LAB 10

Today's Lab has the following objectives:

1. Learn about the OSI Network Model.
2. Learn basics of TCP/IP.
3. Learn the basics of networking on a Linux machine.
4. Learn how applications communicate over a network.

Background:

Recall from lecture that a network is a collection of independent computers that exchange information with each other over a shared communication medium. There are different types of networks, namely: LANs, WLANS, WANs.

Network Types

LAN:

The acronym LAN stands for Local Area Network and it describes a configuration for a network that is typically confined to a limited geographical area (like a single building or a campus). LANs can be small (3 computers) but can link hundreds of computers as well.

WLANs:

Wireless LANs use radio frequency (RF) technology to transmit and receive data over the air. This minimizes the need for wired connections. The IEEE developed the 802.11 specification for wireless LAN technology, which specifies over-the-air interface between a wireless client and a base station, or between two wireless clients directly.

WAN:

Wide Area Networks (WANs) combine multiple LANs (or WLANs) that are geographically separate. Typically, a dedicated physical connection over a hard phone line, satellite link, or fiber optic is used to establish this network, but the amount of data that travels over a WAN is far less than the amount over a LAN, since special protocols usually minimize the amount of data flowing over a WAN as opposed to a LAN.

ESE 150 – Lab 10: Networking

What is a network protocol?

A network protocol defines rules and conventions for communication between network devices. It sets up a standard way for devices to communicate on a network. In the same way that there is a "protocol" (aka a set of rules) for meeting the Queen of England---perhaps your name being officially announced and then you bow or courtesy afterwards---it is a way to establish a standard repeatable and compatible form of communication!

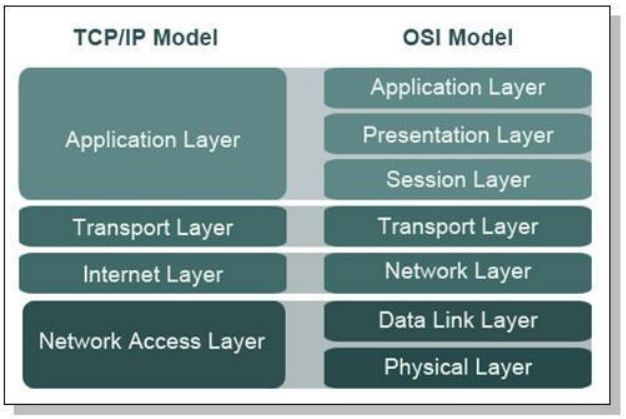
After a physical connection between two computers has been established on a network (e.g. – say a cable has been connected between them), network protocols define the standards that allow computers to communicate over that physical connection. A protocol establishes the rules and encoding specifications for sending data. The protocol defines how computers identify one another on a network, the form that the data should take in transit, and how this information is processed once it reaches its final destination.

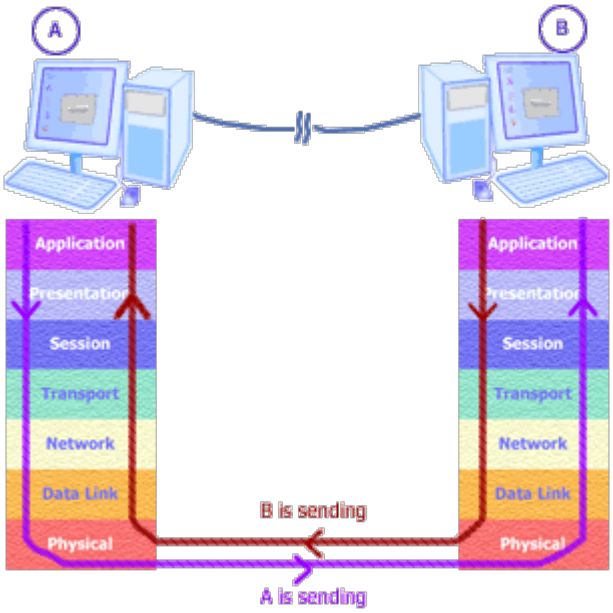
The most famous network protocol is: **TCP/IP (Transmission Control Protocol / Internet Protocol)**. But there are many others: AppleTalk, NetBIOS/NetBUI to name a few. Although each protocol is different, they can all share the same physical cabling (or wireless connection). This is called "protocol independence", which means devices which are compatible at the physical and data link layers in the OSI Model (talked about further below) allow the user to run many different protocols over the same medium.

