifert, 2000, Wiley Computer Publishing.

*Ethernet: The Definitive Guide*, Charles E. Spurgeon, 2000, O'Reilly & Associates, Inc.

*International Standard ISO/IEC 8802-3 ANSI/IEEE Std 802.3*, 2000, The Institute of Electrical and Electronics Engineers, Inc.

*Commercial Building Telecommunications Cabling Standard*, TIA/EIA-568-A, 1995, Telecommunications Industry Association.

*Virtual Bridged Local Area Networks IEEE Std 802.1Q™*, 2003 Edition, The Institute of Electrical and Electronics Engineers, Inc.



**Figure 1—The addition of Bridge 2 creates a loop.**

## Introduction

In an automation application that relies heavily on the health of the Ethernet network that attaches all the controllers and computers together, a concern exists about what would happen if the network fails? Since cable failure is the most likely mishap, cable redundancy is suggested by configuring the network in either a ring or by carrying parallel branches. If one of the segments is lost, then communication will continue down a parallel path or around the unbroken portion of the ring. The problem with these approaches is that Ethernet supports neither of these topologies without special equipment. However, this issue is addressed in an IEEE standard numbered 802.1D that covers bridges, and in this standard the concept of the Spanning Tree Protocol (STP) is introduced.

## IEEE 802.1D

ANSI/IEEE Std 802.1D, 1998 edition addresses the redundancy issue by utilizing bridges to connect two or more local area networks (LANs) at the MAC level, which is layer two in the ISO Reference Model. Generally the MAC type can be different on each LAN, but usually Ethernet LANs are on either side of a bridge. Interconnecting LANs by using bridges creates a Bridged LAN whereby end stations located on different LANs can communicate as if the bridges were not present.

Cable redundancy introduces loops in the topology and, as we will see, these loops must be disabled. An automation user may want loops to guard against a primary cable failure while an office automation user may want to guard against an inadvertent loop. The 802.1D standard addresses both situations.

## Bridge Operation

If you understand how an Ethernet switch works, you know how a bridge operates. However, all the requirements of a bridge (e.g., STP) are not always present in a switch. A bridge needs to relay and filter frames and it must make independent decisions about when to do this.

Look at Figure 1. In a two-port Ethernet bridge, each port has an Ethernet-type MAC port connected to a separate LAN and a filtering database (memory) shared by both ports. Within each LAN is a collection of end stations, repeating hubs and simple plug-and-play switches. Each end station has a unique MAC address. For simplicity, we will assume ordinary integers although true Ethernet MAC addresses are 48 bits long. In our example, three numbered end stations are present in each LAN. Assume Bridge 1 has recently been powered and its memory cleared (Bridge 2 will be added later). Station 1 sends a message to station 11 followed by Station 2 sending a message to Station 11. These messages will traverse the bridge from one LAN to the other. This process is called “relaying” or “forwarding.” The database in the bridge will note the source addresses of Stations 1 and 2 as arriving on Port A. This process is called “learning.” When Station 11 responds to either Station 1 or 2, the database will note that Station 11 is on Port B. If Station 1 sends a message to Station 2, the bridge will do nothing since it realizes that because Stations 1 and 2 are on the same LAN their message does not need to be shared with other LANs. This process is called “filtering.” If Station 1 ceases to initiate messages for a period of time, the bridge will erase Station 1 from its database—requiring the location of Station 1 to be relearned. This is called “aging.”



The above examples are all directed or “unicast” messages—meaning that one station is sending a message to another station. With multicast (one station to many stations) or broadcast messages (one station to all other stations), the bridge will forward messages to all stations since it may not know the actual location of all stations. This process is called “flooding.”

Looking at the same Figure 1, Bridge 2 is now added to parallel Bridge 1. This gives us a redundant path, but it also creates a loop with the following adverse results. When Station 1 initiates a message to Station 11, this message is forwarded by Bridge 1 and appears on LAN B. Bridge 2 interprets this message as originating on LAN B so it forwards the message to LAN A while incorrectly noting that Station 1 is located on LAN B. When Station 1 initiates a second message, Bridge 2 interprets this action as if Station 1 has now moved to LAN A from LAN B and resets its filtering database accordingly. Now assume that Station 1 sends out a broadcast message. Bridge 1 will forward the message to LAN B. Bridge 2 will observe the message on LAN B and forward it to LAN A. Bridge 1 will observe this message on LAN A as a new message and forward it to LAN B again – initiating an endless cycle, totally consuming the bandwidth of both bridges and rendering both LANs useless. To maintain the integrity of our network, we must guard against the formation of loops.

### Tree Topology

To avoid loops we need a tree topology consisting of a root, a succession of branches and then leaves. The leaves represent end stations, and there is one and only one path from a leaf to another leaf. Therefore, the tree is free of loops that can cause havoc in a network. The other requirement is that all leaves are connected. There are no isolated segments. Another term for this topology is distributed star. Within our tree structure will be a series of bridges used to connect the branches and leaves. There are two types. The “root bridge” is the main one of interest because it has a special assignment and there is only one within a network. The other bridges (that are to be used) are all “designated bridges” and there could be many within the network. To have a tree topology, you need bridges with more than two ports.

### Port Designations

Although bridges do not need MAC addresses to operate, a MAC address is needed to identify bridges in the Spanning Tree Protocol. Besides a MAC address for each bridge, each port on each bridge must be identified. For bridges, a unique 64-bit bridge identifier is assigned by appending a 16-bit priority field in front of a unique 48-bit MAC address resulting in a “Bridge ID.” The MAC address comes from the bridge vendor while the priority field can be set by the user. The default priority value of 0x8000 is in the middle of the priority range; if the user fails to assign priority values, the bridges will still have unique assignments. This is important since the bridge with the lowest numerical bridge identifier will become the root bridge. All other bridges have the possibility of becoming designated bridges.

Similarly, a 16-bit “port identifier” exists consisting of an 8-bit port address preceded by an 8-bit priority field. Again, the user sets the priority field while the bridge vendor sets the port addresses usually beginning with one for port one and so on. The default priority field is 0x80. Now we have all the bridges and ports identified including the root bridge.

To avoid loops, there is one and only one bridge that is responsible for forwarding messages from the direction of the root towards branches, which we will call links. If there is only one path from the root to a link, where end stations (leaves) attach, there will be no loops. We need a forwarding policy, which is called the Spanning Tree Protocol. To implement this policy, we need to assign each port to become either a “designated port” or a “root port.” A designated port is a port that forwards traffic away from the root and toward the leaves. A root port carries traffic back to the root bridge with the further requirement that no more than one root port exists on any one designated bridge. Looking at Figure 2, notice that the root bridge (R) has all designated ports (because it is the root) while the designated bridges have root ports when connected to the root bridge or when the direction of flow is towards the root.

### Path Cost

The next item to specify is the cost of each link between branch devices (end stations are not counted). Since we do not want to forward traffic onto low-speed links if we can avoid it, we assign a link cost based upon port speed. The recommended cost figures are shown in Table 1. A 100 Mbps port would cost 19 while a 10 Mbps port would cost 100. Two 100 Mbps cascaded links (two links requiring two bridges to reach the root) would cost 38. This would represent the “path cost.” Therefore, all designated ports on Bridge C in Figure 2 would advertise a path cost of 38 assuming traffic is at 100 Mbps. STP utilizes path cost to the root for arbitrating the ideal route. All other routes are blocked. If path costs for each port on each bridge are manually assigned by the operator, configuration could be tedious. Once these costs are entered, we can let the STP determine the best topology that will not introduce loops.

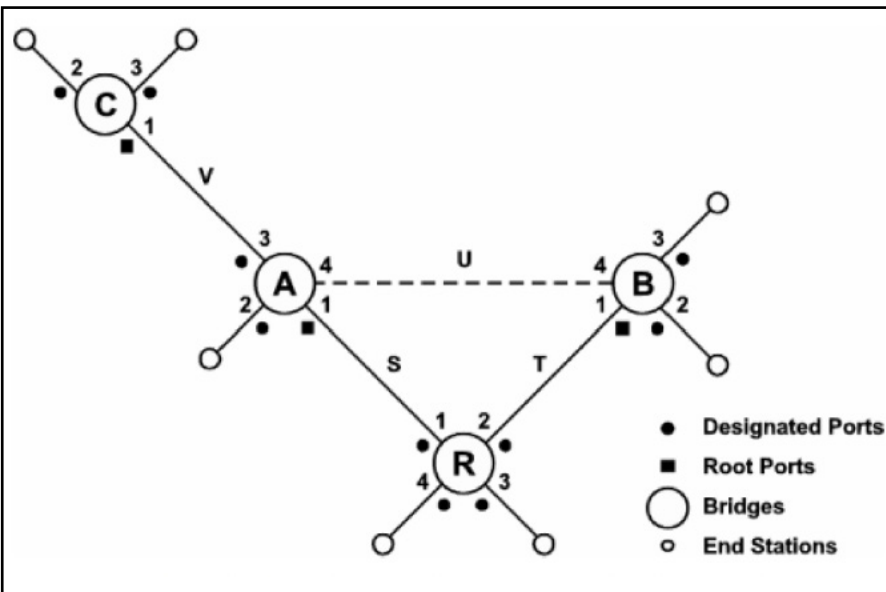


Figure 2—The addition of link U creates a redundant path. Bridge R has the lowest bridge ID and is, therefore, the root.

### BPDU

Bridges must communicate with one another to execute the STP, and they accomplish this by sending configuration messages in the form of “Bridge Protocol Data Units (BPDUs).” The BPDU is sent as a multicast message (01-80-C2-00-00-00) within a reserved range of MAC addresses which are consumed by each bridge and not forwarded. Each bridge must periodically advertise its understanding of the topology and the path cost to the root for each of its ports. The BPDU format is shown in Figure 3. The more important fields will be discussed in the following example.

DATA RATE	RECOMMENDED LINK COST VALUE
4 Mb/s	250
10 Mb/s	100
16 Mb/s	62
100 Mb/s	19
1 Gb/s	4
10 Gb/s	2

**Table 1—Link Cost Recommendations**

<b>Protocol Identifier</b>
<b>Protocol Version Identifier</b>
<b>BPDU Type</b>
<b>Flags</b>
<b>Root Identifier</b>
<b>Root Path Cost</b>
<b>Bridge Identifier</b>
<b>Port Identifier</b>
<b>Message Age</b>
<b>Max Age</b>
<b>Hello Time</b>
<b>Forward Delay</b>

**Figure 3—There are two types of Bridge Protocol Data Unit types. Configuration messages are normally sent; however, a designated bridge can initiate a Topology Change message.**

Refer to Figure 2 where we have four interconnected bridges connected to end stations. We will assume a stable network with a duly elected root bridge (R) and several designated bridges (A, B, C). All ports on the root bridge are designated ports since they emanate to end stations from the root. All other ports are designated ports or candidates for designated ports except for three ports that point towards the root. They are Port 1 on Bridge A, Port 1 on Bridge B and Port 1 on Bridge C. These are root ports. Assume that all ports on all bridges are rated for 100 Mbps except Bridge B which is only rated for 10 Mbps. Examining Table 1 we find that the corresponding link cost for 100 Mbps is 19 and for 10 Mbps it is 100. This information will be needed when constructing the BPDUs. There are four links of interest (S, T, U, and V).

The root bridge begins the process of sending a periodic configuration message based upon the "Hello Time" which is typically two seconds. In this message, the Root Identifier and Bridge Identifier would be the same since this bridge thinks it is the root by having the lowest-value bridge identifier.

The Root Path Cost will be zero because this is the root bridge. Since all ports on the root bridge are designated ports, a configuration BPDU will go out each port along with the corresponding port identifier. This process repeats every Hello Time.

First-tier bridges, those directly connected to the root bridge, (in our case, Bridges A and B) will receive the BPDUs on Links S and T and will analyze the data. Each bridge will verify that the Root Identifier is indeed lower than its own bridge identifier. If that is true, each bridge will assemble its own set of BPDUs for transmission out its designated ports. The Root Identifier field will not change. Each designated port will increase the total cost of getting back to the root by that port's individual Root Path Cost. Since both of these bridges are tier-1 bridges, the Root Path Cost would be the bridge port's link cost (19 for Bridge A and 100 for Bridge B). The bridge and port identifiers would represent data from each bridge. The Hello Time will remain that specified by the root. BPDUs are then sent out to each bridge's designated ports. Second-tier bridges receive the BPDUs (in our case Bridge C) and the path cost is upped again. In our example, the Root Path Cost from Bridge C would be 38. Bridge C will send out its set of BPDUs—ending the process since no more tier bridges remain. End stations do not participate in the process and ignore the messages. Because the propagation of messages from the root will take time, the standard sets an arbitrary limit to the number of cascaded bridges at seven. This limit does not apply to conventional switches that, in any event, should not be present in an STP network.

### Port States

The operational states of ports participating in the STP are a bit different from a conventional switch. Additional states are needed to prevent a loop, and to limit instability during the voting or topology-change process. There are five states.

**Disabled**—The port is completely non-functional in that it cannot receive or transmit any type of frame.

**Blocking**—The port is neither a designated or root port but is recognized as an alternate port to the root. It does not learn addresses, forward frames or transmit BPDUs but it can hear BPDUs being sent since it may be called to action one day.

**Listening**—This port is being prepared for activity by exiting the blocking state. It still does not learn or forward addresses, but it sends and receives BPDUs. It is participating in the voting, but it might not win the election.

**Learning**—This port will become active in forwarding frames but must wait until the Forward Delay timer (typically 15 seconds) expires. This allows the port to add entries in its filtering database so it will not flood ports once it enters the forwarding state.

**Forwarding**—This port is functioning as any other switch port by filtering and forwarding frames.

We will go back to our example in Figure 2. Since we have a tree topology without loops, all active ports (those with a link partner) are in the forwarding state. Now consider the presence of Link U. Unused Port 4 on Bridges A and B will be active but only in the listening state. While in this state, they send each other BPDUs. Bridge A will send out a BPDU to Bridge B indicating a root path cost of 19 while Bridge B reciprocates with a root path cost of 100. Both bridges recognize there is an alternate path to the root, which is unacceptable, and that (for messages not originating at the end stations attached to Bridge B) the best path to the root is through Bridge A and not Bridge B. Therefore Port 4 on Bridge B assumes the blocking state, and Port 4 on Bridge A the learning state and eventually the forwarding state even though this would be useless since all the messages forwarded from Bridge A to Bridge B would be discarded on arrival. The final result is no topology change. The tree before the redundant connection is the same as after the connection was made. However, Link U is now a potential redundant path which may be utilized after a link or switch failure.

### Topology Change

In the above example, adding Link U did not result in a change in the tree topology. However, a topology change can occur due to a lost link, a lost bridge, the addition of a link or bridge or by management changing the priorities of bridges. What happens if the root bridge fails? STP guards against all these occurrences by monitoring configuration BPDUs, observing that a BPDU failed to arrive or by generating a Topology Change BPDU.

There are two types of BPDUs as identified in the “BPDU Type” field. The configuration type is the normal BPDU as shown in Figure 3. The topology-change BPDU is similar to the configuration BPDU except that no data is transmitted below the Type field. This BPDU is generated by one of the designated bridges that changed its topology. An intervening designated bridge will acknowledge the originator’s topology-change message by sending a configuration message with the “Topology Acknowledge” bit set in the “Flags” field. A new topology-change BPDU will then be sent towards the root. Any intervening designated bridge would repeat the process until the root is notified. The root bridge notices the message and informs all its attached designated bridges of the topology change by setting the “Topology Change” bit in the flags field and sending out a new configuration message.

While this flag bit is set, all designated bridges reduce their aging time to that of the Forward Delay timer in anticipation of the topology change. Since the topology change could possibly make the data in the filtering database invalid, it must be quickly cleared and the new location of end stations relearned. Under normal conditions, the standard recommends a default aging time of 5 minutes! Changing the time to 15 seconds would be a great help in relearning address locations. Only after the root clears the topology change bit will the designated bridges resume their normal aging time and begin the learning and forwarding operations for the new topology.

### Summary

This section provides an introduction to STP. Since the protocol is quite complex, not all issues have been addressed. Since protocol timers and aging times can vary, it is impossible to predict the time it would take for a network to stabilize after a topology change. STP has been criticized for being too slow in implementing a topology change in an automation network; however, the concepts are similar to faster redundancy schemes such as Rapid Spanning Tree Protocol— so STP should be learned. One advantage of STP is that it is not specific to Ethernet and can operate over wide area networks (WANs) as well. For supervisory control and data acquisition systems (SCADA), the speed of recovery to topology changes might be adequate when temporary loss of communication will not render local control useless.

### References

*ANSI/IEEE Std 802.1D*, 1998 edition: Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—common specifications—Part 3: Media Access. Control (MAC) Bridges, The Institute of Electrical and Electronics Engineers, Inc *The Switch Book*, Rich Seifert, 2000 Wiley Computer Publishing.

## Introduction

What do Ethernet and Modbus have in common? They were both developed in the 1970s and are still widely used today. Of course they have evolved over time, but their basic operation remains intact. Why change a good thing?

There is one basic difference in the two technologies. Ethernet describes the data link and physical layers without a protocol while Modbus is a protocol that can operate over several data links and physical layers. Originally intended as a point-to-interface between proprietary Modicon products, the protocol has found use in multi-drop and peer-to-peer networks like TCP/IP. It is no longer restricted to just Modicon equipment.

## Modicon Modbus Communications Protocol

Modbus was introduced in 1979 by the company “Modicon,” a leader in the infant programmable logic controller (PLC) market. It was intended as the internal point-to-point communications protocol between Modicon PLCs and programming panels used to program the controllers. After some acquisitions, Modicon is now part of Schneider Automation with the brand names “Modicon,” “Square D,” and “Telemecanique.” You would think the protocol would have long been forgotten but the group Modbus-IDA now carries the banner at its <http://www.modbus.org> web site. The protocol continues to thrive since it is easy to understand, and many engineers have “cut their protocol teeth” on Modbus. Besides, it is an open system and can be used royalty-free. It is not restricted to just industrial automation. Modbus can be found in numerous diverse automation industries including building automation.

The *Modicon Modbus Protocol Reference Guide* dated June 1996 can be found at the Modbus-IDA site along with other Modbus references.

When you read the original Modbus documentation you will notice many mentions of specific Modicon equipment. Only later, did the Modbus-IDA group develop generic standards to assist implementation. The three other references are called the Modbus Application Protocol Specification, Modbus over Serial Line Specification and Implementation Guide, and the Modbus Messaging on TCP/IP Implementation Guide. All are available for free. The Modbus protocol would operate over several network implementations including Modicon’s proprietary peer-to-peer network Modbus Plus. Before we examine the more modern implementations in another section, we will concentrate on the protocol itself.

## Original Modicon Implementation

It is interesting to note that Modicon did not use Modbus in a multi-drop network but instead used point-to-point connections with EIA-232C interfaces installed on their PLCs. The Modbus protocol is a master-slave protocol and the terms “master” and “slave” continue to be used today. Modbus allows only one master and up to 247 slaves. A slave is typically a Modicon PLC with an EIA-232C interface. Masters are typically programming panels or host computers.



Therefore, if one host computer needed to communicate to four PLCs, four serial ports would be required on the host computer. This results in a star topology. EIA-232C cable lengths are short, so if longer distances are required modems can be used. It was not until later that 2-wire and 4-wire EIA-485 multi-drop networks appeared.

With the Modbus protocol, only the master can initiate a message. Slaves cannot. So if a slave notices "that the cooling water pumps to the nuclear reactor have stopped," the slave cannot inform the master until the master happens to send a query to the slave with the effective message "how are things going?" The master has no address, but the slaves are numbered from

1 to 247.

Address "0" is reserved as a broadcast address to all slaves. All slaves will receive the broadcast message but will not respond.

<b>Device Address</b>
<b>Function Code</b>
<b>Data Bytes</b>
<b>Error Check</b>

**Figure 1—Simplified Modbus message format.**

### Query—Response Messaging

The command issued by the master is called a Query and the response from the slave is simply called the Response. The format in Figure 1 shows the simplified structure of the messages that can serve as either a query or a response.

The master has no address so the device address is always the intended slave. If it is a query, the query is directed to the slave with the assigned device address. If the message is a response, the response came from the slave with the indicated device address. Commands are issued by function codes such as 03—Read Holding Registers. In this case the master must indicate the range of registers to be queried. The slave responds with the requested data based upon the indicated range. The message format is similar for all function codes, but of course the data changes based upon the code itself. After each message there is an error check appended by the originating station so that the receiving station can check the integrity of the received message.

The above scenario assumes a successful interchange of a query and a response. If the slave wants to communicate an error condition or an exception case, the function code is modified by the slave by setting the most-significant-bit (MSB) of the function code to a 1. The data field will then contain information specific to the exception. The master can still extract the original function code that it sent.

It should be noted that the query-response cycle is completed before the master sends out the next query to either the same slave or another slave. This is unlike protocols such as DeviceNet that can send out one multicast command and then wait for several devices to respond in no particular order to this one command. Modbus has no multicast capability so time is lost as each master query requires the directed slave to not only receive the message but act upon it and respond before the master moves on to other communications activity.

## ASCII and RTU Modes

The simple Modbus protocol becomes a bit more confusing since there are two serial transmission modes. One is called ASCII for American Standard Code for Information Interchange and the other RTU for Remote Terminal Unit. In this case RTU does not mean “rooftop unit.” The RTU term comes from the Supervisor Control and Data Acquisition (SCADA) industry where the master, called a Central Terminal Unit (CTU), communicates to several RTUs at distant locations. This configuration is similar to that of the original Modicon implementation with one CTU communicating to RTUs using modems in a star topology. The use of either ASCII or RTU modes has nothing to do with topology, but it impacts the framing and timing of the messages. When operating over serial communication links, both modes utilize asynchronous communications with one character sent at a time with defined framing.

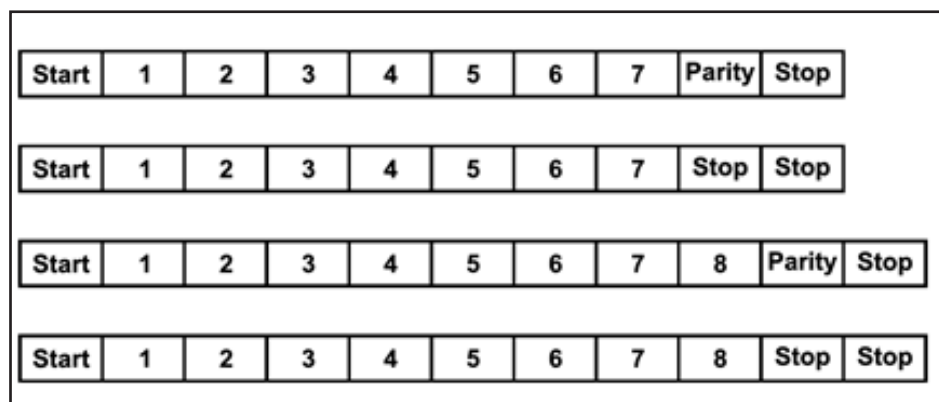


Figure 2 shows how a character is sent using asynchronous serial communications. Each character is sent as a series of bits with a bit time equal to the reciprocal of the baud rate. For example, at 9600 baud the bit time is 104.1  $\mu$ s. When no messages are being sent, the line is said to be marking. The opposite of the marking state is the spacing state. Each character begins with a start bit, as the line drops to the spacing state

**Figure 2—Character framing for 7-bit ASCII and 8-bit RTU with or without parity.**

for one bit time, and ends with one or more stop bits as the line returns to the marking state.

