Hands-on Approach to Teaching Computer Networking Using Packet Traces

Jeanna Neefe Matthews
Clarkson University
8 Clarkson Avenue, MS 5815
Potsdam, NY 13699
315-268-6288
jnm@clarkson.edu

ABSTRACT
This paper describes a novel approach to teaching computer networking through packet traces of actual network traffic. The paper describes a set of exercises that use packet traces to clearly illustrate the activity that takes place on the network under a variety of situations from traces of common network applications like e-mail and web browsing, to traces that illustrate backbone network activity like dynamic routing protocols, to traces of abnormal conditions like computer viruses or worms in action. Students can view each trace using freely available software that runs under both Windows and Linux. This method gives students the hands-on, practical learning style they find most interesting while eliminating the need to make specialized networking equipment available for laboratory exercises. The paper discusses the results of applying this method to a diverse set of courses including semester-long undergraduate courses in information technology and computer science, graduate courses in computer networking and even a workshop course for high school students.

Categories and Subject Descriptors
K.3.2 [**Computer and Information Science Education**]
D.4.4 [**Communications Management**]

General Terms
Documentation

Keywords
Network Protocol Analysis

1. INTRODUCTION
I have taught computer networking to a wide range of student groups from graduate students specializing in computer networking to high school students with little background in networking. I have developed a substantial set of exercises based on packet traces of actual network activity. I have found these exercises remarkably effective with all groups of students regardless of their background.

SIGITE'05, October 20–22, 2005, Newark, New Jersey, USA.

In this paper, I illustrate the type of information available in packet traces and describe how I use packet traces to add a hands-on, practical component to the courses I teach. I compare trace-based exercises to laboratory exercises involving networking equipment.

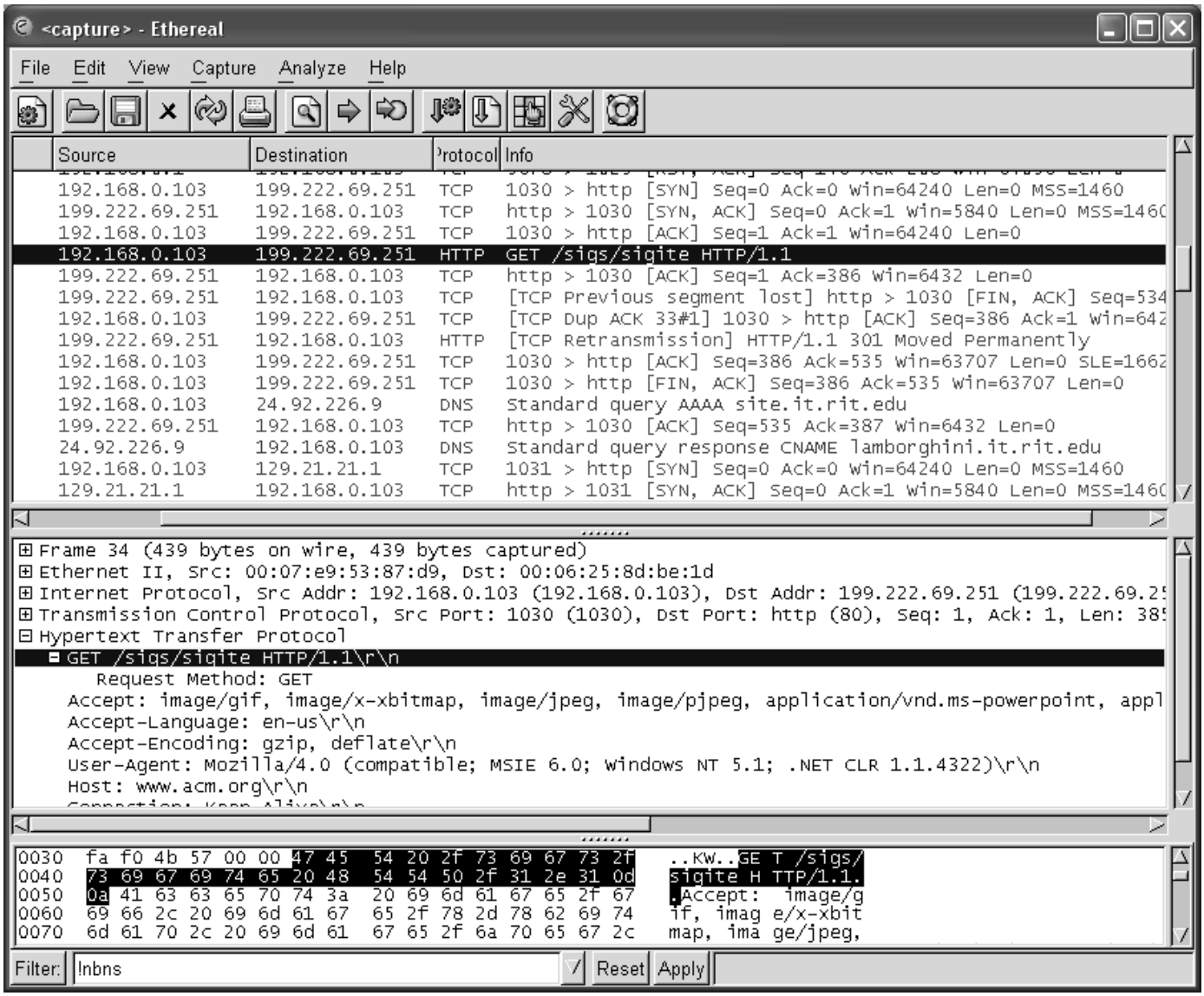
I describe a set of exercises that I have developed and my experiences using these exercises with student groups at different educational levels. I describe how I modify the set of exercises used for each group. Finally, I give advice for taking traces to use as the basis of your own set of exercises.

