Securing and Analyzing Networks with EtherPeek

EXERCISE: BASELINING NETWORKS
EXERCISE: NETWORK STATISTICS FUNDAMENTALS
EXERCISE: CHARACTERIZING NETWORK TRAFFIC

PROTOCOL FUNDAMENTALS

USER SERVICES
TRANSPORT SERVICES
MEDIA SERVICES
EXERCISE: PROTOCOL DISCUSSION AND IDENTIFICATION

PACKET FUNDAMENTALS – HARDWARE AND PROTOCOL ADDRESSES

HARDWARE ADDRESS
PROTOCOL ADDRESSING
EXERCISE: USING THE NETWORK NEIGHBORHOOD TRACE TO BUILD A NAME TABLE
EXERCISE: CONTINUE BUILDING THE NAME TABLE

NETWORK SECURITY MONITORING

USING ETHERPEEK AS AN INTRUSION DETECTION TOOL
USING ETHERPEEK FOR NETWORK SECURITY ANALYSIS
USING ETHERPEEK TO TEST FIREWALL IMPLEMENTATIONS

USING ETHERPEEK TO TEST THE VALIDITY OF A NETWORK ATTACK

USING ETHERPEEK AS THE ULTIMATE NETWORK PROBLEM SOLVER

EXERCISE: SIMPLE FTP PERFORMANCE ANALYSIS
NETWORK UTILIZATION EXERCISE

PROTOCOL ANALYSIS

USING ETHERPEEK AS AN APPLICATION MONITORING TOOL

ADDITIONAL RECOMMENDATIONS AND RESOURCES

ETHERPEEK IMPLEMENTATION ROADMAP

SUMMARY

Introduction

Toward a Faultless Network

Overview

Today's networks are one of the single most complex aspects of the modern world. Computers from different eras, different vendors and different operating systems must interact in a logical and productive way in order to be useful to their operators. The arena in which these diverse machines interoperate falls to network engineers and IT Professionals to design, build and operate, and also to troubleshoot.

Given the mission critical nature and overwhelming diversity, complexity and speed at which today's networks grow, network support personnel can no longer rely on intuition or on unplugging various devices when attempting to find the source of an anomalous network condition. Now they must rely on sophisticated management tools to ease and expedite their troubleshooting efforts. EtherPeek is an invaluable tool created to assist in these efforts.

The material provided in this document is meant to explain all aspects of EtherPeek and how to optimize use of its feature set to secure, analyze, diagnose and troubleshoot networks. Each major feature of EtherPeek will be introduced and described in detail, and in-depth information provided. There is some overlap between this document and the manual. The manual, which is reproduced in on-line Help within the software, is always the best source for learning the mechanical use of EtherPeek, while this document focuses on when and how to best use EtherPeek. Feel free to consult your manual or on-line Help whenever necessary because there are times when it will be a better reference, particularly if you have no prior experience using EtherPeek.

Goals

The purpose of this document is to provide:

1. An overview of Ethernet network design, history and functionality.
2. A thorough understanding of the EtherPeek User Interface.
3. An explanation of the different uses of EtherPeek and where to place it on a network to solve a specific problem.
4. Instruction on how to collect baseline network data.
5. Mastery of the major features and uses of EtherPeek.
6. Information to give you the ability to design different EtherPeek installations for the various uses of EtherPeek.
7. An understanding of the principals of protocol analysis.

8. The instruction needed to approach most network problems with EtherPeek.

EtherPeek has a wide range of uses, and can be a key tool in solving virtually any network problem. Your ability to creatively use EtherPeek will increase as your familiarity and comfort level with the software grows. You should understand that EtherPeek's primary design function is as a protocol analysis tool, and mastering protocol analysis takes time. Although this document is a great start, literature alone cannot teach the intricacies of even a single protocol, much less all of them. For this reason, this document focuses on the basic elements of IP protocol analysis, the most prevalent networking protocol in use today. A thorough understanding of this protocol will provide the most practical, relevant and immediately useful packet analysis instruction, and understanding the way IP works will translate readily to other protocol-based environments.

Basic Ethernet Network Design and EtherPeek Use

Ethernet Fundamentals

Ethernet is currently the most common LAN technology. It provides simple and cost-effective high-performance networking to all types of computer equipment. Ethernet was developed jointly by Digital Equipment Corporation, Intel Corporation, and Xerox Corporation. Introduced in 1980, a time when a typical computer connection ran at 300 bits per second, Ethernet was distinguished by its high speed (10 million bits per second), its unusual signaling methodology, and by the physical medium on which it ran: a thick, high-quality coaxial cable with a bright yellow braided sheath.

Today, the term Ethernet usually refers to the more modern versions of the original signaling technology. Ethernet now runs on a wide variety of physical media, from the original thick coaxial cable to thin coaxial, to twisted pair wires (shielded and unshielded), to fiber optic cables, radio waves, and many others. Two new direct developments of Ethernet are Fast Ethernet and Gigabit Ethernet. For our purposes, the following discussion applies to all three types of Ethernet (10mb/s, 100mb/s, gigabit).

The smallest unit of information on Ethernet is a packet. Packets are discreet units that are containers for information. By today's standards of information, packets are very small. The Ethernet Frame type controls their size, which is typically 1500 bytes (not K bytes, but raw bytes). Therefore, when computers exchange information over a network, they typically use a stream of multiple packets called a session.

How EtherPeek works

EtherPeek takes advantage of a fundamental characteristic of Ethernet networks. On Ethernet, one computer does not send a packet exclusively to another computer. This is very easy to visualize if you remember that 10baseT networks emulate Thinwire Ethernet. When all workstations are attached to the same strand of Thinwire Ethernet, it is obvious that traffic flows to all the workstations on that wire. When a device sends a packet, each network node on the same network segment receives the packet and examines its destination address to determine if that node should process it. If the packet was intended for a particular computer, that machine captures it, puts it in memory, and then passes it to the next layer of the protocol software for processing.

For example, Device A in Figure 1.1 transmits a packet addressed to Device C. All devices on the network receive a packet that contains Device C's address, but Devices B, D, and E ignore the packet. Only Device C processes the packet further.

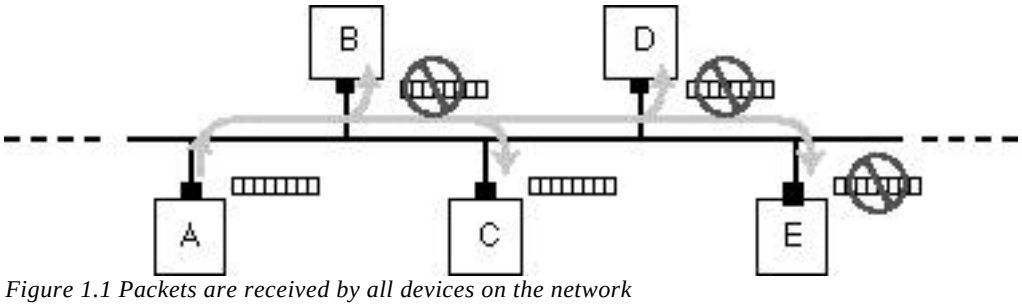


Figure 1.1 Packets are received by all devices on the network

Certain packets contain a special destination address that specifies the packet can have multiple destinations. A broadcast packet has a destination address that allows every node on the network to accept the packet. A multicast packet is similar to a broadcast packet except

that instead of every single node accepting the packet, only certain predefined classes of nodes accept the packet.

