

**University of Pennsylvania**  
**Department of Electrical and Systems Engineering**  
**Digital Audio Basics**

ESE250 Spring 2011

Lab 11: Networking (TCP/IP)

Thursday, March 31, 2011

---

**For Lab Session:** Tuesday, April 5, 2011 in Ketterer Lab.

**Due:** Monday, April 11, 2011 by noon.

**Collaboration:** Work in lab in teams of 2. Perform individual writeups. See course collaboration policy in the [Administrative handout](#).

**Objective:** Observe, quantify, and deconstruct network traffic and network characteristics.

**Prelab Requirements:** Download and read this entire lab assignment. We recommend you read through the lab assignment before arriving in lab. Bring your headphones.

**Deliverable:** Writeup of answering questions (including prelab questions) **handin:** All labs will be turned in electronically through the [Penn Blackboard](#) website. Go to the assignment submission link and follow the instructions. Your writeup should be a **PDF file**.

**Exit Ticket:** Show the TCP conversation for your `ftp` session to your TA.

## Motivation

Modern data and communication networks can be very complex, consisting of multiple computers and other network devices each running a set protocols corresponding to the different layers in the network abstraction. The network provides a virtual point-to-point connection for devices wishing to communicate with each other that uses a layered approach to hide the underlying implementation from the applications on the computers wishing to communicate.

In this lab, you will explore network characteristics and network packets for the TCP/IP protocol.

## Lab Procedure Guide

### Observing network delay and packet loss using ping

Read <http://www.visualware.com/resources/tutorials/tracert.html>.

Use the linux utility `ping` to identify the communication latency with:

- The linux machine next to you
- [www.es.eupenn.edu](http://www.es.eupenn.edu)
- [www.harvard.edu](http://www.harvard.edu)
- [www.berkeley.edu](http://www.berkeley.edu) or [www.stanford.edu](http://www.stanford.edu)
- [www.ucla.edu](http://www.ucla.edu) or [www.usc.edu](http://www.usc.edu)
- [www.unsw.edu.au](http://www.unsw.edu.au) or [www.cuhk.edu.hk](http://www.cuhk.edu.hk)

**FOR QUESTIONS:** Make a table of response times and packet loss rates. Add the physical distances to these destinations to your table.

### Watching network paths using traceroute

Use the linux utility `traceroute` (same as windows `tracert`) to identify the path your packets take to reach the above destinations. You might find the following maps useful in deciphering the machine and network names along some of the paths.

- [http://www.internet2.edu/network/library/deployment\\_phases.pdf](http://www.internet2.edu/network/library/deployment_phases.pdf)
- <http://www.cenic.net/operations/>

To get a more accurate trace, you will need to use the flag `-T` to use TCP for the trace. For example, `traceroute -T www.es.eupenn.edu`

**FOR QUESTIONS:** Add the number of intermediate network hops reported along the path to each destination to the table you created in the previous section.

## Network connections using netstat

Run `netstat -ps`. Considering the statistics provided in the TCP section of the `netstat` output, record:

- the fraction of packets (segments) that required retransmission
- the fraction of packets (segments) that were bad

The “OUTPUT” section of the linux manual page (`man netstat`) explains the columns in the `netstat` output. For the following, we will focus on the “Active Internet connections” portion at the top of the `netstat` output (You should focus on the **Active Internet connections** section that appears at the top and **not** the socket information that appears at the bottom of the output).

Run `netstat -ap`. Record:

- How many ports are listening
- How many ports are open (established)

Here, we open a number of additional ports and observe them using `netstat -p`. The video is an introduction to what you will do in the next section, so you do want to pay attention to it, too.

- `ssh` into your eniac account in one xterm.
- In a separate shell from the `ssh` shell above,  
`scp username@eniac.seas.upenn.edu:/home1/e/e250/html/week11/lab/day1_audio.ppt`  
to `/tmp` on your workstation.
- Browse: <http://www.wireshark.org/>
- With your headphones on, watch the “Introduction to Wireshark” video.
- In a new shell (*i.e.* neither the `ssh` shell nor the one running `scp`) run `netstat -p` while the `ssh` session is open, the `scp` is copying, and the video is playing.

Record:

- the hostnames and ports for the three activities above. Remember to look in the **Active Internet connections** section for this information.
- any other open TCP ports that you identify .

Close down all these connections and your web browser.

