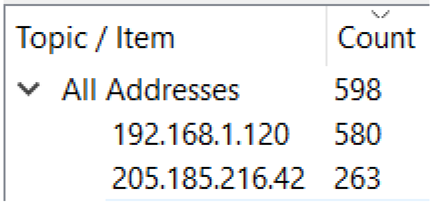
Aiden Duffy 251008412

Networking 4436 Exam 1

Problem 1

Q1

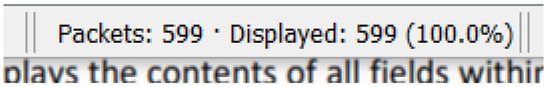
My laptop's IP address IPv4 is 192.168.1.120. My laptop's IPv6 was not shown, this was confirmed by checking the IPv6 statistics in the Statistics tab of Wireshark. The IPv4 address appears in almost every row (580/599), either as the source or destination. For the rows that this address did not appear, the protocol type was always SSDP, with an info of NOTIFY * HTTP/1.1. This protocol type was only used when my device's IP was not the source or destination.



Topic / Item	Count
All Addresses	598
192.168.1.120	580
205.185.216.42	263

Q2

A total of 599 packets were captured.

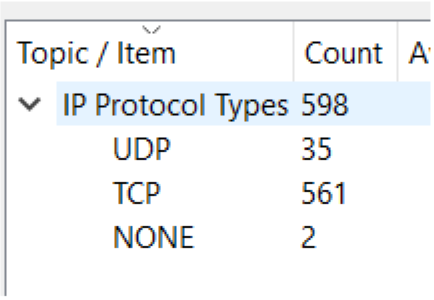


Packets: 599 · Displayed: 599 (100.0%)

plays the contents of all fields withir

Q3

TCP packets accounted for 93.7% (561/599) of all packets. UDP packets accounted for 5.8% (35/599) of all packets. Therefore, the ratio of TCP to UDP is roughly 16:1.

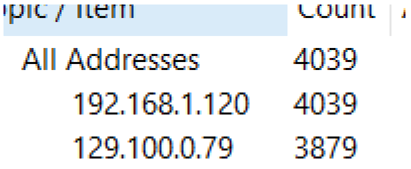


Topic / Item	Count	A
IP Protocol Types	598	
UDP	35	
TCP	561	
NONE	2	

Aiden Duffy 251008412

Q4

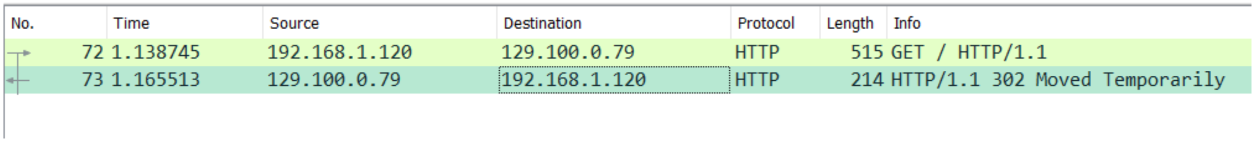
The IP address that corresponds to the http link requested is 129.100.0.79.



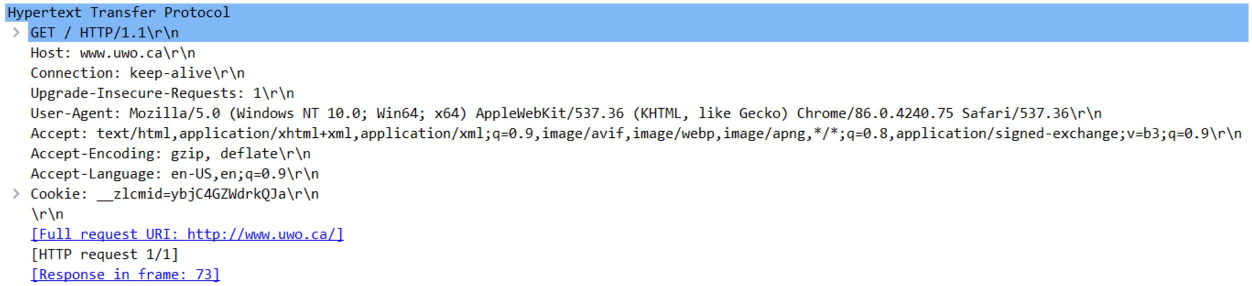
pic / Item	Count
All Addresses	4039
192.168.1.120	4039
129.100.0.79	3879

Q5

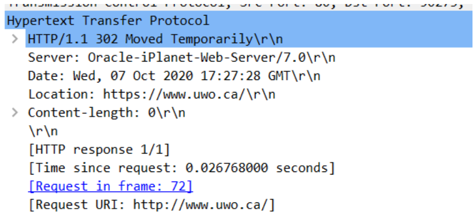
The two HTTP messages returned were GET and 302 Moved Temporarily. Both have been snipped below.



No.	Time	Source	Destination	Protocol	Length	Info
72	1.138745	192.168.1.120	129.100.0.79	HTTP	515	GET / HTTP/1.1
73	1.165513	129.100.0.79	192.168.1.120	HTTP	214	HTTP/1.1 302 Moved Temporarily



```
Hypertext Transfer Protocol
 > GET / HTTP/1.1\r\n
   Host: www.uwo.ca\r\n
   Connection: keep-alive\r\n
   Upgrade-Insecure-Requests: 1\r\n
   User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.75 Safari/537.36\r\n
   Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
   Accept-Encoding: gzip, deflate\r\n
   Accept-Language: en-US,en;q=0.9\r\n
 > Cookie: __zlcmid=ybjC4GZWdrkQJa\r\n
   \r\n
   [Full request URI: http://www.uwo.ca/]
   [HTTP request 1/1]
   [Response in frame: 73]
```



```
Hypertext Transfer Protocol
 > HTTP/1.1 302 Moved Temporarily\r\n
   Server: Oracle-iPlanet-Web-Server/7.0\r\n
   Date: Wed, 07 Oct 2020 17:27:28 GMT\r\n
   Location: https://www.uwo.ca/\r\n
 > Content-length: 0\r\n
   \r\n
   [HTTP response 1/1]
   [Time since request: 0.026768000 seconds]
   [Request in frame: 72]
   [Request URI: http://www.uwo.ca/]
```

Q6

The response took 0.026768 seconds (1.165513-1.138745). This is also visible in the HTTP response header under Time Since Request.

