**Problem #1**

**Q1: What is your device IP address, are there any rows where you do not see it?**

A1: 10.0.0.90, UDP packets did not display my devices IP addresses.

**Q2: How many packets in total were captured?**

A2: 609

**Q3: What are the ratios of each the UDP, and TCP packets from the overall number of packets?**

UDP >>> 530:609

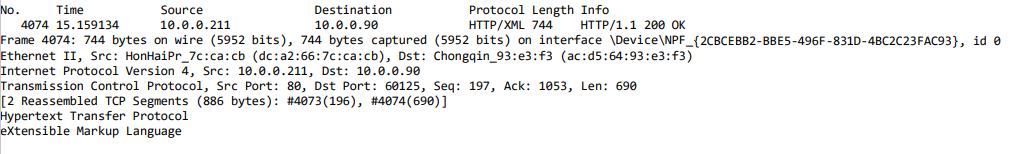
TCP >>> 34:609

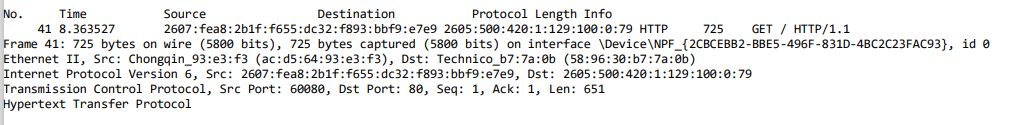
**Q4: What is the IP address that corresponds to the http link requested?**

The corresponding IP address is 129.100.0.79



**Q5: Print the two HTTP messages (GET and OK)**





**Q6: How much time did it take to deliver the website to your device?**

It took approximately 6.79 seconds, based on the timestampgs of the Get and Ok messages.

**Q7: How much longer did it take the second time when you made the same request for the page?**

It took six seconds less than the first time.

**Q8: What do you attribute the change in the time between the two?**

I took less time because the page is already in the memory cache.

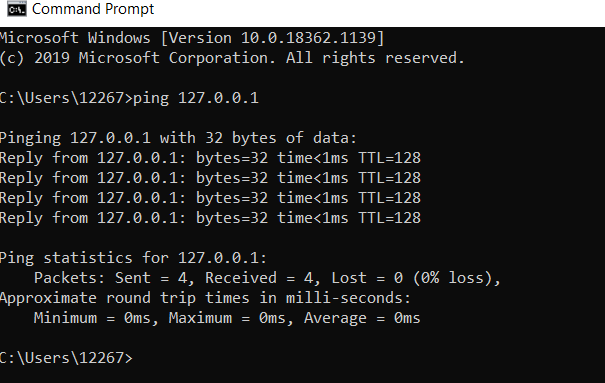
**Problem #2**

**Q1: Use ping on your workstations to familiarize yourself with these commands. List a few of the flags that are used with these two commands?**

ipconfig

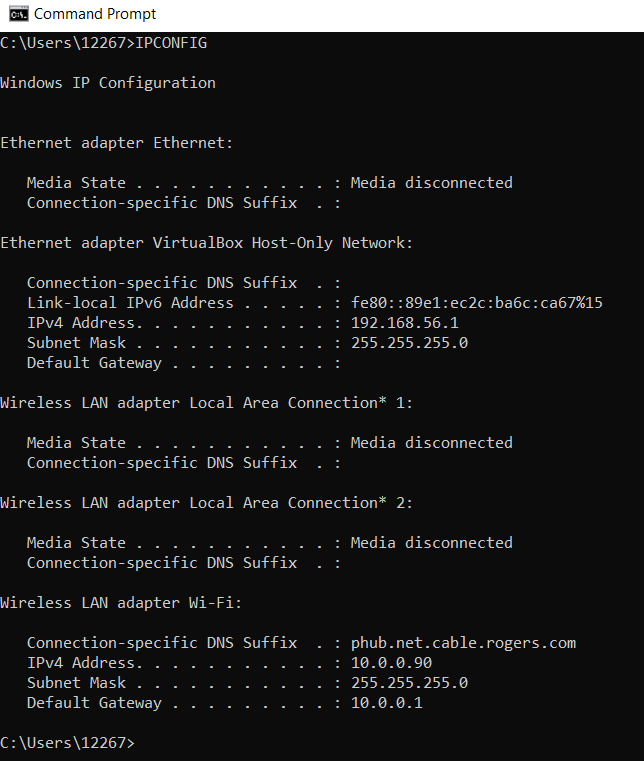
**Q2: Ping your workstation’s loopback interface (localhost). What command did you use? Was the ping successful?**

ping 127.0.0.1, the ping was successful.