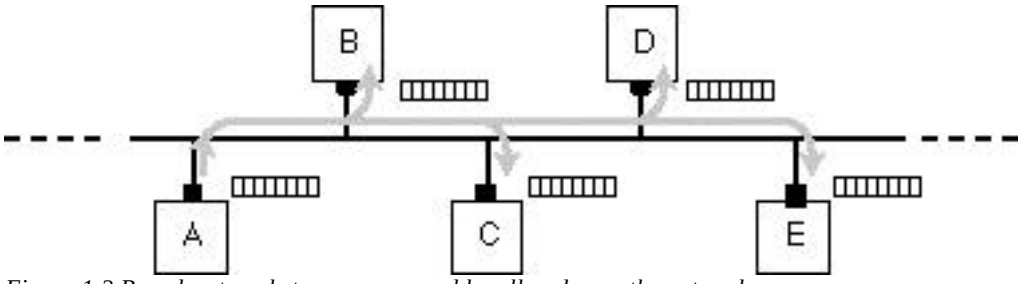


Figure 1.2 Broadcast packets are processed by all nodes on the network

When EtherPeek runs on a workstation, it puts the Ethernet hardware in promiscuous mode, a state that prevents any packet from being ignored. Because the Ethernet hardware is in promiscuous mode, *the workstation running EtherPeek accepts every packet, whether or not it is addressed to that workstation.* To do this, EtherPeek installs itself so that it is the first device to see the packets after they arrive.

For example, if EtherPeek is running on Device D in Figure 1.3 and Device A sends a packet addressed only to Device C, both Device C *and Device D* accept and process the packet. Device C processes the packet normally, by passing it to the next layer of the protocol software. Device D passes the packet directly to EtherPeek so that you can use EtherPeek to analyze traffic patterns and individual packet contents.

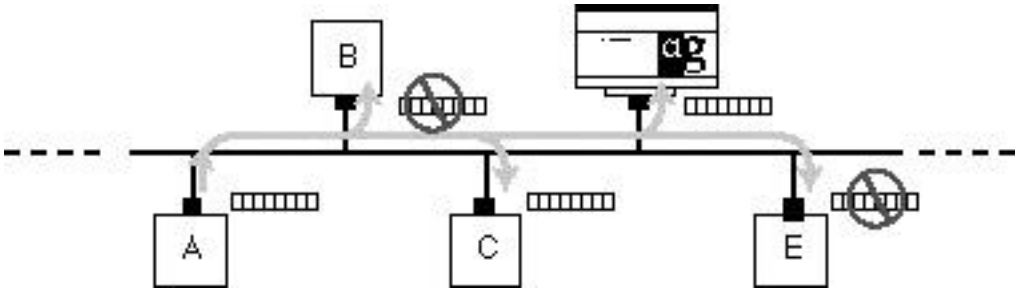


Figure 1.3 EtherPeek accepts all packets ("promiscuous mode")

Another effect of running in promiscuous mode is that EtherPeek does not need a valid protocol address of any type to capture packets. When EtherPeek is in capture mode (EtherPeek has a number of active components that do need a valid protocol address for their various functions), it not only does not need a protocol address of any type, it does not even need a working set of transmit pairs within 10baseT cabling. For this reason, EtherPeek should not be used as an "idiot light" to verify a healthy network connection.

Ethernet networks are constructed using five basic devices: hubs, concentrators, bridges, switches and routers. Each of these interacts with EtherPeek in a very different manner.

EtherPeek & Network Device Interaction

Hubs basically emulate Thinwire Ethernet and are transparent to all network traffic. As long as the machine running EtherPeek has the same transceiver type as the hub, EtherPeek can be placed anywhere within the hub to capture all traffic.

Concentrators are giant hubs with interchangeable modules and may be segmented or segmentable. Concentrators are typically used in large enterprise environments that need either various media types in a single network closet or have a large number of home runs that need to be supported. Stackable hubs function in a similar manner and may be segmented via manual jumpers or a Thinwire Ethernet backplane. These segments are totally separate

Ethernet networks and rely on other devices such as routers to allow traffic flow. Concentrators may contain a single Ethernet segment or multiple independent segments. In passive backplane concentrators, each Ethernet module is totally isolated from the others. If you are using EtherPeek within a Concentrator or managed hub stack, make certain that you are on the correct segment in order to capture all of the network traffic you would like to see.

This Asante Concentrator functions as a single large hub, but may contain two *totally separate* Ethernet networks.

Bridges are another logical, intelligent device used to partition Ethernet networks. Though still seen on some networks, they have fallen out of favor, and have largely been replaced by switches. Switches are multi-port bridges and function in a similar logical manner. Both bridges and switches are transparent to multicast and broadcast packets, but do not forward any other packets. This can lead to some very confusing interaction with EtherPeek. Additionally, many intelligent switches support VLANs or virtual LANs. This also adds to the complexity of traffic seen by EtherPeek. Most VLAN implementations are also capable of forwarding only selected multicast and broadcast traffic.

These Cisco Catalyst Switches support extremely complex VLANs and can *virtually* distribute routed Ethernet LANs just as easily as patch panels can distribute them.

There are several ways to use EtherPeek in switched networks. The first is to run multiple copies of EtherPeek on each segment that is being studied. Similarly, you can run EtherHelp on each segment. The second is to run EtherPeek directly to the node being studied. The third is to use a Farallon EtherWave adapter and splice into the switched run. EtherWave adapters are extremely fast repeaters that forward all packets without re-timing or checking their validity. For this reason, they are an excellent solution for assisting EtherPeek, particularly on notebook computers. Similar to this, you can use a small Fast Ethernet hub with four of five ports in a fast Ethernet environment. Fourth, if the switch supports it, enable port mirroring. This is another excellent option available on managed switches and switches that support VLANs. In this situation, a specific port is either designated by the product, enabled via software, or both, to mirror all traffic on the switched port being studied.

The final logical device is a router. Routers come in all shapes, sizes, and media types. It is important to remember that software-based routers are also common. All major servers, including Solaris, NetWare, NT Server, and Apple have various software-based router products for all three major protocols (IP, IPX, and AppleTalk). It is more likely that you

will experience a problem with a misconfigured software-based router than a hardware-only router.

Routers are completely opaque to EtherPeek. No traffic passes through a router unless the router is specifically configured to forward it. Routers are also responsible for generating most of the network's characteristics.

Firewalls are a new type of intelligent network device. They are basically one-way routers. Most firewalls are software routers running on fast servers with specific feature sets to control in-bound and outbound network traffic. Firewalls typically function in an asynchronous format.

Ethernet Network Design Fundamentals

Ethernet as a technology was designed for networks with periods of short bursts of activity. It is not designed for connecting multiple nodes (more than 2) together for long periods of sustained bandwidth-consuming activity. Token-based network technology is better suited for that application. Through the years, Ethernet's popularity has shaped many of the applications that we currently use. In other words, Ethernet's continued use for twenty years has resulted in applications that run well on Ethernet networks. Today's Intranets are such a technology.

C.1989

Most networks built before 1989 rely heavily on bridges. Many flat-bridged networks were built in the mid-1980s. At the time, router ports were very costly and many network designers believed that they were an unnecessary expense. It was not unusual to have networks with as many as two thousand nodes per router interface! The introduction of broadcast-based protocols made bridged networks a thing of the past.

C.1992

Most networks built in the early 1990s were router based and focused on limiting traffic to servers. In other words, servers really ruled the day, and networks were built around them. Most traffic was seen between nodes and the server, and all traffic was typically confined to one side of a router interface. This was also due to the fact that most routers of this period were relatively slow and performed poorly when heavily loaded.

C.1995

1995 is best described as the year of Internet awakening. Internet growth exploded, as did the traffic to and from it. For the first time, traffic on many networks was destined for places other than internal servers. This forced a change in the way we think about network design.

The other major influence was the introduction of switches. 1995 saw many new designs that depended heavily on flat switching. Just as with bridged networks, many of these designs failed under the load of multiple protocols.