Q7

Since the HTTP response redirected me to an HTTPS page, the remainder of the HTTP traffic was encrypted, and I cannot see the subsequent messages when reloading the page. I would assume that the refresh would take less time if my browser cached the web page after the initial request.

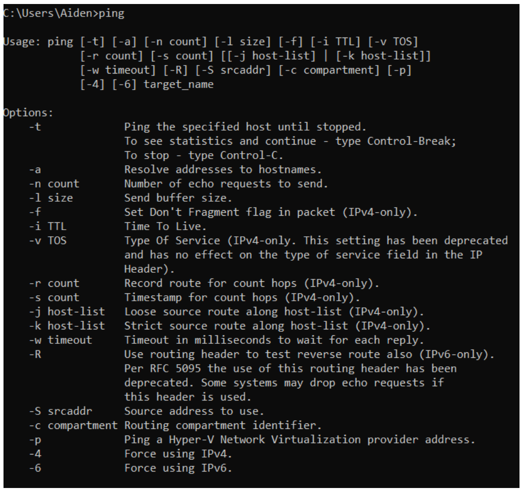
Q8

I was unable to examine the differences in time between the initial request and the refresh request due to the page being encrypted, however if the web page was put into my browser's cache, then I would assume that the refresh would be quicker than the initial request due to the caching.

Problem 2

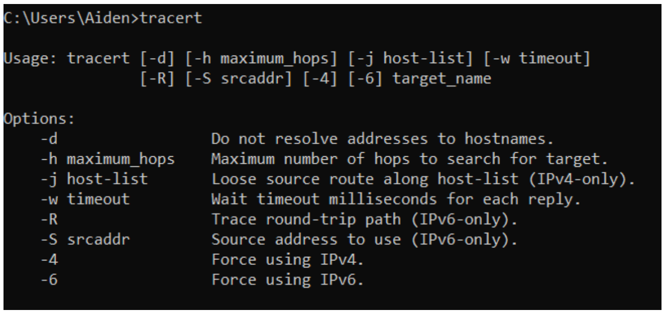
Q1

Ping is a command that is used in your device's terminal. There are several flags that can be used with **ping**. One useful flag is the **-t** flag, which allows the user to ping specific IP addresses continually until stopped. Another helpful flag is **-n** which takes in an integer parameter to specify how many pings you want to send, useful if you want to send a specific number of pings that is not the default number of pings (4 pings).



```
C:\Users\Aiden>ping

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
            [-r count] [-s count] [[-j host-list] | [-k host-list]]
            [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]
            [-4] [-6] target_name

Options:
    -t             Ping the specified host until stopped.
                   To see statistics and continue - type Control-Break;
                   To stop - type Control-C.
    -a             Resolve addresses to hostnames.
    -n count       Number of echo requests to send.
    -l size        Send buffer size.
    -f             Set Don't Fragment flag in packet (IPv4-only).
    -i TTL         Time To Live.
    -v TOS         Type Of Service (IPv4-only. This setting has been deprecated
                   and has no effect on the type of service field in the IP
                   Header).
    -r count       Record route for count hops (IPv4-only).
    -s count       Timestamp for count hops (IPv4-only).
    -j host-list   Loose source route along host-list (IPv4-only).
    -k host-list   Strict source route along host-list (IPv4-only).
    -w timeout     Timeout in milliseconds to wait for each reply.
    -R             Use routing header to test reverse route also (IPv6-only).
                   Per RFC 5095 the use of this routing header has been
                   deprecated. Some systems may drop echo requests if
                   this header is used.
    -S srcaddr     Source address to use.
    -c compartment Routing compartment identifier.
    -p             Ping a Hyper-V Network Virtualization provider address.
    -4             Force using IPv4.
    -6             Force using IPv6.
```

Tracert is the traceroute command used in your device's terminal. It also has multiple flags that can be used with it. **-h** takes in an integer to specify the maximum number of hops that the route will take to search for the target (instead of the default 30 hops). Similarly, **-w** specifies the maximum number of milliseconds that the command should wait before giving a timeout. Both are useful in restricting the traceroute command from taking too long.

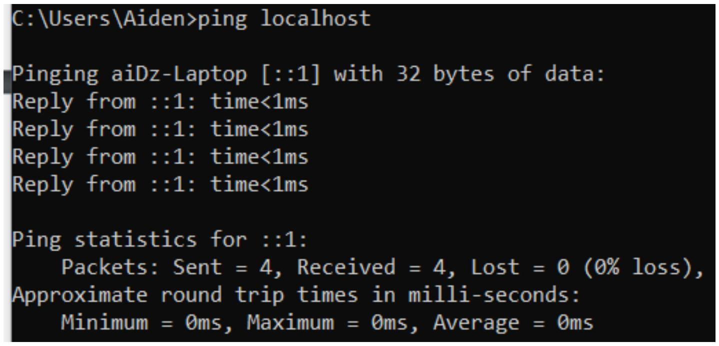


```
C:\Users\Aiden>tracert

Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
               [-R] [-S srcaddr] [-4] [-6] target_name

Options:
    -d                 Do not resolve addresses to hostnames.
    -h maximum_hops    Maximum number of hops to search for target.
    -j host-list       Loose source route along host-list (IPv4-only).
    -w timeout         Wait timeout milliseconds for each reply.
    -R                 Trace round-trip path (IPv6-only).
    -S srcaddr         Source address to use (IPv6-only).
    -4                 Force using IPv4.
    -6                 Force using IPv6.
```