2. ONE ILLUSTRATIVE EXAMPLE
I would like to begin with a simple example to illustrate the type of information that is available in packet traces. Figure 1 contains a screen shot of Ethereal, a freely available and open source network protocol analyzer [1]. The screenshot shows the trace of traffic generated when I used a web browser to open the URL http://www.acm.org/sigs/sigite. The sequence of packets shows the HTTP connection to www.acm.org, the redirection to site.it.rit.edu, the DNS responses that indicate the canonical name for this machine, and finally, the transfer of the page from the machine, lamborghini.it.rit.edu. Even this short trace illustrates how packet traces can be used to provide a look "under the hood" of computer networks.

Many types of exercises are possible with even this simple trace. For example, each frame can be decomposed into an Ethernet frame, an IP datagram, a TCP segment or a UDP datagram and finally an application layer protocol message. The details of each protocol used can be examined. For example, HTTP requests and responses are visible along with their header tags such as Accept, Accept-Language, etc. TCP connection establishment with its three-way handshake of SYN, SYN-ACK and ACK is clearly visible. Data loss and retransmission in TCP can be studied. Even the actual data transferred is visible in the trace including the HTML text and embedded graphics.

Now that I have illustrated some of the potential uses of packet traces, in the next section, I describe how I came to use them as the basis for lab exercises in my computer networking courses.

3. MOTIVATION FOR EXERCISES BASED ON PACKET TRACES
As an instructor of computer networking, I struggled to find an efficient way to bring hands-on exercises to my students. At first, I believed this required a lab full of networking equipment, so I



set about collecting routers, switches, hubs, PCs and the lab space to house them. I was able to provide a great learning experience for a small number of students, but the experience did not scale to the size of a typical undergraduate class. It was too expensive to outfit a laboratory large enough to hold the entire class and holding many small lab sections became an administrative nightmare. Maintaining, upgrading, and managing the lab space required, even for a small class, required significant time to. I found myself facing an unpleasant decision – teach computer networking in a pure lecture style without the hands-on exercises that I found so effective or continue to invest huge amounts of time in funding and managing the facilities required to support them.

At that point, I took a good look at the laboratory exercises I was using. Nearly all of them involved three parts:

1) Configuring the hardware and software to prepare for a networking experiment (e.g. connecting a series of machines with routers, hubs of switches, configuring the machines with hard-coded IP address, etc.).

2) Performing an experiment to generate specific network activity and capturing a trace of the activity.

3) Analyzing the trace to understand the subtleties of what occurred.