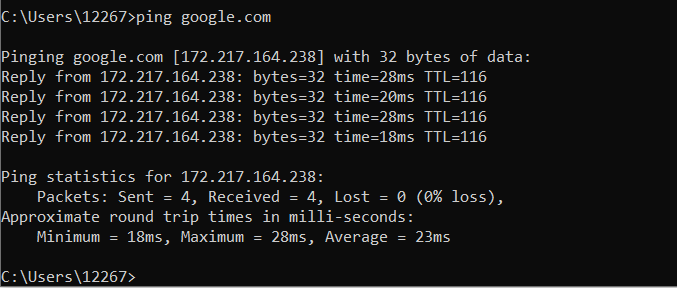
**Q3: Ping your default gateway and your DNS server. What command did you use?**

I used ipconfig.



**Q4: Ping google.com, What is the IP address of the computer you pinged? What are the**

**minimum, average, and maximum round trip times?**



IP address: 172.217.164.238

Minimum rtt: 18 ms

Maximum rtt: 18 ms

Average rtt: 18 ms

**Q5: Use ping to measure Round Trip Time (RTT) for 10 messages 1024 bytes. Use the "-f" configuration switch to make sure that the message is not fragmented. Graph the message size versus RTT for two nodes on a WAN (alibaba.cn, google.ca). Calculate the average and standard deviation for every destination. Discuss the effects of distance and their relationship with latency.**

**Q6: Use the tracert utility on your workstation to find the route to a host**

**i) In Toronto (google.ca): How many hops did it take to reach the destination host?**

Text

Description automatically generated

It took 9 hops.

**ii) In China (Alibaba.cn): How many hops did it take to reach the destination host?**

**Text

Description automatically generated**

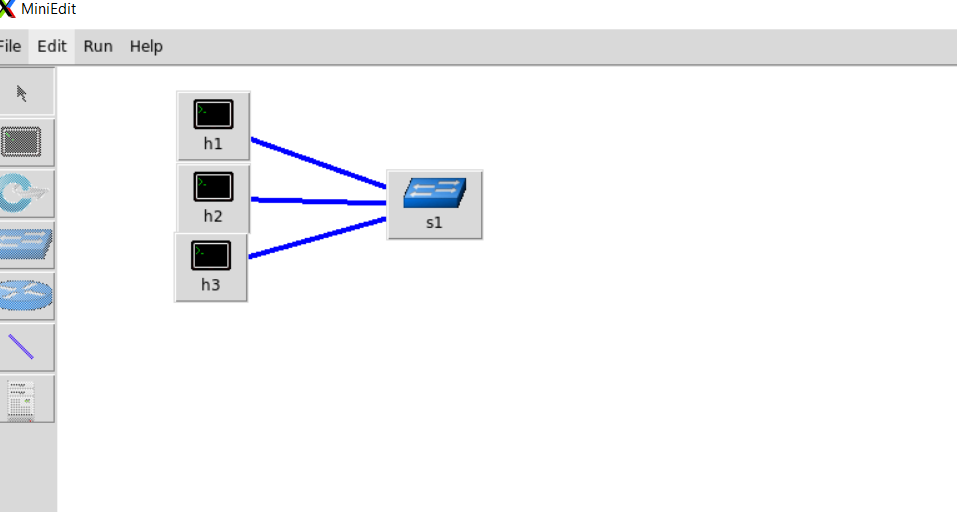
It took 23 hops.