Aiden Duffy 251008412

Q2

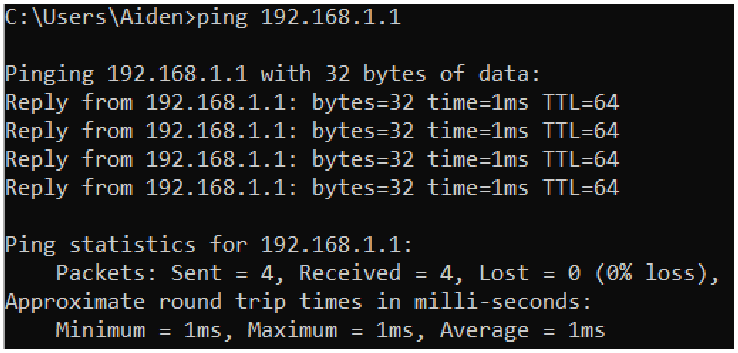
I used **ping localhost**. The command was successful.



```
C:\Users\Aiden>ping localhost

Pinging aiDz-Laptop [::1] with 32 bytes of data:
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms

Ping statistics for ::1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

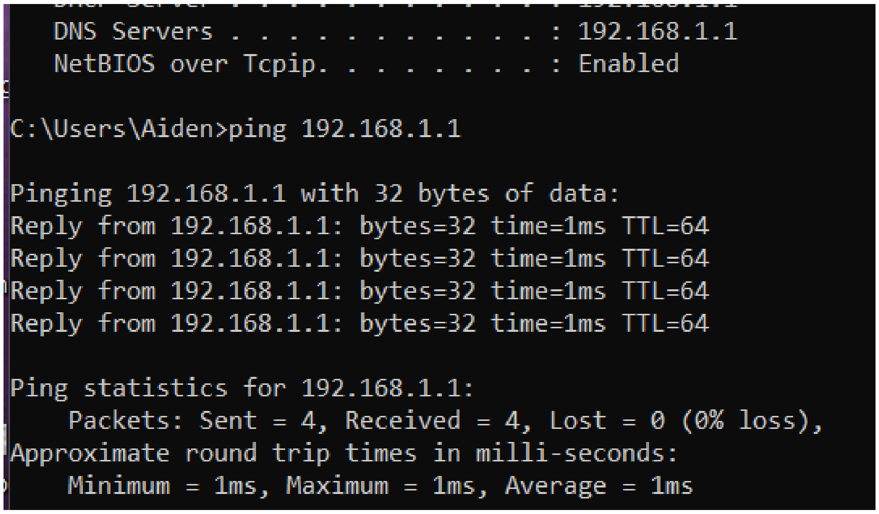
Q3

To ping my default gateway, I first used **ipconfig** to find my default gateway address. I then ran **ping 192.168.1.1**. The command was successful.



```
C:\Users\Aiden>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

To ping my DNS server, I first used **ipconfig /all** to find my DNS server address. I then ran **ping 192.168.1.1**. The command was successful.

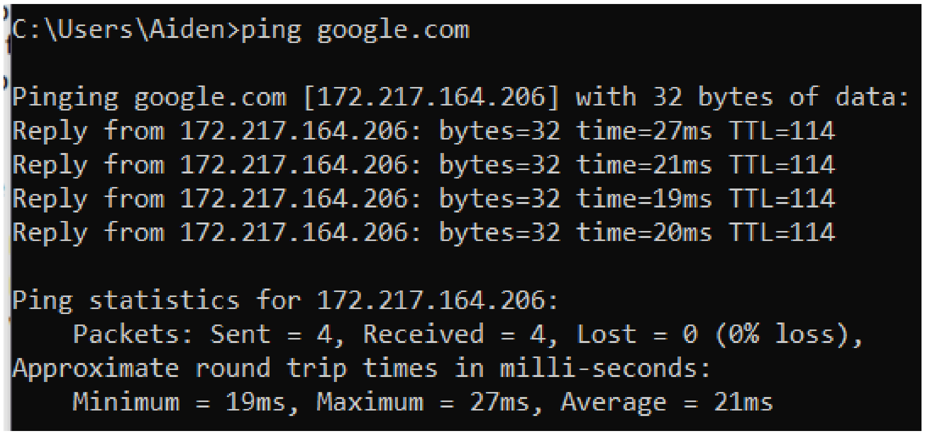


```
   DNS Servers . . . . . . . . . . . : 192.168.1.1
   NetBIOS over Tcpip. . . . . . . . : Enabled

C:\Users\Aiden>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

Aiden Duffy 251008412

Q4

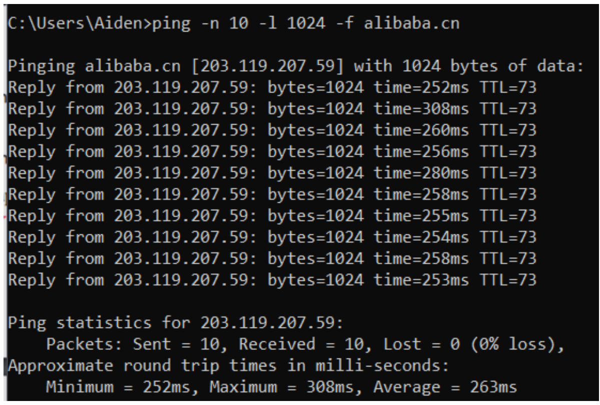
I pinged google.com, whose IP address is 172.217.164.206. The minimum round-trip time was 19ms, the maximum round trip time was 27ms, and the average round trip time was 21ms.



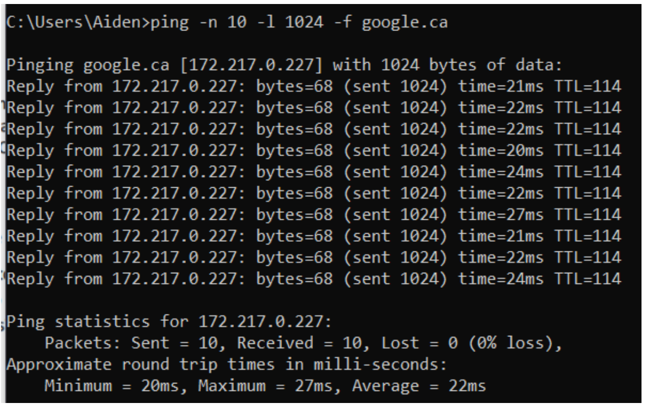
```
C:\Users\Aiden>ping google.com