I realized that most of the learning for students was occurring in step 3 when they analyzed the traces they had captured. This is where students actually examined the details of the networking protocols. Steps 1 and 2 were also quite time-consuming and error-prone for students. Although they did learn practical skills setting up the experiments, they would frequently have to repeat experiments because they failed to capture the proper traffic – especially for more complicated experimental scenarios. This often distracted them from the primary goals of the exercises.

I decided to try a new approach. I would describe the experiment in detail to the students and then distribute a trace of the

experiment that I had taken for them to analyze. I was thrilled with the results. I could maintain a single set of networking equipment on which I could take the traces. I could write an answer key that referred to specific packet numbers because all students looked at the same trace. Even campus system administrators who had been nervous about students collecting packet traces were happy because I collected the traces myself and distributed them to the students. Students were able to focus on the key goals of the exercises rather than struggling with the logistics of setting up the experiments. The trace-based exercises retained the hands-on, "under the hood" of the network flavor that students found so appealing. Students were even able to complete the exercises at home using freely-available software that runs on most computing platforms.

4. EXERCISE SETS

In this section, I describe the trace-based exercises sets that I have developed. Depending on the background of the students and the length of the course, I may not use all of the exercises. In Section 5, I discuss in more detail my strategy for choosing which exercises I use with different student groups.

Most courses in computer networking are organized around the different layers in the network protocol stack. Therefore, I have found it quite natural to design packet trace exercises around the layers of the protocol stack as well. In this section, I describe exercises for each layer of the protocol stack beginning at the top with the application layer through the link layer at the bottom. (Note: these exercise sets can also be used with a bottom-up presentation of the material by beginning with the link layer exercises.) I also describe some sample exercises focused on cross-cutting issues such as network security that have implications at all layers of the protocol stack.

For each exercise, I typically distribute a trace file along with an exercise guide. The exercise guide describes the configuration in which the experiment was performed and the type of network activity that was captured in the experiment. It also typically provides some background material. It ends with a set of questions that students must answer by examining the trace using network protocol analysis software like Ethereal. I typically provide a range of questions including some questions that can be answered easily by examining the trace, some questions that require some additional research, and some open-ended discussion questions.

Space does not allow me to present complete exercises. In this section, I focus on describing the types of traffic I captured for each exercise. Table 1 lists the exercises I commonly use at each layer and the sections that follow elaborate on each one.

Table 1. Exercise topics at each layer of the protocol stack.

Layer	Exercise Topics
Application Layer	HTTP, HTTPS, POP, SMTP, DNS, FTP, AIM, Network games, Peer-to-Peer file sharing
Transport Layer	TCP Basics, TCP Retransmission, Comparing TCP to UDP, TCP Congestion Control, Variants of TCP (SACK, FACK, etc.)
Network Layer	Basics of IP, ICMP, Ping and Traceroute, Interior Gateway Routing protocols (RIP, OSPF, IGRP), Border Gateway Protocol, IPV4 vs IPv6, Acquiring an IP address with DHCP, Network Address Translation (NAT)
Link Layer	Ethernet, Wireless LANs, Wired Equivalent Privacy (WEP), Ethernet Hub vs Switch, Address Resolution Protocol (ARP), FDDI

4.1 Application Layer Exercises

The application layer is arguably the most diverse layer of the protocol stack. I find it most effective to design exercises around the network applications the students use most. For that reason, as in the example in section 2, I often start with a simple web browsing exercise.

I capture a trace of fetching a page in a web browser. In the exercise, I ask students to imagine what a web browser says to a web server to fetch a web page. If they are not already familiar with HTTP, students are often quite surprised to see that HTTP is based on English text and uses understandable commands like "GET". Students who have experience with HTML and web design will be interested to see the raw HTML actually transferred over the network. They can also see each element of a web page – style sheets, pictures, and text – transferred as individual objects. It is also a good opportunity to introduce other important application layer protocols like DNS.

In the exercise, I often fetch several pages with different characteristics. For example, a simple web page with only text and a more complicated web page with graphics, style sheets and cookies. The level of detail can be tailored to the goals of the class and background of the students.

I frequently follow this basic web-browsing exercise with an exercise that compares traffic transferred with HTTP to traffic transferred with HTTPS. Many students are aware that they should look for URLS to begin with https when submitting confidential information like credit card numbers. By actually seeing the difference between plain text and encrypted text transmissions, the importance of this advice is driven home.

