

The ABCs of Using Ethereal 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 Ethereal.

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 Ethereal, 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 **Ethereal** 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 Ethereal, 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, UNIX 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.

Contemporary Control Systems, Inc. • 2431 Curtiss Street • Downers Grove, Illinois 60515 • USA
Telephone 1-630-963-7070 **Fax** 1-630-963-0109 **E-mail** info@ccontrols.com **Web** www.ccontrols.com

Contemporary Controls Ltd • Sovereign Court Two • University of Warwick Science Park •
 Sir William Lyons Road • Coventry CV4 7EZ UK

Telephone +44 (0)24 7641 3786 **Fax** +44 (0)24 7641 3923 **E-mail** info@ccontrols.co.uk **Web** www.ccontrols.eu

Ethereal Packet Sniffing

.The first task in learning how to use Ethereal is to learn how to say the word. Two pronunciations are possible. You can say either *Ether-real*, or say *E-the-re-al*. You can download Ethereal from the web site www.ethereal.com. Ethereal 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 Ethereal 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 bused 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 bused 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 the 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. 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. A specialized switch will probably be required.

Analyzing a Packet Captured by Ethereal

The adjacent screen is the result of capturing BACnet/IP packets over Ethernet. Ethereal 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. Ethereal will continue to capture traffic until requested to stop. Individual frames can be examined while Ethereal 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 Ethereal interpreted the frame. First of all, Ethereal 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. Ethereal 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.

Ethereal 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 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. Port number 47808 is a registered port number given to the BACnet community. Finally, within the wrapper is the BACnet message that begins with its own Type code of 0x81. This Type code refers to Annex J in the ASHRAE 135 BACnet standard.

Annex J defines BACnet over the IP protocol. Eventually, we will learn the BACnet command or response within the message. Ethereal decodes BACnet since it is one of the 750 protocols it supports. Notice that the Ethernet preamble and CRC bytes are not displayed. That is because they are passed from the NIC to the operating system. We know we have a valid frame since the frame was transferred from the NIC.

Not only is Ethereal excellent for troubleshooting networks, it is a great resource in the study of Ethernet and TCP/IP protocols. Ethereal has numerous features such as filtering so that the operator is not flooded with useless and confusing information. The best way to learn how to use this tool is to actually capture packets and study the results against reference books that describe the various protocols.

The screenshot displays the Ethereal interface with the following details:

- Packet List:**

No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.0.0.100	10.0.0.246	BACnet-A	BACnet APDU (Conf)
2	0.003107	10.0.0.246	10.0.0.100	BACnet-A	BACnet APDU (Conf)
- Packet 1 Details (59 bytes on wire, 59 bytes captured):**
 - Ethernet II:** src: 00:07:e9:76:97:0a, dst: 00:50:db: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)
- Raw Capture:**

```

0000  00 50 db aa aa 11 00 07 e9 76 97 0a 08 00 45 00  .P.....V....
0010  00 2d 7d 05 00 00 80 11 a8 61 0a 00 00 04 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 00 c0 00 01 19 55  .....U
  
```

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 benefits 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.

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 no disruption of network traffic being monitored. Another 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 output 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 Ethereal 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

Ethereal Packet Sniffing, Angela Orebaugh, Syngress Publishing Company, 2004