Pinging google.com [172.217.164.206] with 32 bytes of data:
Reply from 172.217.164.206: bytes=32 time=27ms TTL=114
Reply from 172.217.164.206: bytes=32 time=21ms TTL=114
Reply from 172.217.164.206: bytes=32 time=19ms TTL=114
Reply from 172.217.164.206: bytes=32 time=20ms TTL=114

Ping statistics for 172.217.164.206:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 19ms, Maximum = 27ms, Average = 21ms
```

Q5

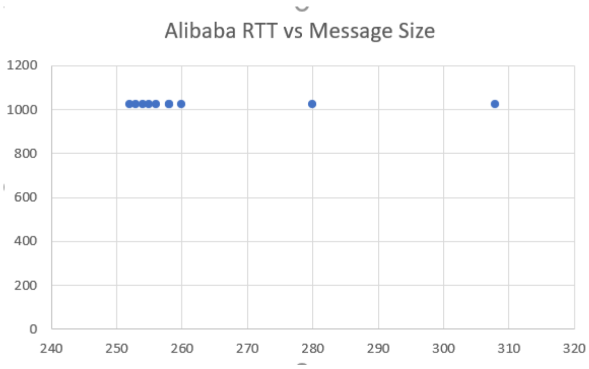
I pinged both alibaba.cn and google.ca ten times with packet sizes of 1024 bytes. The pings to alibaba.cn took about 12 times as long as the pings to google.ca. The average RTTs were 263.4ms for alibaba.cn and 22.5ms for google.ca. The standard deviations were 17.6ms for alibaba.cn and 2.0ms for google.ca.

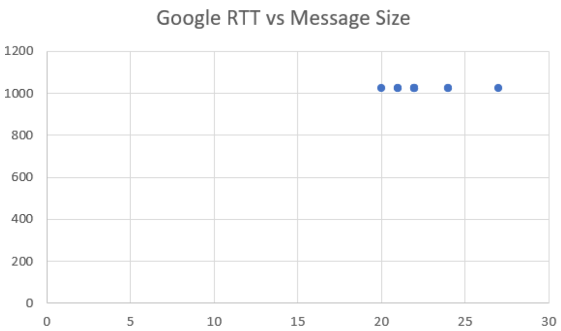


```
C:\Users\Aiden>ping -n 10 -l 1024 -f alibaba.cn

Pinging alibaba.cn [203.119.207.59] with 1024 bytes of data:
Reply from 203.119.207.59: bytes=1024 time=252ms TTL=73
Reply from 203.119.207.59: bytes=1024 time=308ms TTL=73
Reply from 203.119.207.59: bytes=1024 time=260ms TTL=73
Reply from 203.119.207.59: bytes=1024 time=256ms TTL=73
Reply from 203.119.207.59: bytes=1024 time=280ms TTL=73
Reply from 203.119.207.59: bytes=1024 time=258ms TTL=73
Reply from 203.119.207.59: bytes=1024 time=255ms TTL=73
Reply from 203.119.207.59: bytes=1024 time=254ms TTL=73
Reply from 203.119.207.59: bytes=1024 time=258ms TTL=73
Reply from 203.119.207.59: bytes=1024 time=253ms TTL=73

Ping statistics for 203.119.207.59:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 252ms, Maximum = 308ms, Average = 263ms
```



```
C:\Users\Aiden>ping -n 10 -l 1024 -f google.ca

Pinging google.ca [172.217.0.227] with 1024 bytes of data:
Reply from 172.217.0.227: bytes=68 (sent 1024) time=21ms TTL=114
Reply from 172.217.0.227: bytes=68 (sent 1024) time=22ms TTL=114
Reply from 172.217.0.227: bytes=68 (sent 1024) time=22ms TTL=114
Reply from 172.217.0.227: bytes=68 (sent 1024) time=20ms TTL=114
Reply from 172.217.0.227: bytes=68 (sent 1024) time=24ms TTL=114
Reply from 172.217.0.227: bytes=68 (sent 1024) time=22ms TTL=114
Reply from 172.217.0.227: bytes=68 (sent 1024) time=27ms TTL=114
Reply from 172.217.0.227: bytes=68 (sent 1024) time=21ms TTL=114
Reply from 172.217.0.227: bytes=68 (sent 1024) time=22ms TTL=114
Reply from 172.217.0.227: bytes=68 (sent 1024) time=24ms TTL=114

Ping statistics for 172.217.0.227:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 20ms, Maximum = 27ms, Average = 22ms
```

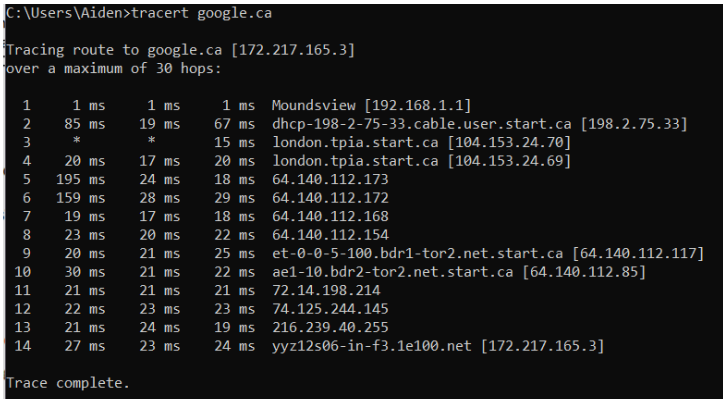




Clearly distance has a huge effect on latency. The relationship is obvious, that the further away the ping must travel, the longer the round-trip time is. This relationship is likely somewhat linear, as the propagation times for this data are generally very similar. Therefore, as the distance to an IP increases, the RTT to that IP will likely increase by a factor of the distance increase * 2 (to account for both trips). In that case, I would guess that Alibaba's servers are roughly 6 times further from my device than Google's servers. Additionally, the further away a server is, the more likely there will be slight differences in the latency as it has more time to encounter difficulties. This explains the much greater standard deviation that alibaba.cn had compared to google.ca.

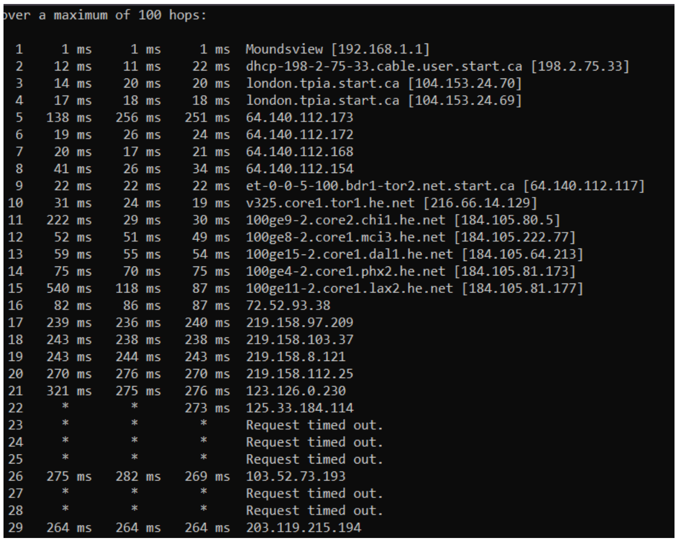
Q6

i) It took 14 hops to reach the destination host. The full route is below.