Examining plain text transfers provides a good opportunity for a discussion of the ethics of trace collection. When tracing on a network, it is possible to collect authentication information like username and passwords, as well as private information like which web pages a person is viewing. I make it clear to students that they should only take traces on a network if they have the permission of the network administrator and if users of the network are aware they are being traced. Many schools are trying to incorporate discussions of professional ethics into their existing curriculum and this is a good opportunity to do so.

I also make a point of knowing the school policy – where and when is it appropriate for students to take their own traces and what are the potential penalties for inappropriate tracing. Many schools have policies that forbid tracing on campus networks. It is natural for students to want to experiment with taking their own traces so I usually try to obtain permission to have a switched network set up in which students do have permission to take traces. Switched networks generally let a tracing machine see only traffic that is sent to and from that machine rather to and from other machines in the network.

After web browsing, email is one of the most familiar network applications for most students. Therefore, I often follow the web browsing exercises with an exercise illustrating how SMTP and POP are used to send and receive email. The email exercise is one example of a trace that can be difficult to collect without revealing private information. In particular, the full contents of emails as well as plain text passwords for POP are often visible in the trace. For this reason, I typically set up a stand-alone network with a mail server and client using accounts created specifically for this purpose.

After web-browsing and email exercises, I often student interest guide my selection of other application layer exercises. Any other network application that students use frequently can make a good basis for an exercise. Students often request traces of Instant Messaging or network games. However, where possible, I choose applications that use open protocols that are clearly documented in the Request For Comments documents [5]. Some protocols like those behind AIM and network games are not always as clearly documented.

The application layer exercises are interesting even to students with limited background in computer networking. The level of detail can be customized to the audience. For more advanced students, there are many subtle details that can be examined. In this case, the RFCs are an especially useful source of detailed documentation. For beginning students, a more basic introduction is typically sufficient.

4.2 Transport Layer Exercises

The transport layer exercises revolve around the two primary transport layer protocols - the User Datagram Protocol (UDP) and the Transport Control Protocol (TCP). Transport layer exercises can range from simply introducing the basics of TCP and UDP to exercises that capture the subtleties of TCP congestion control.

I typically begin with an exercise that introduces the basics of TCP – including connection establishment, use of sequence numbers and general two-way data flow. I take traces of a variety of TCP connections including bulk data transfer like in a large file download and also short, interactive data transfer like in a remote login connection. I also use a tool called ttcp [6] to generate synthetic TCP streams of specified characteristics.

I often follow this with an exercise that focuses on retransmission in TCP. This is an example of an exercise that must be carefully constructed in order to provoke small periodic data loss without disturbing the whole connection. Tools such as NISTNET [3] that drop packets with a specified probability can be helpful for this.

UDP is simple enough that I typically do not devote a complete exercise to UDP alone. Rather, I capture traffic that illustrates the difference between TCP and UDP. For example, I show how TCP retransmits lost data while UDP does not, and I show how TCP has a connection set-up phase while UDP does not.

For more advanced audiences, I use a series of exercises that examine TCP congestion control. I capture traces illustrating what happens when two TCP streams share a bottleneck link. I illustrate that TCP connections adjust to share the link evenly while a UDP stream will take over all the bandwidth, completely starving the TCP stream. I also collect traces illustrating the subtle differences between variants of TCP such as TCP-SACK, TCP-Reno, TCP-FACK and other.

4.3 Network Layer Exercises

There is one dominant protocol at the network layer, the Internet Protocol (IP). At first glance, this lack of diversity might seem to limit the exercise options at the network layer. However, there are actually many options if you include the many companion protocols such as the Internet Control Message Protocol (ICMP), interior gateway routing protocols such as RIP, OSPF and IGRP and the Border Gateway Protocol (BGP). In addition, it is interesting to compare IPv4 with IPv6 and to explore commonly deployed network layer technologies like Network Address Translation (NAT).

In the network layer, I like to begin with something that explores the basics on IP. I find it effective to do this with an exercise that compares IPv4 and IPv6. This requires setting up an IPv6 network segment for trace collection.