**iii) Why do you see “\* \* \*” on some of the output lines?**

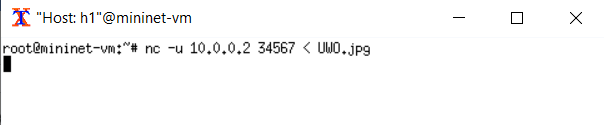
It meant that the request had timed out upon further research this is because “many people block ICMP/ping for security reasons, like preventing hackers from getting information about open ports and staving off denial of service attacks. When ping is blocked, the server doesn’t respond at all, resulting in “request timed out” messages that prevent traceroute from ever being able to map the path to the final destination.” -Susan (Spiceworks)

**Problem #3**

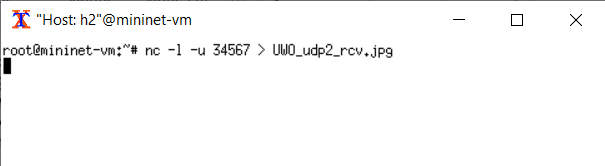
**Q1: create a star topology network consisting of 3 hosts and one legacy switch**



**Q2: Host 1 & 2 Terminal Code**







**Q2-1: How long did it take to transfer the file?**

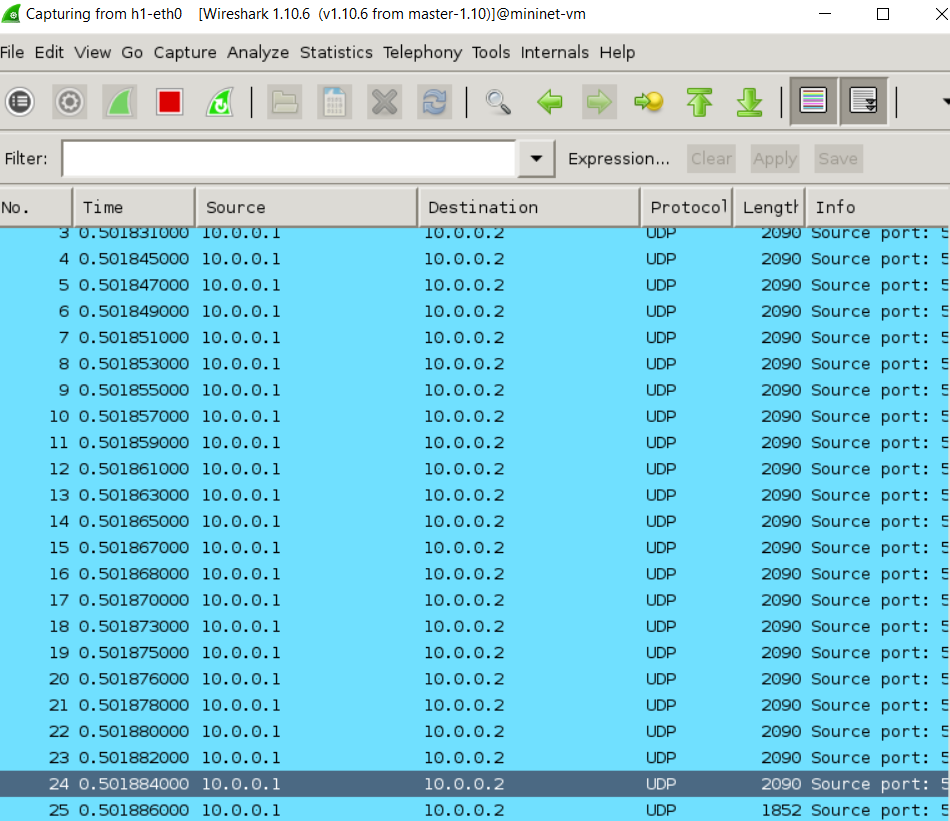
The file transferred at a very fast rate at the start but then just paused a third of the way through transmission. 22 UDP packets were transmitted in 0.000055 seconds, before the program stopped transmitting. The total transmission time till stop was only 0.501886 seconds as shown in the Wireshark screenshots below.





**Q2-2: Are there UDP packets in both directions, i.e. from h1 to h2 and from h2 to h1? Justify your answers.**

No UDP packet were only transmitted from h1 to h2 not in both directions. This is shown in the Wireshark screenshot below.