ESE 150 – Lab 10: Networking

What is the TCP/IP Protocol?

The TCP/IP protocol defines a 4-layer "model" for how devices should communicate over a network. It is compared below to the OSI (Open Systems Interconnect) Model.





TCP/IP lacks the "presentation and session" layer of the OSI model, and the datalink and physical layer of the OSI model are merged into one "network access layer" in the OSI Model.

What is Ethernet?

Ethernet falls into the "physical layer" category of LANs. It is the most popular because it strikes a good balance between speed, cost, and ease of installation (most of you are probably familiar with Ethernet ports and cables!). The Ethernet standard, specified in IEEE Standard 802.3, defines the number of conductors that are required for a connection, a framework for data transmission, and the expected performance thresholds. The standard Ethernet can transmit data up to 10 Megabits per second (Mbps),

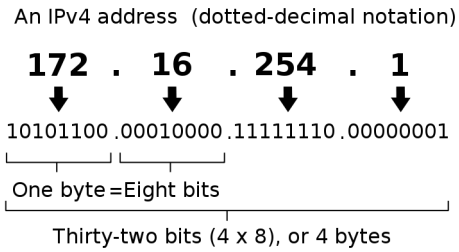
whereas other LAN types, like Fast Ethernet or Gigabit Ethernet, can transmit at 100 Mbps or 1000 Mbps. By following standard definitions of how elements of Ethernet networks are allowed to interact with each other, network equipment and protocols can communicate effectively.

IP Addresses:

Every device that wishes to use the "Internet Protocol" must have a unique number assigned to it known as an IP Address. The number is assigned to whomever administers the network one is a part of. Here at Penn, CETs is in charge of assigning IP Addresses to different devices on the network. The numbers are unique and make it so one computer can identify another computer by its number. It's a lot like a phone number.

IP Addresses come in two lengths: 32-bit and 128-bit. The 32-bit version is called "IP addressing scheme version 4 (IPv4)" and the 128-bit version is called "IP addressing scheme version 6 (IPv6)."

IPv4 addresses have the following breakdown, and you can see why they are 32-bits long, because there are 32 binary numbers. Notice the first example: 172 is the decimal representation of the binary number: 10101100. So, each decimal number we see has "8-bits" underneath it. There are always 4 zero or positive decimal numbers in an IPv4 address, the highest number for any single 8-bit field is 255.



The first two numbers (172.16 in the example above) are typically referred to as the network identifier. And the last two (254.1 in this example) are known as the host identifier. Typically, an organization (like Penn) will be identified by the network identifier and individual machines are assigned host identifiers.

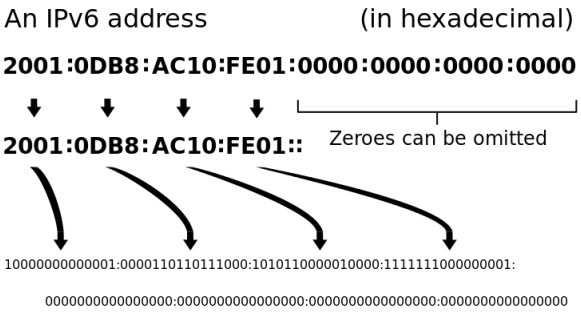
Let's say I purchase internet service from Verizon for my organization and they assign me the network identifier: 145.45. That would be the root for all my IP addresses in my organization. Let's say my first computer would be given the IP Address: 145.45.0.1, and the second: 145.45.0.2 and so on. But let's say I wish to further sub-divide my internal network. I could make the 3rd "octet" of my IP Address represent different departments, say:

145.45.0 represents the English Department.

145.45.1 represents the Math Department.

Then I could have nearly 256 computers in each department and I'd know which was in which department by looking at that third "field" in the IP Address.

In IPv6 there are simply more IP addresses available to assign to different devices. Notice the format of the IPv6:



It is no longer in "decimal", it is in what is known as hexadecimal (base 16).

IP Addresses and DNS Names

As you'll learn later in the lab, IP Addresses can be associated with what are called DNS names or network names. It's kind of like a phone number to name contact list. IP addresses go from very general to very specific. Let's say my organization is "tom.com" and I get the network identifier: 145.45 from Verizon when I purchase internet service from them. I can associate my computer called "www" with an IP Address, let's say I use 145.45.0.1 for my "www" machine. DNS will "map" these two things together:

Example DNS Record for tom.com:

www.tom.com 145.45.0.1

Notice, 145.45 is really associated with "tom.com" while the host identifier: 0.1 is associated with the computer named: "www". So, while DNS names go from specific (www) to general: (tom, then .com), IP Addresses go in the opposite direction: 145.45 (that's tom.com), .0.1 (that's www).