```
C:\Users\Aiden>tracert google.ca

Tracing route to google.ca [172.217.165.3]
over a maximum of 30 hops:

  1     1 ms     1 ms     1 ms  Moundsview [192.168.1.1]
  2    85 ms    19 ms    67 ms  dhcp-198-2-75-33.cable.user.start.ca [198.2.75.33]
  3     *        *       15 ms  london.tpia.start.ca [104.153.24.70]
  4    20 ms    17 ms    20 ms  london.tpia.start.ca [104.153.24.69]
  5   195 ms    24 ms    18 ms  64.140.112.173
  6   159 ms    28 ms    29 ms  64.140.112.172
  7    19 ms    17 ms    18 ms  64.140.112.168
  8    23 ms    20 ms    22 ms  64.140.112.154
  9    20 ms    21 ms    25 ms  et-0-0-5-100.bdr1-tor2.net.start.ca [64.140.112.117]
 10    30 ms    21 ms    22 ms  ae1-10.bdr2-tor2.net.start.ca [64.140.112.85]
 11    21 ms    21 ms    21 ms  72.14.198.214
 12    22 ms    23 ms    23 ms  74.125.244.145
 13    21 ms    24 ms    19 ms  216.239.40.255
 14    27 ms    23 ms    24 ms  yyz12s06-in-f3.1e100.net [172.217.165.3]

Trace complete.
```

ii) It took 29 hops to reach the destination host. The full route is below.



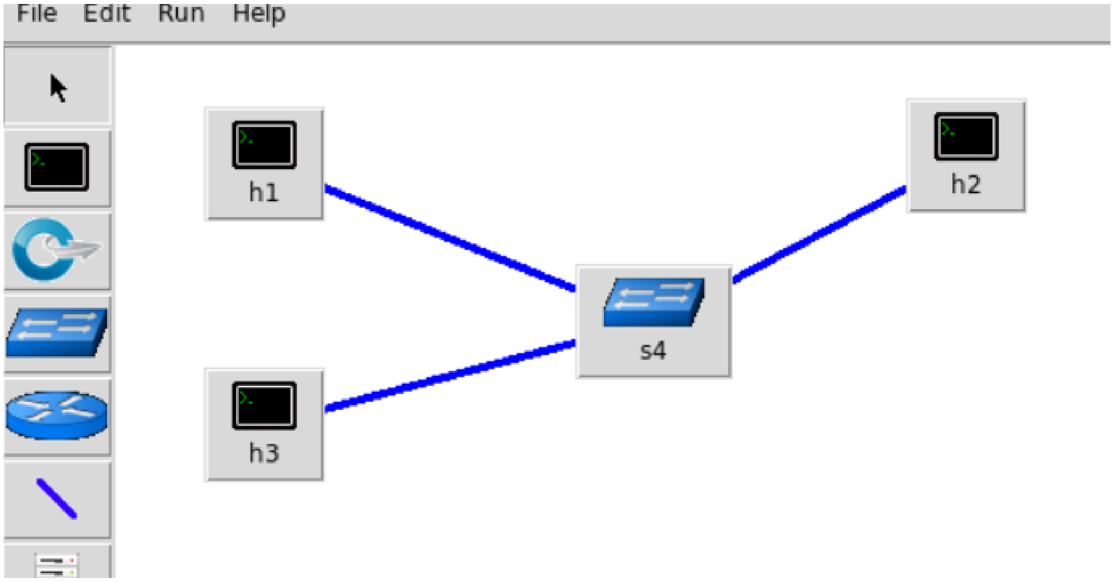
```
over a maximum of 100 hops:

  1     1 ms     1 ms     1 ms  Moundsview [192.168.1.1]
  2    12 ms    11 ms    22 ms  dhcp-198-2-75-33.cable.user.start.ca [198.2.75.33]
  3    14 ms    20 ms    20 ms  london.tpia.start.ca [104.153.24.70]
  4    17 ms    18 ms    18 ms  london.tpia.start.ca [104.153.24.69]
  5   138 ms   256 ms   251 ms  64.140.112.173
  6    19 ms    26 ms    24 ms  64.140.112.172
  7    20 ms    17 ms    21 ms  64.140.112.168
  8    41 ms    26 ms    34 ms  64.140.112.154
  9    22 ms    22 ms    22 ms  et-0-0-5-100.bdr1-tor2.net.start.ca [64.140.112.117]
 10    31 ms    24 ms    19 ms  v325.core1.tor1.he.net [216.66.14.129]
 11   222 ms    29 ms    30 ms  100ge9-2.core2.chi1.he.net [184.105.80.5]
 12    52 ms    51 ms    49 ms  100ge8-2.core1.mci3.he.net [184.105.222.77]
 13    59 ms    55 ms    54 ms  100ge15-2.core1.dal1.he.net [184.105.64.213]
 14    75 ms    70 ms    75 ms  100ge4-2.core1.phx2.he.net [184.105.81.173]
 15   540 ms   118 ms    87 ms  100ge11-2.core1.lax2.he.net [184.105.81.177]
 16    82 ms    86 ms    87 ms  72.52.93.38
 17   239 ms   236 ms   240 ms  219.158.97.209
 18   243 ms   238 ms   238 ms  219.158.103.37
 19   243 ms   244 ms   243 ms  219.158.8.121
 20   270 ms   276 ms   270 ms  219.158.112.25
 21   321 ms   275 ms   276 ms  123.126.0.230
 22     *        *      273 ms  125.33.184.114
 23     *        *        *     Request timed out.
 24     *        *        *     Request timed out.
 25     *        *        *     Request timed out.
 26   275 ms   282 ms   269 ms  103.52.73.193
 27     *        *        *     Request timed out.
 28     *        *        *     Request timed out.
 29   264 ms   264 ms   264 ms  203.119.215.194
```

iii) Some of the output lines contain * * *. This means that the router at that specific hop does not respond to whatever packet type is being used to trace the route.

Problem 3

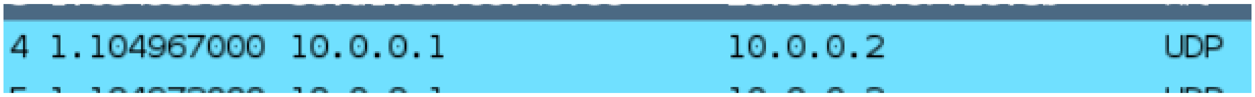
Q1-1

View attached .mn and .py files.

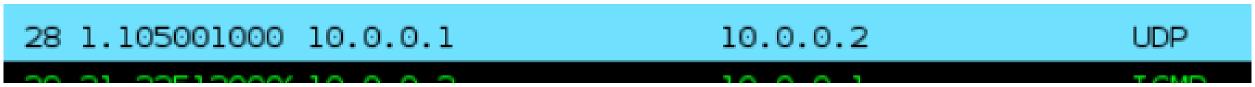


Q2-1

It took 25 packets a total of 0.050062 seconds to transfer this file.



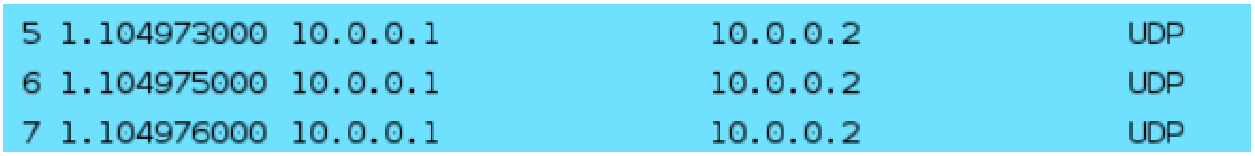
```
4 1.104967000  10.0.0.1          10.0.0.2          UDP
```



```
28 1.105001000  10.0.0.1         10.0.0.2          UDP
```

Q2-2

There are only UDP packets in a single direction, from h1 to h2. This is because UDP does not establish a connection handshake beforehand, and so it simply sends the file to the specified address without a response.

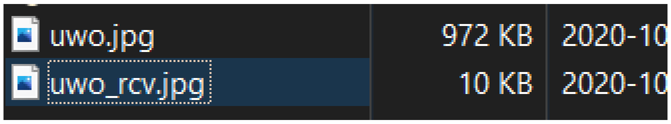


```
5 1.104973000  10.0.0.1          10.0.0.2          UDP
6 1.104975000  10.0.0.1          10.0.0.2          UDP
7 1.104976000  10.0.0.1          10.0.0.2          UDP
```

Q2-3

The received image is not the same as the original test file. The test file opens without issue in an image editor, but the received image does not. Additionally, they are both different sizes. The test image is 972 KB, while the received image is only 10 KB.

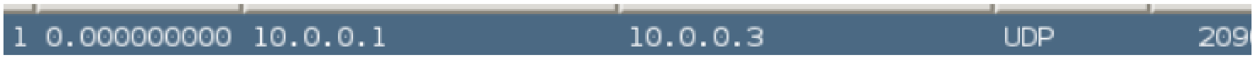
Aiden Duffy 251008412



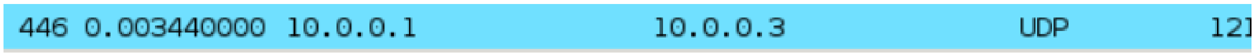
```
uwo.jpg        972 KB  2020-10
uwo_rcv.jpg     10 KB  2020-10
```

Q3-1

It took 446 packets 0.00344 seconds to transfer the file.



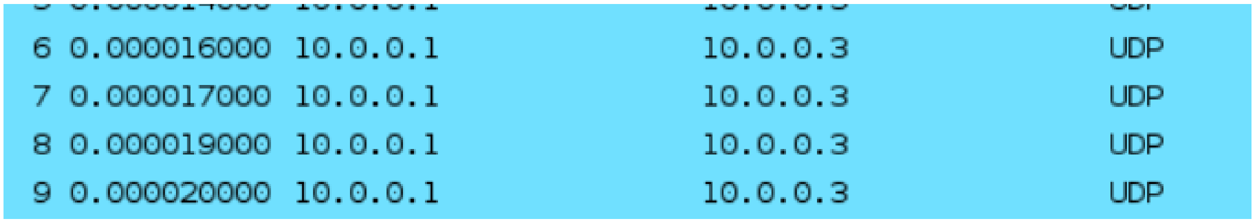
```
1 0.000000000  10.0.0.1                 10.0.0.3                 UDP      209
```



```
446 0.003440000  10.0.0.1                10.0.0.3                 UDP      121
```

Q3-2

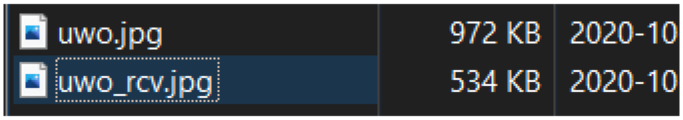
There are only UDP packets in a single direction, from h1 to h3. This is because UDP does not establish a connection handshake beforehand, and so it simply sends the file to the specified address without a response.