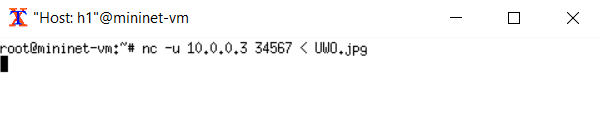


**Q2-3: Is the received image the same as the original test file? Do they have the same size? Indicate the sizes.**

In theory the received image should be the same size as the original test file, however since the transmission did not complete there is a difference. The original file is 125 KB and the received file is 45 KB.



**Q3: Host 1 & 3 Terminal Code**



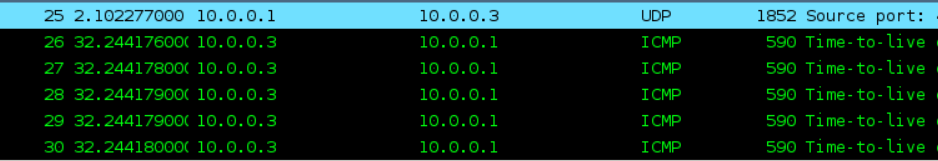




**Q3-1: How long did it take to transfer the file?**

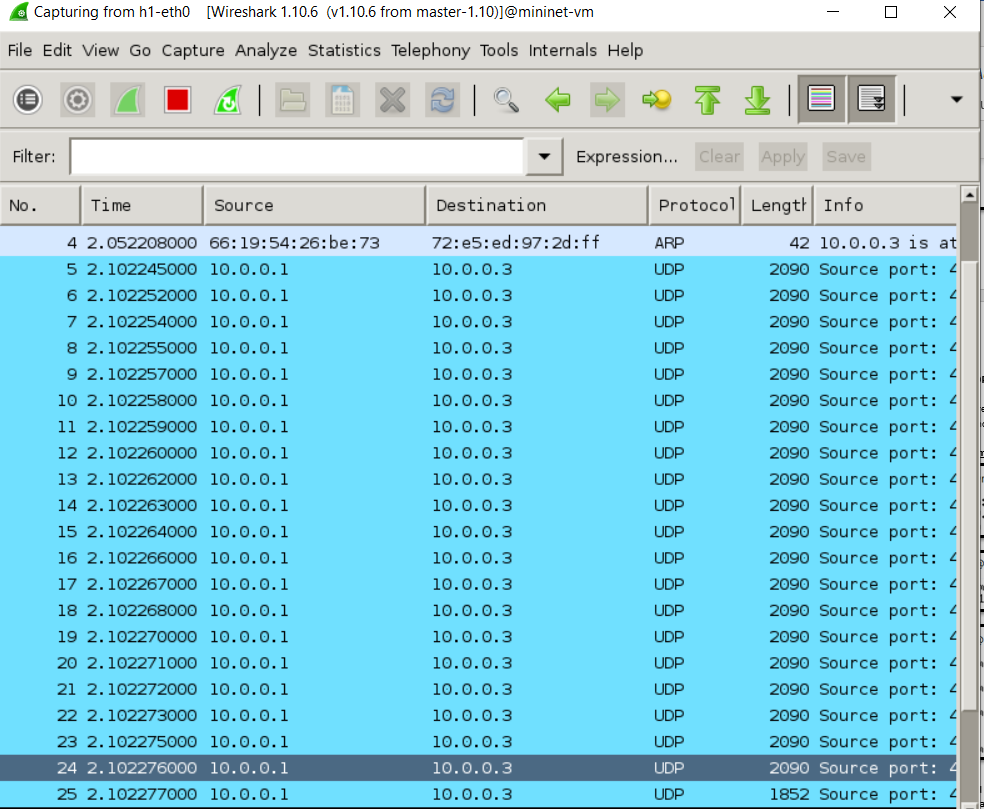
The file transferred at a very fast rate at the start but then just paused a third of the way through transmission. 22 UDP packets were transmitted in 0.000032 seconds, before the program stopped transmitting. The total transmission time till stop was only 2.102277 seconds as shown in the Wireshark screenshots below.





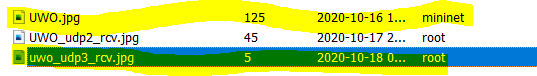
**Q3-2: Are there UDP packets in both directions, i.e. from h1 to h3 and from h3 to h1? Justify your answers.**

No UDP packet were only transmitted from h1 to h3 not in both directions. This is shown in the Wireshark screenshot below.