I typically follow this with an exercise that introduces ICMP in the context of the common network utilities ping and traceroute. Ping uses ICMP echo request and reply and traceroute relies on ICMP TTL expired messages to map the route to a destination. In this exercise, I often take traces at multiple vantage points as the traceroute traffic passes through the network. This enables students to see how the headers of packets change as they pass through intermediate routers along the path.

For more advanced audiences, I also include a series of exercises on dynamic routing protocols. For example, I developed an exercise involving several routers configured to use various interior gateway protocols to dynamically establish routes. This is one of the most challenging exercises to configure as it involves 5 Cisco routers and 6 computers running Ethereal to examine the resulting traffic from many different vantage points. I typically perform the same experiment using the Routing Information Protocol (RIP), Open Shortest Path First (OSPF) and the Enhanced Interior Gateway Routing Protocol (EIGRP) in order to compare and contrast their behavior.

The Border Gateway Protocol is the dynamic routing protocol used between autonomous systems in the Internet. If you have access to tracing on a backbone link BGP conversations between backbone Internet routers can be captured. I often set up a small example of a network using BGP and take traces in that context.

Finally, an increasing number of homes and businesses use cable modem routers that perform Network Address Translation to share a single globally routable IP address between many internal computers. Therefore, many students enjoy an exercise examining traces taken before and after such a router. It is interesting to see the local use addresses such as 192.168.0.1 mapped to and from a globally routable IP address.

4.4 Link Layer Exercises

There are a wide variety of link layer technologies that can be explored in trace-based exercises. I try to focus on the link layer technologies that students are most likely to encounter in their daily lives including Ethernet and wireless networks. I will often include an exercise highlighting one less common link layer technology like FDDI.

I begin with Ethernet because it is so ubiquitously available. I capture traffic that illustrates the difference between the IEEE 802.3 and the Ethernet II frame formats. I also typically capture traces that illustrate the difference between Ethernet hubs and switches. This difference is especially relevant to students in light of network trace analysis because when a machine is connected to a hub it can see traffic sent to all other machines on the hub while a switch forwards traffic only to the specific port used by the destination machine .

I typically follow with a set of exercises focused on wireless traffic. Students are especially interested to see the difference between traffic sent with and without Wired Equivalent Privacy (WEP). This helps students appreciate the risks of setting up an unsecured network access point in their home.

To capture wireless traces that show the IEEE 802.11 details, I have found it necessary to use another network protocol analyzer called Kismet [2] that runs under Linux. When capturing wireless traces under Ethereal, the frames appear as Ethernet frames rather than 802.11 frames. However, once a trace is taken with Kismet, Ethereal can display the 802.11 details just fine.

At the link layer, I also typically do an exercise focused on the Address Resolution Protocol (ARP). In this context, I discuss MAC addresses in general, what happens if duplicate MAC addresses exist on a network and the problem of MAC address spoofing. This helps students realize why using MAC addresses for identification purposes in unreliable.

At the link layer, I also sometimes cover networking technologies such as ATM , X-25 or Frame Relay that although they were designed as network layer protocols are frequently used as link layer technologies from the perspective of IP in the Internet. Traces such as this require some specialized equipment. I have found the WANEdukit cards [7] for PCs made by Sangoma Technologies to be an excellent investment for this purpose.

4.5 Network Security

Network security is an example of a crosscutting topic that affects many layers of the protocol stack. In addition to mentioning security issues in exercises dedicated to a particular protocol layer, I have found it effective to assemble a collection of exercises dedicated specifically to network security. I have found that students are especially interested in security related topics.

I have collected a set of traces of specific computer viruses and worms in action. Taking these traces requires additional care. In particular, it is important to set up a completely stand-alone network to avoid infecting any production computer systems in the process. In addition, I take some extra precautions like ensuring that no copies of the live virus codes are captured in the traces. At times, this can be difficult because many viruses spread by attacking other computers and then sending copies of themselves to compromised systems.

If I have time for several security related exercises, I like to provide examples of some of the major categories of computer malware. For example, I try to provide a trace of a malware that does mass emailing, a trace of a buffer overflow attack and a trace of a back-door program controlled by an attacker.