C.1998

Today's networks are largely hybrids and are best described as meshed networks. Typically, loads are balanced between switches and routers, with no more than 250 per router interface. All ports may have a separate dedicated switched port.

Today's networks are becoming significantly more focused on application support. Networks are now being viewed as a transparent service that should simply work and not get in the way of the user. Most of today's business-critical applications are strongly network dependent, and so networks are now being managed to insure support for those applications. Toward this end, most of today's networks can now be divided into two logical groups. Those that are arranged around workers in a workgroup setting, and those arranged around processes that are so spread out that they require a meshed network.

Workgroup Networks

Workgroup networks tend to be organized around independent departments or are artifacts of networks that were built around servers. Workgroup networks also tend to rely heavily on routers and are therefore typically slower, though more efficient, than meshed networks. Workgroup networks are easy to secure on the network end, but because they are often spread throughout a company, are hard to secure physically.

Meshed Networks

Meshed networks typically share two traits. First, they make use of switched hubs at virtually every level of the network; and, secondly, they frequently make use of server farms located in network operation centers or NOCs. Many meshed networks rely on VLANs as a way to create workgroups within the environment. Building VLANs in this environment is a double-edged sword: in dynamic corporations they can save many trips to the wiring closet, but they can also generate furious amounts of network traffic.

Meshed networks can also be the result of poor planning. Be wary of sites that claim to have meshed networks but don't have physical or logical wiring maps or diagrams of the network. It is possible to build a meshed network that relies on the spanning tree algorithm that has no design whatsoever. This is because the spanning tree algorithm found in higher-end switches automatically cuts off redundant routers. It is intended as a way to build redundant paths, but can also lead to networks that are impossible to troubleshoot.

Building EtherPeek Installations

There are three main ways in which to deploy EtherPeek when attempting to address network issues. You can use it on demand by installing and removing it from a node on a specific physical network segment when experiencing a problem. You can build a large, fast multi-NIC box to install in a NOC or server farm and run EtherPeek from that box. Or, you can install it on a fast laptop computer as a portable network analyzer.

For Immediate, Reactive Response to Network Problems

Installing EtherPeek at multiple points on your network on an as-needed basis is not always recommended. The main problem with this method is that you can get very different results on different installations. It therefore becomes impossible to establish a baseline for the various network segments and determine what things should look like when they are working properly.

There are times when this approach should be considered, however. If you are having problems with a critical machine or server and have exhausted all other avenues of problem solving, then installing EtherPeek directly on the machine to determine exactly what is happening can be an excellent approach. No other method of troubleshooting "gets into the machine's head" as EtherPeek can. It is still necessary to have some idea of what the correct situation should look like, so there are times when this approach can fail.

Enterprise Install

Building an enterprise install of EtherPeek in a NOC is one of the best uses of EtherPeek. In this situation, you dedicate a machine to EtherPeek use and typically install multiple NICs for various network segments. For example, install EtherPeek on a Pentium-based machine with a large monitor. Some installations build a "network console" machine with an SNMP console, VLAN management software and serial cables to various routers. It is a good idea to use the largest possible monitor available for this type of installation.

Please note that EtherPeek cannot use more than one NIC at a time for capture, nor can multiple copies of EtherPeek be running at the same time on the same machine. It is useful to have multiple NICs running so that you can easily switch between LAN segments merely by switching adapters in EtherPeek. It is also useful in that you can generate traffic via one program (AGNetTools) with one NIC on one side of a network device (such as a router or firewall) and simultaneously watch with EtherPeek what happens on the other side of the device via another NIC.

You should also be aware that most operating systems do not like to be multi-homed within one addressing scheme unless it is their "native" protocol. Even then, it's not a very good idea. If you want a stable machine, it is unwise to do this. Installing multiple NICs is fine if you never install multiple NICs into one LAN segment and run the same protocol on each. In other words, if you want multiple NICs on one LAN, run a different protocol through each NIC or use one of the NICs for listening only.

Portable Network Analyzer

Building a laptop-based EtherPeek installation is the most versatile choice. This is an excellent installation for networks without a central NOC or for solving problems on networks spread over a corporate or school campus.

A Pentium/166 or faster laptop with a large active matrix screen is a good minimum platform for this installation. You also want to have a supported 10/100 PC card so that you can easily diagnose networks with different media speeds.

Another very useful NIC for portable analysis is the Farallon EtherWave. This card supports returning error statistics and, more importantly, has an external transceiver that is a very fast repeater. This allows EtherPeek to be inserted into any network, including sites where no available ports exist. With this device, data can be captured from a dedicated switched port where no port mirroring exists. The EtherWave product determines if a crossover is needed and automatically reformats itself in such a situation. It also is fully loaded with diagnostic lights, so it serves as a useful tool for troubleshooting bad physical installations. Note, however, that it is an Ethernet 10baseT-only product.

For Fast Ethernet, a small, 4 port hub can be particularly useful in densely-loaded networks where a spare port may not be available. These are also useful because they eliminate tracing a run back to a closet.

The Many Faces of EtherPeek

Now that we have discussed the various EtherPeek installations, we need to talk about the ways to use the software to assist in aspects of network management. You may combine any of EtherPeek's tool functions with the installation types, but some are better paired with others.

EtherPeek is a very versatile product. We will examine using EtherPeek as a Proactive Network Analysis tool, as a Network Reporting Tool (for network macroanalysis), as an Intrusion Detection and Security Tool, as an Application Monitoring Tool and, finally, as the ultimate Network Problem Solver (for network microanalysis).

You should be aware that there are a number of ways to obtain additional information as you work through this document. The best way is to consult the EtherPeek manual. The next is to use EtherPeek's built in on-line Help. Finally, EtherPeek has an advanced protocol reference encyclopedia built into the program that is available via a number of windows.

Using EtherPeek as a Proactive Analysis Tool

The tutorial portion of this document begins by examining EtherPeek in its use as a Proactive Network Analysis Tool. The goals for this section are:

- To get to know the EtherPeek User Interface
- To learn about the various default settings that need to be set in order to ensure valid capture data
- To ascertain that all data sought is in fact collected

For EtherPeek to be most effective as a troubleshooting tool, it needs a set of baseline capture files that can be used in comparison with problem situations. These baseline files should be generated regularly and at every critical point on a network. This typically means on the outbound leg of the network's Internet connection, inside a firewall if one is present, on segments with critical servers, and on heavily utilized segments such as backbones. Typically, a baseline file is created by capturing for five to ten minutes and at different times of the day. These files will become a record of normal network activity, so the network must be monitored frequently enough to determine what normal activity looks like.

Baseline files should be stored for later comparison, just as backup files are stored for a period of months. If these files are not of a sensitive nature, they can be stored locally, as backup files may be stored. Removable media is a good storage choice since it allows fast access to the files without having to go through a restore process.

Before starting a promiscuous capture session, it is recommended that you check the EtherPeek UI in a specific pattern. . Move to the "Capture" menu and pull all the way down to the bottom. Starting with "Network Speed", verify that it is set for the correct type of Ethernet with which you are working. If you are working with a 10/100 card, it is always best to set it manually. (EtherPeek only checks the card when it initializes itself, therefore if you move from one network type to another without relaunching EtherPeek, you will be using the wrong settings.)

Next check the "Capture Buffer." This is a critical setting. It determines how much memory EtherPeek has to store packets until it has to do something with them. There is no upper limit to the amount of RAM EtherPeek can make use of; so given the relatively low prices on memory, it's a good idea to buy as much as you can afford when building your EtherPeek installation. The next thing to look at is what you want EtherPeek to do with the packets once the memory buffer is full. The default setting is to stop capture when the buffer is full.