ESE 150 – Lab 10: Networking

Prelab:

Prelab – Section 1: IP Addresses

- Learn the basics of IP Addresses and their use in the TCP/IP Protocol.

This section assumes you have read the background above. If you haven't please do now!

1. Login to the ENIAC server using your personal computer. Linux and MAC users can directly ssh into the ENIAC server.

   ```
   ssh pennkey@eniac.seas.upenn.edu
   ```

2. When you login, eniac will tell you something like:

   ```
   Last login: Tue Mar 31 09:14:37 2020 from pool-108-2-150-73.phlapa.fios.verizon.net
   ```

   a. Record the name following "from" (pool-108-2-150-73.phlapa.fios.verizon.net in my example above)
   b. Include this name in your report.

3. Determine the "IP Address" of eniac by typing the following command at the shell window on eniac:

   ```
   /usr/bin/ifconfig
   ```

   a. "ifconfig" stands for "interface configuration". It shows you all the "network interfaces" you have on your computer. Think of your own laptop. You likely have a LAN connection you can plug a cable into, a Wi-Fi connection or a Bluetooth interface.
   b. Take a screenshot of the output of this command.
   c. You'll see the results of ifconfig are grouped by their interface. You may see "eth0", "em1", "em2", "lo" as examples. "eth0" would represent the "0th" or first network card in the computer with an Ethernet connection. Each of these names represent a network interface.
   d. Look at the top of each grouping, after the words: "Link encap".
   This tells you how the network interface is connected to the network. For hardwired and wireless connections, you'll commonly see the word "Ethernet" next to it.
   e. Next, you'll notice "HWaddr"; it is in HEX format, separated by colons:

   Example: C8:1F:66:C4:B6:3A, that's actually 48-bits long, as each number/letter represents 4-bits each. This address is a unique hardware address that the network interface has assigned to it from its manufacturer. This is very different from an "IP address", and it is often called a MAC address.

   f. Finally look for the address after the words "inet addr". The number that follows is your machine's 32-bit address. At Penn this number typically starts with: 158.
   g. Look lastly for your 128-bit address. It is located after the words "inet6 addr" and typically begins with "fe80". There may be many of them, but only one will begin with "fe80".

   h. Lastly, notice the "Local Loopback" interface. This isn't an actual physical interface, it is more like a logical "test" interface that always "loops back" to the network card itself. It always contains the 32-bit address: "127.0.0.1".
4. Answer the following questions (in your writeup):
   a. What was the name (e.g. eth0, vmnet1, em1, etc) of the network interface that contained the 32-bit IP Address starting with 158?
   b. What was the 32-bit IP Address?
   c. What is that 48-bit hardware address in hex form?
   d. What was the 128-bit IP Address (in hex form, as it is displayed in the output)?
5. Determine the IP address(es) of your local machine:
   a. If running on linux or OS-X, bring up a local terminal window and use the same command as on eniac. (or you may need to use `/sbin/ifconfig`)
   b. If running on windows, run the command `ipconfig /all` in PowerShell.
      i. The 32-bit IP address can be found under "IPv4 Address"
      ii. The 48-bit hardware address can be found under "Physical Address"
      iii. The 128-bit IP address can be found under "Link-local IPv6 Address"
   c. Include the IP address(es) you determine in your report.

Prelab – Section 2: Names of Computers

1. IP Addresses are long and hard to remember, kind of like phone #'s. We instead like to associate IP addresses with names! Much like you put phone #'s in your phone and associate them with people's names.
2. Not every IP Address is associated with a name, but it can be if it's an IP Address people need often. Here at Penn, you are familiar with our school's web server: www.seas.upenn.edu.
3. There is something called a "domain name server (DNS)" that keeps the association between names and IP addresses. Each organization maintains their own "name server." For example, Penn has a DNS that maps IP Addresses to names of computers.
4. You can run this section on eniac or on your local Linux, OS-X, or Windows machine
5. You can ask the DNS for the IP address of a computer if you know its name as follows:

   ```
   nslookup eniac.seas.upenn.edu
   ```

   This simply means "name server" lookup eniac.seas.upenn.edu. It will return with:

   ```
   Server: 128.91.18.2 (this is Penn's DNS computer's IP Address)
   Address: 128.91.18.2#53

   Name: eniac.seas.upenn.edu
   Address: 158.130.64.112     (this is ENIAC's IP Address)
   ```

   Some computers have many IP Addresses, this can simply mean that they have lots of "network interfaces" (e.g. connections to the network).
6. The "nslookup" utility also works in reverse. If you know the IP Address of a computer, if it has a name, it can tell you what it is! As an example:

   ```
   nslookup 158.130.64.112
   ```

This returns:

```
Server: 128.91.18.1 (this is Penn's DNS computer's IP Address)
Address: 128.91.18.1#53
112.64.130.158.in-addr.arpa name = eniac.seas.upenn.edu.
```

You may see something besides "eniac" in the name because eniac is actually composed of more than one server!

7. Find the IP Address for the machine that eniac believes you are logged in from.
    a. Perform nslookup on the hostname you recorded in Section 1, step 2.a
    b. Include the IP address in your report.
    c. This may not match the IP address you determined in Section 1, step 5
        i. Most likely your computer is connected behind a NAT (Network Address Translation) that is translating your local network address (Section 1, step 5) and the Internet visible address (what you found in step a).
8. Find the IP Address of Apple's website server: www.apple.com
    a. –submit this with your report too!
9. Your computer can also have a private name that can be different from its official "network name." Sometimes only the computer itself knows its private name. You can find out your computer's "host" name as it's called by typing in:

    `hostname` (submit this with your report)

10. Also, you can find it by typing in:

    `uname –a` (submit this with your report)

    a. *but this also returns all sorts of cool stuff, like what version of Linux you have, what type of computer architecture you have: x86_64 would be an Intel based 80x86 computer with a 64-bit word size for example!*

Prelab – Section 3: Testing Connections

1. PING is a utility that can send a "test" signal from one computer to another computer via the IP Address or its DNS registered name. Administrators like to use this utility to see if another computer is on and properly on a network. You may find it useful, too, if you're trying to do the same!
2. You can run this section on eniac or on your local machine; if possible, run it on your local machine.
3. Try "pinging" www.seas.upenn.edu by typing in the following command:

   `ping -c 4 www.seas.upenn.edu`

   If you're on Windows, use `ping -n` instead of `ping -c`

```
PING eniac-l.seas.upenn.edu (158.130.64.112) 56(84) bytes of data.
64 bytes from plus.seas.upenn.edu (158.130.64.112): icmp_seq=1 ttl=64 time=0.035 ms
64 bytes from plus.seas.upenn.edu (158.130.64.112): icmp_seq=2 ttl=64 time=0.036 ms
64 bytes from plus.seas.upenn.edu (158.130.64.112): icmp_seq=3 ttl=64 time=0.039 ms
64 bytes from plus.seas.upenn.edu (158.130.64.112): icmp_seq=4 ttl=64 time=0.036 ms

--- www.seas.upenn.edu ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.035/0.036/0.039/0.006 ms
```

Note: copying "ping -c" may not work. If you have trouble, retype it instead of trying to copy it.

4. How to interpret the above output?
   a. You asked ping to send "4" packets of data over to eniac.
   b. Each "packet" of data contained 64 bytes of data.
   c. The "round trip" time it took to for the 1st packet get back and forth was 0.035ms.
   d. This can be different each time depending on what route through the network the packet took, or how busy the router was that routed the packet to eniac.
   e. But on the average, it transmitted 4 test packets of data to eniac, and eniac responded each time with an average speed of about 0.036ms!
   f. It's not a precise timing measurement, but it's a good average number.
6. Include a screenshot of your output from above.
   a. What is your average roundtrip time?
7. Try pinging www.apple.com from eniac:
   a. Save the results to hand in with your report (screenshot).
   b. What was the average time?
8. Submit select prelab answers to the Lab 11 prelab assignment on canvas: (Section 1: 2, 4a, 4b, 5; Section 2: 7b, 8; Section 3: Screenshot from 7).

Prelab – Section 4: Wireshark

1. Download the Wireshark software for your computer here: https://www.wireshark.org/#download

Lab Procedure:

Lab – Section 1: Routes between computers

1. If possible, perform this on your local computer; as a fallback, use eniac.
2. Bring up a terminal (Linux, OSX) or PowerShell (Windows).
3. *Traceroute* is another handy shell utility that works like PING, except it records what machines (hubs/switches/routers) it sends the packets through. So let's say you "ping" apple.com, traceroute can show you the "route" the packets took to get there.
4. Run traceroute between your computer and www.uw.edu:
    a. Type one of the following:
        i. `Windows: tracert www.uw.edu`
        ii. Linux/OS-X: traceroute www.uw.edu
        iii. eniac: /usr/sbin/traceroute www.uw.edu

    Following is an example from tracert www.apple.com.