Another popular security related exercise is one that illustrates TCP session stealing. TCP session stealing allows an attacker to end or hijack an on-going TCP session simply by watching the network for what sequence number each side of the connection is expecting and injecting a forged packet with the expected sequence number. This highlights the security risks present in protocols like TCP and IP.

Finally, I often include an exercise that illustrates the use of nmap [4] to scan a computer for open ports. It is important for students to know that any server software they have running on their machine and any open network port represents an avenue of possible attack. Even savvy home computer users are typically not aware that they are running any server software on their home machines; however, the vast majority of home PCs do have open network ports. Exercises like this teach students to shut down unneeded server software and to keep any server software they are running up-to-date.

4.6 Other Exercise Sets

I have also assembled other exercise sets that I have found useful with various audiences. For example, I have a set of exercises designed to introduce students to specific network protocol analysis software like Ethereal. For students with little background in computing, these exercises are essential. However, more advanced students tend to find them overly simple.

I have also assembled exercise sets devoted to multi-media networking and quality of service. For example, exercises focused around Voice Over IP and protocols like the Realtime Transport Protocol (RTP) that support it.

I am constantly adding to and improving my set of exercises based on experiences using them in the classroom. As a result, I have a large collection of exercises and sometimes allow students to choose optional exercises of their choice for extra credit. I have also had good luck using this collection of exercises as the basis of independent study courses where students complete a certain number of exercises independently. Some independent study students have also followed by descriptions of the experimental setup to gain practical experience with hardware and software configuration as well as with network trace analysis.

5. EXPERIENCES USING THIS METHOD WITH A VARIETY OF STUDENT GROUPS

I have used this approach successfully with a wide range of student groups including semester-long graduate course in computer networking through short workshops for high school students. In this section, I describe how I tailor the approach for each group. Table 2 summarizes how I choose exercises and modify the focus for each group.

In order to tailor the exercises to each group, I have found it helpful to have a different goal for the students. When working with high school students, my goal is to help them become educated network users. I want them to understand what type of traffic crosses the network in support of the network applications

they use. In particular, I want them to know what data is passing over the network in plain text so that they can make wise decisions about network use. I also want them to understand the most common types of attacks against networked computers so that they can avoid risky practices.

To achieve these goals, I have found it most useful to choose exercises that focus on application layer of the protocol stack and on security topics. When I have tried to use exercises focused on the details of lower layer protocols such as TCP or dynamic routing protocols, I have found that the majority of high school students are not interested in that level of detail. I do try to give students exposure to the other layer s of the protocol stack primarily through exercises that focus on security at each layer. For example, high school students have responded well to exercises focused on WEP in wireless networks

.

Table 2. Modifications for various educational levels.

Level	Goal	Modifications
High School	Educated Network User	1) Focus on wise use of network technologies. 2) Choose exercises from the application layer and some security topics.
Undergraduate	Computing Professional	1) Focus on ability to research and understand protocol specifications (RFC, etc.) and in using network protocol analysis for trouble-shooting network problems. 2) Choose some exercises from each layer of the protocol stack.
Graduate	Networking Expert	1) Focus on understanding and evaluating subtle points of protocol operation. 2) Choose detailed exercises from transport and network layers.

For undergraduates, my goals are different. I want to help them be well-rounded computing professionals. This certainly includes being an educated network user, but it also includes being able to effectively use online resources to find the address to any networking problem or question they may encounter in the course of their career. As a result, with undergraduates, I provide a well-rounded set of exercises from all layers of the network protocol stack. In comparison to the exercises I use with high school students, I tend to use fewer application layer exercises and replace them with exercises with the transport, network and link layers.

For undergraduates, I frequently include questions about the traces that require them to search for the answer in the RFC document for a specific protocol. This gives them practice in using existing online resources to answer detailed questions. In this case, I am most interested that they become adept at looking for answers in the right places not that they find the particular detailed answer.

Undergraduates tend to appreciate practical computing skills and many students have told me after they graduated that network trace analysis skills have been extremely useful to them in their roles as working professionals. Network trace analysis gives them the tools to diagnose many common networking problems. For example, allowing them to investigate why an attempt to contact another machine is failing or to diagnose the presence of a virus.