The default buffer is about 2MB. You can change this to continuously capture packets and either flush the buffer completely and resume capturing when it fills, or save the packets to a file and then resume capture.

Next, check the network interface from which EtherPeek will capture. Make certain that it is the one you want to use and that it is on the correct LAN segment. (There's nothing like shooting a great roll of pictures only to discover that you haven't loaded the film!) This is done in the "Select Adapter" dialog box. It is a good idea on EtherPeek Enterprise installations to post a note physically on the box with the slot number, NIC type and MAC address of the NIC. This can be used to verify with the "Select Adapter" dialog box that you have, in fact, selected the adapter you wanted.

The last check before moving on to the next menu is to make sure that no Triggers or Filters are selected in the remaining two dialog boxes under the "Capture" menu.

The next menu is "Send." All items under the Send menu are off unless you specifically enable them. Make certain that "Echo on Capture" is *unchecked*.

The next menu is "Options." These are primarily the display options and are not critical for a proactive capture session since the main use of this session is just to gather data for long term reference use.

In keeping with building good habits, we will examine a number of these menus, however. Starting with the bottom, you may want to make certain that no "Notifications" are set. This typically means that you have loaded a pager gateway on the system, as would be typical in an enterprise installation.

If "Scroll During Capture" is enabled, you can get a sense of how active the network is. However, note that EtherPeek wants to put all its clock cycles into capturing and processing packets, so scrolling may slow statistics updates when enabled.

For this exercise, we will not worry about any of the display options and so we will move onto the "Statistics" menu. These are simply windows that can display various other pieces of information.

Finally, you may want to turn off all plug-ins during a proactive capture session. There is really no reason to have them on, since they are fully functional as post-capture analysis tools. No other items under the "Special" or "Window" menus need attention at this point.

You have now set up EtherPeek for a Proactive Capture Session. Select the Start Capture button from the main window and you will begin to see traffic.

Exercise: Configuring EtherPeek for a Proactive Capture Session

Set-up EtherPeek for a Proactive Capture Session, as detailed above, and verify that it has written data to the hard drive.

Using EtherPeek as a Network Reporting Tool

Now that we have gained some basic exposure to the various features of the EtherPeek UI, we will begin focusing on more advanced features and ways to use EtherPeek. EtherPeek can function as an excellent network statistics analysis and reporting tool. Using EtherPeek in this manner is one step above the simple "default" capture that we developed in the first section, and it moves us in the direction of being able to use EtherPeek as a troubleshooting tool. The goals for this section are:

- To understand the relatedness of the various facets of the EtherPeek UI

- To insure that all data sought is in fact collected
- To insure that all data sought is in fact valid data
- To learn to effectively use packet slicing
- To learn to use the name table
- To extend the functionality of EtherPeek by using external tools

The basic premise for using EtherPeek as a network reporting tool is to let it run for a period of time, capturing all or a subset of network traffic, and then either examine the data within EtherPeek or export it to another program for further analysis or enhanced display technology.

In this section, we are not interested in examining what is contained within the packets payload. We are interested only in the type of protocol being used, where it is going, where it has come from, and what it is doing. In order words, we are interested in the specifics of the packet's header.

One way to improve the capture performance of EtherPeek is by using packet slicing. This feature tells EtherPeek to retain only the first user-definable portion of the packet. By saving only a portion of each packet, the capture buffer fills more slowly and EtherPeek has to write to disk less frequently. Since EtherPeek pauses capturing when it writes to disks, the end result of properly using packet slicking is a more accurate picture of overall network traffic.

It is important to note that if you fail to save enough of each packet, you will not be able to run certain post-capture functions such as plug-ins, and you may fail to generate valid network statistics. Unless you know exactly what you are looking for in a packet, you should never set the slice size below 64 bytes. If you want to enable use of plug-ins, you will need to set the size to a higher level, typically 120 bytes. Please be aware that as future plug-ins are released, this may change.

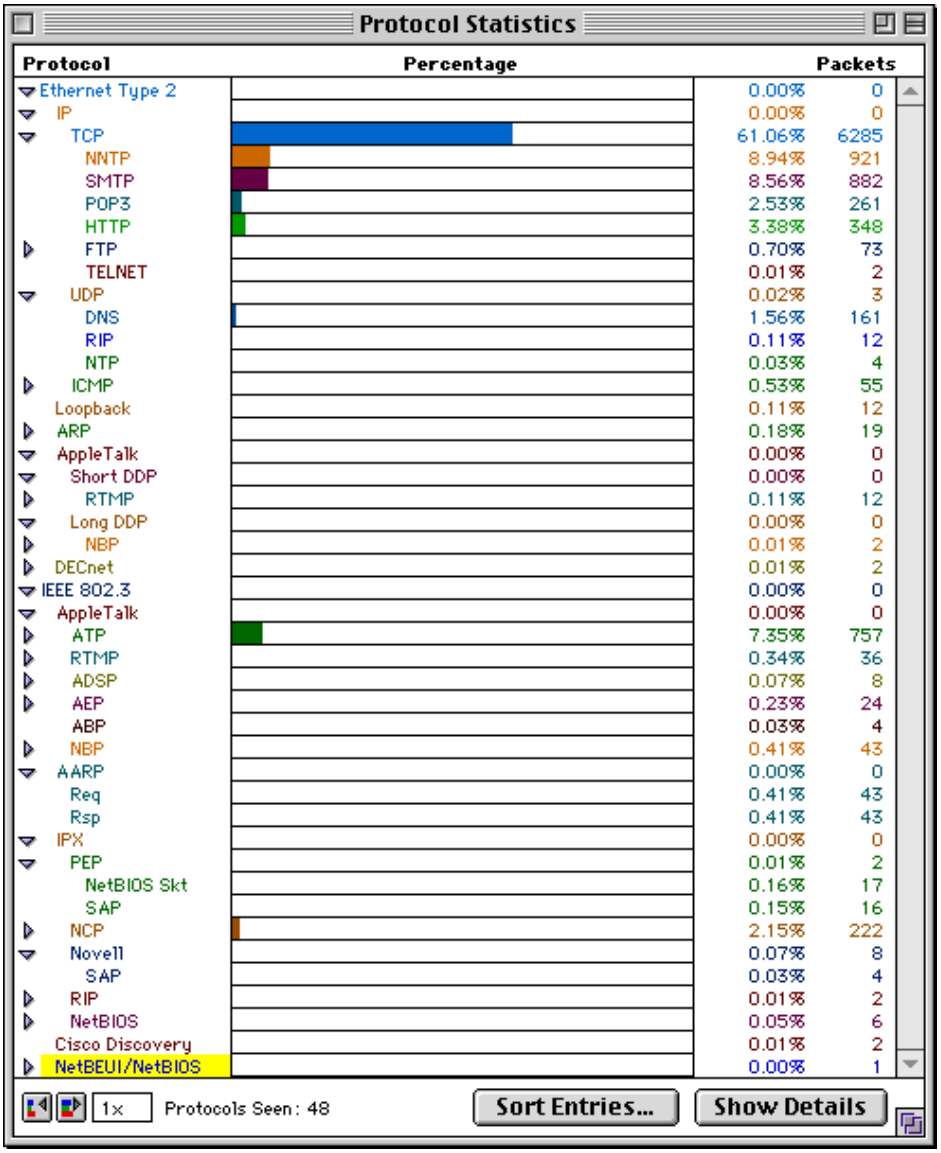
It is important to note that EtherPeek's ability to report error packets is directly dependent on the type of NIC installed within the host running EtherPeek and the driver version of that card. Typically you can get error information from SNMP-based devices that report these same statistics, or from dedicated intelligent cable test units. Both of these compliment the capabilities of EtherPeek without overlapping functionality.