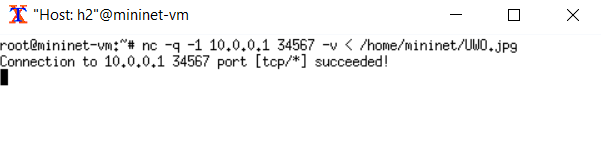
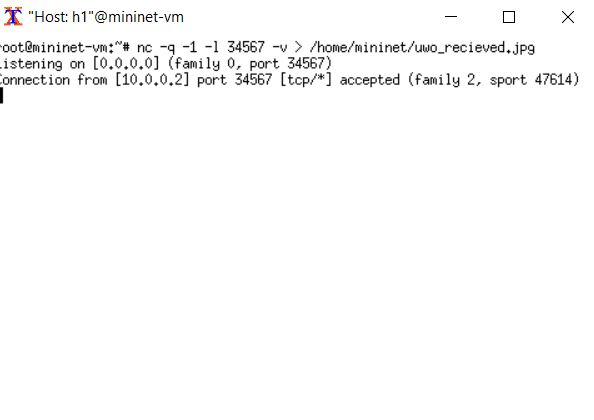


**Q3-3: Is the received image the same as the original test file? Do they have the same size? Indicate the sizes.**

In theory the received image should be the same size as the original test file, however since the transmission did not complete there is a difference. The original file is 125 KB and the received file is 5 KB.



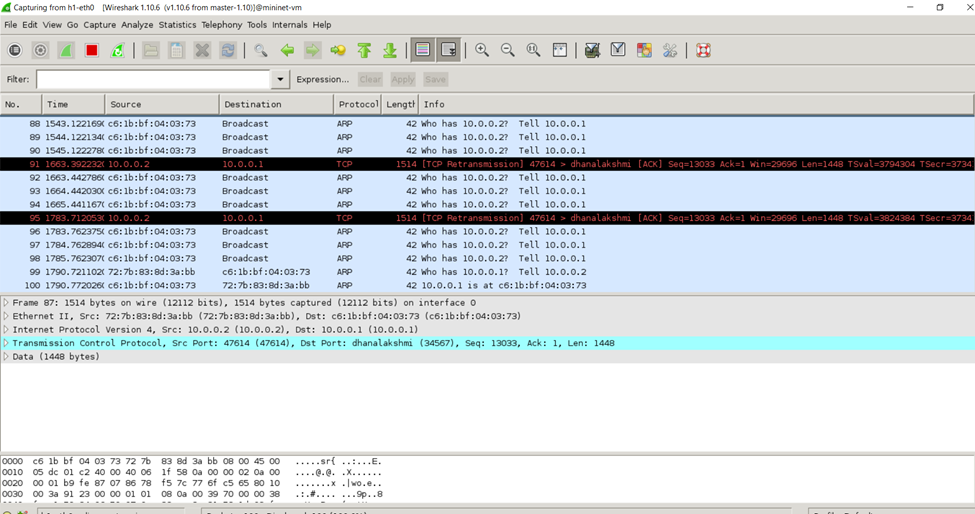
**Q4: Host 1 & 2 Terminal Code**





**Q4-1: How long did it take to transfer the file?**

I stopped the file transfer process at about 1790 seconds (almost half an hour), I had only received 14kB by this point. Extrapolation of bit rate vs. required bits suggests this process could have taken about 5 hours if it continued at this same rate. It is unclear why it was so slow. Please note I double checked the link delay and loss to see if this could explain the issue but to no avail. See Wireshark screenshot below for proof.