Either a 7-bit (ASCII mode) or 8-bit (RTU mode) character is sent in between—with the least-significant-bit (LSB) sent first. After the character comes either a parity bit or another stop bit. Odd, even, or no parity can be selected by the user. In ASCII mode it takes ten bits to send one character while in RTU mode it takes eleven. With asynchronous communications, characters can be sent either back-to-back or with some delay between characters. A series of characters form messages having different structures depending whether ASCII or RTU mode is intended.

## ASCII Message Framing

The seven-bit ASCII code was developed in the early 1960s as a uniform code for displaying English characters on teleprinters such as the Teletype Model ASR-33. As CRT terminals (glass teletypes) began to replace electromechanical teleprinters, the ASCII standard was retained, making the migration easier. ASCII is a U.S. standard for displaying English characters and for sending control codes such as Carriage Return (CR) and Line Feed (LF) which are carryovers from the teleprinter days. The reason a CR precedes a LF was to give the teleprinter more time to physically move the carriage from the end of a line to the beginning of the line.

Start of Frame	Device Address	Function Code	Data	LRC Check	End of Frame
1 character (:)	2 characters	2 characters	n characters	2 characters	2 characters (CRLF)

**Figure 3—ASCII framing of a Modbus message.**

second in the sequence. Electromechanical teleprinters had no buffering so if an LFCR was executed instead of a CRLF you could have printing in the middle of the page instead of at the beginning because the carriage failed to return in time before the next printable character was received. The CRLF sequence was very important for 10 characters per second teleprinters but not for CRT terminals. For Modbus ASCII mode, the CRLF sequence simply indicates the end of a frame. The advantage of ASCII mode is that if you attached a CRT terminal in place of a slave device, you can observe the nicely formatted human readable code sent by the master on the CRT screen.

The line feed would advance a line vertically, but that activity was faster than moving the carriage so this command was sent

Figure 3 shows the ASCII framing for Modbus messages. The start of frame is simply a colon (:) and the end of frame is the CRLF sequence requiring two ASCII characters. ASCII characters are each 7-bits. All other characters in the other fields must be either the numbers 0–9 or the letters A–F since the data is going to be represented in hexadecimal format but displayed as ASCII characters. For example, function code 03 would be displayed as two ASCII characters "0" and "3." The same applies to the data. One advantage of ASCII mode is timing. As much as one second can elapse between characters without a timeout error. Therefore, a good typist could simulate a master by typing out a character string on a CRT terminal to a slave and observing the slave's response.

### RTU Message Framing

When operating in RTU mode the timing is much more critical. There is no specific Start of Frame character. Instead, the message frame begins with four character times of marking. After this interval, the device address is sent followed by the function code and data. There are other differences from that of the ASCII message frame as noted in Figure 4. Instead of a Longitudinal Redundancy Check (LRC) check in the ASCII frame, a more robust Cyclic Redundancy Check (CRC) check of the data is applied in the RTU frame. The End of Frame indication is strictly based upon four character times of marking.

RTU messages must be sent as a continuous stream and any significant gaps between characters could result in a dropped message. Unlike ASCII, the RTU messages are not human readable. However, the messages are quite compact and more efficient to send. The RTU mode remains the more popular format.

Start of Frame	Device Address	Function Code	Data	CRC Check	End of Frame
4 character times	8 bits	8 bits	n x 8 bits	16 bits	4 character times

**Figure 4—RTU framing of a Modbus message.**

I/O Range	Description
00001 – 09999	Read/Write discrete output or coils
10001 – 19999	Read discrete inputs
30001 – 39999	Read input registers – 16-bit registers such as analog inputs
40001 – 49999	Read/Write holding registers – 16-bit storage or I/O

**Figure 5—Typical Modbus Register Map.**

### Modbus Register Map

Before we discuss function codes, we should study the Modbus register map in Figure 5 since certain function codes imply specific register ranges.

Early PLCs were mostly

concerned with discrete inputs and discrete outputs each represented by one-bit in a register map. For Modicon PLCs, discrete outputs begin at location 00001 and discrete inputs begin at location 10001. Each requires one-bit of storage. Inputs, called contacts, can only be read and outputs, called coils, can be read or written. Since early PLCs were considered relay panel replacements, the terms coils and contacts were retained to assist electricians trying to understand these new electronic controllers.

As the complexity of PLCs increased, the ability to handle analog input/output (I/O), and to execute math calculations was added. The I/O range of 16-bit register references begins at 30001 for read-only analog inputs or thumbwheel switches, and 40001 for general purpose read/write registers that can also serve as analog outputs. There are really no restrictions above 40001. Depending upon the vendor of the equipment, they can be internal registers, analog inputs, analog outputs, and even discrete inputs and outputs. However, not all function codes reference this range—but enough do.

### Function Codes

The Modbus function codes are defined in both the Modicon Modbus Protocol Reference Guide and the Modbus Application Protocol Specification. Because there are differences in the function names and the number of function codes the latter document is recommended. Although the function code range spans from 1 to 127, only about 20 are defined public function codes. User-defined function codes are allowed in specific locations within this range. However, many Modbus devices only support a small subset of the available codes. We will only examine those function codes that involve single-bit and 16-bit data access to get a flavor of how I/O is handled. A list is provided in Figure 6.

Code	1/16-bit	Description	I/O Range
01	1-bit	Read coils	00001 – 09999
02	1-bit	Read contacts	10001 – 19999
05	1-bit	Write a single coil	00001 – 09999
15	1-bit	Write multiple coils	00001 – 09999
03	16-bit	Read holding registers	40001 – 49999
04	16-bit	Read input registers	30001 – 39999
06	16-bit	Write single register	40001 – 49999
16	16-bit	Write multiple registers	40001 – 49999
22	16-bit	Mask write register	40001 – 49999
23	16-bit	Read/write multiple registers	40001 – 49999
24	16-bit	Read FIFO queue	40001 – 49999

**Figure 6—Data access function codes.**

You will notice from Figure 6 that 1-bit function codes relate to discrete devices such as contacts and coils. The 16-bit function codes relate to input registers and holding registers. Input registers can only be read while holding registers can be either read or written. Also notice there is an implied I/O range depending upon the function code. For example, function code 06—Write single register, only addresses the relevant range of 40001–49999 and no other range. Therefore it is only necessary to reference the offset from the base range when structuring the message. Instead of indicating register location 40001 we simply say 0000.

This is a good time to explain one of the more confusing aspects of Modbus and that is referencing I/O points in Modbus messages. Modicon elected to number physical points within a range beginning with the number 1 instead of 0. Coil 1 is referenced in a message as location 0000 and not 00001. Likewise, Contact 1 is referenced as 0000 instead of 10001. The same applies to holding register 40001. It is also referenced as 0000. The function code always points to the proper I/O range and only the offset from base address of that range is needed to uniquely identify the point.

### Summary

Modbus is popular for its simplicity. With so many users in the field with Modbus knowledge and a Modbus-IDA association backing this open standard, it will continue to remain popular.

### References

*Modbus Application Protocol Specification V1.1b*, <http://www.Modbus-IDA.org>, December 28, 2006.  
*Modbus over Serial Line Specification and Implementation Guide*, V1.02, <http://www.Modbus-IDA.org>, December 20, 2006.  
*Modbus Messaging on TCP/IP Implementation Guide V1.0b*, <http://www.Modbus-IDA.org>, October 24, 2006.  
*Modbus Protocol Reference Guide* Rev J, <http://www.Modbus-IDA.org>, June 1996.

## Modbus Serial and Modbus TCP

### Introduction

There are two implementations of the Modbus protocol that remain important. The first implementation is the traditional implementation of Modbus over a serial line. The second implementation is more modern with Modbus operating over a TCP/IP network. Both implementations remain popular.

### Modbus over Serial Line

Modbus.org has released a *Modbus over Serial Line Specification and Implementation Guide V1.02* that provides guidance when using Modbus with serial links. As mentioned in a previous section, Modbus was originally intended to be used with point-to-point EIA-232C interfaces with the master being a Human Machine Interface (HMI) and a PLC as the slave. Multiple slaves connected to one master would then require multiple links which is inconvenient

and expensive. It would be only natural to change the point-to-point link to a multipoint serial infrastructure such as EIA-485 which would allow one master to communicate to multiple slaves over a common serial line. This approach is encouraged in the Modbus.org document, but not mentioned in the original *Modicon Modbus Protocol Reference Guide*.

Layer	ISO/OSI Function	Modbus Function
7	Application	Modbus Application Protocol
3–6	Various	Null
2	Data Link	Modbus Serial Line Protocol
1	Physical	EIA-485, EIA-232C

Table 1—Modbus over Serial Line uses a three-layer model.

### Three Layer Model

Instead of the traditional seven-layer ISO Open Systems Interconnection Reference Model, the Modbus over Serial Line model is collapsed to three layers as shown in Table 1. At the top is the application layer that was discussed previously. This is called the Modbus Application Protocol or simply the Modbus Protocol. Layers 3–6 are not used—instead, the model relies on the application layer to ensure end-to-end delivery of a message. The data link (layer 2) is occupied by the Modbus Serial Line Protocol. Finally, the physical layer (layer 1) allows for either the EIA-232C or EIA-485 implementation. With only three layers, Modbus over Serial Line is easier to understand than other automation protocols. Since the Modbus Application Protocol was discussed previously, it will not be repeated here. Instead, only the data link and physical layers will be discussed.

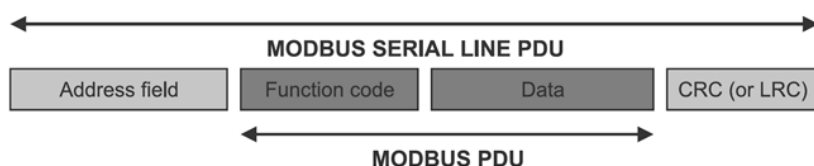


Figure 1—A slave address field and error check wrap around a Modbus PDU.

Slave Address	Function Code	Data	CRC
1 byte	1 byte	0 up to 252 bytes(s)	2 bytes CRC Low   CRC Hi

Figure 2—RTU framing is more condensed than ASCII framing.

### Data Link Layer

Much about the Modbus over Serial Line Protocol Data Unit (PDU) was mentioned in a previous section but will be summarized below. Referring to Figure 1, note that the PDU consists of four elements. In the middle is the Modbus PDU consisting of a function code and data. Most Modbus implementations only use a subset of all the available function codes. The data structure may change depending upon the function code. Wrapping the Modbus PDU is an address field and an error check field. The address field only contains slave addresses or the broadcast address. The master address is not required and not referenced since this is a master/slave protocol with commands originating from a unique master.



As previously mentioned, the actual framing of Modbus over Serial Line messages depends on whether ASCII or RTU transmission mode is used. RTU is the most popular mode and is shown in Figure 2. It is a very compact frame with only one byte reserved for the slave or broadcast address, one byte for

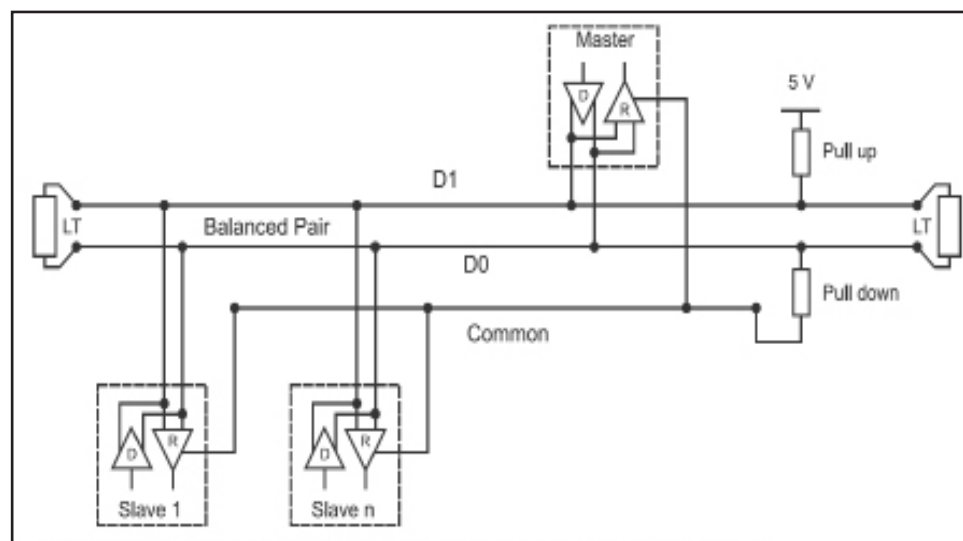
Start	Slave Address	Function Code	Data	LRC	End
1 char	2 chars	2 chars	0 up to 2x252 char(s)	2 chars	2 chars CR,LF

**Figure 3—ASCII framing requires start-of-frame and end-of-frame characters.**

the function code and two bytes for the CRC error check. A single byte carries the function code. Notice that there is no end-of-frame sequence. With RTU, end-of-frame is indicated by 3.5–4.5 character times of silence.

The largest frame occupies only 256 bytes. With RTU each byte is sent using 11 bits. Each data character requires eight bits. There is one start-bit, one stop-bit and one parity-bit. If parity is not used, another stop-bit is sent in its place. If parity is employed, it can be odd or even.

The ASCII message format in Figure 3 requires two bytes for the slave address as well as for the function code. Unlike RTU, the ASCII frame uses a two-byte LRC for the error check field. The advantage of the ASCII format is that it is human readable. Notice that there is an end-of-frame sequence composed of carriage return and line feed (CR, LF). Inter-frame spacing is not critical. Data is



**Figure 4—A two-wire serial line implementation actually requires three wires.**

represented as a hexadecimal value coded as ASCII. Therefore, only 7 bits are required for every ASCII character, but two characters are required for each byte of data. One start-bit and one stop-bit are used. If a parity-bit is used, either odd or even parity can be sent. If no parity is sent, then another stop-bit is sent. This means each byte in ASCII is sent as 10 bits.

### Physical Layer

The original Modbus protocol called for a point-to-point EIA-232C link

between a host computer and a PLC. This option remains today. But the Modbus over Serial Line specification encourages the use of the multipoint EIA-485 standard—supporting up to 32 devices over a common bus. This can be implemented with either a two-wire or four-wire cabling configuration. With any of the serial line implementations, a wide range of baud rates from 1.2 kbps to 115 kbps are allowed, but all implementations must at least support 9.6 kbps and 19.2 kbps. The default rate is 19.2 kbps.

### Two-Wire Network

Figure 4 shows a recommended two-wire interface for EIA-485 networks with applied line polarization. As expected, there is one master transceiver and multiple slave transceivers connected to a common 2-wire bus with the wires labeled D1 and D0.

At a minimum, a total of 32 devices must be supported. With a 2-wire bus, the output of the transmitter is directly tied to the input of a receiver at each device. Even though this is called a 2-wire bus, there is a common reference connection labeled common. Each device must share its common with all other devices on the bus to ensure that the maximum common-mode voltage rating of the device is not exceeded. The line polarization network (consisting of a pull-up and pull-down resistor) is shown near the master, but its location is not a requirement—only a recommendation.

Line polarization is used to force the bus into a known state when no drivers are active. EIA-485 receivers require a 200 mv failsafe bias to ensure they detect a floating line as an “off” state. This is why line polarization is typically referred to as failsafe bias. At each end of the bus are line terminators (LT) to match the natural impedance of the bus. The pull-up and pull-down resistors interact with the two termination resistors to create the failsafe bias. The Modbus over Serial Line specification recommends that the pull-up and pull-down resistors have values between 450 and 650 ohms, and that only one network is used. This assumes that failsafe bias is needed at all. Some transceivers have built-in bias so external bias is not needed.

### Four-Wire Network

Figure 5 shows a recommended four-wire interface still using EIA-485 devices. The transmitter of each device is separated from the device’s receiver. The master has its transmitter connected to all the slaves’ receivers while all the slaves’ transmitters are connected to the master’s receiver. Failsafe bias and termination are still used — but their requirements are doubled in a four-wire network. Even the four-wire arrangement requires a “fifth” wire and that is the common.

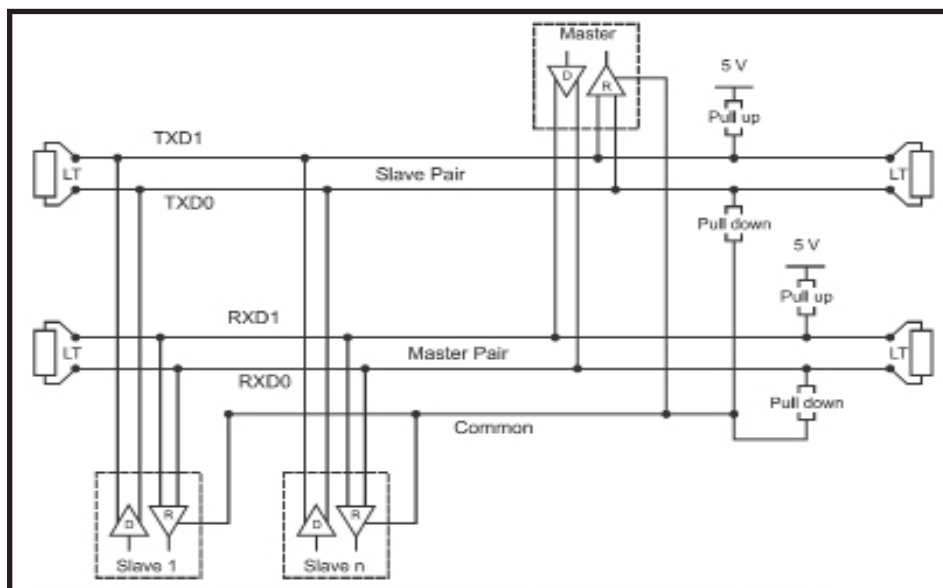


Figure 5—A four-wire serial line implementation actually requires five wires.

Although the Modbus over Serial Line specification supports both a 2-wire and 4-wire arrangement, the 2-wire implementation is the most popular. Although it is possible to have a full-duplex link with 4 wires, the Modbus protocol is strictly half-duplex. The master imitates commands to a particular slave while awaiting the slave’s response. This is handled quite effectively with a 2-wire implementation.

## Modbus TCP

The Modbus protocol continues to survive in an automation world more interested with connecting to Ethernet networks and more specifically, IP/Ethernet networks. Modbus.org authored the Modbus Messaging on TCP/IP Implementation Guide V1.0b for this very purpose. Instead of a three-layer model that was used for Modbus over Serial Line, a five-layer Internet model was used for Modbus TCP as shown in Table 2. Instead of a long discussion on physical and data link layer

Layer	ISO/OSI Function	Modbus Function
5,6,7	Application	Modbus Application Protocol
4	Transport	Transmission Control Protocol
3	Network	Internet Protocol
2	Data Link	IEEE 802.3
1	Physical	IEEE 802.3

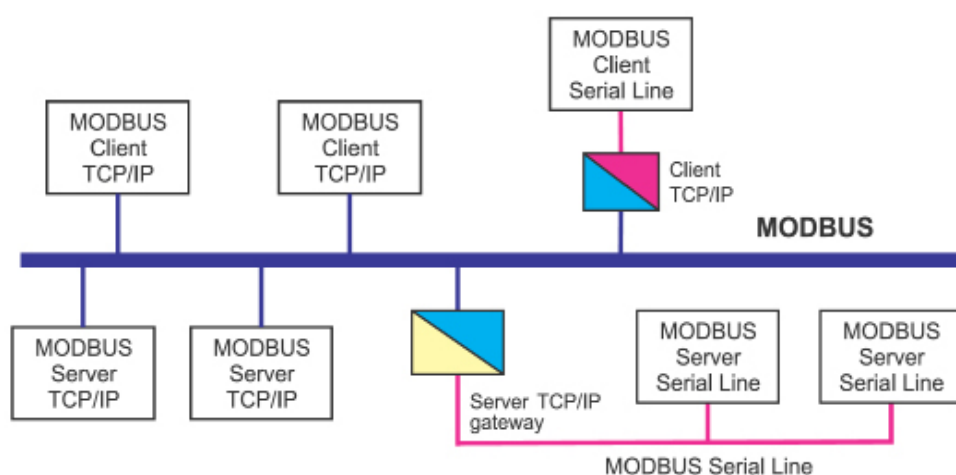
Table 2—Modbus TCP uses a five-layer Internet model.

issues, the standard only needs to point to the IEEE 802.3 standard. There is no mention of how to physically attach stations or what cabling or connectors to use.

This messaging standard only talks about how a Modbus PDU (consisting of a function code and data) is encapsulated into a higher-level protocol.

Another significant change (shown in Figure 6) is that the Modbus bus is actually an IP bus. The physical and data link layers are not specified. Instead of having one master attached to multiple slaves, the terms client and server are used. Clients could be HMIs or PLCs, while servers could be input/output racks. Like a master, clients initiate commands to a server. Like a slave, servers respond to client commands. However, the proper terminology with client/server communications is that clients initiate requests with servers providing responses. It is actually a bit more involved than that.

- A Request is sent by the client to initiate a transaction.
- An Indication is sent by the server to confirm that a request was received.
- A Response is sent by the server to comply with the client request.
- A Confirmation is sent by the client to acknowledge receipt of the response.



What is significant in this model is that several clients can reside on the IP network and access a common set of servers. This is a fundamental change in how the Modbus protocol works. There is no single master controlling a defined set of slaves. Any number of clients can access any number of servers. Is it possible to have conflicts with clients making contradictory requests of a particular server? Yes, that is the risk this model presents with its newly gained flexibility.

Figure 6—Instead of using master and slaves, the Modbus TCP model uses clients and servers.

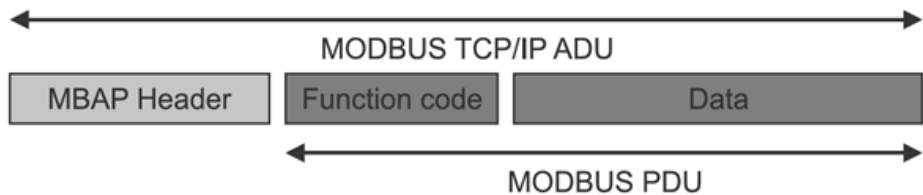


Figure 7—The Modbus Application Protocol header is added to the Modbus PDU.

Transaction Identifier	Protocol Identifier	Length	Unit Identifier
2 bytes	2 bytes	2 bytes	1 byte

Figure 8—The MBAP header is seven bytes long.

### The MBAP Header

Figure 7 shows how a new Modbus TCP/IP Application Data Unit (ADU) is formed. The traditional Modbus PDU of the Modbus over Serial Line method is still present. The function code and data definitions remain intact. What is appended to this PDU is a Modbus Application Protocol (MBAP) header, details of which are shown in Figure 8. The Transaction Identifier is supplied by the client and used to keep track of specific requests.

The server is to send back the same identifier with its response. The client is allowed to send multiple requests to a server without waiting for individual responses. The Protocol Identifier would allow support for multiple protocols. For Modbus the value is zero. The Length field identifies the length of all remaining fields including the Modbus PDU fields. Finally, the Unit Identifier provides the address of a Modbus Serial Line slave that must be accessed through a gateway.

With Modbus TCP clients and servers, station addressing occurs by using IP addresses. However, if a Modbus slave is attached to a serial line, the actual slave address needs to be specified. The gateway address would then be an IP address. In order to send the ADU over TCP, a registered TCP port number must be used. Modbus.org registered port 502 for this purpose.

### Summary

Modbus is popular for its simplicity. Because so many users with Modbus knowledge are in the field and because a Modbus-IDA association backs this open standard, it will continue to remain popular.

