Exercise 1: Wireshark

Name: ______________________________

Sunet ID: ______________________________@ stanford.edu

Background

Wireshark is a tool for inspecting packets sent/received on a network interface. There are two modes: Open and Capture. Capture mode shows you a live stream of the packets currently going to/from the interface, which you can then save to a pcap file if you like.
Open allows you to inspect a pcap previously generated by some capture.

If you want to look at two pcap files simultaneously, the best way I've found is to start two instances of Wireshark -- e.g. on Mac, `open -n /Applications/Wireshark.app`

Pcap 1: Ping

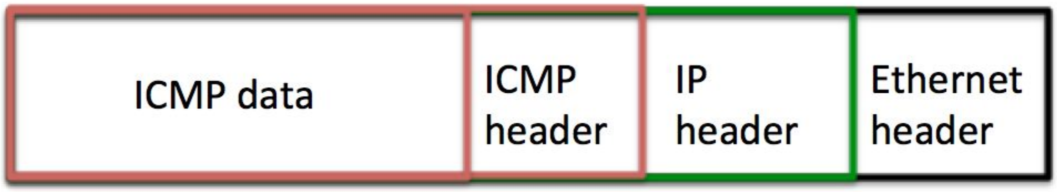
Open `ping.pcap`, which captures a single ping from one host to another.

Don't worry about the details of the ARP packets for now. We'll learn later that ARP is a discovery protocol for finding the Ethernet address to use when sending to a local IP address.

1. What is the IP address of the host being pinged? **192.168.2.1**
2. What are the 3 layers in packet 1, starting with the outermost?
   Outermost: **Ethernet** Middle: **IP** Innermost: **ICMP**
3. Does the innermost protocol identified in (2) use ports? **No; ICMP doesn't need to multiplex across multiple applications like TCP does**
4. For packet 1, label the length (in bytes) of each portion on the diagram.
   Hint: The lengths should sum to 98 (the total length of the packet)

Wireshark shows lengths in bottom left when you click on the header, although for the ICMP header it includes the payload length)

48 16 (64-48) 20 14



Pcap 2: SMTP

Open `smtp.pcapng`, which captures an SMTP conversation similar to the one in lab 0.

In your answers, use the Wireshark packet number (the "No." column) to identify packets.

To make the TCP sequence/acknowledgement numbers easier to understand, set up Wireshark to display them relative to the first packet: Wireshark -> Preferences -> Protocols -> TCP -> check “Analyze TCP sequence numbers” and “Relative sequence numbers”

1. What port does the SMTP server run on? **25**
2. What port does the client run on? **51319**
3. What protocol does SMTP run on top of? **TCP**
4.
    a. Which packet represents the telnet request? **1**
    Hint: you won’t see the word “telnet” explicitly - but remember that the telnet request initiates a connection over the protocol you identified in (3).
    b. Which flag in the packet identified in (a) tells the SMTP server that this is the beginning of the connection? **SYN**
    c. Which packet contains the 220 response from the SMTP server? **4**
5.
    a. In which packet does the client first acknowledge the 220? **5**
    b. What is the ACK number of the packet acknowledging the 220 (i.e. the packet identified in b)? **50**
    c. You should see that the **ACK** number is one more than the length of the 220 response’s payload, meaning the client had **received** one byte in addition to the 220 by the time it acknowledged the 220. What data was in the byte the client **received** before the 220? **SYN**
    d. Notice that the **sequence** number of the packet acknowledging the 220 is 1, meaning the client had already **sent** one byte by the time it acknowledged the 220. What data was in the byte the client **sent** before the 220? **SYN**
6. Wireshark has flagged packets 15 and 17 as duplicates. Which packets do they duplicate?
Packet 15: **14 (same seq/ack/length; Wireshark also tells you under SEQ/ACK Analysis)** Packet 17: **16**
7. Notice that this pcap only contains packets involved in the email conversation, even though the computer that sent the email had lots of other network traffic going on at the same time. That’s because the capture was created using the capture filter “`tcp port smtp`”. In addition to capture filters, Wireshark also has display filters, which narrow down the packets displayed. For instance, we can filter out TCP packets with no payload, leaving only the packets containing the client’s requests and the server’s responses. Type this into the display filter box below the toolbar: “`tcp.len > 0`”. How many packets are displayed when this filter is applied? **19**

