Using Network Analysis Software To Teach the Internet Protocol Stack in the Laboratory

Richard E. Pfile, William T. Lin

Purdue School of Engineering and Technology at Indianapolis

Abstract

The stacked protocol concept is difficult to understand and to many students it's an abstract notion. Although students are masters at memorizing tasks the different protocol layers perform, they don't grasp fundamental concepts of how the protocol stack functions in computer communications. Message addresses reside at different layers in a stack, several stack layers perform different integrity checks, and handshakes can take place at different stack layers: all very confusing to students. In addition, several messages must usually be exchanged to get necessary information to make a connection and send data to a remote computer. The complexity causes students to loose sight of the big picture involved in the communication process.

This simple series of exercises in which communications packets are captured and displayed allows students to see the dynamics of the operation of a stacked protocol and mechanisms involved in computer communications. By capturing the real-time packets, the various mechanisms that allow messages to be sent between two computers within a room as well as between two computers in different states or countries are unfolded on the computer screen. Software designed for network troubleshooting can be easily used to bring the communications messages alive in the laboratory.

Introduction

These laboratories are used in a beginning networking laboratory to give students first-hand experience examining network data packets. The software program, LANWatch, is used to capture the packets. LANWatch is a program designed to help troubleshoot networks and is capable of filtering and monitoring packet types specified by the user. This filtering capability makes it ideal for capturing network packets in a networking laboratory. Summary information about the packet is given in a header and different packet layers are displayed in different colors to easily identify them. The data packets captured in the laboratories are complete with all the raw data from all layers. The purpose of these exercises is to allow students to see packets and better understand layered communications protocols.

The laboratory should be connected through a hub for these exercises since a switch can eliminate computers from the communications path, not allowing all students to see the communications sequences.

"Proceedings of the 2001 American Society for Engineering Education Annual Conference & Exposition

Exercise 1 Internet Protocol (IP) Packet

In the first exercise, students capture IP packets and comment on the fields in the Internet and network interface layers. These layers include Internet addresses, physical addresses, and protocol types as well as other message fields.

To capture the packets, the network card on the student's computer is placed in the verbose mode by the LANWatch software, which allows all packets streamed past the network card to be read. Since most networks are busy, filters must be setup so that only packets of interest are captured and stored. If packets are not filtered, the input buffer would quickly fill with packets from other network traffic and the desired packets would not likely get stored.

Packet filtering can be done on physical addresses, Internet addresses, and protocol types. To implement a filter, the filter menu is selected and parameters entered to setup the filter types. Some examples of filter parameters are:

IP packets only: ip
Source protocol address: ip src 134.68.225.22
Destination protocol address: ip dst 134.68.225.22
Source hardware address: src fffffffffff
TCP protocol with source address specified: ip src 134.68.225.22 tcp
ARP messages only: arp ip

AND or OR operators can match two separate conditions: ip src 134.68.225.22 and ip dst 134.68.225.6

LANWatch is setup to capture only IP protocol packets for this first exercise. After capturing an IP packet, students comment the target hardware address, source hardware address, and protocol type fields.

A typical packet captured is shown below.

```
|--------------------Network Header---------------------------|

|----Target Hardware Address----|----Source Hardware Address----|--Protocol Type--|
  00 00 C0 A0 51 24     00 00 0C 4D 7D 2C      0800

|-----------------------------------------IP Header-----------------------------------------|
|Version Length|Precedence/TOS|-Length-|-----ID------|Fragment|TTL|Protocol|Checksum|------IP Source------|--IP Desintation--|
    45           00          00 28  D3 25  00 00  2F   06    22 50  C7 03 41 12   86 44 08 01
```

By examining the network layer students can see the six byte Ethernet addresses and determine that an IP packet follows the Ethernet layer. Students examine the Internet layer and determine the class of the school's Internet address. I also have them comment on the packet fragmentation, time-to-live, the protocol that follows the Internet layer, message precedence and type of service.

Exercise 2 Address Resolution Protocol (ARP) Protocol

In the next exercise, an ARP[i] packet is examined. This protocol lets a host get the address of a remote host when it has an IP address. It is very simple protocol that allows students to see how local addresses are resolved.

Students filter on IP protocol, traffic to and from the instructor's computer, and ARP requests. Students are given the NIC card address for the instructor's computer and they setup a filter so that all messages to and from this computer are captured. As an instructor, I empty the ARP buffer on my computer and then originate a communication that will generate an ARP request. A ping works well for this.

The examination of an ARP sequence reveals the local address resolution process. The ARP request packet is sent out as a hardware broadcast to everyone on the network. It contains the IP address of the remote computer whose physical address is needed as well as the IP address of the computer responsible for handling ARP requests. The computer responsible for resolving addresses returns a message with the physical address. Students are able to note the sequence of events including the original ARP request, the ARP reply, and then a message sent to the computer of interest. They can also see the ARP protocol type specified in the network layer, and fields that identify the ARP request and reply. The ARP request lets students observe a broadcast message as well. In just a few minutes, students have a good overview of the address resolution process. We can then examine the computer pinged and verify that the hardware address returned in the ARP reply is correct.