Ethernet Frame Types

Ethernet was developed by a consortium from DEC, Intel, Xerox (DIX) in the 1970s and 1980s and was originally considered “proprietary” but spread rapidly due to its comparatively low cost. Ethernet has always had detractors, but it is easy and inexpensive to implement. The standard grew over the years to include fast speed (> 10 Mb) and more cable types. The original standard has now been used as the basis for two new technology advancements, Fast Ethernet and Gigabit Ethernet. These technologies are again proving that they can outdistance their competitors by being easier and more inexpensive to deploy.

802.3 was developed by IEEE as part of the OSI movement in the 1980s. The goal was to build a link standard that was as media-independent as possible.

The diagram below is an example Protocol Statistics window. Locate the various frame types in the window and notice that some protocols appear in 2 places. How is this possible?



Exercise: External Graphing

Set up EtherPeek to capture all packets with a slice size set to 120 bytes and then export the data. Import the data into Excel and generate a 3D pie chart.

Exercise: Summary Statistics Use

Use the "Summary Statistics" window to generate five network profiles at one-minute intervals over 5 minutes. Start capture and take a snapshot of the network. Then, stop capture, restart, and take a second snapshot. The two Summary Statistics columns displayed will have separate start times and statistics reflecting the traffic captured during their respective capture sessions. Repeat at stated intervals to get a sense of network changes in key traffic elements in a short period of time.

Exercise: Baselining Networks

Determine what characteristics you want to baseline on your network. How are these related to the network design?

Exercise: Network Statistics Fundamentals

Start a live capture session. Then, answer these questions.

What was the average utilization during this capture? (Network Statistics window)

How many nodes are sending packets during this capture? (Source Statistics window)

How many addresses are receiving packets during this capture? (Destination Statistics window)

How many Ethernet broadcasts are being sent during this capture? (Source Statistics window)

How many multicasts are being sent during this capture? (Source Statistics window)

How many protocols are seen? (Protocol Statistics window)

Exercise: Characterizing Network Traffic

Statistics windows provide detailed analysis of communicating devices, communication partners, and protocols and subprotocols in use on the wire.

Open the Dual Statistics window within EtherPeek to display a list of all communicating nodes on the network during a live capture session. The sort button allows you to order your view of network devices by packets received, packets sent, or in alphabetical order. By choosing to sort by packets sent, you can readily see the top contributors to network utilization.

The hierarchical structure of the Protocol Statistics window provides fine detail about protocol and subprotocol use on the wire. Double-click any of the protocol names to discover which network devices are responsible for the protocol traffic chosen.

Protocol Fundamentals

Protocols are the basic language of computer communication. Every protocol shares a number of traits. By developing a conceptual model of what is needed to communicate, and the traits that underlie these needs, it is possible to develop an intuitive sense that will greatly aid in learning protocol analysis.

Computers, like humans, have to overcome language and cultural barriers. These language barriers are different protocols. The easiest way to overcome this is to insure that all computers on the network use the same protocol for all communication and the same network type. This is one of the reasons why many networks are turning off all protocols except IP. In the case where the listener and the speaker have different cultural make-ups, some translation must be provided so that a common communication is possible. This is a standard function of network gateways. Gateways are typically used when legacy systems simply cannot be made to speak another protocol, or when it would cost too much in time or materials to convert critical systems.

A node must share the available bandwidth of the network with all other nodes. Efficient communication systems guarantee that only one individual speaks at a time. There must be an orderly way to take control of the media and resolve conflicts between simultaneous speakers. This is critical to understanding protocol analysis, but it not actually a part of protocols. The physical type of network medium determines this. Ethernet uses Carrier Sense Multiple Access - Collision Detection. Other types of communications systems are token based (TokenRing and FDDI), for example.

All protocols must have a way of identifying individuals. It is impossible to carry on an orderly conversation without knowing whom to address. This is the role of resolving protocol addresses.

All protocols must have a way to structure their communication. In language, structure is important because it provides a way for speaker and listener to be sure that what is being received is the same as what's being sent. By structuring communication, the possibility of misunderstanding decreases. This is the role of packet formats and specifications. You will need a detailed reference for each protocol you want to learn.

All protocols must have a way to contact the network entity with which a user chooses to communicate. The user must be able to communicate at all times, even under changing conditions. You must also know how to approach the entity with which you wish to communicate, how to begin the interaction, how to transfer the information, and how to end the communication. This is the role of protocol sub-types and may also include formats designed by the applications that lie above the protocols themselves. In some cases, EtherPeek will not be able to fully decode a packet because there are multiple correct answers that depend on application-specific uses.

All protocols must have a way to specify how long you can talk and how fast. Time is limited and you want to make the best use of the time that you are allowed to communicate. If you go on too long, you waste the time available to talk for others. Also, if you have to resend your communication because of transmission errors, it's better to have a reasonably-sized chunk to resend. This information is typically determined by applications above the protocols, but this varies with the protocol type. It may also vary within a protocol, but be determined by the application speaking. In other words, some protocols have multiple ways to handle this communication.

User Services

Application	Maps user meanings to actions	**User Services**
Presentation	Maps actions to code, handles data representation	**User Services**
Session	Resource and connection management	**User/Transport Services**
Transport	Manages data delivery process to process through internet	**Transport Services**
Network	Delivers data process to process through internet	**Transport/Media Services**
Data Link	Manages data delivery node to node on a single link	**Media Services**
Physical	Delivers data node to node on a single link	**Media Services**

Interface between human and machine

The user must have a clear and consistent way of requesting services and getting feedback from the machine. The user also needs an easy, intuitive way to locate and engage the services that the network provides, such as printers, file servers, computer servers, access gateways, etc.

File access and management

The user needs to be able to use information regardless of where it is located and on which system it runs.

Handle differences in system characteristics

The protocol must have a way of coping with the fact that the user and the service they are using are on different platforms, have different communication needs, and use different file formats.

Transport Services

Application	Maps user meanings to actions	**User Services**
Presentation	Maps actions to code, handles data representation	**User Services**
Session	Resource and connection management	**User/Transport Services**
Transport	Manages data delivery process to process through internet	**Transport Services**
Network	Delivers data process to process through internet	**Transport/Media Services**
Data Link	Manages data delivery node to node on a single link	**Media Services**
Physical	Delivers data node to node on a single link	**Media Services**

Get the bits from process to process

Reliably, swiftly and consistently transfer the information from source to destination. Perform this function under changing conditions in dynamic environments.

Resource and connection management

To allocate resources and establish the connection. Every device has a limited set of resources. It is important for the protocol system to use those resources efficiently, and be able to disengage automatically when it is no longer reasonable to communicate.

Managing reliability/performance tradeoffs

Typically a high performance, yet somewhat unreliable delivery method is managed by another, complimentary protocol that is constructed to provide the best performance for a single kind of connection type, whether stream oriented, transaction oriented or display oriented.

Cope with variations imposed by physical network

Different networks have different requirements for frame sizes, and the delay time can vary considerably from network type to another. Connection timers can be adjusted accordingly and packets can be fragmented to fit on networks with shorter maximum packet lengths. AppleTalk typically does not provide these services.

Provide routing services

Because the communication network may consist of many interlocking pieces, the transport mechanism must have a way of guiding the information from source to destination. The route may need to change as the internet structure varies over time, so the connection components must learn new (better) routes and forget about old routes that are no longer optimal.

Provide name services
Users find it much easier to cope with alphanumeric names for services and network areas, while the devices prefer numerical location information. Name services resolve these two systems. In AppleTalk, this is also necessary to provide consistent services, since addresses are dynamic. Names are only changed deliberately by someone with the proper software.