### References

*Modbus Application Protocol Specification V1.1b*, <http://www.Modbus-IDA.org>, December 28, 2006  
*Modbus over Serial Line Specification and Implementation Guide V1.02*, <http://www.Modbus-IDA.org>, December 20, 2006  
*Modbus Messaging on TCP/IP Implementation Guide V1.0b*, <http://www.Modbus-IDA.org>, October 24, 2006  
*Modbus Protocol Reference Guide Rev. J*, <http://www.Modbus-IDA.org>, June 1996

# Modeling a BACnet Physical Device

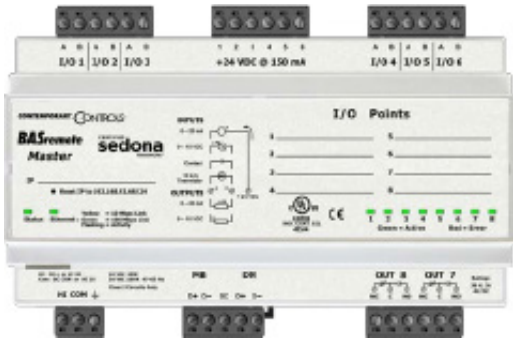


Figure 1—The BASremote has six universal input/output points and two relay outputs.

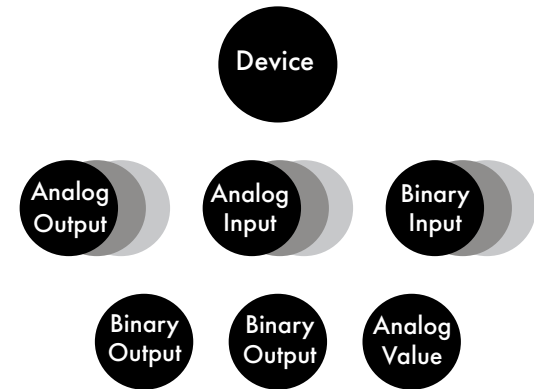


Figure 2—The BASremote can be described as a collection of objects with one identifying the device itself.

## Introduction

Modern control networks such as Rockwell Automation’s EtherNet/IP, Modbus/TCP and BACnet/IP use Ethernet for communications due to its high speed, lowering cost and in some instances, the necessity to operate over structured wiring. Although the study of Ethernet does not require an understanding of application protocols, knowledge of application protocols has become increasingly important as modern networks are deployed. The latest protocols are all based upon Object Modeling which can be quite confusing to someone who has not been exposed to this abstract concept. This section introduces object modeling, object properties, and services as they pertain to a physical BACnet/IP device. Although the majority of BACnet devices support the master-slave/token-passing (MS/TP) network, newer devices now function over Ethernet.

## BASremote as a Physical Device

Although any other product could have been used, we will use as our physical device example the BASremote by Contemporary Controls for this discussion. This product is intended as a remote input/output device complying with the BACnet/IP standard. When connected to an Ethernet infrastructure, the BASremote provides eight input/output points that can be accessed by any device on a BACnet network. Six of those points are universal input/outputs—meaning they can be field configurable to be either an analog or digital input or analog output. Two are fixed as relay outputs. The BASremote supports the BACnet/IP standard as defined in Annex J of the BACnet standard and therefore, requires no router in order to connect to Ethernet. But how do other devices access the BASremote? To explain how this works, one needs to study BACnet’s object model and services.

## The BACnet Standard

The current revision of the BACnet standard is ANSI/ASHRAE 135-2012 from the American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. If you purchase the standard in print from the ASHRAE bookstore, you will receive a 1000+ page bound copy. Revisions to the standard occur about every two years proving that the BACnet standard continues to evolve. Like any other standard, the BACnet standard is not written as a tutorial on BACnet although some exist on the web. However, since the BASremote is a relatively simple BACnet device, studying this product offers an opportunity to learn how a device becomes BACnet compliant.

## Object Modeling the BASremote

Observing the BASremote sitting on a table, it appears in its physical form as a plastic enclosure with screw connectors for powering the unit and for connecting to field devices such as sensors and actuators (see Figure 1). According to the BASremote data sheet, in its simplest form it has six universal input/output points and two relay outputs. The six universal I/Os can be individually configured to be an analog input, analog output or digital input. In addition, the circuitry is flexible in that it can accommodate several voltage and current levels such as 0–10 VDC or 4–20 mA. Contact closures or thermistors can be sensed directly. There is one RJ-45 connector for the Ethernet network. To someone in the industry, the BASremote would be considered an eight-point remote I/O device with an Ethernet connection.

## BACnet Object Types

### Basic Device Object Type

Device  
Analog Input  
Analog Output  
Analog Value  
Binary Input  
Binary Output  
Binary Value  
Multi-state Input  
Multi-state Output  
Multi-state Value  
File

### Process-related Object Types Access Zone

Averaging  
Loop  
Program

### Control-related Object Types

Command  
Load Control

### Meter-related Object Types

Accumulator  
Pulse Converter

### Presentation-related Object Types

Group  
Global Group  
Structured View

### Schedule-related Object Types

Calendar  
Schedule

### Notification-related Object Types

Event Enrollment  
Notification Class  
Notification Forwarder  
Alert Enrollment

### Logging Object Types

Event Log  
Trend Log  
Trend Log Multiple

### Life Safety and Security

Object Types  
Life Safety Point  
Life Safety Zone  
Network Security

### Life Safety and Security Object Types

Access Point  
Access Door  
Access User  
Access Rights  
Access Credential  
Credential Data Input

### Simple Value Object Types

CharacterString Value  
DateTime Value  
Large Analog Value  
BitString Value  
OctetString Value  
Time Value  
Integer Value  
Positive Integer Value  
Date Value  
DateTime Pattern Value  
Time Pattern Value  
Date Pattern Value

### Lighting Control Object Types

Channel  
Lighting Output

**Table 1—There are 54 BACnet object types. The BASremote uses five which are highlighted.**

Obviously, the Ethernet connection allows network access to the various I/O points but how is this accomplished in the BACnet world? Once configured for the needs of the application, how can each physical point on the BASremote be queried from any point on the network without any prior knowledge of the capabilities of the BASremote? Object modeling is the answer.

Although the BASremote is an eight-point I/O device, we know little more. To understand this device or any other BACnet device, it is best to view these devices as a collection of objects. An object is an abstract concept that allows us to identify both physical items such as I/O points, and non-physical concepts such as software and calculations. In Figure 2, we notice that the BASremote can be represented as a collection of objects of varying types such as Analog Input, Analog Output, Analog Value, Binary Input, Binary Output and Device. If no virtual points are created for the BASremote, there will be no Analog Values created. Assume for simplicity that none were created. Since the BASremote has eight points of I/O, there will be eight objects created plus one for the device itself. Since six of the I/O points are universal, the number of each object type will vary depending upon the final configuration. This is not true for the Device object since only one is allowed per physical device. There will be two Binary Output objects to correspond to the two relay outputs on the BASremote. In total, the BASremote can be modeled as a collection of nine objects with up to five object types – six object types if virtual points are created. As shown in Table 1, the current BACnet standard identifies 54 object types that would typically be found in building automation systems. BACnet compliant devices are not required to implement all 54. standard identifies 54 object types that would typically be found in building automation systems. BACnet—compliant devices are not required to implement all 54.

## Objects Have Properties

Although objects can represent physical points, they can also describe processes or internal operations. Each object has a listing of properties that tell us about the nature of the object. This object listing varies with the object type. What the BACnet standard does is define the object types and the property fields for each object type. In this way, uniformity is ensured among vendors that will lead to interoperability between different vendor systems. The intent is to make the objects “network visible”—any object can be queried from any point along the network. As the standard evolved, object types have been added from the initial 18 to the current 54. These 54 object types are referred to as standard objects. A BACnet device need not support all object types, but if an object type supported, it must comply with the standard object model for that object type. Each object type has a list of required properties and optional properties. Optional properties can be included at the manufacturer’s discretion.



Property Identifier	Value
Object_Identifier	(Device, Instance 2749)
Object_Name	"RE1 Penthouse"
Object_Type	DEVICE
System_Status	(OPERATIONAL)
Vendor_Name	"Contemporary Controls"
Vendor_Identifier	245
Model_Name	"BAS-8M"
Firmware_Revision	"1.0"
Application_Software_Version	"1.0"
Protocol_Version	2
Protocol_Revision	
Protocol_Services_Supported	(List of Services)
Protocol_Object_Types_Supported	(List of Object Types)
Object_List	(List of all Objects)
Max_APDU_Length_Accepted	1476
Segmentation_Supported	(NO SEGMENT)
APDU_Timeout	(3000 MSEC)
Number_Of_APDU_Retries	0
Device_Address_Binding	
Database_Revision	1

**Table 2—The required properties of the Device object are listed. From this object, much can be learned about the device.**

There is one object type that must be included in any BACnet-compatible device and that is the Device object. There must be one and only one instance of the Device object whose structure is shown in Table 2. An instance number is the way of identifying items using object modeling. The Object Identifier must be unique within the complete BACnet network. It is comprised of the object type Device and an instance number.

The default instance is 2749, but there is no significance to this value since it is changed by the installer through the web page during commissioning. The Object Name must also be unique within the complete BACnet network. The remaining fields are set by the manufacturer.

From the other properties you can learn much about the device. All the object types present in the device are listed under Protocol Object Types Supported. All the objects within the device along with their instance number are included under Object List.

In our example of the BASremote, there could be five object types and nine objects listed. We will study one of those objects of the type Analog Input. In Table 3 you will see all the required properties of one Analog Input instance which the manufacturer identifies as AI, Instance 2 meaning that the second physical point on the BASremote happens to be configured as an analog input. All properties in any object must be able to be read. Point 2 on the BASremote is connected to a thermistor located on the roof of the building in order to measure outside temperature. The actual temperature appears in the Present Value property as 69.3. The unit of measure is degrees Fahrenheit as indicated in the Units property. The installer has identified the point as "Outside temperature" in the Object Name property meaning that another BACnet device on the network can search for "Outside temperature" and obtain the actual outside temperature without caring which device or which point had the result. This demonstrates the power of object modeling.

Property Identifier	Remarks
Object_Identifier	(Analog Input Instance 2)
Object_Name	Outside Temperature
Object_Type	ANALOG INPUT (0)
Present_Value	69.3
Status_Flags	(All FALSE)
Event_State	NORMAL (0)
Out_of_Service	FALSE
Units	DEGREES_FAHRENHEIT (64)

**Table 3—The required properties of the Analog Input object for Instance 2. Notice that the value of the point is maintained in engineering units.**

### Devices Provide Services

Having just objects with properties is not sufficient. We need services so that one BACnet device can access data from another or to command another device to take action. In order to learn what the outside temperature is, we would use the service called Read Property which allows the reading of properties within an object. In fact this service is the one service that all BACnet devices must process. For example, the BASremote's point 2 knows the outside temperature. Another device functioning as a client makes a Read Property request to the BASremote which functions as the server. The request is made to the Present Value property in the Analog Input, Instance 2 object. The BASremote receives the request and executes the command by providing the information.

A confirmed service is one where a service request is initiated and a response with data is expected. An unconfirmed service expects no reply. Depending upon the capabilities of a device, it may be able to initiate a confirmed service request or respond to a confirmed service request. Some devices can do both. The same applies to unconfirmed services.

There are 38 possible service requests and they are grouped into five categories (see Table 4).

- Alarm and Event Services (11)
- File Access Services (2)
- Object Access Services (10)
- Remote Device Management Services (12)
- Virtual Terminal Services (3)

The Read Property service is included in the Object Access Services group as well as the Write Property service necessary to set outputs. These are both confirmed services. With these two simple operations, we can read and write I/O points.

Other interesting services are the Who-Is and I-Am which are unconfirmed services used to “discover” devices. With the Who-Is request, an initiating device is trying to determine the device object identifier; the actual network address of a device, or both. A range of device object identifiers can be sent to restrict the search. In the most basic request, the Who-Is is sent as a broadcast message with no device object identifier restrictions. Since it is a broadcast message, all devices on the network will hear the request, but only those devices that can execute an I-Am service will respond. Each device capable of responding will send out their device object identifier along with some limited information on the device itself including the Vendor Identifier. With BACnet/IP, the response is sent as a broadcast UDP message so the source address can be learned from the response.

The initiating device or any other listening device on the network can then construct a table of device object identifiers versus IP addresses so that future communication would not require this discovery process. Although, the I-Am response logically follows the Who-Is request, an I-Am can be initiated at any time. It is usually sent when a device joins the network and announces to devices on the network that it is present.

A similar set of services exist with the Who-Has and I-Have. In this case, the initiator is trying to determine the device object identifier, network address or both, which contains a particular object identifier or object name. For example, we are trying to learn where we can access outside temperature by entering the Object Name “Outside temperature.” Since we do not know the object identifier, we will simply send out the object name in the Who-Has request. The response to the Who-Has is an I-Have which returns the device object identifier, object identifier, and object name. The network address can be learned as well. In our example, the BASremote has the Object Name “Outside temperature” so it would respond with a I-Have message with the device object identifier of 2749, the object identifier Analog Input, Instance 2, and the object name “Outside temperature.” Although an I-Have is the proper response to a Who-Has, an I-Have can be initiated at any time without a Who-Has.

BACnet Services

<b>Alarm and Event Services</b> AcknowledgeAlarm ConfirmedCOVNotification UnconfirmedCOVNotification ConfirmedEventNotification UnconfirmedEventNotification GetAlarmSummary GetEnrollmentSummary GetEventInformation LifeSafetyOperation SubscribeCOV SubscribeCOVProperty	<b>Remote Device Management Services</b> DeviceCommunicationControl ConfirmedPrivateTransfer UnconfirmedPrivateTransfer ReinitializeDevice ConfirmedTextMessage UnconfirmedTextMessage TimeSynchronization UTCTimeSynchronization Who-Has I-Have Who-Is I-Am
<b>File Access Services</b> AtomicReadFile AtomicWriteFile	<b>Virtual Terminal Services</b> VT-Open VT-Close VT-Data
<b>Object Access Services</b> AddListElement RemoveListElement CreateObject DeleteObject ReadProperty ReadPropertyMultiple ReadRange WriteProperty WritePropertyMultiple WriteGroup	

Table 4 —38 services are grouped into five categories.

With 38 possible services, discussing each service would make for a lengthy discussion. However, the concepts are the same as with the simple Read Property and Write Property services. As we will learn later, your simpler devices are only required to implement a fraction of the available services.

Summary

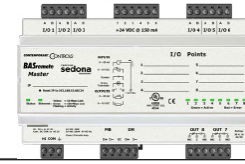
Although object modeling appears complex and awkward, it is the modern method of making devices “network visible.” Objects are used to describe physical and the types of objects are defined in the BACnet standard. Objects have properties and the property list varies with the object type. In order to use objects to our advantage, devices must be able to provide services. The types of services are also defined in the BACnet standard. Objects, object properties, and services begin to define BACnet-compliant devices.

References

ANSI/ASHRAE Standard 135-2012  
BACnet—A Data Communication Protocol for Building Automation and Control Networks, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.  
Direct Digital Control of Building Systems Theory and Practice, H. Michael Newman, John Wiley and Sons, Inc., 1994.  
<http://www.polarsoft.biz/langbac.html>, The Language of BACnet—Objects, Properties and Services, Bill Swan, Alerton Technologies, Inc.  
BACnet Explained, H. Michael Newman, ASHRAE Journal, November 2013.

## BASremote

Versatile BACnet/IP Controller/Gateway



### BACnet Protocol Implementation Conformance Statement (Annex A)

Date: October 24, 2013  
 Vendor Name: Contemporary Controls  
 Product Name: BASremote  
 Product Model Number: BASR-8M  
 Applications Software Version: 3.7.0    Firmware Revision: 3.7.0    BACnet Protocol Revision: 2  
 Product Description: BACnet/IP compliant 8-point Sedona Framework controller with Modbus Gateway.

#### BACnet Standardized Device Profile (Annex L):

- |   |  |
|---|--|
| <input type="checkbox"/> BACnet Operator Workstation (B-OWS)          | <input type="checkbox"/> BACnet Advanced Application Controller (B-AAC)            |
| <input type="checkbox"/> BACnet Advanced Operator Workstation (B-AWS) | <input checked="" type="checkbox"/> BACnet Application Specific Controller (B-ASC) |
| <input type="checkbox"/> BACnet Operator Display (B-OD)               | <input type="checkbox"/> BACnet Smart Sensor (B-SS)                                |
| <input type="checkbox"/> BACnet Building Controller (B-BC)            | <input type="checkbox"/> BACnet Smart Actuator (B-SA)                              |

#### List all BACnet Interoperability Building Block Supported (Annex K):

- |  |   |
|--|---|
| DS-RP-B Data Sharing — ReadProperty — B          | DM-DDB-B Device Management — Dynamic Device Binding — B       |
| DS-WP-B Data Sharing — WriteProperty — B         | DM-DOB-B Device Management — Dynamic Object Binding — B       |
| DS-RPM-B Data Sharing — ReadPropertyMultiple — B | DM-DCC-B Device Management — Device Communication Control — B |
| DS-COV-B Data Sharing — ChangeOfValue — B        | DM-TS-B Device Management — Time Synchronization — B          |

#### Segmentation Capability:

- |  |              |
|--|--------------|
| <input type="checkbox"/> Able to transmit segmented messages | Window Size: |
| <input type="checkbox"/> Able to receive segmented messages  | Window Size: |

#### Standard Object Types Supported:

Object Type Supported	Can Be Created Dynamically	Can Be Deleted Dynamically
Analog Input	No	No
Analog Output	No	No
Analog Value	No	No
Binary Input	No	No
Binary Output	No	No
Device	No	No

No optional properties are supported.

#### Data Link Layer Options:

- |  |   |
|--|---|
| <input checked="" type="checkbox"/> BACnet IP, (Annex J)                     | <input type="checkbox"/> MS/TP slave (Clause 9), baud rate(s):              |
| <input checked="" type="checkbox"/> BACnet IP, (Annex J), Foreign Device     | <input type="checkbox"/> Point-To-Point, EIA 232 (Clause 10), baud rate(s): |
| <input type="checkbox"/> ISO 8802-3, Ethernet (Clause 7)                     | <input type="checkbox"/> Point-To-Point, modem, (Clause 10), baud rate(s):  |
| <input type="checkbox"/> ATA 878.1, 2.5 Mb. ARCNET (Clause 8)                | <input type="checkbox"/> LonTalk, (Clause 11), medium:                      |
| <input type="checkbox"/> ATA 878.1, EIA-485 ARCNET (Clause 8), baud rate(s): | <input type="checkbox"/> BACnet/Zigbee (Annex O)                            |
| <input type="checkbox"/> MS/TP master (Clause 9), baud rate(s):              | <input type="checkbox"/> Other:   |

#### Device Address Binding:

Is static device binding supported? (This is currently necessary for two-way communication with MS/TP slaves and certain other devices.) ☐ Yes ☒ No

#### Networking Options:

- ☐ Router, Clause 6 — List all routing configurations, e.g., ARCNET-Ethernet-MS/TP, etc.  
☐ Annex H, BACnet Tunnelling Router over IP  
☐ BACnet/IP Broadcast Management Device (BBMD)  
 Does the BBMD support registrations by Foreign Devices? ☐ Yes ☐ No  
 Does the BBMD support network address translation? ☐ Yes ☐ No

#### Character Sets Supported:

Indicating support for multiple character sets does not imply that they can all be supported simultaneously.

- |   |   |                                     |
|---|---|-------------------------------------|
| <input checked="" type="checkbox"/> ISO 10646 (UTF-8) | <input type="checkbox"/> IBM™/Microsoft™ DBCS | <input type="checkbox"/> ISO 8859-1 |
| <input type="checkbox"/> ISO 10646 (UCS-2)            | <input type="checkbox"/> ISO 10646 (UCS-4)    | <input type="checkbox"/> JIS X 0208 |

If this product is a communication gateway, describe the types of non-BACnet equipment/network(s) that the gateway supports:  
 Modbus gateway support.

#### Network Security Options:

- ☒ Non-secure Device — is capable of operating without BACnet Network Security  
☐ Secure Device — is capable of using BACnet Network Security (NS-SD BIBB)  
☐ Key Server (NS-KS BIBB)

Table 5—PICS Statement for BAS Remote

# Achieving BACnet Compliance

## Introduction

The first section on BACnet introduced object modeling, object properties, and services as they pertain to a physical BACnet/IP device. This section continues the discussion by addressing the requirements for achieving BACnet compliance.

BACnet Service	Initiate	Execute
ReadProperty	X	

Table 1—BIBB-Data Sharing-ReadProperty-A (DS-RP-A)

BACnet Service	Initiate	Execute
ReadProperty		X

Table 2—BIBB-Data Sharing-ReadProperty-B (DS-RP-B)

## BACnet Interoperable Building Blocks (BIBBs)

A primary goal of the BACnet standard is interoperability among vendors of BACnet equipment. Users also need to make sense of the vast range of BACnet products from simple sensors, workstations and building controllers. Not all devices need to provide the same services so how do we classify devices? The BIBB concept was introduced later in the standards development process as a way of classifying the numerous available services into more manageable processes as would be needed in a building automation application. The resulting 107 BIBBs can be found in Annex K and are classified into five groups:

- Data Sharing (21)
- Alarm and Event Management (24)
- Scheduling (8)
- Trending (13)
- Device and Network Management (41)

	B-AWS	B-OWS	B-OD	B-BC	B-AAC	B-ASC	B-SA	B-SS
<b>Data Sharing</b>	DS-RP-A,B DS-RPM-A DS-WP-A DS-WPM-A DS-AV-A DS-AM-A	DS-RP-A,B DS-RPM-A DS-WP-A DS-WPM-A DS-V-A DS-M-A	DS-RP-A,B DS-WP-A DS-V-A DS-M-A	DS-RP-A,B DS-RPM-A,B DS-WP-A,B DS-WPM-B	DS-RP-B DS-RPM-B DS-WP-B DS-WPM-B	DS-RP-B DS-WP-B	DS-RP-B DS-WP-B	DS-RP-B
<b>Alarm, Event Management</b>	AE-N-A AE-ACK-A  AE-AS-A AE-AVM-A AE-AVN-A AE-ELVM-A <sup>2</sup>	AE-N-A AE-ACK-A  AE-AS-A AE-VM-A AE-VN-A	AC-N-A   AE-VN-A	AE-N-I-B AE-ACK-B AE-INFO-B AE-ESUM-B	AE-N-I-B AE-ACK-B AE-INFO-B			
<b>Scheduling</b>	SCHED-AVM-A	SCHED-VM-A		SCHED-E-B	SCHED-I-B			
<b>Trending</b>	T-AVM-A	T-V-A		T-VMT-I-B T-ATR-B				
<b>Device &amp; Network Management</b>	DM-DDB-A,B DM-ANM-A DM-ADM-A DM-DCB-B DM-DCC-B DM-MTS-A  DM-OCD-A DM-RD-A DM-BR-A	DM-DDB-A,B  DM-DOB-B DM-MTS-A	DM-DDB-A,B  DM-DOB-B	DM-DDB-A,B  DM-DOB-B DM-DCC-B DM-TS-B or DM-UTC-B  DM-RD-B DM-BR-B	DM-DDB-B  DM-DOB-B DM-DCC-B DM-TS-B or DM-UTC-B  DM-RD-B	DM-DDB-B DM-DCC-B	DM-DDB-B <sup>1</sup> DM-DOB-B <sup>1</sup>	DM-DDB-B <sup>1</sup> DM-DOB-B <sup>1</sup>

<sup>1</sup>Not required if the device is a BACnet MS/TP Slave.

<sup>2</sup>Not required for devices claiming conformance to a Protocol Revision less than 7.

Table 3—BIBBs required for various standard BACnet device profiles.