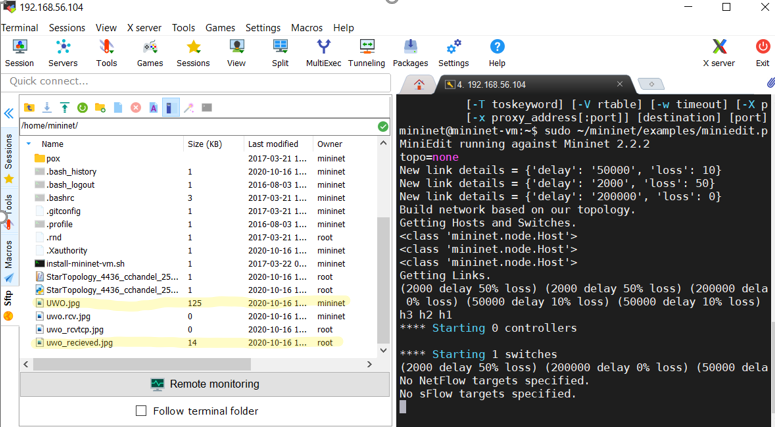


**Q4-2: Are there TCP packets in both directions, i.e. from h1 to h2 and from h2 to h1? Justify your answers.**

When I applied the TCP filter In Wireshark I received results which went in both directions. See Wireshark screenshot below for proof.

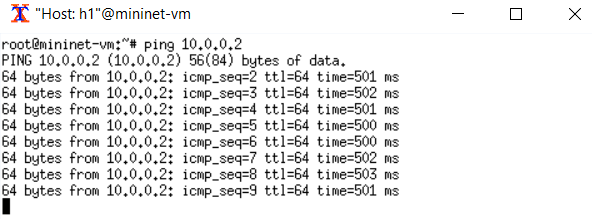
**Q4-3: Is the received image the same as the original test file? Do they have the same size? Indicate the sizes.**

Technically yes, the file sizes should be equal but because it took so long I had to cut the experiment short so not all of the image loaded. The original file is 125KB and I only got 14 KB loaded in the time I used – I acknowledge that it should 125 KB. See VM screenshot below for proof.



**Q4-5: What is the RTT for the TCP connection.**

The RTT for the TCP connection is 501 milliseconds.



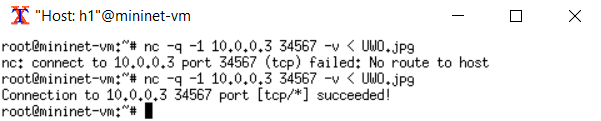
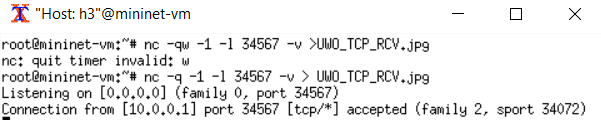
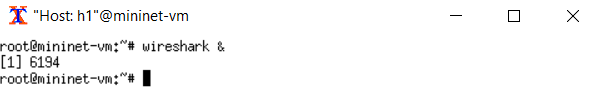
**Q4-6: Show the packet loss of TCP in Wireshark with a screenshot.**