    The utility will output a list, showing each piece of equipment it went through to get to apple.com, and then to the "www" machine on apple.com's network:

```
traceroute to www.apple.com (104.97.94.156), 30 hops max, 60-byte packets
 1 158.130.0.233 (158.130.0.233) 25.881 ms 34.633 ms 25.846 ms
 2 isc-ist.seas.upenn.edu (158.130.0.250) 0.278 ms 0.293 ms 0.284 ms
 3 vag-brdr.i2trcps-ashb.router.upenn.edu (128.91.238.222) 4.480 ms 4.495 ms 4.4ms
 4 18.ae15.pr0.dca10.tbone.rr.com (107.14.16.81) 4.478 ms 4.469 ms 4.460 ms

      … (I've deleted some of the route)

       8 bu-ether12.nycmny837aw-bcr00.tbone.rr.com (66.109.6.27) 7.463 ms 12.549 ms 12.983
      ms
       9 0.ae0.pr1.nyc20.tbone.rr.com (107.14.17.216) 12.039 ms 7.553 ms 66.109.1.59
      (66.109.1.59) 7.525 ms
10 a104-97-94-156.deploy.static.akamaitechnologies.com (104.97.94.156) 7.724 ms 7.754 ms
      7.544 ms
```

    b. This is just the first 4 "hops" on your packet's route to apple.com and the last 3 as well (8, 9, 10). Yours will likely be different than mine. The packet leaves a Penn Moore School computer and goes most likely to a switch identified by IP address: 158.130.0.33. Then that devices forwards the packet and it forwards it to a UPENN router named: vag-brdr. This router then forwards your packet to our internet provider and it keeps on going all the way to "akamaitechnologies.com"---a content provider that is likely hosting apple.com's webserver!
    c. Each of these IP addresses is location somewhere different around the country.

   d. Some of the hops may come back * * * --- something timed out on these hops; it's ok to ignore these assuming you got a good set of hops between the source and the destination to decipher the route that your packets take along the way.
   e. Record the full set of hops for use in the postlab.

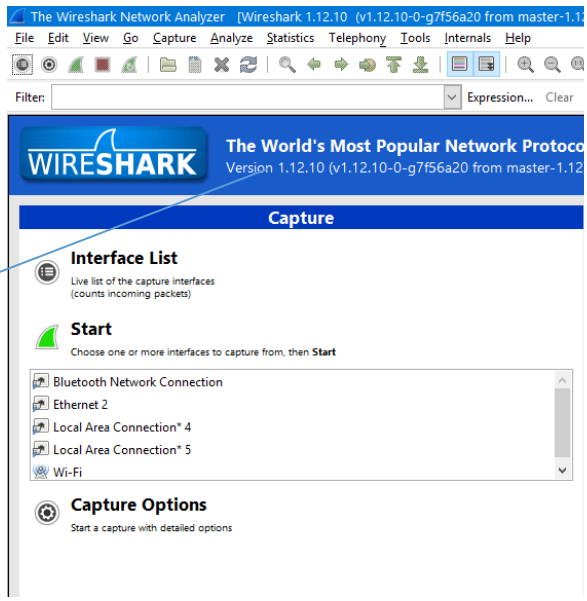
5. Run tracert between your computer and "eniac.seas.upenn.edu" (unless you are running this on eniac, in which case run tracert to the address you found in Preclass Section 1, step 2.a).
   a. Record the route.
   b. In your lab report, note the differences between your route and that in step 4.

Lab – Section 2: Investigating the packets

- Recall from lecture that a packet is the actual data that you send from one computer to another. Usually it is a fixed size. Let's say you're sending 128 kBytes to another computer. TCP/IP may break that up into 128 packets with about 1 kBytes each. This way if something goes wrong with one of the packets, it can just re-send one small packet, instead of re-sending the entire 128kBytes in one shot.
- Also recall from lecture that a packet not only contains the data you want to send, but information about where the packet came from and where it went to!
- We can look at the packets if we like using either an oscilloscope, or something a bit higher level, a software "packet sniffer" or a "network analyzer"
- In this section, we'll use a utility called: Wireshark (it is a software-level packet sniffer)

1. Use ifconfig to determine the IP address for your laptop and the name of the configured network interface.
    a. This is what you did in Prelab Section 1, Step 4.a – if you did that on your own machine, you should already have the answer.
2. Start Wireshark from wherever you installed it. You may need to select which network interface to use.