```
  6 0.000016000  10.0.0.1          10.0.0.3          UDP
  7 0.000017000  10.0.0.1          10.0.0.3          UDP
  8 0.000019000  10.0.0.1          10.0.0.3          UDP
  9 0.000020000  10.0.0.1          10.0.0.3          UDP
```

Q3-3

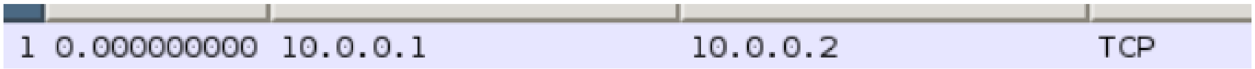
The received image is not the same as the original test file. The test file opens without issue in an image editor, as does the received image, except the received image is simply a jumble of characters. Additionally, they are both different sizes. The test image is 972 KB, while the received image is 534 KB.



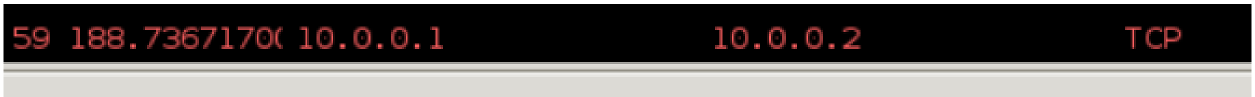
```
uwo.jpg        972 KB  2020-10
uwo_rcv.jpg    534 KB  2020-10
```

Q4-1

It took 65 packets 188.736 seconds to transfer the file.



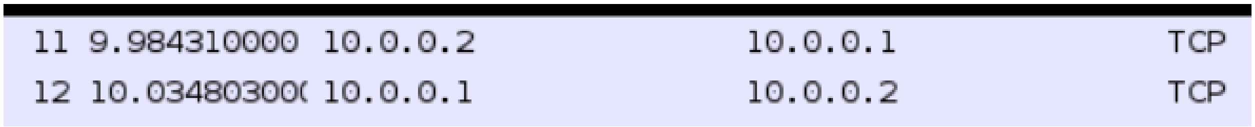
```
1 0.000000000  10.0.0.1                  10.0.0.2                  TCP
```



```
59 188.7367170( 10.0.0.1                  10.0.0.2                  TCP
```

Q4-2

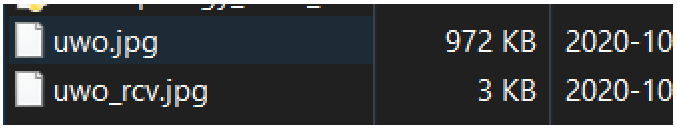
There are TCP packets in both directions, from h1 to h2 and h2 to h1. This is because TCP establishes a connection before sending any data. Therefore, there is communication of packets in both directions.



```
11 9.984310000  10.0.0.2          10.0.0.1          TCP
12 10.03480300( 10.0.0.1          10.0.0.2          TCP
```

Q4-3

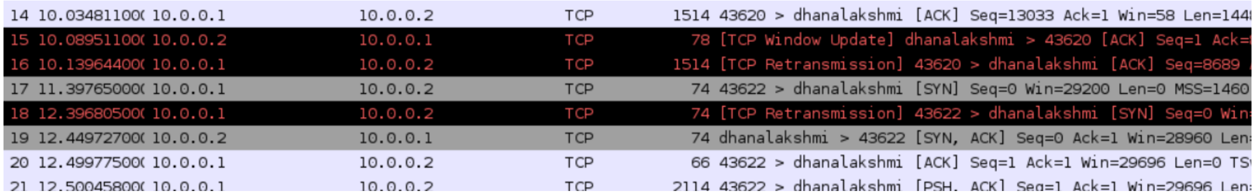
The received image is not the same as the original test file. The test file opens without issue in an image editor, but the received image does not. Additionally, they are both different sizes. The test image is 972 KB, while the received image is only 3 KB.



```
uwo.jpg        972 KB   2020-10
uwo_rcv.jpg      3 KB   2020-10
```

Q4-4

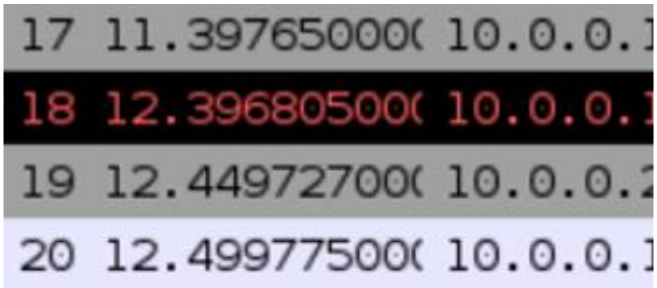
Below is the TCP handshaking process.



```
14 10.03481100( 10.0.0.1     10.0.0.2     TCP     1514 43620 > dhanalakshmi [ACK] Seq=13033 Ack=1 Win=58 Len=144
15 10.08951100( 10.0.0.2     10.0.0.1     TCP       78 [TCP Window Update] dhanalakshmi > 43620 [ACK] Seq=1 Ack=
16 10.13964400( 10.0.0.1     10.0.0.2     TCP     1514 [TCP Retransmission] 43620 > dhanalakshmi [ACK] Seq=8689
17 11.39765000( 10.0.0.1     10.0.0.2     TCP       74 43622 > dhanalakshmi [SYN] Seq=0 Win=29200 Len=0 MSS=1460
18 12.39680500( 10.0.0.1     10.0.0.2     TCP       74 [TCP Retransmission] 43622 > dhanalakshmi [SYN] Seq=0 Win
19 12.44972700( 10.0.0.2     10.0.0.1     TCP       74 dhanalakshmi > 43622 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len
20 12.49977500( 10.0.0.1     10.0.0.2     TCP       66 43622 > dhanalakshmi [ACK] Seq=1 Ack=1 Win=29696 Len=0 TS
21 12.50045800( 10.0.0.1     10.0.0.2     TCP     2114 43622 > dhanalakshmi [PSH, ACK] Seq=1 Ack=1 Win=29696 Len
```

Q4-5

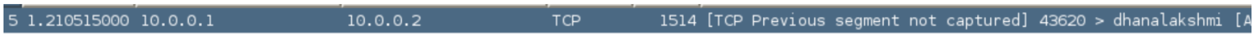
The RTT for the TCP connection is 1.102125 seconds (12.499775-11.39765).