At first glance it would appear that things are getting more complicated than less. How could we get 107 BIBBs from only 38 possible services?

One answer is to understand who the requesting device is and who the executing device is.

In the jargon of BIBBs, an “A” device is one who uses the data (client) while the “B” device is the one who provides the data (server). Study Tables 1 and

2 which show two Data Sharing BIBBs. Notice that both BIBBs use the same Read Property but in one operation Client A is initiating the request while in the second operation Server B is executing the operation.

Devices are not required to both initiate and execute services, but some do.

BIBBs are indeed “building blocks” to interoperability. They are used to classify the capability of the device and for creating standard device profiles. The number of required BIBBs increases with the complexity of the device. To fully understand the workings of each BIBB you need to consult Annex K. Table 3 provides a listing of required BIBBs for each device profile.

## BACnet Standard Device Profiles

To ease the process in determining BACnet compliance, the standard classifies devices into eight categories:

- Advanced Operator Workstation (B-AWS)
- Operator Workstation (B-OWS)
- Operator Display (B-OD)
- Building Controller (B-BC)
- Advanced Application Controller (B-AAC)
- Application Specific Controller (B-ASC)
- Smart Actuator (B-SA)
- Smart Sensor (B-SS)

BIBB	DESCRIPTION
DS-RP-B	Executes a ReadProperty Request
DS-WP-B	Executes a WriteProperty Request
DM-DDB-B	Executes a Who-Is Request
DM-DOB-B	Initiates an I-Am Response
DM-DCC-B	Executes a Communications Control Request

**Table 4—B-ASC Device Profile**

In order for a device to be classified as a standard BACnet device, it must comply with a set of defined BIBBs. The listing of required BIBBs for each device classification is called a Device Profile. The simplest device is the Smart Sensor and, as shown in Table 3, it is only required to support the DS-RP-B BIBB. It only needs to execute the request for data since it is a sensor. The BAS Remote is more complex, being classified as a B-ASC requiring the BIBBs listed in Table 4.

The Read Property and Write Property BIBBs are straightforward since they involve data sharing, but in both cases the device only responds to a request to read or write. The Device & Network Management BIBBs are also limited in that the device only responds to requests. It basically functions as a server and not a client. It should be noted that the number of BIBBs listed in Table 3 only represent minimum requirements. A vendor can choose to support more, so the device profile for his device would include more BIBBs. In order to fully understand what a device is capable of doing, you need to study the device's Protocol Implementation Conformance Statement (PICS).

## BACnet Protocol Implementation Conformance Statement

To help the customer in determining the compliance level of a device, a vendor must supply a PICS statement of compliance. The basic format of the statement is provided in the standard and this was the format used for the BAS Remote. For the BAS Remote, its PICS indicates that it complies with the B-ASC standard device profile with no additional BIBBs supported. The actual BIBBs are listed. The data link layer supported is also listed as BACnet/IP Annex J. A vendor is not required to use the identical format, but the relevant information must be provided. Table 5 shows the PICS statement for the BAS Remote. Although the vendor has stated his product's compliance with a PICS statement, how does a user know that the product actually complies? The ideal approach is for the vendor to submit the product to the BTL Laboratories for compliance testing. These labs were formed by BACnet International, a trade association devoted to advancing the BACnet standard around the world.



## BACnet International

According to its website at <http://www.bacnetinternational.org/>, BACnet International (BI) is an organization that encourages the successful use of BACnet in building automation and control systems through interoperability testing, educational programs, and promotional activities.

Membership is open to both companies and individuals interested in the BACnet standard. There are two key activities BI sponsors in its mission to encourage successful installations. BI organizes annual Plugfests where vendors can bring their products to an event in order to verify that their devices interoperate. The other initiative is the creation of the BACnet Testing Laboratories (BTL) where products can bear the BTL mark upon successful completion of conformance testing.

### Plugfests

Plugfests are a convenient way for vendors to test-drive their products before actually incurring the expense and effort of a formal conformance test. Usually run once a year, the Plugfest is open to any BACnet device manufacturer, but participants pay a fee. The trade press is not invited, and results are not tabulated. It is an opportunity for vendors to freely discuss the intricacies of the BACnet protocol and to verify that their products can communicate among compliant equipment as required by the standard. It is expected that there could be some glitches occurring during testing and experts are at hand to clarify the requirements of the standard. This is all being done to improve the interoperability between competing products.

One BI member offers to be the host for the annual event and assumes responsibility for securing space for the event. To make it quick and easy to setup, communication among vendors should be BACnet/IP although other data links can be tested. This requires that routers be used to support the various data links. To avoid conflicting BACnet Device IDs, each vendor is assigned a range of 1000 IDs based upon their ASHRAE assigned vendor ID. Each vendor needs to provide a printed document listing a product's BACnet objects that the vendor intends to test. For each object, the object ID, object name and any optional properties must be listed.

Each vendor has to provide an Ethernet connection along with an Ethernet repeating hub and some sort of protocol analyzer for observing the data being sent and received. Repeating hubs facilitate the use of protocol analyzers since all traffic can be observed on all hub ports. Software changes could be made during breaks if the vendor brings along a full development system. This is the one chance during the year to do some real interoperability testing, so it is best to come prepared.



The Plugfest occupies two to three days of organized testing. On the first day are speed sessions requiring ten minutes of setup and fifty minutes of testing. Vendors are assigned to teams with one team meeting with another team for the full session. One of the teams must have equipment that can read another vendor's device. For example, not much would occur if the two teams only had sensor devices. Little testing would occur. For any one session there could be 24 tables each occupied by two teams. Upon completion of the session, another session immediately starts with different teams occupying the same tables. These sessions go on all day.

There are mini-roundtables consisting of four or more participants communicating to one host participant. There are only four of these sessions lasting for two hours each. Also there are panel discussions from the experts on how to design for interoperability. With easy access to the experts, this is the time to ask questions and to take advantage of this excellent training opportunity.

When the Plugfest ends, the vendor can now return with added confidence of a successful submission to the BTL Labs.

### **BTL Mark**

The BACnet Testing Laboratories was created by BI to support compliance testing and interoperability testing. BTL published Implementation Guidelines which provides excellent information on achieving compliance. This document can be downloaded from the BI website. Another source of information on compliance can be gained by joining the BACnet mail list by visiting:

<http://www.bacnetinternational.org/>

Compliance testing and listing is overseen by the BACnet Working Group. A vendor can receive a set of testing procedures from the labs to do some pre-testing before actual submission to the labs. Although you do not need to be a BI member to submit a product for testing, BI members receive preference and a discount on testing. Once the product has been successfully tested by the labs, it can be listed and bear the BTL Mark as shown. This mark gives the user assurance that the product complies with the BACnet device profile to which it was tested. An added benefit to the vendor is that some bid specifications require only BTL listed products.

## Summary

Although object modeling appears complex, it is the modern method of making devices “network visible.” Through the use of objects, properties and services, the BACnet standard defines BACnet compliance through defined BIBBs that lead to standard device profiles. Using a PICs statement, a user can determine the capability of a device and its compliance level. Users can be assured that a product is BACnet compliant if it completed conformance testing. Vendors also use compliance before submitting the product to the BACnet Testing Laboratories.

## References

**ANSI/ASHRAE Standard 135-2012 BACnet—A Data Communication Protocol for Building Automation and Control Networks**, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.

**Direct Digital Control of Building Systems Theory and Practice**, H. Michael Newman, John Wiley and Sons, Inc. 1994.

<http://www.polarsoft.biz/langbac.html>, **The Language of BACnet— Objects, Properties and Services**, Bill Swan, Allerton Technologies, Inc.

<http://www.ccontrols.com/pdf/TD0403000D.pdf>, **Building Automation System Remote**, Contemporary Controls.

<http://www.bacnetassociation.org>, BACnet International.

**BACnet Explained**, H. Michael Newman, ASHRAE Journal, November 2013.

Introduction

An interesting attribute of the Building Automation and Control Network (BACnet) is that Ethernet is supported along with several data links, and communication between the various data links is possible through the use of routers. Each data link is treated as a separate network and the collection of data links is considered a single BACnet internetwork. BACnet routers facilitate the connections — but when BACnet/IP is used as one of the data links, adjustments must be made to how BACnet routers are used in an IP network.

BACnet’s four-layer Communication Model

Within the first few pages of the 1000-page BACnet standard (ANSI/ASHRAE Standard 135-2012), the BACnet communications model is introduced as it compares to the Open Systems Interconnection — Basic Reference Model with its seven-layers. BACnet uses a collapsed architecture with only four layers. One of those four layers is the network layer (layer 3) and there is where routers live. Although many

automation communication architectures use the more simplified three-layer model, the BACnet community incorporated a network layer from the very beginning — even before the popularity of the Internet Protocol (IP).

In Figure 1 you will find the BACnet communications model. Notice that BACnet supports several data links including Ethernet – IEEE 802.3. The standard allows for the mixing of data links — which consequently requires the use of BACnet routers. Above the data links is the network layer, but this network layer is not the Internet Protocol. IP networks are supported through a BACnet Virtual Link Layer (BVLL). BACnet devices attach to segments and segments are interconnected through the use of repeaters. Repeaters operate at the physical layer since they manipulate the symbols sent over the medium in the form of “1s and 0s”. A good example of a repeater would be an EIA-485 repeater that extends two MS/TP segments. The data link layer is responsible for organizing the transmission of data in the form of frames from one station to another. Stations require addresses so that access to a shared medium is controlled. This is called **Media Access Control (MAC)** with each station being assigned a unique MAC address. A collection of MAC-level devices — each with a unique MAC address — comprise a local area network (LAN) or simply a network. Devices that connect the same data link technologies together and provide some MAC filtering are called bridges. Bridges maintain the integrity of the single MAC domain requirement and do not duplicate MAC addresses. A good example of a bridge is an Ethernet switch.

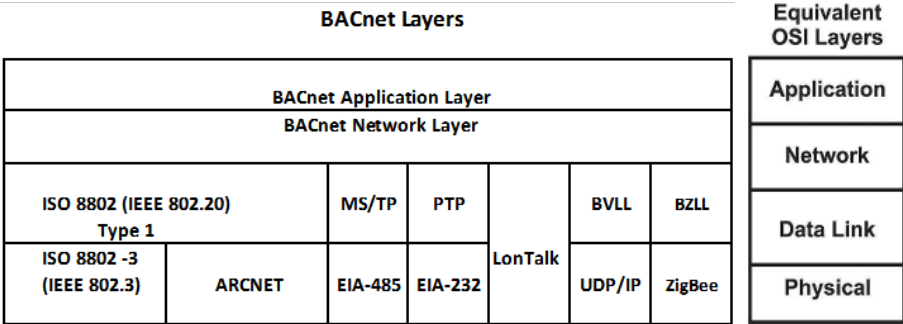


Figure 1—BACnet Supports several data links including Ethernet



Figure 2—An NPDU is comprised of a header and APDU payload.

Assume you have two isolated Ethernet segments that you want to connect together. You can use an Ethernet hub which functions as a repeater or you can use an Ethernet switch that functions like a bridge. In both instances the integrity of the single MAC domain is maintained. Now suppose we want to connect an Ethernet segment to an ARCNET segment. The MAC domains are different and incompatible so both MAC domains must be treated as separate networks. BACnet defines an *internetwork* as a collection of two or more networks interconnected by *routers*.

### Device Instance and Network Addressing

One requirement of BACnet is that a BACnet device address, called a *device instance*, must be unique within a BACnet internetwork. This means that regardless of the network location of the BACnet device, its address must be unique. With a 22-bit address field allowing over 4 million possible addresses, this requirement should not be too hard to achieve. The BACnet application layer does not understand the concept of networks and therefore it is not necessary to assign a network number to a BACnet device. Only BACnet routers and the BACnet network layer need to know the 16-bit BACnet network numbers. With 65,534 possible network addresses plus one for broadcasts, there are plenty of possibilities. There is no address assignment for a BACnet internetwork. Although BACnet devices are uniquely defined, you still need to find them if they are located on a particular network. BACnet routers and devices generally work together to discover the location and MAC address of target devices through a procedure known as *dynamic device binding*.

### The BACnet Network Layer

According to the BACnet standard, 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. Devices that interconnect two networks together are called routers and their activities are called routing. When we are talking about routers, we are talking about layer-3 (network) functionality. Messages that are sent at the data link and physical layer are usually called frames — although BACnet uses the term *MAC Layer Protocol Data Unit (MPDU)*. At the network layer, BACnet sends out *Network Protocol Data Units (NPDUs)* which are shown in Figure 2 at left.

An NPDU is comprised of a header and APDU payload. Encapsulated into the NPDU is an Application Protocol Data Unit (APDU) which represents the BACnet payload. It consists of a header and service-related data. The NPDU has a header as well — and it is called the Network Protocol Control Information (NPCI). The length of the NPDU depends upon the type of message and its destination. If the message is being sent from one BACnet device to another on a local bus involving no routers, the NPCI collapses to only two bytes. However, if a message is being sent to a remote device (a device on another network), the NPCI expands since the network layer must specify the destination network in the NPCI. When a router relays this same remote message to its attached destination data link, the source network is substituted for the destination network in the NPCI. The longest messages involve router-to-router communication where both source and destination network addresses are specified in the NPCI.

Usually one of the requirements of the network layer is to be able to route messages from one network to another over different paths. However, BACnet is explicit about this requirement; it does not allow it. There can be only one path between devices on the BACnet internetwork. Another requirement is message segmentation and reassembly. The maximum message lengths for the various data links differ — so it is possible a message will need to be broken into two or more packets, sent sequentially then reassembled at the other side. BACnet skirts this problem at the network layer by insisting that no segmentation occur at this level. It must be done at the application layer.

### Segmentation

Each data link has a **Maximum Transmission Unit (MTU)** specification that cannot be exceeded. A message larger than the MTU figure requires the message be broken into segments — each smaller or equal to the MTU of the receiving device. The receiving device is then obligated to put the segments back together in correct order for the message to be understood. This is called segmentation although the term fragmentation is often used. Unlike the IP protocol, the order of the segments is guaranteed since BACnet allows only one path between any two BACnet devices. The BACnet specification stipulates an MTU value of 501 bytes for ARCNET, MS/TP and Point-to-Point (see Figure 3). Ethernet has the largest MTU with 1497 bytes, while LonTalk has the smallest with 228 bytes. If an Ethernet device communicates with another Ethernet device, there should be no problem since each device is capable of receiving and transmitting the same size message. This assumes that the receiving device has allowed sufficient buffer space to receive the largest packets. However, there could be problems with a transfer from Ethernet to another data link — resulting in an error message. The source device could try again by segmenting the message which assumes the destination device supports segmentation. Note that with BACnet, segmentation occurs at the application layer and not the network layer.

Data Link Technology	Maximum NPDU Length (bytes)
IEEE 802.3 Ethernet	1497
ARCNET	501
MS/TP	501
Point-to-Point	501
LonTalk	228
BACnet/IP	1497

**Figure 3—Maximum Transmission Units vary with the data link.**

## Router Operation

Routers do not need to be stand-alone devices. Router functionality can be built into building controllers or other control equipment — and it frequently is. BACnet routers automatically build and maintain their routing tables based on network-layer communications with other routers. By definition, a router connects networks — with each connection called a port. Each port has a MAC address that is maintained in the router's routing table. Also included in the routing table are the network numbers of the two attached networks — as well as any remote network number that can be reached via each port. When a router first comes up, it broadcasts an *I-Am-Router-To-Network* message out each port identifying what network numbers are reachable from its other port(s). This allows other routers to update their routing tables. When a router receives a message it first checks the NPCI to see if a destination network number is specified. It then looks up the MAC address of the router that provides the path to this network number. If this search is successful, the message is forwarded. *Who-Is Router-To-Network* message is sent by the router to learn the route. If that is not successful, a *Reject-Message-To-Network* message is returned to the originator. Routed messages travel

from router to router as directed messages until the routed message reaches its final destination network. If the originating message was a broadcast, the router will substitute the appropriate broadcast command for the attached data link and send it accordingly.

## The Impact of BACnet/IP upon BACnet Routers

As the popularity of TCP/IP exploded, the BACnet community needed a strategy for using the BACnet protocol in an IP world without a major re-write of the standard. The result was **BACnet®/IP (B/IP)** and it is described in Annex J of the 135-2012 standard. The body of the BACnet standard makes exclusive use of MAC addresses for all data links. (This holds true for Ethernet as well.) But in the BACnet/IP world, IP addresses are needed. For BACnet/IP, Annex J defines an equivalent MAC address comprising of a four-byte IP address followed by a two-byte UDP port number. The BACnet community registered a range of 16 UDP port numbers as hexadecimal BAC0 through BACF. This port number must accompany both directed and broadcast messages sent by BACnet/IP devices. If, for some reason, two independent BACnet/IP networks are to share the same IP subnet, it is possible to change the UDP port number to another of the reserved ports in the range.

BACnet/IP devices that share the same UDP port number are considered part of the same BACnet network — even if it is sub-netted. A second BACnet/IP network with a different UDP port number is deemed another BACnet network, just as an Ethernet 802.3 or ARCNET or MS/TP segment would be. It should be noted that

BACnet does not use TCP transmissions — instead relying upon connectionless UDP messages.

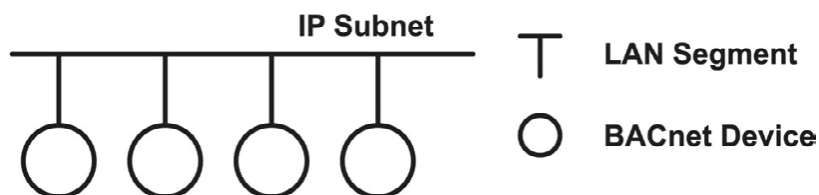


Figure 4—BACnet /IP traffic is unrestricted with one subnet.



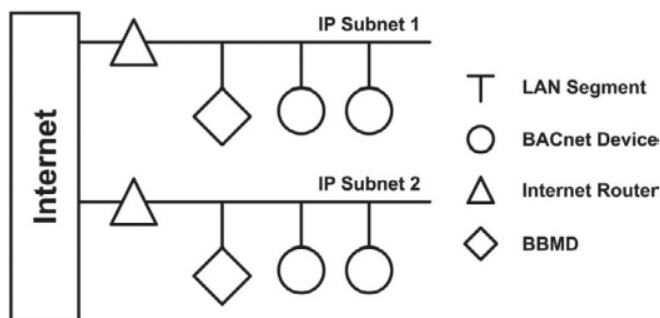


Figure 5—Ip routers usually block broadcast messages.

BACnet/IP incorporates the same four-layer ISO model as shown in Figure 1. Routers operate at the network layer. Since routers were already defined in “The Network Layer” (clause 6 of the standard), Annex J makes reference to this clause. Annex J introduces the concept of the **BACnet Virtual Link Layer (BVLL)** which provides an interface between clause 6 equipment and another communications subsystem. BVLL messages can be either directed or broadcast messages. A directed message is exchanged between two IP addresses and no others. A broadcast message originates from one IP address and is sent to all other IP addresses on the subnet.

In the general case, a BACnet network consists of one or more IP subnets containing BACnet/IP devices using the same UDP port number. It is possible the BACnet network has only a single subnet. This is the simplest case since both directed and broadcast messages are sent along the subnet with no restrictions. Refer to Figure 4.

However, if more than one subnet comprises the BACnet/IP network, there is an issue. Referring to Figure 5, an IP router is required to connect the IP subnet to the Internet or to an internetwork. IP routers perform differently than BACnet routers. Broadcast messages could be (and usually are) blocked by these IP routers if the IP routers do not support what is called a **directed broadcast**. For a directed broadcast to pass through the IP router, the IP router must support a feature called **bridging** in which the IP router treats these directed broadcasts more like a switch than a router. If this type of IP router cannot be found, then a way of managing broadcasts in a sub-netted BACnet/IP network must be devised.

### Directed and Broadcast Messages

The BACnet standard describes a **directed message** is a message sent from one device to another device within the BACnet internetwork. When it comes to **broadcast messages**, the standard defines three types. A **local broadcast** is a message sent from one device to all other devices on the same network. A **remote broadcast** is a message sent from one device on one network to all devices on another network. A **global broadcast** is a message sent from one device on one network to all devices on the internetwork. A BACnet/IP network with a shared UDP port number must function just like any other BACnet network in regard to directed and broadcast messages. If all devices are located within the same subnet, there is no problem sending or receiving either directed or broadcast messages since all parties to these messages face no communication restrictions. Infrastructure devices such as hubs, repeaters, and switches do not interfere with the sending and receiving of directed or broadcast messages. But if a local broadcast message is sent to all subnets within the BACnet/IP network, there could be trouble if the IP routers do not support bridging.

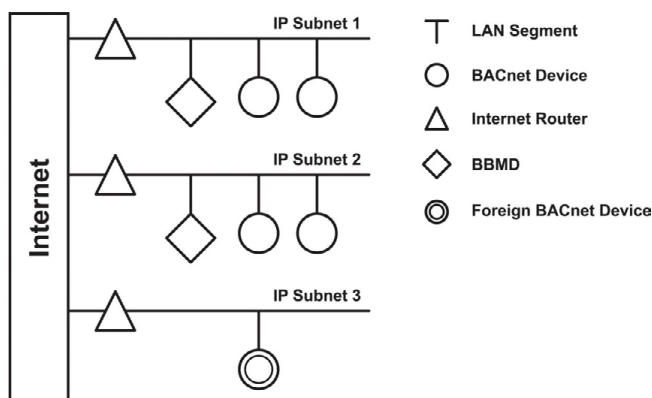
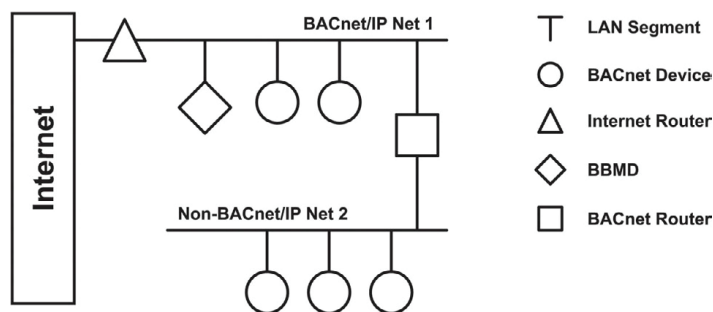


Figure 6—A foreign device can reside on a subnet without a BBMD.



**Figure 7—The addition of a non-BACnet/IP data link results in two networks.**

### BACnet/IP Broadcast Management Device (BBMD)

With a BACnet/IP network comprised of two or more IP subnets, a local broadcast may not be able to span multiple subnets — if so, a device called a BBMD must be used. A BBMD located on an IP subnet monitors the origination of a broadcast message on that subnet and, in turn, constructs a different broadcast message *disguised* as a **directed message** in order to pass through an IP router. This disguised message is directed to other BBMDs (located on the various subnets) that receive the directed message and retransmit the broadcast on their attached subnets. Since the BBMD messages are directed messages, *individual messages*

must be sent to *each* BBMD. Each BBMD device maintains a **Broadcast Distribution Table (BDT)**, the content of which is usually the same for all BBMDs within the network. BBMDs must know the IP address of all other BBMDs in the network. Note in Figure 5 that there is one BBMD on each subnet.

It is possible to communicate to a device on a subnet that does not have a BBMD as in Figure 6. This type of device is called a foreign device since it resides on a different IP subnet from devices attempting to communicate with it. Usually in BACnet lingo, a foreign device is on a different network — but with BACnet/IP; a **foreign device** is on a different subnet. If the foreign device registers with the BBMDs, it can be party to communication with all other devices on the network. The BBMD must then maintain a **Foreign Device Table (FDT)**.