Media Services

Application	Maps user meanings to actions	**User Services**
Presentation	Maps actions to code, handles data representation	**User Services**
Session	Resource and connection management	**User/Transport Services**
Transport	Manages data delivery process to process through internet	**Transport Services**
Network	Delivers data process to process through internet	**Transport/Media Services**
Data Link	Manages data delivery node to node on a single link	**Media Services**
Physical	Delivers data node to node on a single link	**Media Services**

Put information into legitimate network packets
Every network type (Ethernet, Token Ring, etc.,) has a particular frame format. The information from higher protocols must be assembled into that format and be placed on the network media in the proper way.

Gain network access
Once the information is assembled and formatted properly, the system must take control of the media in order to send the information. This must be done according to the media access rules of the network, whether collision avoidance, collision detection, or token passing.

Signaling and receiving
After the information has been assembled and the system has control of the network media, it must encode the information in the format required by the network. This involves creating a voltage wave of a particular shape that is transmitted down the wire. It must also receive and decode signaling from other systems.

Physical error detection
The signaling hardware must reject or correct errors that come from the media. Processing a packet costs processor cycles and it is important not to waste that CPU time for meaningless packets.

There are two types of addresses that can be used to identify nodes within Ethernet networks and therefore within EtherPeek: Hardware Addresses and Protocol Addresses.

Hardware Address

Hardware Addressing is used to identify the host at the base level of the OSI stack (physical) and is actually burned into the Ethernet Network Interface Card (NIC) by the manufacturer. This is commonly referred to as the MAC address.

Each manufacturer establishes their own range of codes and registers them with the IEEE. This address remains with the NIC throughout its lifetime. The address is made up of 6 hexadecimal octets. The first three octets always identify the manufacturer of the NIC. These ID octets are constantly being updated as new companies enter the market and old address spaces are used up. For this reason, we recommend you check for updates periodically by visiting <http://www.cavebear.com>.

It is important to remember that, if the NIC is changed within a host, the host's Ethernet address will change and this may force all protocol addresses to change. The one exception to this is Cisco routers. The MAC address for most Cisco routers will not change if the Ethernet interfaces are changed.

The hardware address is independent of where it is placed in an internet.

The hardware address is used by bridges and switches to determine whether to pass a packet to another segment

Ethernet Hardware Address Example

48 Bit Hardware address - 6 bytes - 12 Hex digits
Expressed as: 00:00:81:2C:44:09
First 3 bytes indicate manufacturer
00:00:81 indicates Synoptics Communications

Protocol Addressing

The protocol address is established when the host initializes its protocol stack. It is stored in RAM only when the device is powered and remains with the device while it is running. This address is *dependent* on placement of the host within the internet. The protocol address is used by routers to determine whether to pass a packet to another network.

Example: IP Protocol Address Structure

32 bit protocol address - 4 bytes
number of bits for network, subnet and host portion of address varies
network expressed as 192.216.124.0/24
24 bits allocated for network and subnet
8 bits allocated for host
node expressed as 192.216.124.35
4 octets - decimal representations of 8 bit number

When nodes on a network communicate, they must translate between their Protocol Address and their Ethernet Address. Each protocol uses a separate way of accomplishing this task.

IP uses Address Resolution Protocol (ARP). Each node keeps an ARP cache to map between addressing systems. If a mapping is unknown, the transmitting node will broadcast an ARP

request for the needed address. The intended receiver, if it is present, will respond with an ARP Response. ARP Cache entries eventually age out and are removed from the cache.

EtherPeek can be instructed to translate the names in the various windows into any of the other name types. EtherPeek uses its own name table to make this translation. The default name table includes a number of Ethernet broadcast protocol types, and the program includes a number of ways that you can use to add entries to the Name Table.

- You can manually add names on a case by case basis.
- You can have EtherPeek automatically resolve IP addresses.
- You can import the current MAC address vendor ID file.

MAC Address Management

In order to fully understand and quickly diagnose network anomalies, it is critical to develop a system of recording and tracking all the MAC addresses on your network. MAC addresses only change when you change NICs in the host. For this reason, they are the most accurate way to identify a device on a network. If MAC addresses are not tracked, it may not be possible to correctly determine what device is actually being viewed within EtherPeek.

Most NICs now ship with the MAC address as their serial number. This may be identified on the outside of the box, on the NIC itself and/or on the shipping receipt for the NIC. You should go to great lengths to develop an accurate database of this information. Within Windows 95 and Windows NT, you can directly query the host to find out the MAC address of each NIC. EtherPeek with report this information in the “Network Interface” window. There are also a number of shareware programs that can be used to record the MAC address to a tab-delimited file on a floppy.

Some older networks still run bootP, which is a good source of this information.

Name Table Management

When EtherPeek for Windows is installed, a default name table is added to the EtherPeek “NAMES” directory. Should your name table become corrupted, you can use this file to restore the name table to its original settings. The Names File used by EtherPeek is stored in the “CONFIG” directory. You should not access this file directly. You can select different names files by using the right mouse button within the “Names” dialog box and choosing the Import command.

You should get in the habit of generating a separate Names File for each LAN on which you use EtherPeek. If you want to use names based on protocol addresses, you may need to generate this information each time you use EtherPeek.

Building the Name Table

Manually building the Name Table is the most accurate way to add names. This is particularly true in networks using dynamic addressing such as AppleTalk or DHCP. The names in these networks may change rapidly. You should consult with the network manager to determine what timeouts have been set for these services. This will also mean that the Name Table within EtherPeek will need to be flushed regularly.

Resolve Names Function

The EtherPeek Resolve Names function works for IP if the network has domain name services present.

Once names are resolved, they are automatically added to the name table, and names replace logical address entries for devices in the main window and in all node statistics windows.

The EtherPeek manual and on-line Help provide step-by-step instructions for resolving names.

Network Security Monitoring

EtherPeek is an excellent security analysis tool. It is one of the few tools that can actively check firewall and router configurations and verify that the configuration is correct and that the firewall or router is functioning as intended.

Typical network designs for Internet commerce now involve at least two firewalls and two levels of web servers or transaction servers. The outside web servers are considered untested and must authenticate through a second firewall. The first firewall is used to build an authentication network and to protect those hosts and limit their vulnerability. Remote access for administrators is provided via virtual private network services, and authentication on a detected secure host. This network uses a perimeter defense model and relies on tight control of the network. No modems, routers or other devices that breach the perimeter can be set up inside the network.

Using EtherPeek as an Intrusion Detection Tool

Most networks today are in the process of increasing their overall level of security and some are implementing sophisticated network designs offering multiple levels of security, redundancy and intrusion detection. The majority of local area networks today consist of a flat-switched Ethernet network connected through a firewall to a router and then to the Internet. In some cases the firewall and router may be combined into a single product. EtherPeek can be used as a separate undetectable (passive network analysis based) intrusion detection system. Typically, it is run on a separate host and records information to its own hard drive. The typical interface for this system would be a pager running on the same box.

To implement EtherPeek in this manner, you will want to build an enterprise install with as much RAM as you can afford, and a very large HD that is capable of storing at least a week's worth of data. You will not necessarily need a large monitor, as the typical UI for EtherPeek as an intrusion detection tool is a pager or the log file. Be careful when setting up the paging characteristics of EtherPeek via plug-ins, because it is possible to generate a flood of pages and back up your pager gateway, or the paging system itself.

You should also carefully consider where you want to install EtherPeek as an Intrusion Detection tool. If you want to observe and record all hostile network activity, then you should install EtherPeek on the outside of your firewall and set it to record all attack information only to the log file. This is an excellent way to have an additional record of attacks to supplement the log file from your firewall itself. Firewalls vary greatly in the degree to which they actively log information. Some flood the log files with arcane information, while some have no logging capability whatsoever. By adding EtherPeek to your overall security plan, you can supplement your logging activities or customize them in a more accessible format.