Pcap 3: Traceroute

Open `traceroute.pcap`, which captures a traceroute from a VM to MIT. Below is the partial output of the traceroute:

```
traceroute to mit.edu (104.83.252.128), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  1.384 ms  1.288 ms  1.141 ms
 2  192.168.0.1 (192.168.0.1)  16.188 ms  16.088 ms  16.021 ms
 3  96.120.91.229 (96.120.91.229)  10.174 ms  10.112 ms  10.849 ms
 4  be-20052-rur02.santaclara.ca.sfba.comcast.net (68.87.196.49)  12.210 ms  12.515 ms
12.448 ms
 5  162.151.78.129 (162.151.78.129)  11.981 ms  12.304 ms  12.223 ms
 6  be-232-rar01.santaclara.ca.sfba.comcast.net (162.151.78.253)  11.784 ms  9.530 ms
12.152 ms
 7  be-3651-cr02.sunnyvale.ca.ibone.comcast.net (68.86.91.73)  12.071 ms  11.576 ms  11.788
ms
 8  be-11083-pe02.529bryant.ca.ibone.comcast.net (68.86.84.14)  11.357 ms  11.604 ms
11.507 ms
 9  75.149.231.242 (75.149.231.242)  12.913 ms  13.152 ms  12.740 ms
10  203.208.149.250 (203.208.149.250)  21.001 ms  20.906 ms
    203.208.172.233 (203.208.172.233)  12.462 ms
11  203.208.149.254 (203.208.149.254)  23.141 ms *  22.830 ms
12  * * *
13  203.208.192.162 (203.208.192.162)  166.854 ms
    22rrnpr02-hu0-6-0.npr.optusnet.com.au (210.49.108.54)  165.978 ms
    203.208.190.138 (203.208.190.138)  182.836 ms
14  * * *
15  22rrnpr02-hu0-7-0.npr.optusnet.com.au (210.49.108.62)  166.082 ms
    22rrnpr01-hu0-6-0-1.npr.optusnet.com.au (210.49.112.114)  166.618 ms
    22rrnpr02-hu0-7-0.npr.optusnet.com.au (210.49.108.62)  172.404 ms
```

Note: you can ignore packets 1-4 in the pcap; they are part of another communication.

1.
    a. What is the IP of the host requesting the traceroute? **192.168.0.102**
    b. How does this host determine MIT's IP address? Hint: See packets 5-8. **DNS**

2. After determining MIT's IP, the source host begins sending packets to MIT.
    a. What is the innermost protocol of these packets? **UDP**
    b. How many packets does it send before getting the first response? **13**
    What is the TTL of the last packet sent before the first response? **5**

What do you notice about the source and destination ports of the packets sent to MIT? **Source is random and different for each; destination starts with a random number and increases by one for each**

c. Which packet is the first response responding to? **9**
Hint: The ICMP payload of the response packet contains part of the packet which prompted the response. The ports may be helpful in differentiating packets.

3. Look at the traceroute output for hops 10, 13, and 15. What is different about the output for these hops?
Multiple hosts responded (multipath)

    a. Which packets did the source send to prompt the responses from hop 10? To confirm your answer, check that the source/destination ports match. **61, 62, 63 (destination ports are 33461, 33462, 33463, coincidentally)**
    Hint: You can filter for a TTL of ***x*** with `ip.ttl ==` ***x***.
    Also note that the traceroute was run from a VM, so the first “hop” is to the laptop running the VM (IP 10.0.2.2). Unlike a router, the laptop doesn’t decrement the TTL, so the router listed in the traceroute output as hop 2 (IP 192.168.0.1) is actually responding to packets sent with TTL 1.
    b. Subtract the timestamp of the packet sent to host 203.208.172.233 from the timestamp of the corresponding response. How does this compare to the RTT to 203.208.172.233 reported by traceroute? (It should match to the nearest integer number of milliseconds). **About the same; 12 ms (subtract packets 74 and 63)**
    Hint: You can filter for ICMP TTL exceeded packets with `icmp.code == 0`

NOTES:

Example traceroute

```
nickm@yuba.Stanford.EDU (nickm) 21 > traceroute www.mit.edu
traceroute to www.mit.edu (23.213.120.46), 30 hops max, 40 byte packets
 1  csee-west-rtr-vl3874.SUNet (171.64.74.2)  0.183 ms  0.224 ms  0.190 ms
 2  dc-svl-rtr-vl2.SUNet (171.64.255.190)  0.542 ms  0.522 ms  0.445 ms
 3  dc-svl-agg4--stanford-100ge.cenic.net (137.164.23.144)  1.528 ms  2.280 ms  2.247 ms
 4  dc-svl-agg4--svl-agg8-100ge-1.cenic.net (137.164.11.28)  0.864 ms
svl-agg4--svl-agg8-100g.cenic.net (137.164.11.64)  0.928 ms
dc-svl-agg4--svl-agg8-100ge-1.cenic.net (137.164.11.28)  1.295 ms
 5  10-1-1-91.ear1.SanJose1.Level3.net (4.15.122.45)  0.995 ms  1.214 ms  1.530 ms
 6  * * *
 7  NTT-level3-4x10G.SanJose.Level3.net (4.68.62.206)  2.390 ms  2.326 ms  2.289 ms
 8  ae-1.r02.snjsca04.us.bb.gin.ntt.net (129.250.3.59)  1.645 ms
ae-1.r01.snjsca04.us.bb.gin.ntt.net (129.250.2.229)  2.461 ms
ae-1.r02.snjsca04.us.bb.gin.ntt.net (129.250.3.59)  1.602 ms
```

```
 9  ae-1.a02.snjsca04.us.bb.gin.ntt.net (129.250.3.103)  2.754 ms  2.482 ms
ae-0.a02.snjsca04.us.bb.gin.ntt.net (129.250.2.3)  1.312 ms
10  * * *
```