**Q4-7: Show the packet retransmission request of TCP in Wireshark with a screenshot.**

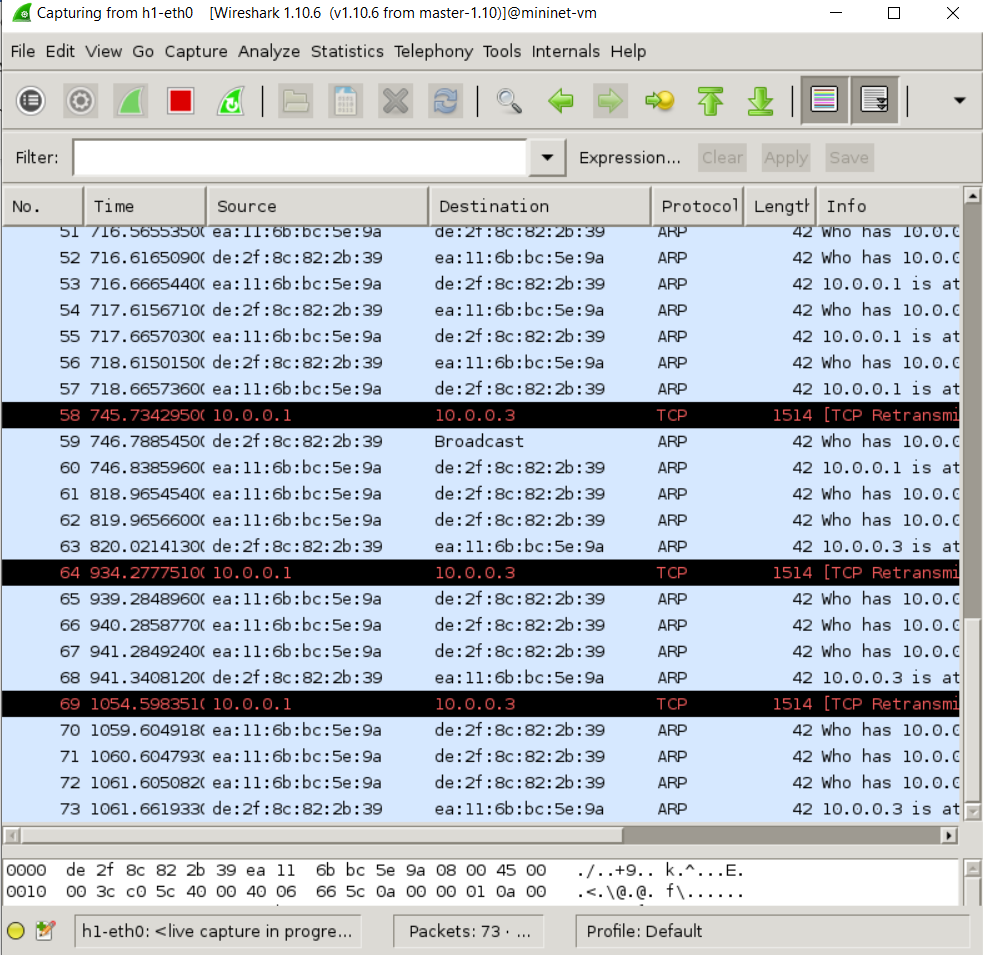


**Q5: Host 1 and 3 Terminal Code**



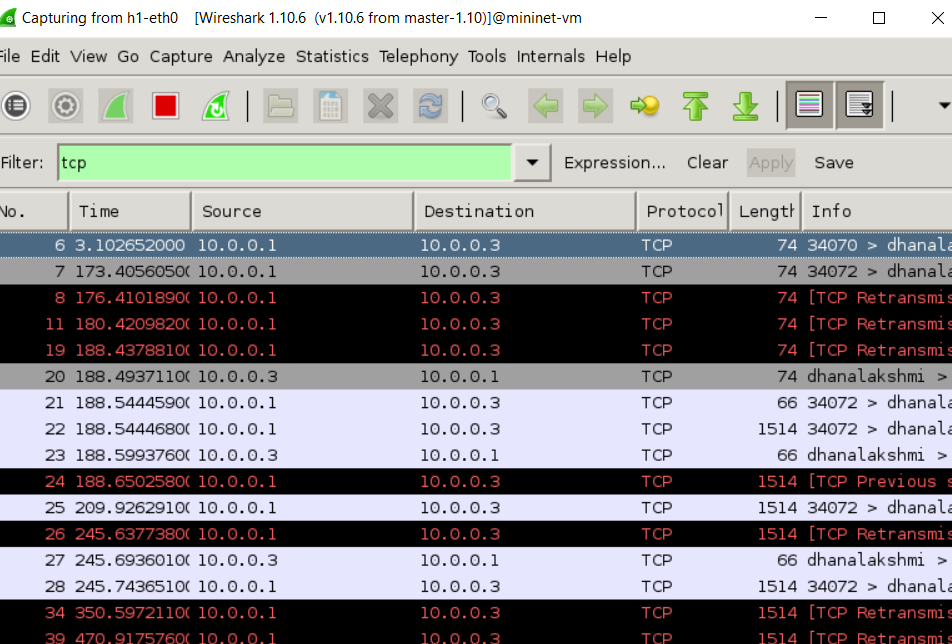
**Q5-1: How long did it take to transfer the file?**

I stopped the file transfer process at about 1061 seconds (about 20 minutes), I had only received 5kB by this point. Extrapolation of bit rate vs. required bits suggests this process could have taken about 8 hours if it continued at this same rate. It is unclear why it was so slow. Please note I double checked the link delay and loss to see if this could explain the issue but to no avail. See Wireshark screenshot below for proof.