If properly installed you'll see something like this:



Click on the network interface name that corresponds to the IP address from Step 2:
Next, click on the "start" or green shark fin button.

At this point, you should be seeing a growing list. It is a copy of every packet being handled by your network interface!

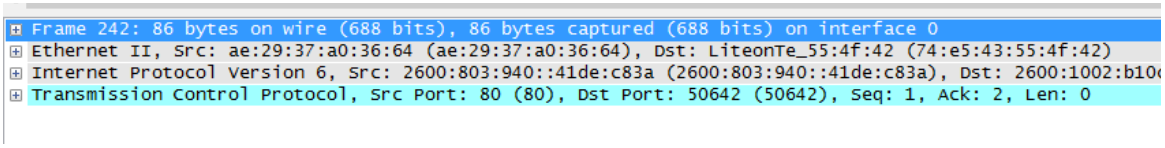
3. Back in your Linux terminal type in *(or your computer's terminal if you are using your own version of Wireshark)*.

   ```
   nslookup www.seas.upenn.edu
   ```

   Record the IP address.

   ```
   ping –c 4 (the IPv4 address of eniac)
   ```
   or for Windows,
   ```
   ping -n 4 (the IPv4 address of eniac)
   ```

4. Press the "stop" button in Wireshark.
   a. Wireshark has just copied all of the ping "packets" you sent to eniac
   b. It's kind of like a software/network oscilloscope!
5. Next, find the PING packets:
   a. There are many packets leaving/coming to your computer all the time. So, finding them can be difficult.
      i. If there are too many, you can try starting Wireshark again without saving, pinging eniac again, and stopping immediately after.
   b. Sort the list at the top by "destination" and scroll until you see eniac's IP address.
   c. Make sure that the "source" is the IP address of your computer.
   d. Note, the protocol is "ICMP" that's for Internet Control Message Protocol. It's a protocol used with "IP" in TCP/IP for "control" signals send between computers. Like "ping".
   e. You should see 4 packets (so four entries).
6. Click on the first packet: look at the "info" column on the right-hand side, look for "echo (ping) request" but the one with the lowest "sequence #" (seq).
   a. With the packet highlighted, look to the middle window, which shows you four components of the packet:



7. Let's examine these components:
   a. Click on the first line: "frame" and look to the window at the bottom:

   Notice all that "hexadecimal" data? That is the actual message that was sent between your computer and eniac. When you click on "frame" it highlights the entire message, as this packet is considered a "frame".
   b. Now click on the second component in the middle window: "Ethernet II"
      i. Notice how it only highlights some of your packet in the window below?
      ii. That's because only that part of the packet is for the "Ethernet II" layer of the TCP/IP model.
      iii. Click on the little "plus" or "arrow" sign next to Ethernet II to expand it.

1) Notice the "source" information: that's your computer. Click on it and see which part of the packet is highlighted.
2) Look at the destination information as well! Which part of the packet makes up the destination?

c. Click on the "Internet Protocol Version 4…" component to note which part is highlighted below. Expand this section and scroll through to look at the different components.
d. Lastly, expand and look at the "Internet Control Message Protocol" component:
    i. This is the Data inside the ICMP (or ping) packet. Identify the sequence number and type encoding that says it is a ping packet. Include these in your report.
    ii. Identify all the ping response packets. Inside a ping response packet, identify the type encoding for the response. Identify the round trip time and relate it to what ping printed in your console window. Include these in your report.
e. Lastly, record the entire request packet (the hex data)
    i. You can copy the data in the bottom panel by right clicking it selecting "Copy Bytes as Hex + ASCII Dump"
f. Make an annotated version indicating which parts of the hex belong to the following parts of the frame (you can click on the section in the middle pane, and the hex associated with it will be highlighted in the bottom):
    1) Ethernet II
        a. Source
        b. Destination
    2) IP Protocol Version 4
        a. Source
        b. Destination
    3) ICMP
        a. Data
        b. Sequence number
        c. Type encoding

Lab – Section 3: Connecting to a webserver on port 80

- A webserver is just a "server" program running on remote computer (say apple.com).
- Its purpose is to "serve" up webpages to clients requesting a webpage within a website.
- A webserver program is set to "listen" for incoming clients on port #80 on a host (like www.apple.com).
- When a client program (like a web browser - Chrome for example) opens up a socket to www.apple.com on port #80, the web server program on www.apple.com is identified by port #80 and allows it to connect. Then data is sent back (in the form of web pages) to the client.