The final example is shown in Figure 7. In this case a non-BACnet/IP data link is attached via a BACnet router — creating two networks but one BACnet internetwork. This example demonstrates the flexibility of BACnet in that legacy data links or lower-cost data links can be supported along with more modern IP networks. There is nothing to preclude the embedding of a BBMD within the BACnet router — thereby eliminating one device on the subnet.

### Summary

By providing network layer functionality into the BACnet stack, support for various data links was achieved through the use of BACnet routers. Annex J of the BACnet standard identifies the methods for attaching BACnet/IP networks to non-BACnet/IP networks while achieving interoperability. Although BACnet/IP provides some challenges, these challenges can be met by fully understanding the network architecture.

### Reference

**ANSI/ASHRAE 135-2012 BACnet — A Data Communication Protocol for Building Automation and Control Networks**, 2012, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.

### Acknowledgment

The author would like to thank David Fisher of PolarSoft for his assistance on this section.

# Using Managed Switches

CTRLink® EISX\_M Series

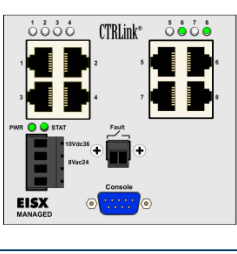
Home

[System Configuration](#) [SNMP Configuration](#) [Performance Monitor](#) [Save Settings](#) [CTRLink Webpage](#) [Logout](#)

CTRLink Compact Managed Switching Hub

EISX8M Information:

Name	Managed Switch V4.16
Location	
Contact	
MAC Address	00-50-DB-00-13-3D
Firmware Version	4.16
Uptime	3188366 seconds
Switch Temperature	43°C



Note: The port and relay status and settings can be accessed by clicking on their images above.

## Introduction

A managed switch is defined as one that supports the Simple Network Management Protocol (SNMP). Sophisticated Ethernet controller technology with numerous features exists in managed switch products. Unlike an unmanaged switch, a managed switch must be configured in order to make use of its capabilities. This is usually accomplished via a serial port or via a web browser. Managed switch features vary among vendors. The following features are common in most managed switch but they are explained using our company's EISX series of managed switches.

## Port Configuration

By default, all copper ports will auto-negotiate speed, duplex and flow control. However, port settings can be pre-set to suit specific needs. SNMP Management Information Base (MIB) data can be displayed for each switch port in order to gain a complete understanding of the performance of each port.

## IP Address Assignment

A default private IP Address, Subnet Mask and Default Gateway Address are factory installed but they can be changed by the user. Instead of a fixed IP address, a DHCP client in the unit will request dynamic settings from a DHCP server. A method exists for resetting the unit to factory default settings.

## Trunking

In order to improve uplink throughput, ports can be aggregated in one of two groups so as to function as one higher performing port. Up to four copper ports can be assigned to each trunk group. Cable redundancy with extremely fast recover times is inherent in trunk groups.

CTRLink® EISX\_M Series

Home

[System Configuration](#) [SNMP Configuration](#) [Performance Monitor](#) [Save Settings](#) [CTRLink Webpage](#) [Logout](#)

CTRLink Compact Managed Switching Hub Port 1

Refresh Help

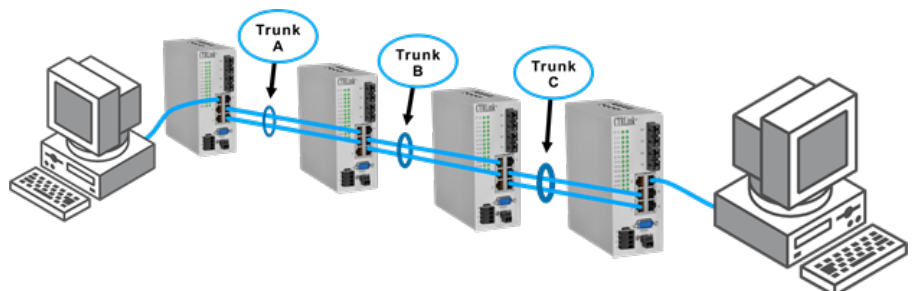
Current State:

Port State	<input checked="" type="radio"/> Enable <input type="radio"/> Disable
Media Type	Copper
Mode	Auto Negotiate
Auto-MDIX	<input checked="" type="radio"/> Enable <input type="radio"/> Disable
Flow Control	<input type="radio"/> Enable <input checked="" type="radio"/> Disable

Apply

Port Packet Statistics:

Unicast Packets Received	53146
Unicast Packets Sent	29810
Multicast Packets Received	0
Multicast Packets Sent	130
Broadcast Packets Received	0
Broadcast Packets Sent	9829
Dropped Packets	0
Oversize Packets	0
Undersize Packets	0
Fragments	0
Jabbers	0
Collisions	0
Deferred Transmissions	0



Configure Port Mirroring	
Status	<input type="radio"/> Enable <input checked="" type="radio"/> Disable
Mirror Port	Port: <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 <input type="radio"/> 7 <input type="radio"/> 8
<b>Ingress (Ingoing) Mirror Rules:</b>	
Source Ports	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/> 6 <input type="checkbox"/> 7 <input type="checkbox"/> 8
Divider	1 (1-1023)
MAC Address Filter	<input checked="" type="radio"/> Capture ALL <input type="radio"/> Capture by Source <input type="radio"/> Capture by Destination
MAC Address	
<b>Egress (Outgoing) Mirror Rules:</b>	
Source Ports	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/> 6 <input type="checkbox"/> 7 <input type="checkbox"/> 8
Divider	1 (1-1023)
MAC Address Filter	<input checked="" type="radio"/> Capture ALL <input type="radio"/> Capture by Source <input type="radio"/> Capture by Destination
MAC Address	
Apply	

## Port Mirroring

Ethernet switches improve throughput by restricting directed traffic only to those ports party to the intended traffic. Although performance is improved, network troubleshooting is more difficult because a packet sniffer attached to another port may not be able to monitor all traffic. The solution is to create a mirror port to the ports party to the traffic being monitored. A mirror port can monitor any of the other ports with filtering based on source or destination addresses or even a particular MAC addresses.

## Virtual Local Area Network (VLAN)

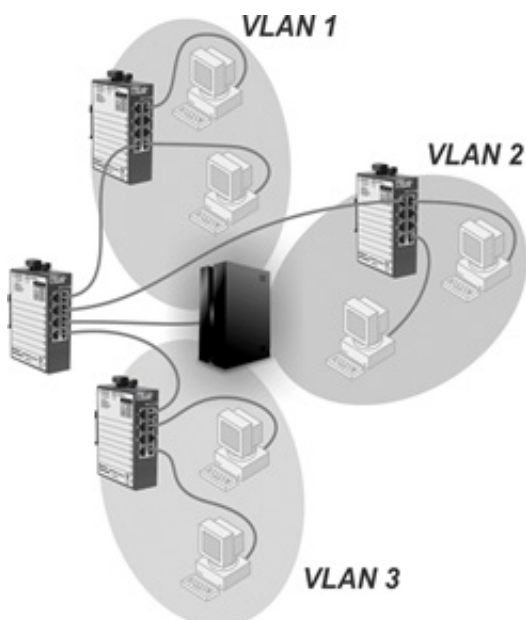
VLANs allow the same Ethernet infrastructure to accommodate concurrent but separate networks dedicated to different functions — such as accounting, security and automation. Each VLAN supports IEEE 802.1Q tagging where each VLAN is assigned a unique VLAN tag (VID). For each VID, ports on the switch become members of the group or they are marked as non-members. Switch ports can be instructed to append a VLAN tag to an ingress (inbound) Ethernet frame or drop VLAN tags on egress (outbound) frames — providing the greatest flexibility in establishing VLANs. Overlapping VLANs can be created if strict isolation is not wanted.

## Port Forwarding and Filtering Database

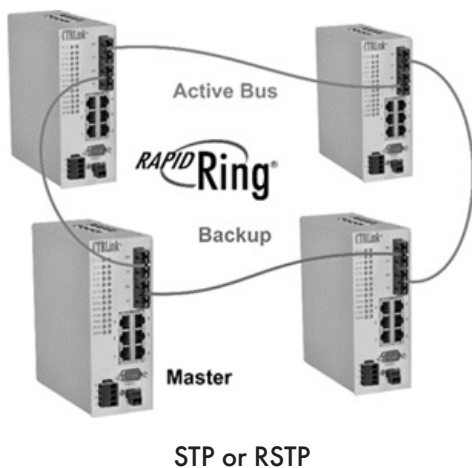
Ethernet switches learn the port upon which an Ethernet station can be reached and this information is entered into its filtering database. Subsequent traffic to Ethernet stations recorded in the database is then restricted to these known ports. While this activity is automatically accomplished as a background task, the filtering database can be modified to meet specific needs. The Aging of the filtering database entries is configurable. Static entries based upon MAC addresses can be entered into the database. The same applies to multicast addresses. Four levels of priority can be set based upon MAC addresses.

## Quality of Service (QoS)

By enabling Quality of Service, Ethernet frames can be given varying degrees of priorities when messages are being queued. There are several QoS methods, which can be enabled. QoS can be established on strictly a port basis where some ports are given priority over others. IEEE 802.1p priority levels can be honored or ignored on a port basis. Although there are eight 802.1p priority levels, these levels are mapped to four levels used by the switch. Support also exists for Type of Service (TOS) and Differentiated Services (DiffServ). Although both TOS and DiffServ priorities have been pre-mapped into four levels, these assignments can be modified.



Example of Overlapping VLANs



### Cable Redundancy

Besides trunking, three other forms of cable redundancy are possible — Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP) and Contemporary Controls' proprietary RapidRing®. For mesh networks, either STP or RSTP (recommended) is available and their parameters can be configured accordingly. For ring topologies, RapidRing is the best option yielding the fastest recovery time — typically less than 300 ms with 100 switches.

### Rate Limiting

Data throughput can be throttled on a port basis for both ingress and egress ports in order to reduce the number of dropped frames on highly loaded networks. Traffic restrictions can be applied individually to Broadcast, Multicast or Unicast messages or to all types of messages.

### Port Security

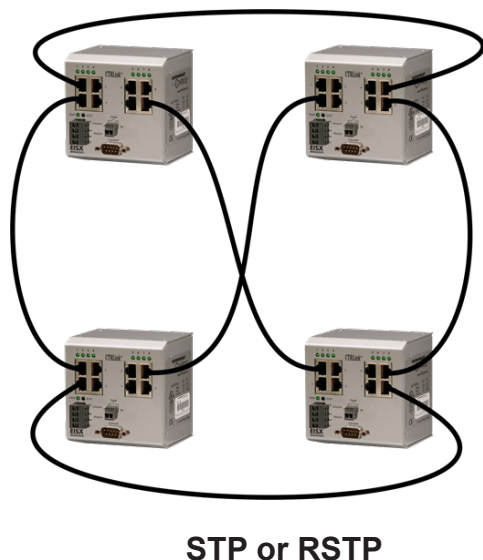
Increased security settings can be enabled on a port basis. Specific MAC addresses can be assigned to particular ingress or egress ports.

### Internet Group Management Protocol (IGMP) Snooping

Both IGMP snooping and IGMP querier are supported in order to reduce multicast traffic to devices which have no interest in this traffic. An IGMP forwarding map can be created on a port basis. The Multicast Filtering Database Aging time is configurable as is the Query Interval time. As a managed switch, the switch supports SNMP and can be configured for System Name, Location and Contact. Private and Public Community String access can be configured for read-only or read/write access. Up to four IP Trap Receivers can be identified. MIB data is available for each port.

### Performance Monitor

A performance monitor exists to assist in troubleshooting. The filtering database can be browsed for entries. When enabling the Spanning Tree Protocol, the forwarding or discarding states of each port can be monitored. Finally, a trap log exists for any SNMP traps that have occurred.



### Introduction—Using IP Routers

What follows are nine application examples for using IP routers to demonstrate how the technologies described in this book can be used. For these examples we had to use a commercial product in order to show the webpage setups. We chose our EIPR-E IP Router because that is the product we understand but most IP routers can be used for these examples. First we will explain how an IP router works.

The EIPR links two 10/100 Mbps Internet Protocol (IPv4) networks — passing appropriate traffic while blocking all other traffic. One network is the local area-network (LAN); the other is the wide-area-network (WAN). The built-in stateful firewall passes communication initiated on the LAN-side while blocking WAN-side initiated communication. With Port Address Translation (PAT), LAN-side clients can access the Internet. Network Address Translation (NAT) allows a one-to-one translation between LAN-side and WAN-side devices. With Port Forwarding, LAN side devices can be accessed from the Internet. The EIPR incorporates a four-port Ethernet switch for multiple LAN-side connections. An external Ethernet-based modem — cable or DSL — can be used to connect to the Internet. DSL modems connect via the PPPoE protocol. Configuration is via a web browser.

### Stateful Firewall — Promotes Secure Communication

A firewall controls the passing of messages from one side of router to the other. A stateful firewall makes decisions based upon the structure of the message and who is initiating and who is responding. The lower part of the router connects the LAN side. The upper part connects the WAN side. A firewall (which can be disabled by the user) separates the two parts.

Originating requests from the LAN side and corresponding responses from the WAN side pass through the firewall. But traffic originating from the WAN side is blocked from the LAN side unless the firewall is adjusted to allow it. This protects the LAN side from unauthorized WAN access.

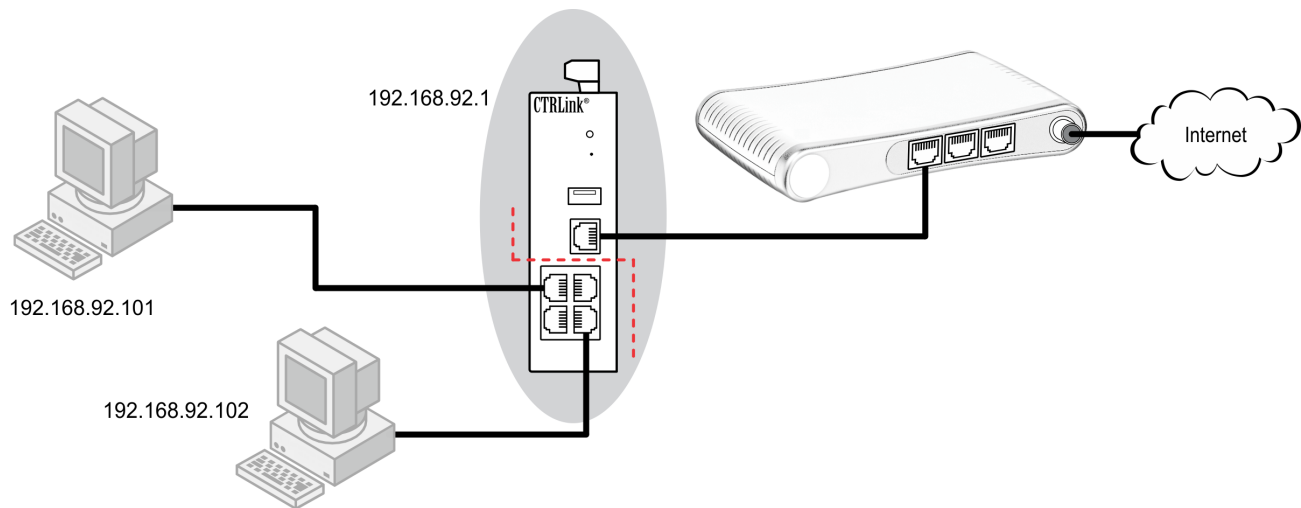
Refer to the following nine application examples for more IP router usages.



### Application #1 — A Cable Modem Connection to the Internet

In the WAN Setup, the default Connection Type is DHCP — where a DHCP server on the WAN side will automatically assign an IP address, subnet mask, default gateway address and one or more DNS addresses to the WAN side of the IP router. Some cable modems have DHCP server functionality.

If a DHCP server is unavailable on the WAN network, you must make static IP entries for the WAN side of the router. Enter the IP address, subnet mask, default gateway address and one or more DNS addresses when using the Static IP option.



**WAN Setup**

Connection Type: **PPPoE**

Username:

Password:

Service Name:

☐ Connect on Demand: Max Idle Time 5 Min

☒ Keep Alive: Redial Period 30 Sec

Optional Settings (required by some ISPs)

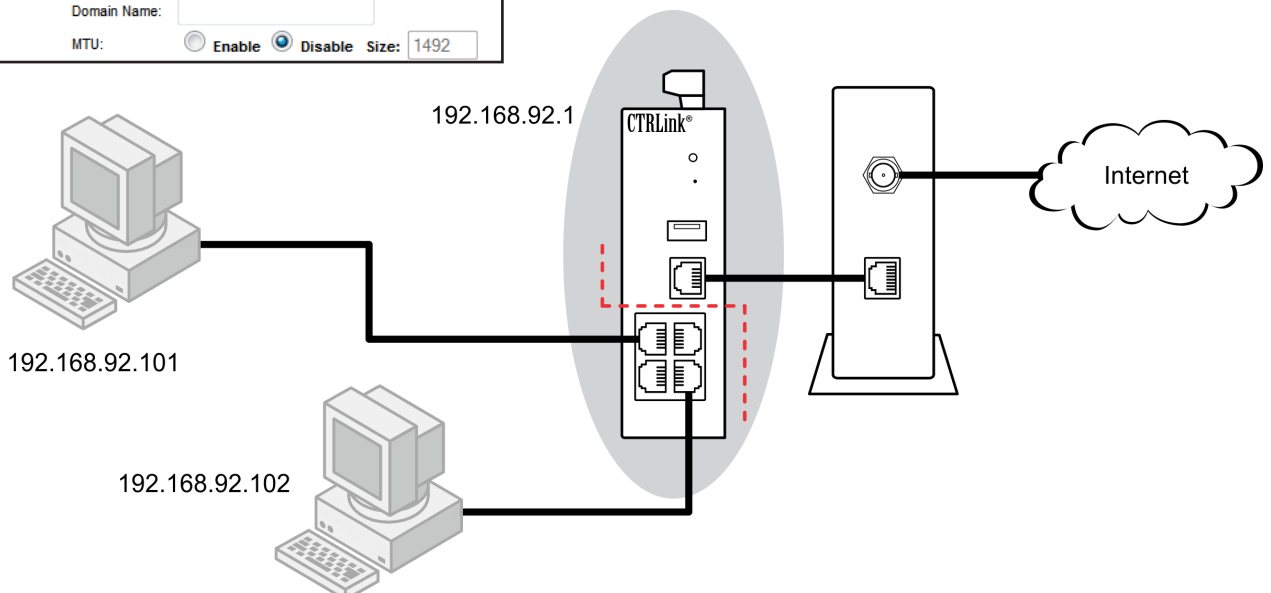
Host Name:

Domain Name:

MTU: ☐ Enable ☒ Disable Size: 1492

### Application #2 — A DSL Modem Connection to the Internet

With DSL modems, the PPPoE protocol must be selected — and a username and password provided. Once a connection is established, the ISP furnishes all the needed WAN IP address assignments.

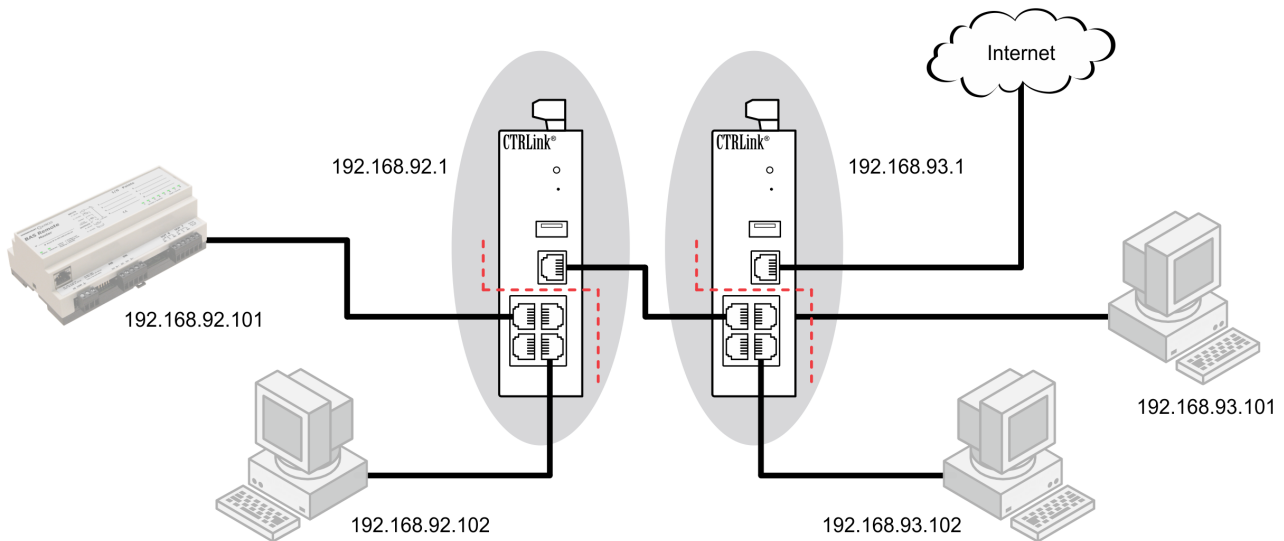




### Application #3 — Cascaded Routers for Additional Isolation

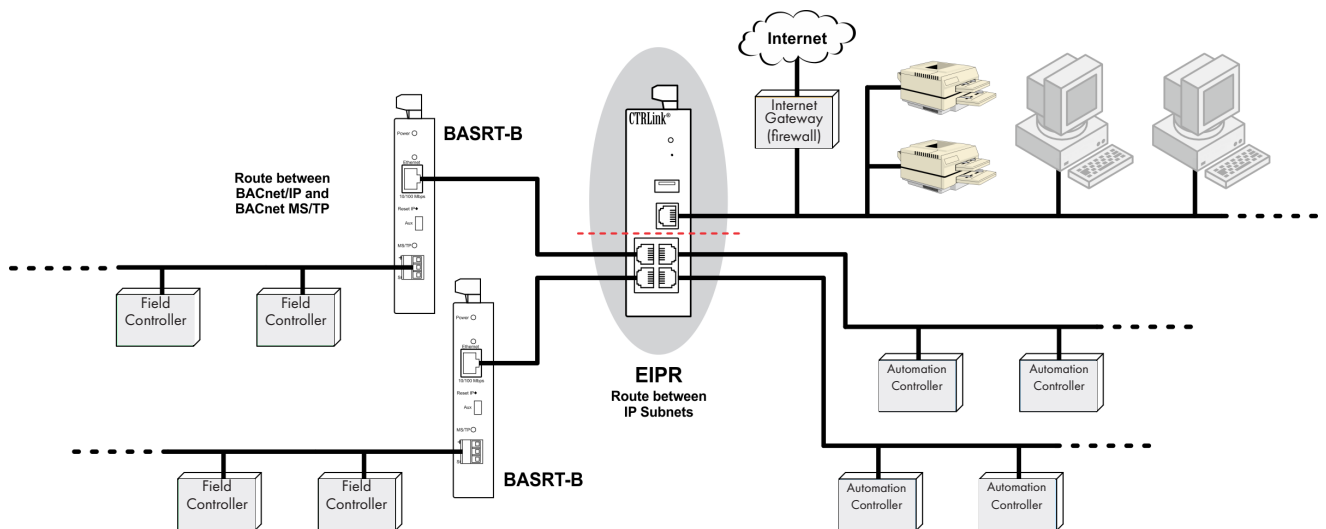
For increased security and isolation, IP routers can be cascaded. Make sure that each LAN-side subnet address is unique when cascading IP routers. The left-most IP router can have its WAN-side IP address assigned using DHCP client or by using static IP address assignment.