Exercise 3 Internet Control Message Protocol (ICMP)

ICMP is used to troubleshoot and determine the health of a network.[ii] In this exercise LANWatch is setup to filter on Internet Control Message Protocol (ICMP) packets and the IP address of the instructor's computer (ip src a.a.a.a or ip dst a.a.a.a). A ping can be used to generate an ICMP message.

Students examine the ping request and reply and then comment on the purpose of the message ID number, data sent, pattern of the sequence numbers, and the number of ping messages sent.

An ICMP error message is then generated by pinging an IP address that does not respond. Pinging an unused IP address works for this. Students examine the ICMP error message to determine why the ping didn't make it through. They also examine the ICMP message type, code, ID Number, sequence number, the data in the IP header that failed and destination trying to be reached.

Exercise 4 Transmission Control Protocol (TCP)

In this exercise, students examine the host-to-host layer. A Transmission Control Protocol (TCP) layer is observed. Students filter on TCP packets and the instructor's IP address. A telnet session is used to generate TCP packets for this exercise.
After the TCP sequence is collected, students are asked to explain the significance of the TCP source and target ports, the source sequence/acknowledge numbers and how they increment throughout the communications sequence, the TCP header length throughout the session, the session flags during session startup and termination, sender window size, urgent data size, and the number of data bytes sent in each packet. TCP is a closed loop communications with packet

checking, which make it a reliable data communications mechanism. The data checking and verification features are easily seen when students capture and analyze an entire TCP session.

Exercise 5 BOOTP

For this exercise, LANWatch is setup to examine BOOTP packets. Students filter on the hardware address of the instructor's computer and IP packets. A BOOTP transmission is generated by booting up a computer. Students note the port numbers of the source and target, the protocol BOOTP is carried on, and the source protocol address and its significance for both the request and reply. They are to note the source and target hardware address and type, the transaction ID, and the IP address assigned to the client.

BOOTP is also sent out as both an IP broadcast and physical broadcast and is an excellent example of the utility of an IP broadcast. It is also carried on the User Datagram Protocol (UDP). Students can note the ports for the DNS service in the UDP layer.

Exercise 6 Domain Name Service (DNS)

This exercise involves examining application level packets communicated by a DNS application to resolve an Internet domain name and helps students understand how names are translated to IP addresses on the Internet.[iii] Students examine a DNS request and reply by filtering on the hardware address and IP packets to and from the instructor's computer. A DNS request can be generated by starting World Wide Web (WWW) session to a remote host. I generally start a WWW session with a large recognizable corporation. The last time I did this with Microsoft and they reported back with IP addresses of nine name servers.

Students examine the protocol (UDP) that DNS is carried on, report on the well-known port number for DNS, the IP address of the local DNS server, the protocol DNS is carried on, the number of name servers reported in the reply, IP addresses of the name servers, and the time to live for the messages. They are also required to explain the name resolution process, starting with the desktop computer, the local domain servers, and the organizational name servers at the target site.

Exercise 7 TELNET

TELNET is at application level that can be observed by filtering on an IP protocol address and TELNET packets. Since TELNET is designed to operate with a variety of terminal types, negotiations take place that are used to setup the terminal types and other options to be used for the session. Students collect the packets and report on the negotiations involved in setting up the communications.

Conclusion

Students have made many comments that the dynamics of seeing both sides of the data transfer process have helped them to better understand the communications process, particularly multi-message processes such as a TCP session startup and termination or a name resolution. Although these concepts are taught in lecture before the laboratory exercise, students better comprehend the dynamics of the processes by capturing and reporting on real-time events and walking over to the computer doing the communications and examining physical and Internet

addresses on that machine. The exercises have made the TCP/IP communications process come alive.

Richard E. Pfile
Richard E. Pfile is a professor of Electrical Engineering Technology at IUPUI. He received his B.S. from the University of Louisville and his M.Eng. from the University of Michigan. He has won the Outstanding Teaching Award and has received Teaching Excellence awards from the School of Engineering and Technology at IUPUI. He teaches courses in microprocessor systems, computer networks and digital signal processing. He has 17 years of teaching experience and eight years of industrial experience, including three years as a systems engineer.

William T. Lin
William Lin is currently a faculty member in the Purdue School of Engineering & Technology at Indiana University, Purdue University at Indianapolis. Prior to IUPUI, Bill has served as faculty members at The Pennsylvania State University, Wayne State University, and DeVry Institute of Technology. Before joining IUPUI, he was a Full Professor in EET at DeVry Institute in North Brunswick, New Jersey. During his tenure at DeVry, in addition to teaching and developing courses in technical area he has been involved in the development and teaching of the Team problem-solving curriculum since its inception. Dr. Lin received his Ph. D. in Electrical Engineering from The Pennsylvania State University, a M.S. degree in Physics from University of Southern Mississippi, and a B.Ed. in Science Education from Taiwan.

---

[i] Comer, Douglas E., Internetworking with TCP/IP Volume 1, Prentice Hall, 1995, pages 73-79.
[ii] Haywood, Drew, Networking with Microsoft TCP/IP, New Riders, 1997, pages 134-136.
[iii] Ramteke, Timothy, Networks, Prentice Hall, 1994, pages 459-466.