## Packet watch and decoding using wireshark

- Start wireshark capturing traffic on your network port, make sure to run it as the super user with `sudo`.
- Restart your web browser.
- Browse [www.ese.upenn.edu](http://www.ese.upenn.edu)
- `ftp` and list the top level directory at `ftp.gnu.org`. `ftp` is short for “File Transfer Protocol.” This provides a way for an internet machine to make files remotely available. In the days before the WWW, this was how you would share data remotely over the internet. It is still in use for distributing software and technical reports today, but is dwarfed by more user-friendly web interfaces, richer interconnection in HTML document, and more flexible `http` transport. Today’s browsers generally also interact with `ftp` servers providing an alternate, more abstracted interface. For this experiment, `ftp` has the advantage that it will maintain an open connection to the remote host for the duration of a conversation—something `http` does not.
  - `ftp ftp.gnu.org`
  - When it prompts for a name, give it `anonymous`.
  - `ls` will list the directory.
  - `quit` to exit the `ftp` session.
- Stop capturing traffic.

Continue using wireshark to complete the following:

- Record the number of packets you capture
- Identify any packets that show evidence of TCP’s ordering and reliability mechanisms (*e.g.* retransmission, duplicates out of order) that come into play. Record the number of each kind (If you find none, you might come back after completing the rest of the lab and capture a larger set of traffic to analyze.)
- Find the HTTP get request for [www.ese.upenn.edu](http://www.ese.upenn.edu)
  - Identify and report source and destination IP address from the decoded IP layer information.
  - Identify and report source and destination ports from the decoded TCP layer information.
  - Identify the sequence number and length from the decoded TCP layer information.
  - Identify and report the first 16 bytes of the message payload.
- The Domain Name Service (DNS) provides translations between symbolic names (*e.g.* [www.ese.upenn.edu](http://www.ese.upenn.edu)) and IP addresses (*e.g.* 158.130.70.64). Use the `dns` filter to select only the DNS packet traffic.

- Record the number of DNS queries that occur in your session
- Record the first 8 names resolved
- Find a packet associated with the `ftp` connection. Use the “Follow TCP Stream” (under “Analyze”) to extract the conversation for this stream. Record the number of packets in this stream. Identify how the packets correspond to your `ftp` interaction.
- Identify any other network traffic that occurred during your captured session. If you find any, describe it.

[Exit Ticket] Show the conversation to your TA.

## Lab Writeup Guidelines

### Theory: Pre-lab

1. What is the straight-line distance from Philadelphia to Los Angeles?
2. Assuming speed-of-light signal propagation, how long will it take a signal to travel from Philadelphia to Los Angeles?
3. Assuming 1 Gbit/s network links and 1500 Byte IP Packets, how long does it take to inject one IP packet onto a network link?
4. Assuming speed-of-light signal propagation latency dominates transmission delay, if we had to wait for each packet to be acknowledged by the destination before sending the next packet, what is the maximum throughput (actual realizable speed) we could communication between Philadelphia and Los Angeles?
5. Assuming no congestion, if we do not wait for an acknowledge before sending subsequent packets, about how many packets can we send over this link before we receive the first acknowledgment from the other end?

### Analysis

**Question 1:** Present your table for your `ping` and `traceroute` sections

#### Observing network delay and packet loss using `ping`

**Question 2:** Compare the communication delay with the speed-of-light transit delay to the locations in your table.

## Watching network paths using traceroute

**Question 3:** Use the detailed path traveled from Philadelphia to Los Angeles to explain the network delays. You might want to identify the time you can assign to speed-of-light signaling delays and estimate the delay added by each router along the way.

## Network connections using netstat

**Question 4:** Report your observations for your `netstat -ps` run

**Question 5:** Report your observations for your `netstat -ap` run

**Question 6:** Report your observations for your `netstat -p` run observing the additional open ports.

## Packet watch and decoding using wireshark

**Question 7:** Report all observations you were asked to record in this section

## Further Thought

*This section is **not** part of your required assignment. Along with each week, we will offer directions and questions for further thought. Due to the nature of this course, we can only begin to glimpse the depth and richness of each of the topic areas. These questions offer some headings to contemplate further depth. These will often be open-ended questions. The Internet protocols were originally developed for best-effort (“send it as fast as possible”), traffic for applications like ftp, email, and remote login. The basic design we sketched follows this capability. How would the protocols need to change to handle real-time traffic (e.g. a phone call? video-stream?). How would you accommodate both kinds of traffic on the same physical network?*