The illustration shows a pair of EIPR routers, but the right-most router could also be some other type of router — perhaps one already existing in the business system — because the EIPR supports standard Internet protocols.



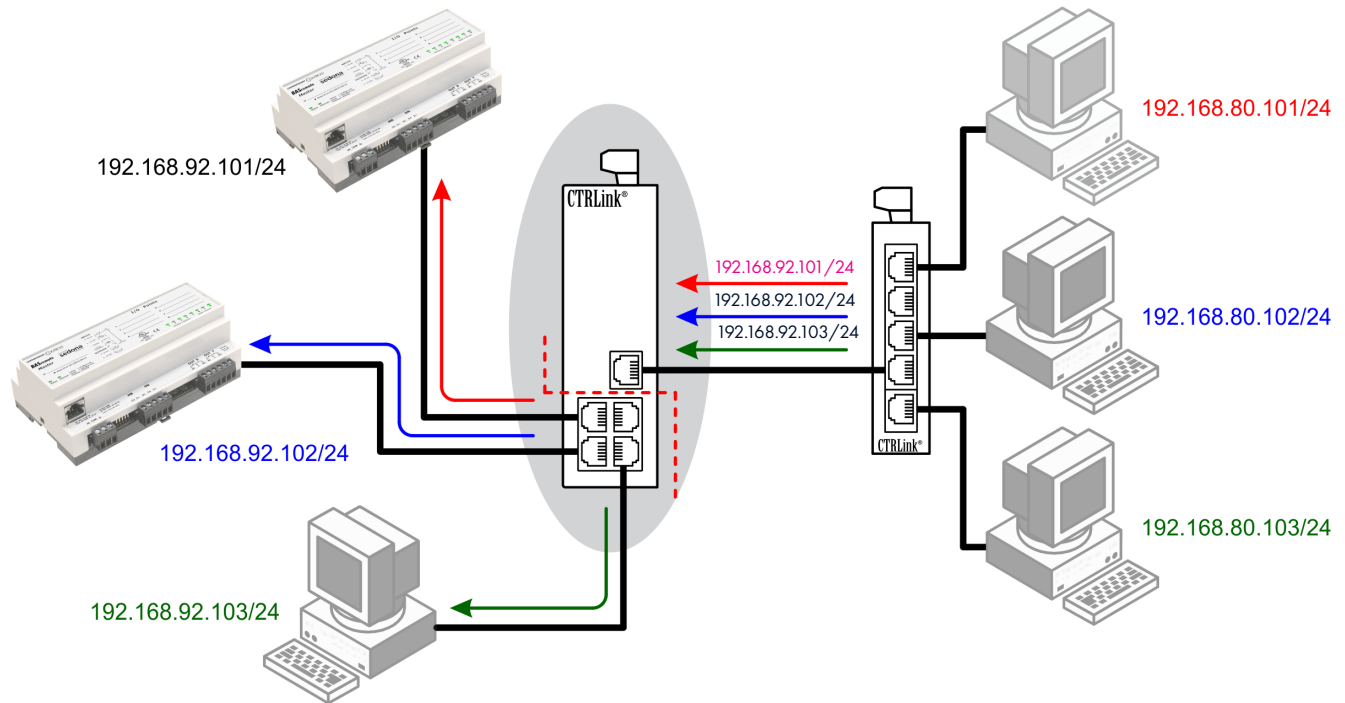
#### Application #4 — Limiting BACnet Traffic

When attaching BACnet devices to IP networks it is possible that the IP network has been sub-netted through the use of IP routers. Most IP routers will not pass broadcast messages which are crucial to BACnet's operation. The solution is to incorporate BACnet/IP Broadcast Management Device (BBMD) functionality within the BACnet internetwork. The BBMD concept requires that a broadcast message originating on one subnet be encapsulated into a directed message and sent to all remote subnets since these directed messages will pass through IP routers. Once the encapsulated messages are received on the remote subnets, a BBMD device will decode the message and resend it on its local subnet as a broadcast message. Therefore it would appear that a BBMD device must be present on each subnet in order to provide this encoding and decoding function. However, this is not the case if all the BACnet/IP devices support Foreign Device Registration (FDR). At a minimum, one BBMD device is required to be located on one of the subnets with FDR devices registering to this one BBMD. This is what is shown in the example with a BAS Router providing BBMD functionality while allowing for foreign devices registration. Notice that connecting to a BACnet MS/TP network is an option.



### Application #5 — Disable the Firewall for Unrestricted Routing

Under the Advanced Tab, you may choose to disable the firewall. Typically the firewall is disabled when the LANs on both sides of the router are within one organization. That is, there is no public side — both sides are essentially private, so no firewall is needed.



### Application #6 — Port Forwarding to Access a Private Web Server

The firewall will normally block all WAN-side requests. Port forwarding allows computers on the WAN side to access devices on the LAN side by opening up selected WAN IP ports. The only WAN-side requests that will be forwarded through the IP router are those that specify both the router's WAN address and a destination IP port number that exists in the router's IP port forwarding table. When this match is made, the message is forwarded to the indicated IP address on the LAN side.

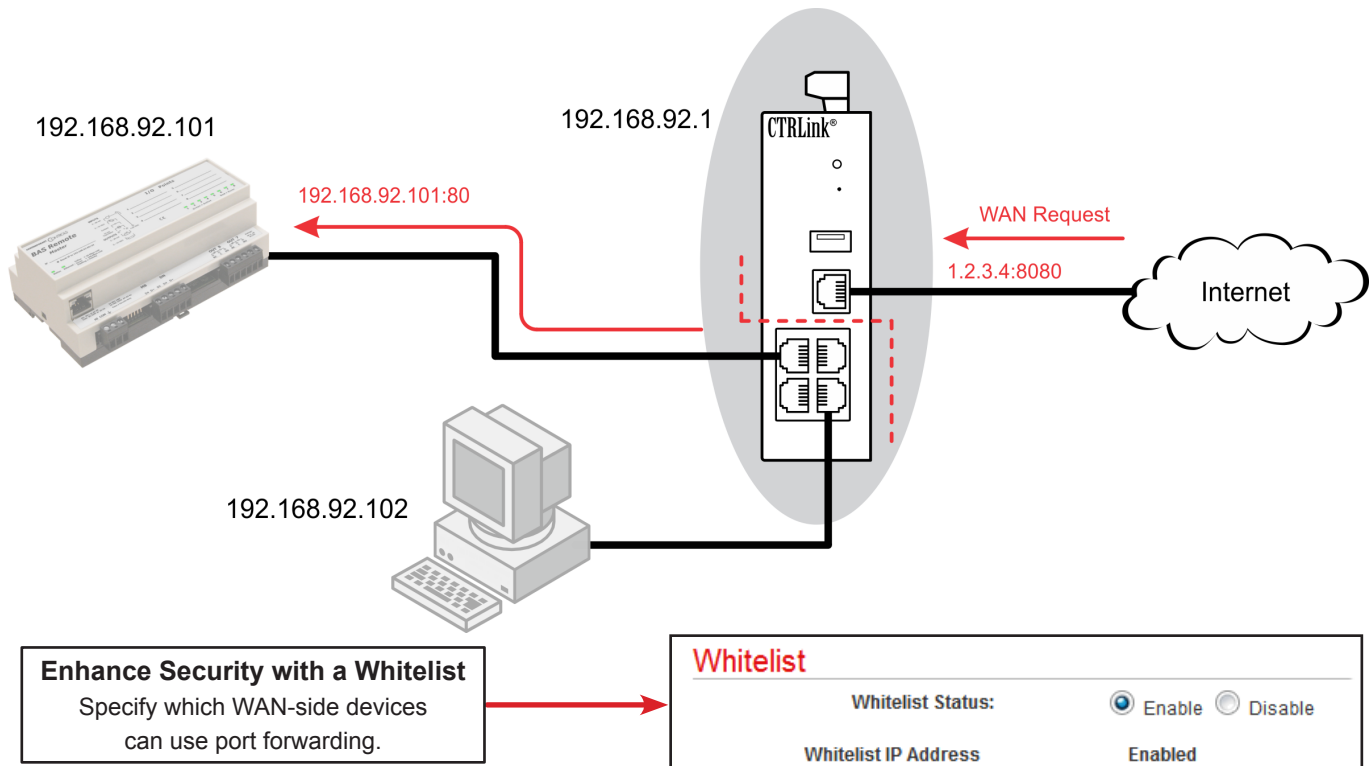
This is very useful when only one public IP address is available, but there is a need to access multiple LAN-side devices. In this example, we want to access a private web server at 192.168.92.101 which is normally invisible from the Internet. Using port forwarding, we allow a WAN-side request made to the router's public (WAN) address. For additional security, the port numbers have been translated.

You can also select Port Range Forwarding to allow an entire range of addresses through the firewall. Note that any WAN-side device can use port forwarding — but you can greatly enhance security by creating a whitelist of allowed WAN-side devices. This is illustrated at the bottom of the page.

Internal IP Address	LAN IP Port	WAN IP Port	External IP Address
192.168.92.101/24	80	8080	1.2.3.4

**Port Forwarding**

WAN IP Port	TCP/UDP	TO	LAN IP Address	LAN IP Port	Enabled	NAT Loopback
8080	Both	TO	192.168.92.101	80	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Both	TO			<input type="checkbox"/>	<input type="checkbox"/>



### Application #7 — Router Access from a WAN-side Device

In some situations you may want a WAN-side device to access and possibly configure the router. This is enabled via the Remote Router Access control (shown below) found under the Administration tab.

Caution: Enabling this control grants access to any device on the public or WAN-side. To restrict access to just certain WAN devices, you must construct a whitelist such as the example below which specifies an outside (public or WAN-side) device that has the IP address of 4.3.2.1.

**Remote Router Access** Administration Port:

Enable: ☒

**Enhance Security with a Whitelist**  
Specify which WAN-side devices  
can configure the router.

**Whitelist**

Whitelist Status: ☒ Enable ☐ Disable

Whitelist IP Address

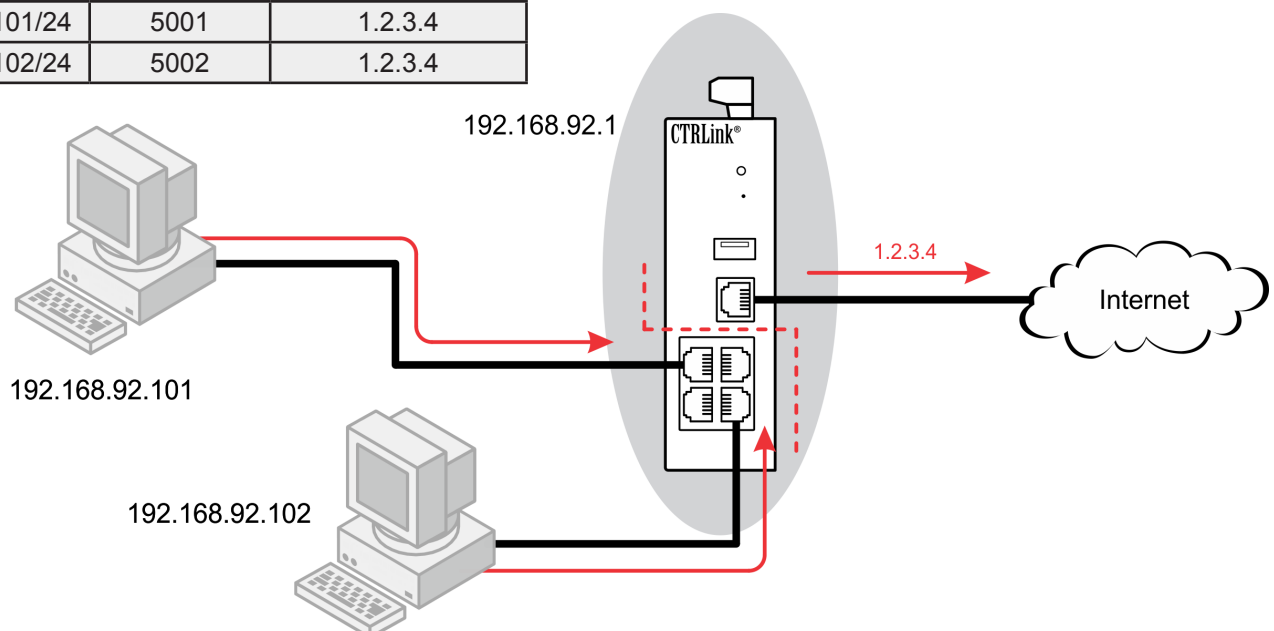
4	3	2	1
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Enabled ☒

### Application #8 — Port Address Translation (PAT)

PAT (also known as a firewall) allows a many-to-one mapping of private IP addresses to one public address. Not only does this provide enhanced security for the devices on the LAN side, it also allows multiple LAN-side devices to communicate to devices on the WAN side using only one WAN IP address. When the WAN network is connected to the Internet, this allows the LAN devices to communicate on the Internet via one public IP address. Most ISPs will limit the number of public IP addresses provided to their customers. PAT is done by the use of port assignments — thus, granting private IP addresses access to the Internet. In this example, the ISP provided the router the public address of 1.2.3.4. Both LAN-side PCs have automatically been assigned local IP ports and granted access to the Internet — and no configuration was needed.

Internal IP Address	LAN IP Port	External IP Address
192.168.92.101/24	5001	1.2.3.4
192.168.92.102/24	5002	1.2.3.4

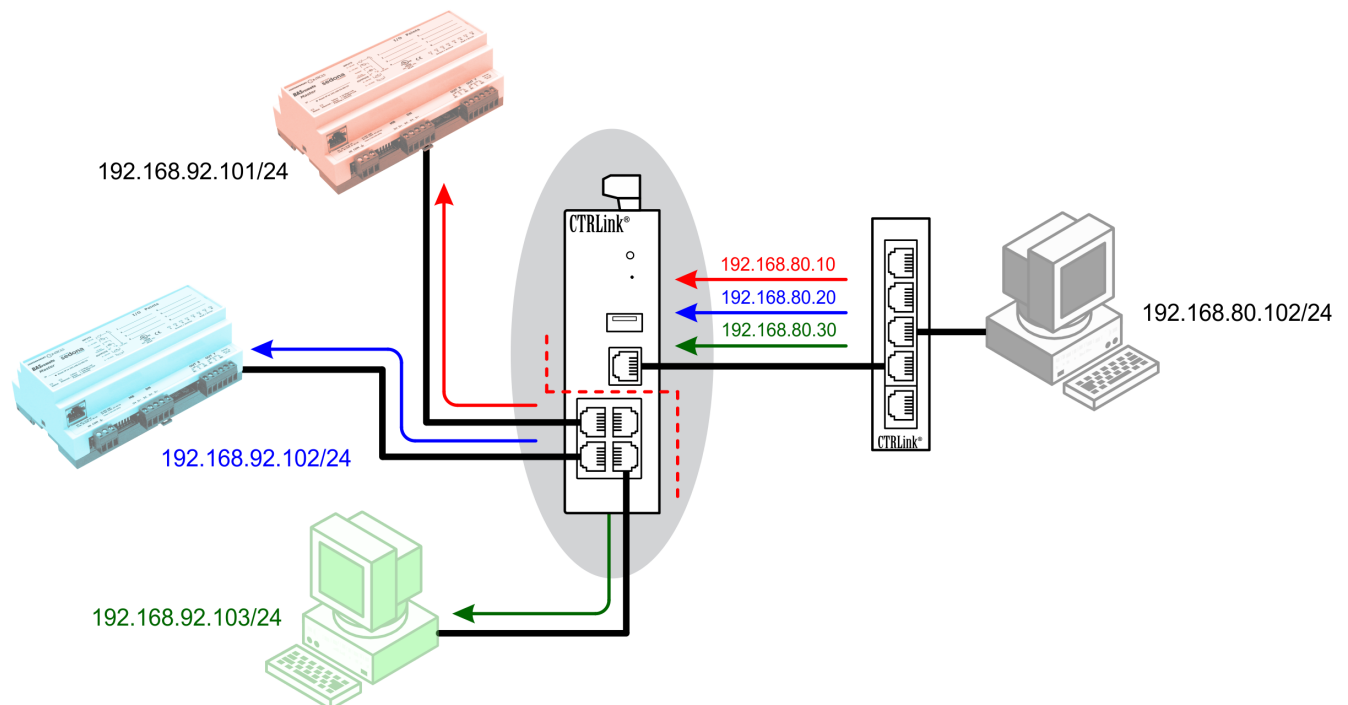


### Application #9 — Network Address Translation (NAT)

NAT allows for a one-to-one mapping of internal IP addresses to external IP addresses. This could be helpful when accessing duplicate systems that are configured the same. The actual LAN-side addresses are hidden. Notice that the LAN and WAN subnets are different.

NAT											
WAN IP Address						LAN IP Address					
192	168	80	10	TO		192	168	92	101		<input checked="" type="checkbox"/>
192	168	80	20	TO		192	168	92	102		<input checked="" type="checkbox"/>
192	168	80	30	TO		192	168	92	103		<input checked="" type="checkbox"/>
				TO							<input type="checkbox"/>

Internal IP Address	External IP Address
192.168.92.101/24	192.168.80.10/24
192.168.92.102/24	192.168.80.20/24
192.168.92.103/24	192.168.80.30/24



# Using Wireshark for Network Troubleshooting

## Introduction

What do you do when you suspect a network problem? Would you grab a multimeter, oscilloscope or a network analyzer? What would you do if the network is not nearby and you are receiving complaints that production is down? This could be a nightmare scenario. Networks are great when they are running but when they are down, or even suspected as being down, they can be extremely difficult to troubleshoot. Network diagnostic tools can be very expensive, but there is one tool that is quite effective with the added bonus that it is free! This tool is called Wireshark.

## Network Protocol Analyzers or Sniffers

The word Sniffer is actually a trade name of a commercial network analyzer from Network Associates. However, the term “sniffer” is commonly used to identify a class of tools that examine packets or frames sent across the network. These tools are called packet sniffers, protocol analyzers or network analyzers. They all capture traffic traversing the network and display the traffic in meaningful ways. Capturing and displaying raw data frames may not be very helpful or efficient so protocol analyzers will also display the meaning of the data sent as packets. To do this, the sniffer must understand the protocol being captured in order to decode the data. With Ethernet frames, there can be numerous protocols operating over Ethernet such as TCP, UDP, IP, and application layer protocols such as BACnet. You need to be sure that the sniffer you intend to use supports the protocol of interest. Sniffers are not restricted to just viewing Ethernet networks. Many, including Wireshark, will work with other popular networks including wireless networks. However, we will restrict our discussion to that of Ethernet. There are several commercial and freeware products that will do sniffing on Ethernet networks, but we will concentrate on this one product called Wireshark because of its wide support.

## Software Analyzers

A commercial network analyzer may consist of specialized hardware and software or it may be completely “soft.” A software analyzer, such as Wireshark, would operate on a desktop or laptop computer relying upon the installed Ethernet network interface controller (NIC) to provide the network interface. The network analyzer software would operate in a Windows or Linux environment capturing packets and storing them in the computer’s memory. It is simply another application that runs on the computer that eliminates the necessity and expense of having a dedicated device for just network traffic capture. There are limitations to this approach.



By using a resident Ethernet NIC in a desktop or laptop computer, you are limited to what a NIC communicates to the operating system. For example, a NIC will not receive a frame that is not destined to its own media access address (MAC). This is the 48-bit address that is unique to every NIC that is made. If the destination MAC address differs from that of the NIC, the NIC discards it. For network analysis, we want to observe all the traffic on the wire and not just the traffic destined to our computer. Therefore, we must put the NIC into a "receive all" mode called "promiscuous" mode. Similar to a NIC receiving broadcast frames, a NIC in promiscuous mode will receive all other directed traffic even though it is not destined to this particular NIC. Of course by doing so, the NIC and the computer will be heavily burdened by capturing all this traffic and the potential of dropped frames exists. When running a sniffer on a desktop or laptop computer, it is best to restrict applications on the computer to just sniffing so that all computer resources can be directed to this processor-intensive activity. You also need to verify that the installed NIC can be set to promiscuous mode.

Another shortcoming of using a standard NIC for capturing traffic is that data link layer problems will not be seen by the sniffer. A deformed frame received by the NIC will be discarded without any notification to the operating system. This could be a frame that is shorter than the minimum size allowed by Ethernet or one that failed the cyclic redundancy check (CRC). These types of framing errors are rejected as a normal course of NIC operation. Although it would be useful to know these types of problems are occurring, specialized hardware would be needed to capture these events. Therefore, a limitation of software analyzers is one in which only frames of the highest integrity can be examined. This means software network analyzers are not suitable for troubleshooting physical layer problems such as faulty wiring or excessive cable length.

### Wireshark Packet Sniffing

You can download Wireshark from the web site [www.Wireshark.org](http://www.Wireshark.org). Wireshark is open source software released under the GNU General Public License. Originally authored by Gerald Combs in 1997, the current list of contributors from all over the world spans several pages. The number of protocols supported now is over 750! Included in the list are automation protocols BACnet, CIP (EtherNet/IP), and Modbus/TCP. The success of this effort certainly points to the benefits of the open source movement.

### Attaching a Sniffer to the Network

Once Wireshark or any other network protocol analyzer is installed on a desktop or laptop computer, it needs to be attached to the network that is to be monitored. This would appear to be a straight forward task for an Ethernet network but there are several issues. It is not as simple as attaching the sniffer to an unused port on a switch. Failure to understand the actual network operation will lead to faulty analysis.

## Using Repeating Hubs

When Ethernet was first developed, it was intended to operate as a bus network where multiple stations shared a common backbone. With this topology, the sniffer could be attached anywhere along this backbone since all stations on the backbone could hear one another. They all reside in the same collision domain. This is called Shared Ethernet or half-duplex Ethernet. Each station would participate in the Carrier Sense, Multiple Access with Collision Detection (CSMA/CD) access rules. A collision would be sensed by all stations and the appropriate action taken. A sniffer does not normally transmit nor should it be the recipient of a directed message. Therefore, it would not participate in the CSMA/CD access rules. However, it could since a regular Ethernet NIC is being used for the network interface. Sniffers are considered passive devices since they simply observe traffic and are not part of the traffic. Since all traffic on shared Ethernet is broadcast, the sniffer with a single NIC can be used to capture all the traffic.

Adding a sniffer to a bus network disrupts the physical wiring of the network so it would be best to use a permanently installed repeating hub with a spare port for making the sniffer connection. This does not disrupt cabling, making the connection of the sniffer transparent to the network. Even with a repeating hub, the sniffer can observe all traffic since the repeating hub remains part of the same collision domain as the backbone with all of its attached stations. Repeating hubs participate in the CSMA/CD access rules and reinforce collisions.

The problem with repeating hubs is that they are not popular and finding multi-speed repeating hubs is difficult. The more recognized connection device is the switch, but switches have their own set of issues.

## Switched Ethernet

A switched Ethernet network creates a distributed star topology where network segments exist between ports on a switch to either stations or ports on other switches. Although the intention is not to use bus segments, bus segments can attach to switch ports. Unlike repeating hubs, switches store-and-forward messages received on one switch port to all other switch ports. The result of this action is that collision domains terminate at switch ports. Removing the collision domain restriction allows switched Ethernet networks to expand geographically without limit from that of shared Ethernet. This characteristic of switched Ethernet would not, by itself, restrict the use of a sniffer. However, switches have another feature that does limit the use of sniffers.