1. Make a new directory (mkdir) called ese150_lab10b and change the directory (cd) into it.
2. Restart Wireshark capture (refer to Section 2 for more detailed instructions).
3. Open up a web browser (any one will do) and type in the following address manually: http://ic.ese.upenn.edu:80/research
   Once you get a response, stop packet collection in Wireshark.
   a. What does this mean? By specifying the ":80" your browser is acting as a client, asking the server named: ic.ese.upenn.edu to talk directly to the process listening on PORT #:80.

   You can also simply type in: http://ic.ese.upenn.edu/research and the same thing will happen; web server's typically use port 80 to listen for incoming requests.
   b. You could also type the webserver's IP Address directly. Your browser doesn't need to work as hard. You specified the IP Address of the host, so now it doesn't need to go to the DNS server first!
   c. What does "http" mean? It stands for "**h**yper **t**ext **t**ransfer **p**rotocol".

   You have specified what protocol you'd like the browser to communicate with the remote server. This "http" protocol is used for exchanging "hyper text" files, or what is called: "HTML" files (Hyper Text Markup Language).
   d. A "web browser" is a client that requests and can even display ".HTML" files. When it receives a .HTML file from a remote server, it opens it up and interprets the information inside as the text and graphics you see on your screen!
   e. Include a screenshot of your web browser output from http://ic.ese.upenn.edu/research in your report.
   f. From the Wireshark capture, select the http protocol packets and identify the request to ic.ese.upenn.edu and the response.
      i. Within Wireshark, sort by "Protocol".
      ii. Next, locate the HTTP protocol, and click into the packets. Within each packet, there should be more information within the "Hypertext Transfer Protocol" section in the middle pane.
      iii. Look through each of these packets to identify the request/response. You may have to open up some of the subsections.
      iv. Take a screenshot of the request and beginning of the response and include with your report. Make sure to include the hex code in the bottom pane.
      v. To save the captured information, make sure the application is stopped (red square in top left), and then go to File -> Save As -> filename. To see this information again, go to File -> Open Recent -> filename.

vi. If you did not manage to capture it on the first try, capture again. It's possible your browser and web server will figure out that the content hasn't changed and won't resend a full response. In that case, it may be necessary to flush the browser cache to see the response the second or subsequent times.

4. Let's use a different "client" to look at the ic website.
   a. If you're on Windows, you'll have to SSH into eniac to do this.
   b. Open a terminal and run the following command:

```
telnet 158.130.67.172 80
```

   c. "TELNET" is a simple client program that can open up a socket to the IP address you specify and to the PORT # you've specified. Above, you've opened up a socket to www.seas.upenn.edu, and telnet is connected directly with the process running on port 80 (which is the webserver).
   d. Telnet shows its progress:

```
Trying 158.130.67.172...
Connected to alliance-vhosts.seas.upenn.edu.
Escape character is '^]'.
```

--This means it connected to ic.ese.upenn.edu:80.
--But, ic.ese.upenn.edu:80 expects a "web browser" as a client, not telnet.
--So, we must pretend to be a web browser and send the type of information a web browser typically sends to a web server.

   e. Send the following information to ic.ese.upenn.edu, by typing it in (or copying and pasting):

```
GET /research.html HTTP/1.1
Host: ic.ese.upenn.edu
User-Agent: telnet
```

   f. To paste into PowerShell, copy the text, then right click the icon in the top left, then select Edit > Paste
   g. If you're not working on eniac, you can watch what happens on Wireshark.
   h. Clear your Wireshark packets (green rewind fin or blue fin), then press <enter> twice.
   i. If you typed things in correctly, you'll see the contents of the ".HTML" file research.html on the ic.ese.upenn.edu server. Pause your Wireshark at this point.
   j. TELNET doesn't really know how to display that data, other than just showing its contents to you. It doesn't understand that the .HTML file that has returned could be interpreted as graphics and formatted text; only a web browser knows how to do that.
   k. You can exit out of telnet by typing the word "exit".
   l. Identify the packet content that corresponds to this in the Wireshark packet capture you made in Step 3.