You may want to install EtherPeek just inside your firewall to monitor what is actually getting through. In this case, you probably want everything set as a pager alert. This is an excellent way of keeping tabs on your firewall to insure that it has not been mis-configured, breached or become outdated.

Finally, you may want to set up a monitoring station to observer all traffic to and from your servers. This is particularly true if you are running a server farm, and have a great concentration of servers within one protected subnet. Repeated studies have shown that more than 90% of all successful attacks come from *within* networks from trusted users.

Using EtherPeek for Network Security Analysis

EtherPeek can also be used as security analysis tool. Virtually all network file servers, mail systems, and databases have a default installation that allows for clear text passwords. In

today's world, it is critical to disable these defaults and insure that at least the login passwords to these systems is encrypted. EtherPeek can be used to examine these various logins to determine whether they are correctly configured and do, in fact, encrypt the password fields.

Using EtherPeek to Test Firewall Implementations

Firewalls are extremely difficult to configure correctly; particularly if used in addition to access lists (filters) on routers. EtherPeek can be used to probe firewalls, hosts, and routers by use of the "Tools" menu and, more importantly, can directly observe the results of probing.

To use EtherPeek in this manner, you want to "inject" various queries and observe the results on one or both sides of the router or firewall. EtherPeek has a "Tools" window that serves as a way to launch other applications to supplement the program's native capabilities. It is a good idea to add AGNetTools, which ships with EtherPeek, as a base set of external tools to access through this window. AGNetTools can be used to generate port scans and service scans. By selecting the same NIC for both EtherPeek and AGNetTools, you can simultaneously probe and observe responses on one network. To do this, simply start a capture, run a scan and record the results. The results should then be compared with what is expected.

A port scan generates a ping for each port within TCP and UDP. It is commonly used to determine what services are running on a UNIX workstation or server. With the introduction of Intranet technology, it is now common to run various IP services on a wide range of hosts and operating systems. By port scanning a device, you can find misconfigured or unknown services. The wide range of devices now running web services (port 80) may surprise you. Among the more unusual devices on the market, APC now makes a power strip with a built in web server!

A service scan is used to look for a single TCP or UDP port within a range of IP addresses. It is a good idea to conduct a service scan monthly for FTP, WWW, and Telnet services within all your subnets. Users frequently introduce new services with no security whatsoever that can easily be used to compromise other devices on your network.

Firewalls are typically configured to allow certain services to be "advertised" on the outside interface. All other services and IP addresses should be blocked. Many firewalls can be configured for network address translation (NAT) and/or port address translation (PAT).

Another way to test a router or firewall is with a two NIC box, using both EtherPeek and AGNetTools at the same time. In this configuration, you would "inject" a probe on the outside of a firewall with AGNetTools and "listen" on the inside with EtherPeek. Make certain that you have correctly configured both products with separate NICs and that the NICs are attached to the intended LANs or network interfaces of the firewall.

Using EtherPeek to Test the Validity of a Network Attack

EtherPeek can also be used to test for various denial of service attacks or to determine exactly what is causing specific network services to fail. To use EtherPeek in this manner, you need to capture what you believe to be the offending packet and then resend that packet in a test setting and observe the results.

There are several ways in which you can attempt to capture the "broken" packet. One way is to run a full capture and then discard packets that are known to be "good." You can then resend the rest of the packets and observe the results. By doing this in a series of steps, you can eventually discard "good" packets until you are left with the "broken" packet. Another way to do this is by use of the triggers. Triggers can be set to start capturing at a specific time or upon the occurrence of a packet transmission that matches predefined filter criteria if you know the network service has a specific port that seems to be vulnerable.

Using EtherPeek as the Ultimate Network Problem Solver

When EtherPeek is first installed for use as a proactive analysis tool, the default setting is to capture all network traffic. However, as problems occur, it will often speed the search for the source of the problem to zero in on specific network traffic.

This section focuses on honing a search for network traffic and ways of making that traffic more understandable. Ultimately you should be able to use EtherPeek to find and diagnose problems. Much of this tutorial is focused on finding problems by comparing "normal" network traffic with "broken" network traffic. There are three basic ways to do this.

- You can use EtherPeek to establish statistics measurements and then determine that there is a problem when the network moves outside these baseline statistics.

- You can use EtherPeek to make performance measurements and then determine when performance has degraded below an established threshold.

- You can use EtherPeek to do protocol debugging. This is the most difficult because it is the most microscopic and complex. You will need to frequently refer to protocol reference guides to obtain the information needed to adequately debug a protocol session.

The process of diagnosing an unhealthy network can proceed along these lines:

Compare the network's current behavior to the average or desired behavior.

Notice the time intervals between the events -

What are the response times and throughput rates?

Are they normal or average?

Are they outside specifications or averages?

What is the source of the network slowdowns?

Look for irregularities in the overall traffic -

Deviations from the normal intensity or protocol mix

Unusual protocols

Unusual sequences

Broadcast / Multicast Analysis -

Is the broadcast / multicast rate within average limits?

Is this level OK or is it interfering with network efficiency?

Is it expected based on the number and types of nodes?

To troubleshoot a network process -
Watch all of the network events and compare them to the normal flow of events

To analyze a network performance problem -
Observe the response times and data flow characteristics of the process

Observe where the slowness is: client, server or network

To analyze network statistics -
Compare to normal usage characteristics or use to check basic network functioning

To understand a network peculiarity -
To see how one node's network activity differs from others on the network

To see why the network's behavior changes over time

To catch something or someone you suspect is causing a problem -
Watch all telnet sessions to the router to see who is reconfiguring

Watch who is logging in as a particular user from which stations

There are a number of features within EtherPeek that can be used to focus on specific network traffic patterns. One of the simplest and easiest to use features is the "Select Related Packets" command under the "Edit" menu. This command attempts to select the entire network session based on the user selecting a single packet from the session. To use this command, select any packet in the main capture window and then initiate the command from the Edit menu. EtherPeek will display the results.

Another way to zero in on a session is to start with the any of the "Statistics" windows and work backward from there into the main capture window. Any feature (such as network address) can be selected and then used to select packets in the main window.

Perhaps the single most powerful feature of EtherPeek is its ability to use filters. Filters can be used both during capture and in post-capture analysis sessions. EtherPeek ships a wide range of predefined filters and can also generate filters quickly and easily using several different methods that are delineated in on-line help.

The following exercises are designed to increase overall abilities with EtherPeek and focus on specific network troubleshooting issues.

Exercise: Simple FTP Performance Analysis

Start capturing IP packets with EtherPeek. Find a packet that is part of an FTP file transfer by using post-filtering and then use the "Select Related Packets" command. Hide the Unselected packets and then change the display options for time to relative. Within this session, which node is faster?

Network Utilization Exercise

Using the Network Statistics Window, answer the following -

When the devices are unoccupied, what is the normal, background utilization?

- Minimum:
- Average:
- Maximum:

What kinds of network processes are going on during this time?

During a file download, what utilization do you see?

- Minimum:
- Average:
- Maximum:

How would you characterize the utilization curve?

During a web page download, what utilization do you see?