A switch goes through a learning process where it builds an internal table of MAC addresses known to be attached to a particular switch port. This is done for all switch ports. Once a switch determines the location of a station, any directed communication to that station will be limited to the switch port known to have access to that station. All other ports on the switch will not participate in the transmission unless they were party to the transmission. This reduction in communication can yield higher throughput since unnecessary traffic is reduced. However, since the sniffer is not directly involved with the transmission, it would most likely not see the communication. In fact, it is quite possible when you connect a sniffer to a vacant switch port, the sniffer will see nothing except broadcast messages or transmissions to stations that the switch has yet to learn. The switch “floods” these types of transmissions to all switch ports.

There seems to be a trick here. If we can prevent the switch from learning, the switch will continuously flood all ports with any received transmission. In this way the switch is functioning much like a repeating hub and we could connect our sniffer to any port and see all the traffic from this single port. This is true, but it only applies to the switch we are attached to and not to the other switches in the network. The other problem is getting the switch to continuously flood. This is not a standard feature on a switch. Plug-and-Play switches have no mechanism for effecting a change in operation. We call a specialized switch that does not learn a Diagnostic Switch.

### **Port Mirroring**

“Port mirroring” or “port spanning” is the generally accepted method to attach a sniffer to an unused port on a switch. With this feature, all the traffic present on one port can be replicated on another. On some switches, traffic on multiple ports can be replicated onto a single port. By attaching the sniffer to the port which is to receive mirrored traffic, the traffic on adjacent ports can be monitored without disrupting the configuration of the network. Good practice is to leave one port on a switch vacant for the single purpose of attaching a sniffer. The port-mirroring feature is usually only found on managed switches. Invoking the feature is usually done through a console screen or web browser.

There are issues with port mirroring. With Switched Ethernet, the benefit of full-duplex transmissions is possible. By connecting only one station to one port on a switch while defeating collision detection, two simultaneous high-speed connections are possible. A 100 Mbps link on Shared Ethernet is limited to 100 Mbps throughput. However, a 100 Mbps link on Switched Ethernet has an effective throughput of 200 Mbps. When you attach a sniffer to an unused 100 Mbps port on a switch and mirror traffic from another port to the sniffer port, the sniffer can only receive a maximum of 100 Mbps. If the mirrored port is operating full-duplex and is fully utilized, the switch electronics will attempt to supply 200 Mbps data to the sniffer resulting in dropped frames. To effectively use port mirroring, the mirrored port should only be operating no more than 50% of its throughput if full-duplex traffic is being captured.

## Analyzing a Packet Captured by Wireshark

The adjacent screen is the result of capturing BACnet/IP packets over Ethernet. Wireshark displays information in three window panes. The top pane is the Summary, the middle pane is the Detail, and the lower pane is the Data. Each line in the top pane represents a captured Ethernet frame. Wireshark will continue to capture traffic until requested to stop. Individual frames can be examined while Wireshark is stopped or while it is capturing. In our example, one of the frames in the upper pane is emphasized, resulting in highlighted data in the bottom pane. The bottom pane displays 16 octets per line in hexadecimal format. To the immediate right is the same data shown in ASCII format. To the untrained observer, the data in both formats looks meaningless. More can be gained by looking at the middle pane.

Information in the middle pane can be expanded by clicking on the + button. For the sake of discussion, all items have been expanded so we can understand how Wireshark interpreted the frame. First of all, Wireshark recognized that it has captured an Ethernet II frame and identified both the destination and source MAC addresses. The destination address is a broadcast intended for all stations. Notice that the 48-bit source address identifies the vendor of the NIC involved in the transmission. The first part of the 48-bit address is the vendor code. Wireshark knows the vendor assignments. The Type field (this is an Ethernet II frame) contains 0x800 which indicates that an IP packet has been captured. Return to the bottom pane and notice the location of the two MAC addresses and Type field in the raw capture. The destination address was sent first, followed by the source address, just like we would expect in an Ethernet frame. The Type field immediately follows the source address. What should follow now is the IP header.

Wireshark decodes the IP header for you. Both source and destination IP addresses are named as Class A private addresses. The header length of 20 bytes is the normal length for an IP header. Other header information such as Time to Live, Version, and other fields are decoded as well. You need to consult a TCP/IP reference in order to understand these terms.

The payload data inside the IP wrapper is actually a UDP datagram and not a TCP segment. Datagrams are not acknowledged as are segments. BACnet relies upon the application layer to acknowledge receipt of a message and not the transport layer of the TCP/IP stack. The datagram begins with a UDP header. This time, source and destination ports are identified.

The image shows a Wireshark packet capture analysis. The top pane (Summary) shows two captured packets. Packet 1 is selected, showing details in the middle pane and raw data in the bottom pane.

**Summary Pane:**

No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.0.0.100	10.0.0.246	BACnet-A	BACnet APDU (Confirmed)
2	0.003107	10.0.0.246	10.0.0.100	BACnet-A	BACnet APDU (Confirmed)

**Detail Pane (Frame 1):**

- Ethernet II, Src: 00:07:e9:76:97:0a, Dst: 00:50:db:aa:aa:11 (Contempo\_aa:aa:11)
  - Destination: 00:50:db:aa:aa:11 (Contempo\_aa:aa:11)
  - Source: 00:07:e9:76:97:0a (Intel\_76:97:0a)
  - Type: IP (0x0800)
- Internet Protocol, Src Addr: 10.0.0.100 (10.0.0.100), Dst Addr: 10.0.0.246 (10.0.0.246)
  - Version: 4
  - Header length: 20 bytes
  - Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  - Total Length: 45
  - Identification: 0x7d05 (32005)
  - Flags: 0x00
  - Fragment offset: 0
  - Time to live: 128
  - Protocol: UDP (0x11)
  - Header checksum: 0xa861 (correct)
  - Source: 10.0.0.100 (10.0.0.100)
  - Destination: 10.0.0.246 (10.0.0.246)
- User Datagram Protocol, Src Port: 47808 (47808), Dst Port: 47808 (47808)
  - Source port: 47808 (47808)
  - Destination port: 47808 (47808)
  - Length: 25
  - Checksum: 0xc49b (correct)
- BACnet Virtual Link Control
  - Type: 0x81 (Version BACnet/IP (Annex J))
  - Function: 0x0a (Original-Unicast-NPDU)
  - BVLC-Length: 4 of 17 bytes BACnet packet length
- Building Automation and Control Network NPDU
  - Version: 0x01 (ASHRAE 135-1995)
  - Control: 0x04
- Building Automation and Control Network APDU
  - APDU Type: 0 (Confirmed-Request-PDU)
  - Data (10 bytes)

**Data Pane:**

Offset	Hex	ASCII
0000	00 50 db aa aa 11 00 07 e9 76 97 0a 08 00 45 00	.P.....V....E.
0010	00 2d 7d 05 00 00 80 11 a8 61 0a 00 00 64 0a 00	.-}.....a...d..
0020	00 f6 ba c0 ba c0 00 19 c4 9b 81 0a 00 11 01 04	.....
0030	00 00 0c 0c 0c 0c 00 01 19 55	.....U

Filter: / Add Expression... Clear Apply Ethernet (eth), 14 bytes

### Cable Taps

The ideal approach to capturing data is to have a “passive tap.” These taps are inserted between two devices on the network—so installation will disrupt operation. Once installed, it is best to leave the cable tap in place. Actually, the cable taps are not completely passive. Electronics are needed to transfer the sniffed traffic to the monitoring electronics. If power is lost to the tap, there is disruption of network traffic being monitored. One advantage is that the tap will operate over varying data rates without adjustment. It is possible to monitor 10/100/1000 Mbps traffic. A single cable tap provides two monitoring outputs. One output monitors the traffic from point A to B while the other monitors the traffic from B to A. Therefore, monitoring both sides of a full-duplex transmission requires the use of two NICs and a dual-channel sniffer. Although a more complicated approach, the throughput issues are eliminated. Cable taps offer another advantage. Framing errors on the monitored segment can be passed onto the sniffer. Of course, the sniffer would need the specialized hardware that can detect the framing errors.

### Summary

Sophisticated industrial networks that use technologies such as Ethernet require a troubleshooting tool that rises to that same level of sophistication. One of the best tools for troubleshooting networks is a network sniffer or protocol analyzer which can translate the traffic on the network into meaningful data to the operator. One such tool is Wireshark which is available for free on the Internet. Connecting a sniffer to a network is no simple task. A misunderstanding of how a switched Ethernet network operates can lead to faulty analysis. Making a connection using either a repeating hub, switching hub or cable tap is possible. All methods have their advantages and disadvantages. However, once a sniffer is properly attached to a network, meaningful data regarding the health of the network can be gained.

### References

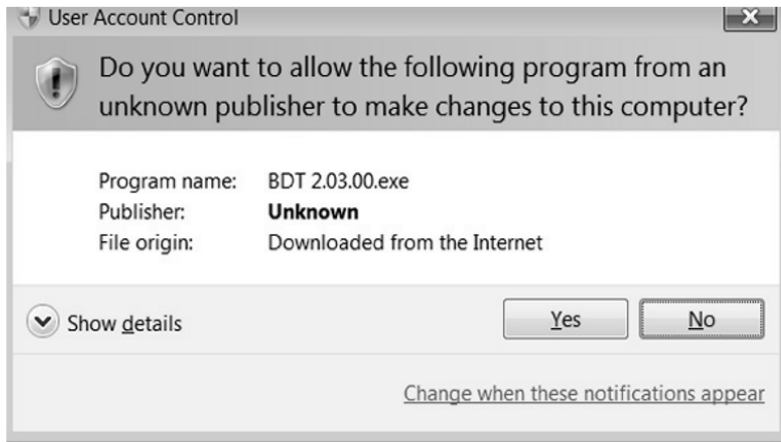
*Wireshark Packet Sniffing*, Angela Orebaugh, Syngress Publishing Company, 2004.

## Using The BACnet Discovery Tool (BDT)

### Introduction

The BACnet Discovery Tool (BDT)—is a vendor-neutral tool to determine if a BACnet router is successfully communicating to attached devices. BDT is a BACnet/IP application for Windows® that is easy to install and use. It is an excellent means for discovering and verifying communication with MS/TP devices that are being accessed through BACnet/IP routers.

Download this free, handy, application at Contemporary Controls' website:  
<http://www.ccontrols.com/sd/bdt.htm>

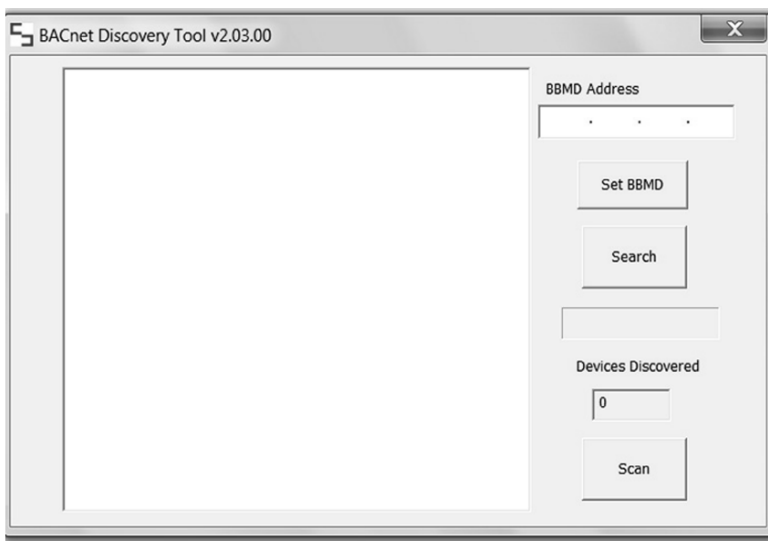


After downloading BDT, unzip its file set to any location on your host PC hard drive (be sure to keep all the files together at the chosen location). The file set will include an instruction sheet in PDF format and the following four files:

- bacnet-stack.dll
- BDT 2.03.00.exe
- mfc71d.dll
- msvcr71d.dll

If you are not using Windows 7, double-click on the **BDT2.03.00.exe** file to launch BDT. But if you are using Windows 7, right click the file name, choose to "Run as Administrator" and click Yes in the dialogue which appears to the right—otherwise, BDT will not function properly.

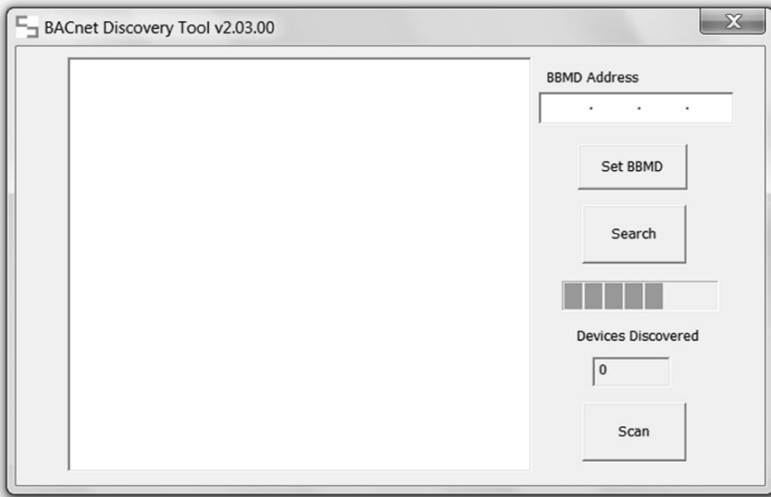
When you attempt to open BDT, Windows may notify you that the file has no valid digital signature. If so, it is safe to ignore this warning and proceed. As BDT starts, it opens in its own window similar to that shown.



Before initiating any BDT activity, you should determine the scope of what you wish to accomplish. The first thing to consider is whether or not your BACnet internetwork crosses IP subnets—and is therefore using a BACnet/IP Broadcast Management Device (BBMD). If a BBMD is present, you may wish to use it to examine more of the BACnet internetwork than just your local subnet. You can do this by using the BDT's **Foreign Device Registration (FDR)** function to register the BDT application as a Foreign Device with a BBMD in a remote subnet.

**Setting the BBMD, if needed.** In the BBMD Address field, enter the IP address of the BBMD with which you wish to register. (You do not specify a subnet mask—this is determined by other equipment in the network.) In some larger networks with several subnets, there may be multiple BBMDs.





In such cases, you would normally register with the central BBMD and thus access the entire BACnet internetwork. You could perform FDR with a non-central BBMD on a specific subnet, but you may not know the extent to which the various BBMDs are sharing information—and you might not access the devices that you need to contact. Targeting the central BBMD is almost always the best option. When you click on the Set BBMD button, registration is completed.

**Using the Search function.** After setting the BBMD (if needed) and with your host PC attached to the network of interest, click the Search button—*not the Scan button*, which should only be used after a search has built a database of objects.

As the Search function runs, it transmits BACnet Who-Is messages and a progress bar appears as in the following screen. **NOTE:** BDT sends a **BACnet/IP Who-Is**—not a **BACnet/Ethernet Who-Is**—so it will not discover devices that only support BACnet/Ethernet.

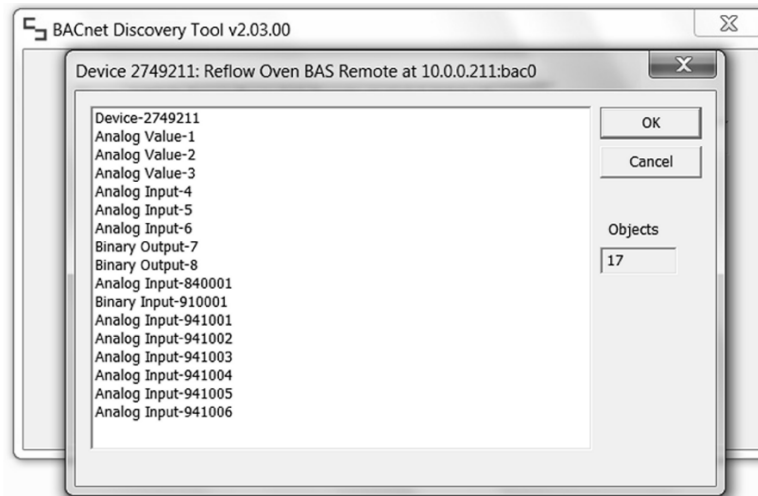
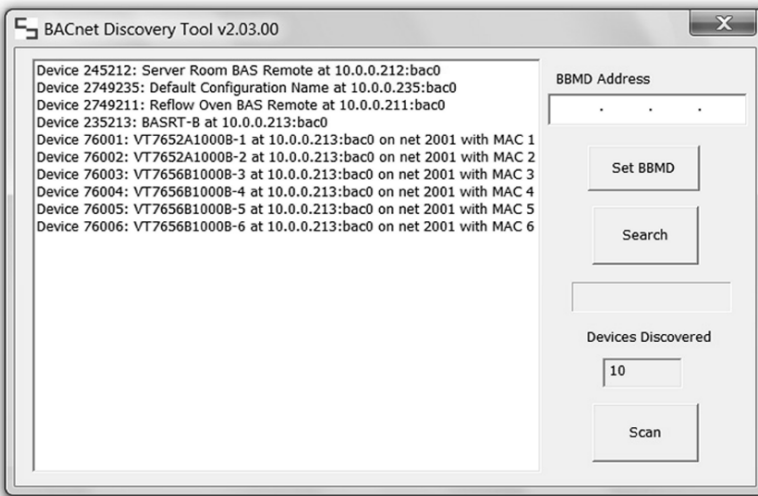
When BDT has completed its search, the progress bar will disappear, the discovered BACnet devices will be listed in the main window and the number of discovered devices will be reported in the Devices Discovered field.

As shown in the sample screen, each I-Am response identifies:

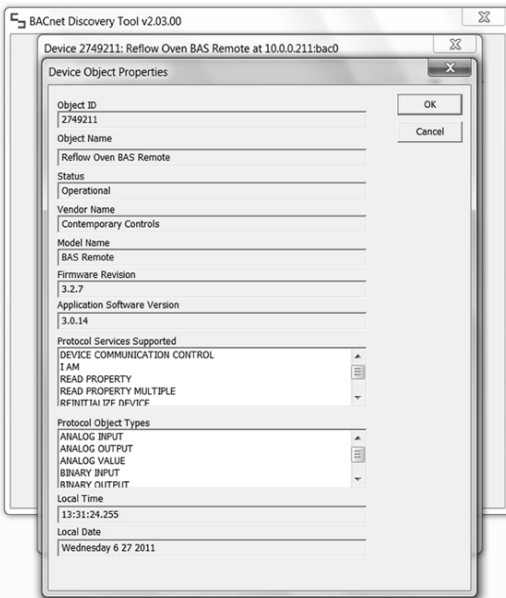
- the Device Instance number of the responding device
- the Device Name of the device
- the IP Address and UDP Port number through which the device was contacted and (for MS/TP master devices) the MS/TP Network number and MS/TP MAC Address

Each piece of equipment producing an I-Am response could be simply a BACnet/IP device (as is the case with the first three respondents in the screen)—or it could be a BACnet/IP to BACnet MS/TP router acting as an intermediary for devices on the MS/TP side (which is the case with Device 235213 above). If the reported Device Instance is that of an MS/TP device, then the MS/TP Network and MAC information will complete the line.

To investigate the objects contained by any discovered device, double-click that device's line in the discovered device list. As shown below, a new window will appear and display a list of the discovered **objects** within the selected device.







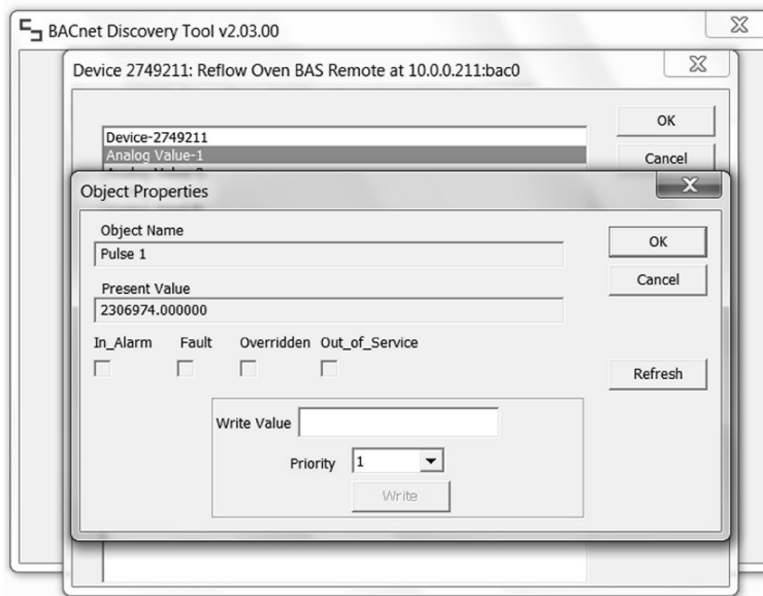
Also, the number of discovered objects will be reported in the Objects field. BDT can support up to 1000 devices and each device can have up to 2000 objects.

You can double-click the device object line to examine the Device Object Properties (Device 2749211 in our example) which brings up a window similar to that which is displayed to the right. As shown, this window reports many helpful Device Object Properties.

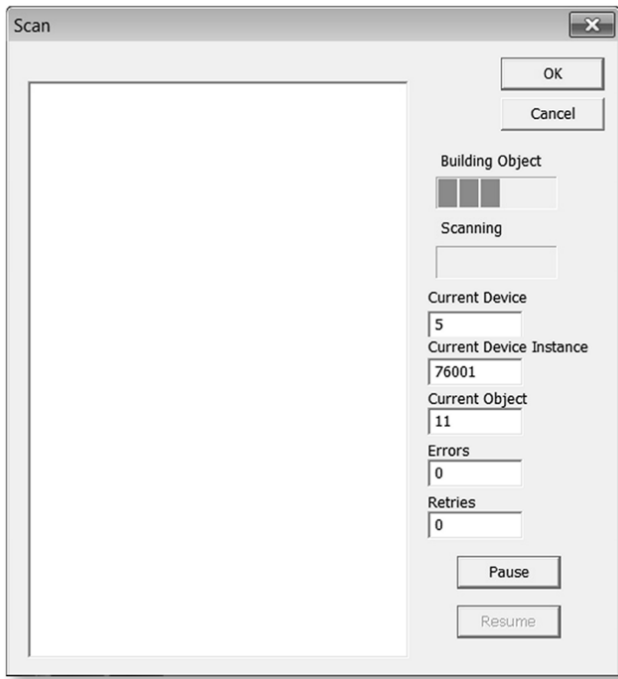
If you the window shown to the left and again display the list of objects contained by the device, you can double-click one of its objects to bring up an Object Properties window similar to that below.<sup>1</sup>

The Object Properties window displays read-only values—Object Name, its Present Value and four on/off status reports (checked means on)—and two writable elements lower down which, although always visible, only yield results for **output** objects. If the Write button is not dimmed, the examined object is writable—in which case, you can enter a number in the Write Value box, set a Priority value and click the Write button to apply your changes. Set a priority value by clicking the Priority field then adjusting the level displayed below the field by scrolling through the available levels using either your keyboard up/down arrow keys or the miniature up/down scroll controls. Clicking the Refresh button on the right will reacquire and re-display the read-only information.

After you are done with the Object Properties window, you can close it with either the OK or Cancel button. Then you can double-click another object to view and/or adjust its properties.