**Q5-2: Are there TCP packets in both directions, i.e. from h1 to h3 and from h3 to h1? Justify your answers.**

When I applied the TCP filter In Wireshark I received results which went in both directions. See Wireshark screenshot below for proof.

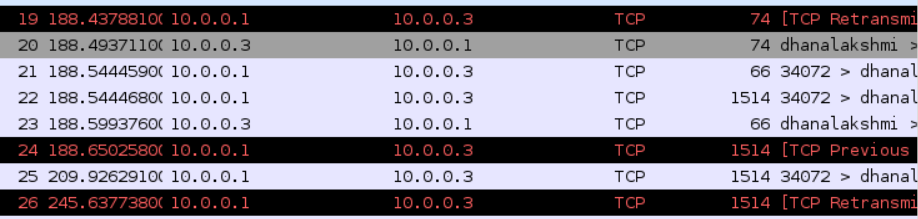


**Q5-3: Is the received image the same as the original test file? Do they have the same size? Indicate the sizes.**

Technically yes, the file sizes should be equal but because it took so long I had to cut the experiment short so not all of the image loaded. The original file is 125KB and I only got 5 KB loaded in the time I used – I acknowledge that it should 125 KB. See VM screenshot below for proof.

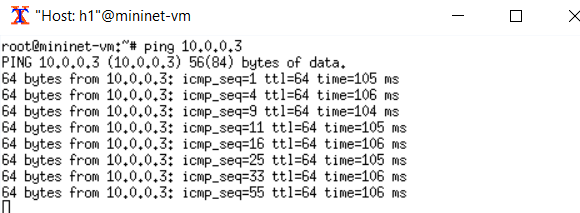


**Q5-4: Show the handshaking process of TCP in Wireshark with a screenshot.**

Shown below in Wireshark screenshot.

**Q5-5: What is the RTT for the TCP connection.**

The RTT for the TCP connection is 105 milliseconds.



**Q5-6: Show the packet loss of TCP in Wireshark with a screenshot**.



**Q5-7: Show the packet retransmission request of TCP in Wireshark with a screenshot.**



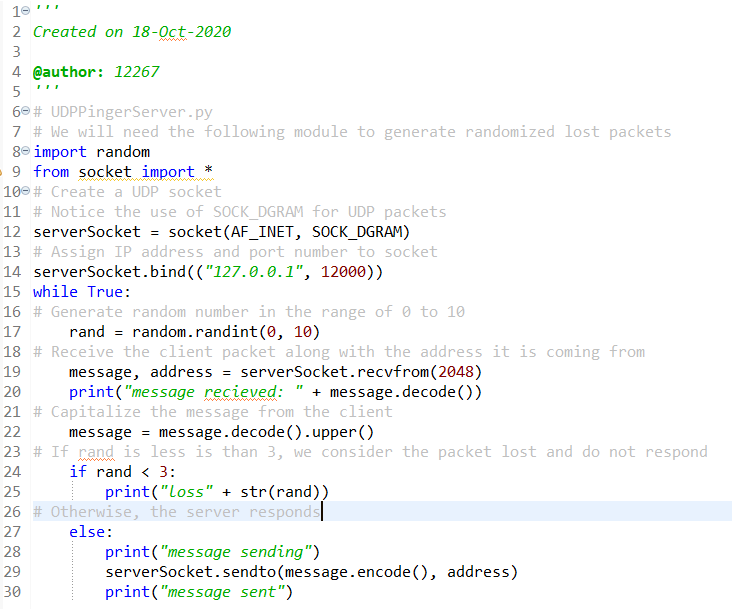
**Problem #4**

The client code

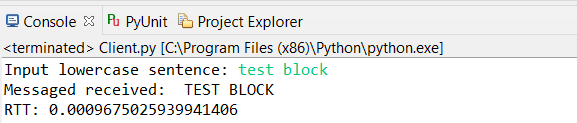
Graphical user interface, text, application

Description automatically generated

The server code



The Client console



**Problem #5**