- Minimum:
- Average:
- Maximum:

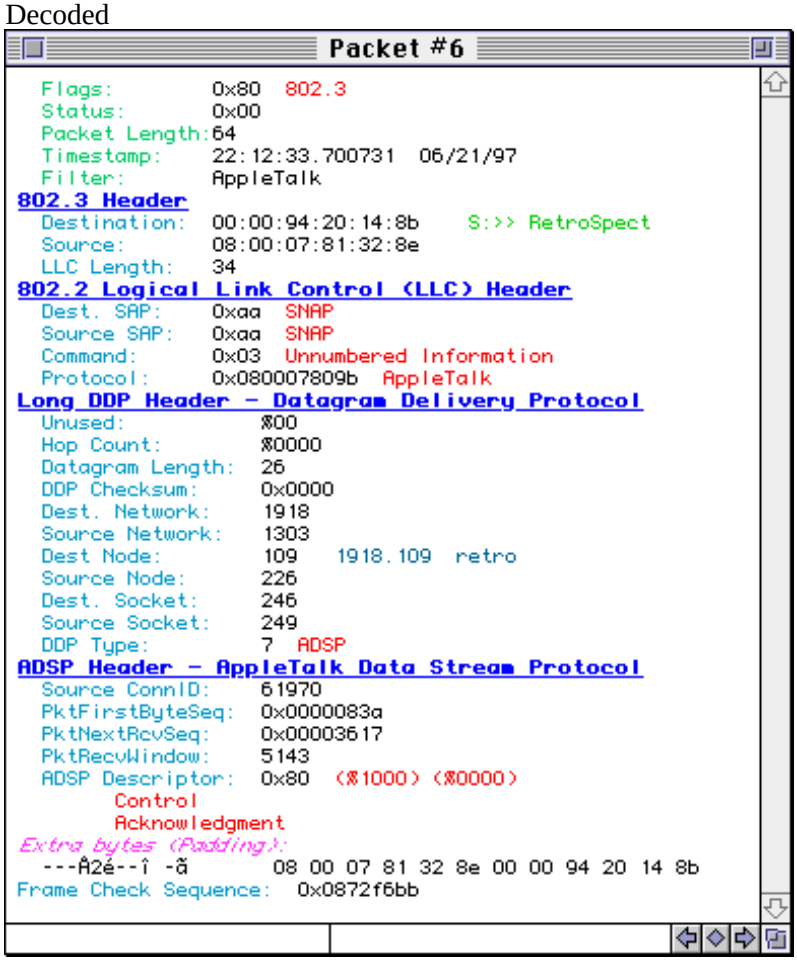
How would you characterize the utilization curve?

Protocol Analysis

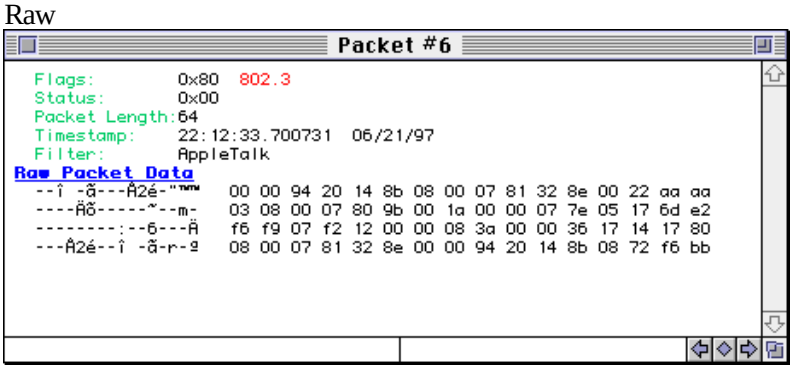
EtherPeek is a protocol analyzer. Protocol analysis is not an easy skill to learn, and the difficulty in learning can be increased by struggling with whatever protocol analyzer you are using. At this point, you should feel comfortable with the EtherPeek UI. Packets, as is the case with all digital data, are binary in form. However, most protocol information is hexadecimal in form. When you begin examining the payload data, you will see data that it is in decimal and ASCII form. For this reason, there are many ways to view packets within EtherPeek.

The Decode Window - Decoded vs. Raw

Decoded



Raw



As you can see, the decoded data is much easier to work with. The ability of a protocol analyzer to decode packets is largely what you pay for when you by one.

The second critical feature needed to make sense of packets is displaying just the packets you want to see. EtherPeek has a number of ways to accomplish this. You can use filters, you can use the select tools, you can skip from specific statistics windows to the main window, and you can use plug-ins. The most straightforward of these is using the variety of filtering mechanisms within the program.

Using EtherPeek as an Application Monitoring Tool

As networks mature, users expect network processes to become more transparent. At the same time, the Intranet revolution has made many network processes and subprocesses critical to the everyday function of corporations and other organizations. For this reason, the network management paradigm is shifting away from network management as an end into itself and toward network application services monitoring. The idea is that business-critical applications with a network component or server should have a guaranteed level of network bandwidth. They should also be monitored for attacks and for general up time.

EtherPeek can be used to monitor many network applications “out of the box” and, more importantly, can be extended to monitor any network application via custom plug-ins. In this section, we will examine a few of the existing plug-ins that ship with EtherPeek.

To enable application monitoring, you will want to set up an enterprise installation of EtherPeek. You will then want to determine what functions are critical and the best way to go about monitoring them. You will need to install EtherPeek on a section of the LAN that has access to the critical traffic. One way to do this is via port mirroring on a switch-based network, but any of the other discussed methods should also be considered.

You will also want to determine if you need any packets to be saved to disk. This is typically not necessary. Unless you are looking for a specific packet, you will want to set up EtherPeek for continuous capture and discard buffer contents. EtherPeek has to pause in order to save the packet contents to disk.

Additional Recommendations and Resources

This document attempts to provide a solid understanding of the current issues in network design, troubleshooting, and network security, and how EtherPeek can be an effective tool in addressing these management issues.

Regular use of EtherPeek will help to maintain and solidify the troubleshooting and analysis skills that you have acquired after working through this paper. Typically, this means using EtherPeek at least once a week. The key here is to remember the different ways of using EtherPeek and vary its use. EtherPeek has a straightforward UI but this also means that it may be easy to forget some of the subtleties of various problem-solving techniques. If you want to master protocol analysis you should use EtherPeek daily and you should attempt to follow through various processes within each protocol you are learning.

For learning general protocol analysis of the most common LAN protocols, you will need to have a number of reference books. Start by reading **Troubleshooting Internetworks**, by Mark Miller. If you want to learn IPX, we recommend **NetWare LAN Analysis** by Laura Chappell. Finally if you want to learn IP, try **Troubleshooting TCP/IP**, also by Mark Miller.

All network engineers should subscribe to both the CERT and CIAC security mailing lists and should check both web sites at least once a month. These lists are primarily concerned with describing new attacks on various systems; however, most of these attacks involve exploiting bugs in various pieces of network equipment and protocol stacks within various operating systems. The single most frequent way outside access is obtained to corporate sites is via known and published bugs in operation system software or via applications that run at a high level of access.

EtherPeek Implementation Roadmap

The following lists the recommended steps for implementing EtherPeek or use in most enterprise networks.

1. Determine the most common and relevant use for EtherPeek as a tool in your network environment.

2. Build one or more EtherPeek installations that best allows EtherPeek to be used as the tool in #1.

3. Use EtherPeek regularly. You should use EtherPeek at least as frequently as you do backups and run virus protection software (hopefully, at least once a week). Think of EtherPeek proactive captures as network data backups.

4. Develop a regular schedule to update your network names file. Build a MAC address table for your network. Download the latest Ethernet Vendor NIC 3-Octet MAC file from www.cavebear.com.

Summary

EtherPeek is a complex tool that takes time and, more importantly, regular use to master. This document was written to help advance a network manager's skill set and comfort level with EtherPeek. To get the most out of the information provided here, exercises with the software should be repeated on a live network several times. To master protocol analysis, a good set of references for specific protocols should be compiled, and a commitment to regular exploration of these protocols on a live network should be made.