For graduate students taking a course in computer networking, my goal with packet trace exercises is to give them a window into the subtle interactions of networking technologies in the real world. For graduate students, I typically spend little time at the application layer or the link layer. Instead, I assign exercises focused on the subtle details of the transport layer and the network layer. For example, a great deal of research has been done on various approaches to congestion control in TCP. I have my graduate students read these papers and then examine traces of network activity that illustrate the concepts..

For students at all educational levels, trace analysis brings a hands-on flavor that helps make the networking concepts they are learning more concrete. Students at all levels routinely report that exercises based on network traces are their favorite part of the course, and that they help cement their understanding of the lecture material. They say this style of assignment encourages them to do additional research on their own to understand each detail of the traces. Students often take their own traces and come to me with wonderful in-depth questions about what they see.

6. LOGISTICS FOR TRACE COLLECTION

In this section, I give some helpful hints for collecting traces for your own exercises.

1) When taking traces, it is easy to collect traffic that is not directly relevant to the intended exercise. I recommend stripping out any nonessential traffic using filters and saving only the essential traffic.

2) When taking traces that you will be distributing to students or posting online, it is important to ensure that the traces do not contain any private data that should not be revealed. In addition to stripping out nonessential traffic, you should examine the data collected to make sure you are not revealing any confidential data. Knowledge of the protocol under study can help you identify things like usernames and passwords that are likely to be present.

3) When in doubt, I recommend setting up a stand-alone network tailored to support the exercise. For example, I setup a stand alone mail server to collect traces of SMTP and POP rather than risk a confidential email being included by mistake. This has an additional advantage of allowing you to strip away information

that is specific to a particular school or environment. For example, the mail server can be named something generic like popserver@school.edu rather revealing the names and IP addresses of servers on your internal network. Stand-alone networks also make it much easier to remove any nonessential traffic since you only expect to see traffic that is directly related to the experiment.

4) Be aware of the actual data you capture and your right to distribute a copy of it. For example, if you capture a trace of a download of a copyrighted file, the entire file would be present in the trace. You can avoid this by downloading only freely available data. Another useful technique is to capture only the headers of each packet rather than the full data payloads.

5) Be aware that some traces require specialized and complex hardware and software configurations. For example, capturing a trace of BGP traffic requires access to a backbone network or access to routers and the ability to configure them correctly to establish BGP sessions. Similarly, capturing traces of a computer virus requires special care to avoid spreading the virus unintentionally.

6) When you have successful configured an experimental configuration, I recommend performing the experiment multiple times and capturing several traces. Additional copies of the traces can be useful in several ways. First, if there is data loss or errors in any one of the traces, you have alternate copies. Second, if you distribute solutions to your exercises, it can be useful to have the exact answers vary from year to year. Maintaining multiple copies of a similar experiment is an easy way to prevent students from reusing answers from a previous year. You can even distribute multiple traces within the same year to make it harder for students to copy answers from each other. In addition to homework grades, I use exams to assess individual learning. Students that copy answers are unlikely to be able to answer similar questions on an exam.

7. CONCLUSIONS

I have found exercises based on network packet traces to be an excellent addition to my computer networking courses for students of all educational levels. They add a hands-on, practical learning style that students enjoy without the need to make specialized networking equipment available for laboratory exercises. This paper explains how I use network trace analysis in my classes. In particular, I have described the set of exercises I have developed and the ways in which I modify their delivery for various students groups. Finally. I have provided some helpful hints for collecting traces for your own courses.

8. REFERENCES

[1] Ethereal, http://www.ethereal.com, Viewed July 1, 2005.

[2] Kismet, http://www.kismetwireless.net/, Viewed July 1, 205.

[3] NISTNET, http://www-x.antd.nist.gov/nistnet/, Viewed July 1, 2005.

[4] Nmap, http://www.insecure.org/nmap/, Viewed July 1, 2005.

[5] RFC Editor Homepage, http://www.rfc-editor.org, Viewed July 1, 2005.

[6] Test TCP (TTCP) Benchmarking Tool for Measuring TCP and UDP Performance, http://www.pcausa.com/Utilities/pcattcp.htm, Viewed July 1, 2005.

[7] WanEdukit, Sangoma Technologies, http://sangoma.com/education/p_wan_edukit.htm, Viewed July 1, 2005.