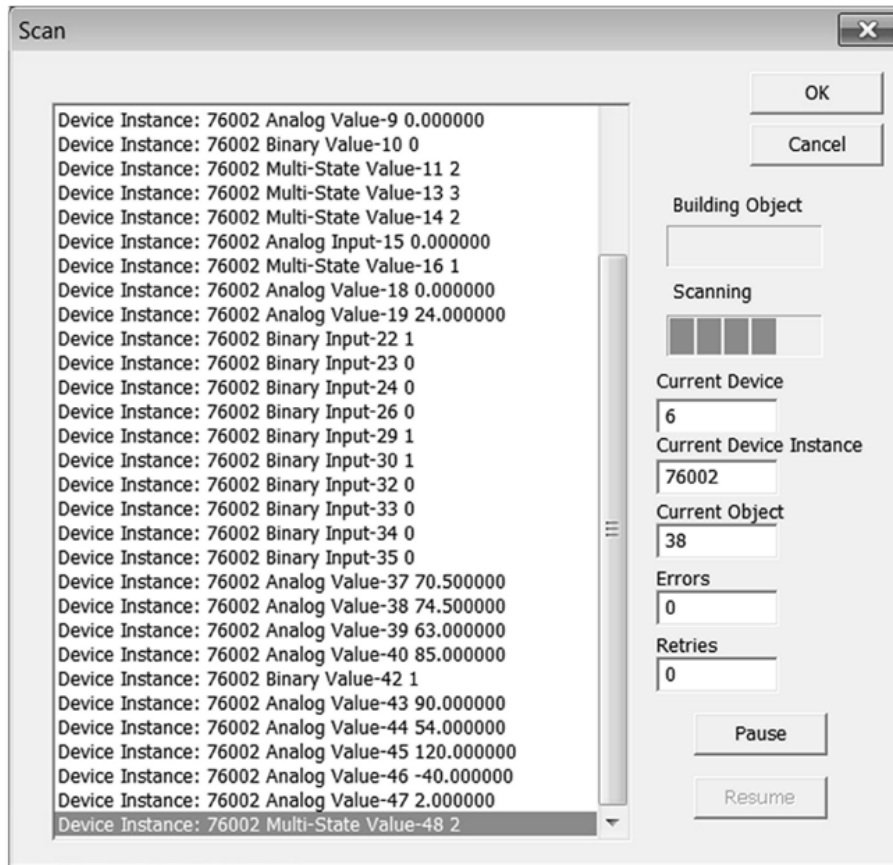
**Only click the Scan button after the Search function has been run.** Usually you should avoid the Scan function because it often creates more network activity than desired by continuously reading the Present Value of most objects that have this property (and alert you if an attempt fails). However, when commissioning or troubleshooting a network, the Scan function can usefully confirm that all devices and objects are properly installed, working and updating. This could be especially helpful if you do not have a building controller at your disposal to check this for you. NOTE: The Scan function only stops when you close the Scan window. Leaving it running can consume significant network bandwidth—and if a building controller is running, communication errors could result.



Executing the Search function creates an object database that is used by the Scan function. Clicking the Scan button opens a window similar to that below where you will see a bar indicating the progress in Building Object—that is, creating a scan list of objects extracted from the object database. As the list is built, the current Device and its Object are identified in their respective fields.

After the scan object list has been built, the scan proceeds as shown in the sample screen.

After the scan object list has been built, the scan proceeds as shown in the sample screen. As the list is scanned, the current Device and its Object being scanned are identified in their respective fields. If an object fails to report its data, BDT retries the data acquisition four times before registering an error and moving on to the next object. These two fields—Errors and Retries—are incremented without reset for as long as the scan proceeds. You can also Pause the scan, if needed.



When scanning for present-values, BDT will display the currently read values. When all objects have been read, BDT will start displaying again—at the first line. Because present-values will appear in the same place in the display, current values can be viewed by scrolling the list up or down. If a present-value cannot be read, an error statement will occupy a line in the list.

When you are finished with scanning, click the OK button to terminate the scan. When you are finished with the BDT, close all of its windows.

## Glossary

The interest in Industrial Ethernet has brought about a completely new dictionary of terms. Here are some of the most important terms we have introduced in earlier Extension articles.

**4B/5B**—A block encoding scheme used to send Fast Ethernet data. In this signal encoding scheme, 4 bits of data are turned into 5-bit code symbols for transmission over the media system.

**4D-PAM5** Encoding scheme used with Gigabit Ethernet where two-bits of data is transmitted as a five-level symbol over twisted-pairs.

**10BASE-T**—10 Mbps Ethernet system based on Manchester signal encoding transmitted over Category 3 or better twisted-pair cable.

**10BASE2**—10 Mbps Ethernet system based on Manchester signal encoding transmitted over thin coaxial cable. Also called Cheapernet or ThinNet.

**10BASE5**—Original 10 Mbps Ethernet system based on Manchester signal encoding transmitted over thick coaxial cable. Also called ThickNet.

**10BASE-F**—10 Mbps Ethernet system based on Manchester signal encoding transmitted over fiber optic cable. This is a base standard for three fiber optic implementations.

**10BASE-FP**—10 Mbps passive MAU fiber optic implementation which is not popular.

**10BASE-FB**—10 Mbps backbone MAU fiber optic implementation which is not popular.

**10BASE-FL**—Popular 10 Mbps link fiber optic implementation which replaces the older FOIRL implementation.

**100BASE-FX**—100 Mbps Fast Ethernet system based on 4B/5B signal encoding transmitted over fiber optic cable.

**1000BASE-SX** 1000 Mbps system using 8B/10B encoding transmitted over short wavelength fiber optic cable.

**100BASE-T**—Term used for the entire 100 Mbps Fast Ethernet system, including both twisted-pair and fiber optic media types.

**100BASE-T2**—100 Mbps Fast Ethernet system designed to use two pairs of Category 3 twisted-pair cable. Not a popular technology.

**100BASE-T4**—100 Mbps Fast Ethernet system designed for four pairs of Category 3 twisted-pair cable. Not a popular technology.

**100BASE-TX**—100 Mbps Fast Ethernet system based on 4B/5B signal encoding transmitted over two pairs.

**100BASE-X**—Term used when referring to any Fast Ethernet media system based on 4B/5B block encoding. Includes 100BASE-TX and 100BASE-FX media systems.

**1000BASE-LX** 1000 Mbps system using 8B/10B encoding transmitted over long wavelength fiber optic cable.

**1000BASE-T** 1000 Mbps system using 4D-PAM5 encoding resulting in a five-level pulse amplitude modulated signal sent over four twisted-pairs.

**802.2**—IEEE Working Group for Logical Link Control (LLC).

**802.3**—IEEE Working Group for CSMA/CD LANs (local area networks).

**8B/10B** A signal encoding scheme used in Gigabit Ethernet fiber where 8-bit bytes are turned into 10-bit code groups.

**AUI**—Attachment Unit Interface. The 15-pin signal interface defined in the original Ethernet standard that carries signals between a station and an outboard transceiver.

**Auto-Negotiation**—A protocol defined in the Ethernet standard that allows devices at either end of a link segment to advertise and negotiate modes of operation such as the speed of the link, flow control or half- or full-duplex operation.

**Backbone**—A network used as a primary path for transporting traffic between network segments. A backbone network is often based on higher capacity technology, to provide enough bandwidth to accommodate the traffic of all segments linked to the backbone.

**Bandwidth**—The maximum capacity of a network channel. Usually expressed in bits per second (bps). Ethernet channels have a bandwidth of 10-, 100-, and 1000 Mbps.

**Baud**—A baud is a unit of signaling speed representing the number of discrete signal events per second and depending upon the encoding can differ from the bit rate.

**Bit**—A binary digit. The smallest unit of data, either a zero or a one.

**Bit Rate**—The amount of bits that can be sent per second. Usually described in units of kbps or Mbps and frequently referred to as the data rate.

**Block Encoding**—Block encoding is a system whereby a group of data bits are encoded into a larger set of code bits. Block encoding is used in Fast Ethernet.

**BNC**—A bayonet locking connector used on 10BASE2 thin coaxial cable segments and is commonly found in communication systems.

**Bridge**—A device that connects two or more networks at the data link layer (layer 2 of the OSI model).

**Broadcast Domain** —The set of all stations in a network that will receive each other's broadcast frames. A single segment or set of segments connected with a repeater or switches are included in a broadcast domain.

**Broadcast**—A transmission initiated by one station to all stations on a network.

**Bus**—A shared connection for multiple devices over a cable or backplane.

**Category 3**—Twisted-pair cable with electrical characteristics suitable for carrying 10BASE-T. Not recommended for new installations.

**Category 5**—Twisted-pair cable with electrical characteristics suitable for all twisted-pair Ethernet media systems, including 10BASE-T and 100BASE-TX. Category 5 and Category 5e cable are the preferred cable types for structured cabling systems.

**Category 5e**—An enhanced version of Category 5 cable, developed to improve certain cable characteristics important to Gigabit Ethernet operation. It is recommended that all new structured cabling systems be based on Category 5e cable; however, this cable may not be the best for use in industrial installations because of noise susceptibility.

**Coaxial Cable**—A cable with an outer conductor, surrounding an inner conductor. Coaxial cables are used for 10BASE5 and 10BASE2 Ethernet systems.

**Collision**—The result of having two or more simultaneous transmissions on a common signal channel such as half-duplex Ethernet or shared Ethernet.

**Collision Domain**—The set of all stations and repeaters connected to a network where faithful detection of a collision can occur. A collision domain terminates at switch ports.

**CRC**—Cyclic Redundancy Check. An error checking technique used to ensure the accuracy of transmitted data.

**Crossover Cable**—A twisted-pair patch cable wired in such a way as to route the transmit signals from one piece of equipment to the receive signals of another piece of equipment, and vice versa. This allows communication between two DTEs or two DCEs. The opposite of a crossover cable is the straight-through cable.

CSMA/CD—Carrier Sense Multiple Access/Collision Detect. The medium access control (MAC) protocol used in Ethernet.

Data Link Layer—Layer 2 of the OSI reference model. This layer takes data from the network layer and passes it on to the physical layer. The data link layer is responsible for transmitting and receiving frames. It usually includes both the media access control (MAC) protocol and logical link control (LLC) layers.

DCE—Data Communications Equipment. Any equipment that connects to Data Terminal Equipment (DTE) to allow data transmissions between DTEs. DCEs are not considered end devices or stations.

DTE—Data Terminal Equipment. Any piece of equipment at which a communication path begins or ends. A station (computer or host) on the network is capable of initiating or receiving data.

Encoding—A means of combining clock and data information into a self-synchronizing stream of signals.

Error Detection—A method that detects errors in received data by examining cyclic redundancy checks (CRC) or checksum.

Ethernet—A popular local area networking (LAN) technology first standardized by DEC, Intel, and Xerox (or DIX) and subsequently standardized by the IEEE through the 802.3 committee.

Ethernet I/O—A system in which analog and/or digital inputs and outputs are connected to a host computer via some type of Ethernet link. The host computer then accepts data from the inputs and sends out data which controls the outputs.

Fast Ethernet—A version of Ethernet that operates at 100 Mbps. Although 100 Mbps is not considered fast, this reference is still used.

Fast Link Pulse—A link pulse that encodes information used in the Auto-Negotiation protocol. Fast link pulses consist of bursts of the normal link pulses used in 10BASE-T.

FDDI—Fiber Distributed Data Interface. An ANSI standard (ANSI X3T12) for a 100 Mbps token passing network (Token Ring) based on fiber-optic and twisted-pair cable. Some of this technology is used in the Fast Ethernet standard.

Fiber Optic Cable—A cable with a glass or plastic filament which transmits digital signals in the form of light pulses.

Flow Control—The process of controlling data transmission at the sender to avoid overfilling buffers and loss of data at the receiver.

FOIRL—Fiber Optic Inter-Repeater Link. An early version of fiber optic link segment replaced by 10BASE-FL

Forwarding—The process of moving frames from one port to another in a switching hub.

Frame—The fundamental unit of transmission at the data link layer of the OSI mode.

Full-Duplex Operation—A communications method that allows for the simultaneously transmission and reception of data.

Gigabit Ethernet—A version of Ethernet that operates at 1000 Mbps.

Half-Duplex Operation—A communications method in which transmissions and receptions can occur in either direction but not at the same time.

Hub—A device with three or more ports at the center of a star topology network. Hubs can usually be cascaded with a hub-to-hub connection. Frequently this name is used to mean repeating hub.

IEEE—Institute of Electrical and Electronics Engineers. A professional organization and standards body.

Interframe Gap—An idle time between frames, also called the interpacket gap.

Intranet—The Internet is the worldwide collection of networks based on the use of TCP/IP network protocols.

Jabber—The act of continuously sending data. A jabbering station is one whose circuitry or logic has failed, and which has locked up a network channel with its incessant transmissions.

Jitter—Also called phase jitter, timing distortion, or inter-symbol interference. The slight movement of a transmission signal in time or phase that can introduce data errors and loss of synchronization.

Late Collision—A failure of the network in which the collision indication arrives too late in the frame transmission to be automatically dealt with by the medium access control (MAC) protocol. The defective frame may not be detected by all stations requiring that the application layer detect and retransmit the lost frame, resulting in greatly reduced throughput.

Link Integrity Test—This test verifies that an Ethernet link is connected correctly and that signals are being received correctly. This is a helpful aide but does not guarantee the link is completely functional.

Link Layer—Short for Data Link Layer. This is layer 2 on the OSI model

Link Light—An optional status light on a DTE or DCE that indicates the status of the link integrity test. If this light is lit on both ends of the link, it indicates that the link is passing the link integrity test.

Link Pulse—A test pulse sent between transceivers on a 10BASE-T link segment during periods of no traffic, to test the signal integrity of the link.

Link Segment—A point-to-point segment that connects only two devices and is “capable” of supporting full-duplex operation.

LLC—Logical Link Control. A standardized protocol and service interface provided at the data link layer and independent of any specific LAN technology. Specified in the IEEE 802.2 standard.

MAC—Medium Access Control. A protocol operating at the data link layer used to manage a station’s access to the communication channel.

MAC Address—A unique address assigned to a station interface, identifying that station on the network. With Ethernet, this is the unique 48-bit station address. Same as the physical address.

Manchester Encoding— Signal encoding method used in all 10 Mbps Ethernet media systems. Each bit of information is converted into a “bit symbol” which is divided into two halves. One half is high and the other is low. Manchester encoding results in a 20 Mbaud stream although data is only being sent at 10 Mbps.

MAU—Medium Attachment Unit. The MAU provides the physical and electrical interface between an Ethernet device and the medium system to which it is connected. Also referred to as a transceiver.

MDI—Medium Dependent Interface. The name for the connector used to make a physical and electrical connection between a transceiver and a media segment. For example, the RJ-45-style connector is the MDI for 10BASE-T and 100BASE-TX.

MDI-X—An MDI port on a hub or media converter that implements an internal crossover function. This means that a “straight-through” patch cable can be used to connect a station to this port, since the signal crossover is performed inside the port.

**MII**—Medium Independent Interface. Similar to the original AUI function, but designed to support both 10 and 100 Mbps. An MII provides a 40-pin connection to outboard transceivers (also called PHY devices). Used to attach 802.3 interfaces (MACs) to a variety of physical media systems.

**Media Converter**—A device that converts signals from one media type to that of another.

**Mixing Segment**—A bus segment capable of supporting two or more devices on the same bus. Coaxial cable segments are classified as mixing segments.

**Multicast**—A transmission initiated by one station to many stations on the network.

**Network Layer**—Layer 3 of the OSI reference model. At this layer routing of packets between multiple networks occur.

**NIC**—Network Interface Card. Also called an adapter, network interface module, or interface card. The set of electronics that provides a connection between a computer and a network.

**Octet**—Eight bits (also called a byte). This term is typically used in communication protocol descriptions.

**OSI**—Open Systems Interconnection. A 7-layer reference model for networks, developed by the International Organization for Standardization (ISO). The OSI reference mode is a formal method for describing the interlocking sets of networking hardware and software used to deliver network services. It is a good model to refer to but strict compliance to the model is seldom accomplished.

**OUI**—Organizationally Unique Identifier. A 24-bit value assigned to an organization by the IEEE. Ethernet vendors use the 24-bit OUI they receive from the IEEE in the process of creating unique 48-bit Ethernet addresses. Contemporary Controls has been assigned a vendor OUI.

**Packet**—A unit of data exchanged at the network layer. This is a much abused definition and the terms frame and packet are frequently interchanged.

**Patch Cable**—A twisted-pair or fiber optic jumper cable used to make a connection between a network interface on a station or network port on a hub and a media segment, or to directly connect stations and hub ports together.

**PHY**—Physical Layer Device. The name used for a transceiver in Fast Ethernet and Gigabit Ethernet systems.

**Physical Layer**—The first layer in the OSI seven layer reference model. This layer is responsible for physical signaling, including the connectors, timing, voltages, and related issues. Data sent over the physical layer are termed symbols.

**Plenum Cable**—A cable that is rated as having adequate fire resistance and satisfactorily low smoke-producing characteristics for use in plenums (air handling spaces). Air handling spaces are often located below machine room floors, or above suspended ceilings requiring the use of plenum rated cable.

**Point-to-Point Technology**—A network system composed of point-to-point links. Each point-to-point link connects two and only two devices, one at each end. Devices could be DTEs or DCEs but no more than two can be connected on one link.

**Power over Ethernet (PoE)** Standardized method of transmitting both data and power over twisted-pair cabling.

**Port**—A connection point for a cable. Repeater hubs and switching hubs typically provide multiple ports for connecting Ethernet devices.



**Promiscuous Mode**—A mode of operation where a device is configured to receive all frames on a network regardless of its destination address. Typically used by network analyzer tools.

**Propagation Delay**—The signal transit time through a cable, network segment, or device. Important in making collision domain calculations.

**Protocol**—A set of agreed-upon rules and message formats for exchanging information among devices on a network. **Repeater**—A physical layer device used to interconnect segments within the same network. An Ethernet repeater can only link Ethernet segments that are all operating in half-duplex mode and at the same speed. Some repeaters can offer media conversion as well.

**Repeating Hub**—A repeater with more than two ports. This name is frequently shortened to simply hub.

**RJ-45**—An 8-pin modular connector used on twisted-pair links.

**Router**—A device or process based on Layer 3 network protocols used to interconnect networks at the network layer.

**SC**—Subscriber Connector. This is a type of fiber optic connector used in 100BASE-FX fiber optic media systems. The connector is designed to be pushed into place, automatically seating itself.

**Segment**—A cable made up of one or more cable sections and connections joined together to produce the equivalent of a continuous cable. **Slot Time**—A unit of time used in the medium access control (MAC) protocol for Ethernet.

**ST**—Straight Tip. This is a type of fiber optic connector used in 10BASE-FL and FOIRL links, but can be found on 100BASE-FX systems as well. The male end of this connector has an inner sleeve with a slot cut into it, and an outer ring with a bayonet latch. The inner sleeve is aligned with a mating key in the socket and the outer ring is turned to complete the bayonet latch.

**Star Topology**—A network topology in which each station on the network is connected directly to a hub.

**Straight-through**—Refers to a cable where cable connections at both ends of the cable are pinned the same way. Used to connect a DTE to a DCE.

**Station**—A unique, addressable device on a network. Sometimes referred to as a node.

**Switching Hub**—A switching hub is another name for a bridge, which is a device that interconnects network segments at the data link layer. Switching hubs are typically located in the center of a star topology, and provide multiple ports for connections to network stations. Frequently this name is shortened to switch.

**Terminator**—A resistor used at the end of copper network cables to minimize reflections.

**Topology**—The physical layout of a network.

**Transceiver**—A combination of the words transmitter and receiver. A transceiver is the set of electronics that sends and receives signals on a media system. Transceivers may be internal or external. Sometimes called a MAU.

**Twisted-Pair Cable**—A multiple-conductor cable whose component wires are paired together, twisted, and enclosed in a single jacket. A typical Category 5 twisted-pair segment is composed of a cable with four twisted pairs contained in a single jacket. Each pair consists of two insulated copper wires that are twisted together.

## References

*Ethernet The Definitive Guide*, Charles E. Spurgeon, 2000, O'Reilly & Association, Inc.

*International Standard ISO/IEC 8802.3 ANSI/IEEE Std 802.3*, 2000, The Institute of Electrical and Electronic Engineers, Inc.

# Original Design Manufacturing (ODM) Service

We can provide the product you require under your brand. The year 2015 marks our 40<sup>th</sup> year of experience in electronics design, development and manufacturing. We have a rich inventory of intellectual property that can be tapped for your next project.

Two design and manufacturing locations provide private label, ODM and electronics manufacturing services. Leverage our design and manufacturing resources to reduce your costs and time-to-market.

## Designed to Worldwide Standards

Two design centers — one in China and the other in the United States — cooperate on product designs from concept to production.

Capabilities include:

- Schematic capture & printed circuit board layout
- Firmware and programmable logic development
- Mechanical design
- Design for Test (DFT)
- Design for Manufacturing (DFM)
- Environmental testing
- Electromagnetic Compatibility (EMC)
- Safety and performance testing

We assist in obtaining regulatory approvals, including UL, CE and CCC markings.

## Worldwide Electronics Manufacturing

Contemporary Controls offers lead-free surface-mount-technology (SMT) electronics manufacturing in the United States and China while complying with the requirements for the Restriction of Hazardous Substances (RoHS) European Union directive. Through-hole assembly and wave soldering are also supported. Contemporary Controls adheres to the workmanship standards established by IPC — Association Connecting Electronics Industries.

The Downers Grove, Illinois, manufacturing plant focuses on lower-volume, higher-mix products or those products requiring Made-in-America compliance or a North American Free Trade Agreement (NAFTA) certificate.

For higher-volume, lower-mix, cost-sensitive requirements, our Suzhou, PRC plant offers the highest production capacity and global logistics support. The Suzhou plant is ISO 9001:2008 registered. Both plants are under Underwriters Laboratories (UL) surveillance. Your intellectual property (IP) is protected at either plant location.



## OUR QUALITY POLICY

Contemporary Controls develops, manufactures and markets innovative networking and control products to the benefit of our automation customers worldwide. We are committed to delivering products and services that meet customer requirements and strive to exceed their expectations through our continuous improvement efforts.

Trademarks — Contemporary Controls, ARC Control, ARC DETECT, BASautomation, CTRLink, EXTEND-A-BUS and RapidRing are trademarks or registered trademarks of Contemporary Control Systems, Inc. BACnet is a registered trademark of the American Society of Heating, Refrigerating, and Air-Conditioning Engineers Inc. (ASHRAE). Powered by Sedona Framework and Powered by Niagara Framework are trademarks of Tridium, Inc. Wireshark® and other product names may be trademarks or registered trademarks of their respective companies.

**BASautomation®**  
Building on BACnet®

**CTRLink®**  
Ethernet for Automation

## Worldwide Locations

### **Contemporary Controls Ltd**

14 Bow Court  
Fletchworth Gate  
Coventry CV5 6SP  
United Kingdom  
+ 44 (0) 24 7641 3786  
[ccl.info@ccontrols.com](mailto:ccl.info@ccontrols.com)  
[www.ccontrols.eu](http://www.ccontrols.eu)

### **Contemporary Controls GmbH**

Fuggerstraße 1 B  
04158 Leipzig, Germany  
+ 49 (0) 341 520359 0  
[ccg.info@ccontrols.com](mailto:ccg.info@ccontrols.com)  
[www.ccontrols.eu](http://www.ccontrols.eu)

### **Contemporary Control Systems, Inc.**

2431 Curtiss Street  
Downers Grove, IL. 60515 USA  
+1 630 963 7070  
[info@ccontrols.com](mailto:info@ccontrols.com)  
[www.ccontrols.com](http://www.ccontrols.com)

### **Contemporary Controls (Suzhou) Co. Ltd**

11 Huoju Road  
Science & Technology Park  
New District, Suzhou  
PR China 215009  
+ 86 512 68095866  
[info@ccontrols.com.cn](mailto:info@ccontrols.com.cn)  
[www.ccontrols.asia](http://www.ccontrols.asia)