5. Let's "save" what telnet gets back from ic.ese.upenn.edu:

a. Repeat the command from 4a, but this time we'll "save" the output to a file with the following modification:

```
telnet 158.130.67.172 80 > research.txt
```

b. This time, you won't see telnet's status, instead, that output is being stored in research.txt.
c. Manually type in the following (EXACTLY AS IT'S SHOWN):

```
GET /research.html HTTP/1.1
Host: ic.ese.upenn.edu
User-Agent: telnet
```

d. Press <enter> twice, wait for about 60 seconds and it will close the connection.
e. You won't see any data returned from ic.ese.upenn.edu, but it was all stored in the file: research.txt.
f. If you're working on eniac, you'll have to copy the file over to your machine with scp: `scp PENNKEY@eniac.seas.upenn.edu:~/research.txt research.txt`
g. Open up research.txt in a text editor.
h. Notice, it is that "HTML" stuff from before! This has all come back from ic.ese.upenn.edu and was stored in your file: "research.txt".
i. Delete the top lines of the file, as shown below (your version may be a little different):
    i. You want to delete up to the line containing the <!DOCTYPE HTML … header

```
Trying 158.130.67.172...
Connected to alliance-vhosts.seas.upenn.edu.
Escape character is '^]'.
HTTP/1.1 200 OK
Date: Sun, 15 Apr 2018 00:16:58 GMT
Server: Apache
Last-Modified: Sun, 15 Jan 2017 02:50:39 GMT
ETag: "1c24-5461921daa0cf"
Accept-Ranges: bytes
Content-Length: 7204
Content-Type: text/html
```

j. Now, save the file. You will need to include research.txt in your submission.
k. Copy the file and change the file name to "research.html".
l. Now, open the file `research.html` in a web browser.
    i. If you have a default browser set up, you can usually just double-click on the file, and the browser will open automatically.
    ii. If this doesn't work, try dragging the file into the address bar of your browser.

Look at the "address" bar and see the address looks like this:

`file:///mnt/castor/seas_home/c/pennkey/.../research.html`

You've asked your web browser to look at a file stored on your computer (not a website ic.ese.upenn.edu).

Notice, some of the images do not show up. That's because telnet didn't download all the graphics associated with the webpage, but for the most part, we've manually done what a web browser does with .HTML files manually!

m. Turn in research.txt and research.html with your writeup. Explain what they are and what they contain.

6. **Exit checkoff** : Show a TA the following and answer questions:
   a. Packet recorded from Wireshark in Section 2: step 8e and annotated in step 8f; be able to point out the fields identified there.
   b. `research.html` file you created in step 5 by editing the `research.txt` that you captured.

Post-lab:

Most of the answers to these questions come from the lab procedure itself, or the lab background. You won't need google for much of anything here!

1) Can a computer communicate using different network protocols over the same physical medium (i.e. – cables?)

2) What is the advantage of IPv4 vs. IPv6?
    a. How many devices total could be represented with IPv4? How many with IPv6?

3) Trace the geographical locations for your tracert/traceroute to www.uw.edu (from Section 1, Step 4) using https://www.iplocation.net/, and create a map displaying each.
   You can use a tool like this to plot all points on a single map: http://www.darrinward.com/lat-long/
   Note: iplocation.net won't give meaningful locations for the Internet2 sites. Instead use: https://routerproxy.wash2.net.internet2.edu/routerproxy/
   You may also find this map useful: https://www.internet2.edu/media/medialibrary/2018/07/16/Internet2-Network-Infrastructure-Topology-All-legendtitle.pdf
   Internet2 is a not-for-profit United States computer networking consortium led by members from research and education communities, industry, and government. For the relevant "hops", use the Internet2 resources.

4) From your Wireshark message, what exactly did a "ping" message contain? It is actual data, readable by a human, what is it?

5) Why is a PORT # necessary? Why can't a client process simply indicate the IP Address of the computer it would like to communicate with?

ESE 150 – Lab 10: Networking

HOW TO TURN IN THE LAB

- Each individual student should submit a PDF document to canvas containing:
    - Recorded data and descriptions/explanations where asked.
    - There is output we require in each section; it is your responsibility to look through this writeup and ensure you turn in each component required. They should be highlighted in yellow.
    - Clearly label what each screenshot is in your report (or you lose credit).
    - Answers to questions asked in the prelab and lab.
    - Answers to post-lab questions.