```
17 11.39765000( 10.0.0.1
18 12.39680500( 10.0.0.1
19 12.44972700( 10.0.0.2
20 12.49977500( 10.0.0.1
```

Q4-6

There were multiple cases of packet loss in the TCP connection.

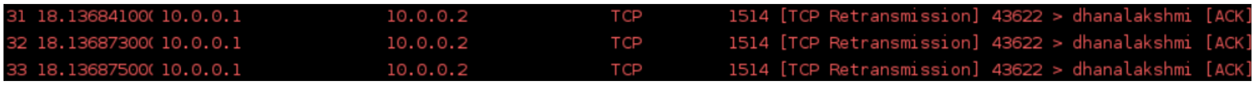


```
5 1.210515000 10.0.0.1       10.0.0.2       TCP     1514 [TCP Previous segment not captured] 43620 > dhanalakshmi [A
```

Aiden Duffy 251008412

Q4-7

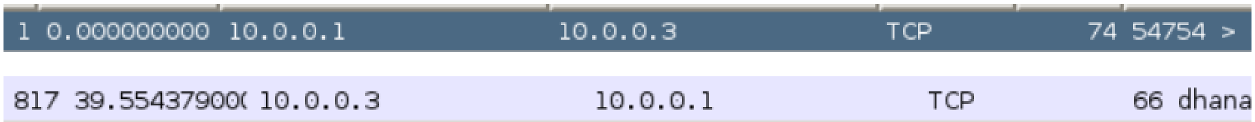
As with packet loss, there were a lot of packet retransmissions in the TCP connection.



```
31 18.13684100( 10.0.0.1       10.0.0.2      TCP     1514 [TCP Retransmission] 43622 > dhanalakshmi [ACK]
32 18.13687300( 10.0.0.1       10.0.0.2      TCP     1514 [TCP Retransmission] 43622 > dhanalakshmi [ACK]
33 18.13687500( 10.0.0.1       10.0.0.2      TCP     1514 [TCP Retransmission] 43622 > dhanalakshmi [ACK]
```

Q5-1

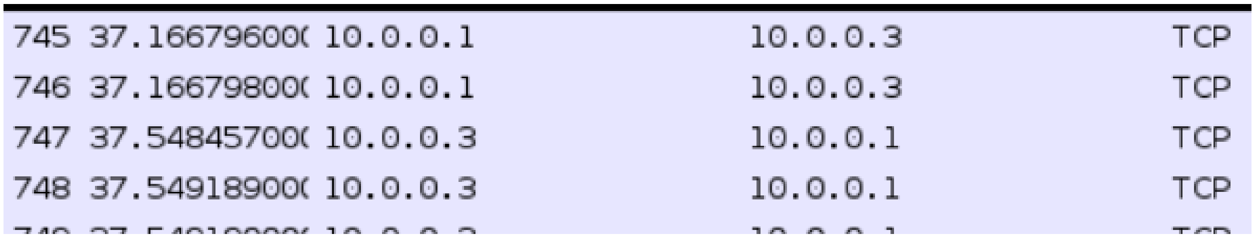
It took 817 packets and 39.554379 seconds to transfer the file.



```
1 0.000000000 10.0.0.1        10.0.0.3        TCP      74 54754 >

817 39.55437900( 10.0.0.3       10.0.0.1        TCP      66 dhana
```

Q5-2

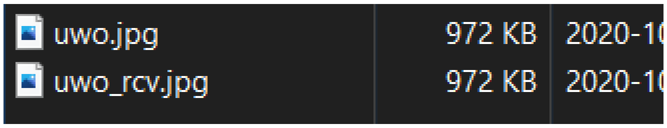
There are TCP packets in both directions, from h1 to h3 and h3 to h1. This is because TCP establishes a connection before sending any data. Therefore, there is communication of packets in both directions.



```
745 37.16679600( 10.0.0.1       10.0.0.3       TCP
746 37.16679800( 10.0.0.1       10.0.0.3       TCP
747 37.54845700( 10.0.0.3       10.0.0.1       TCP
748 37.54918900( 10.0.0.3       10.0.0.1       TCP
```

Q5-3

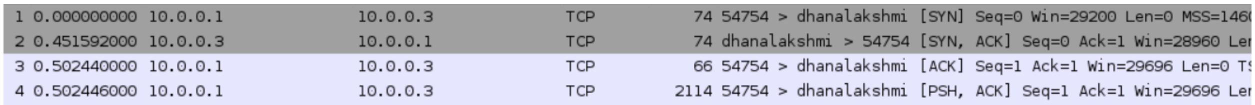
The received image is the same as the original test file. Both open without issue in an image editor. Additionally, they are both the same size, 972 KB.



```
uwo.jpg          972 KB   2020-1
uwo_rcv.jpg      972 KB   2020-1
```

Q5-4

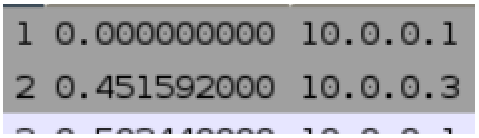
Below is the TCP handshaking process.



```
1 0.000000000 10.0.0.1      10.0.0.3     TCP       74 54754 > dhanalakshmi [SYN] Seq=0 Win=29200 Len=0 MSS=146
2 0.451592000 10.0.0.3      10.0.0.1     TCP       74 dhanalakshmi > 54754 [SYN, ACK] Seq=0 Ack=1 Win=28960 Le
3 0.502440000 10.0.0.1      10.0.0.3     TCP       66 54754 > dhanalakshmi [ACK] Seq=1 Ack=1 Win=29696 Len=0 T
4 0.502446000 10.0.0.1      10.0.0.3     TCP     2114 54754 > dhanalakshmi [PSH, ACK] Seq=1 Ack=1 Win=29696 Le
```

Q5-5

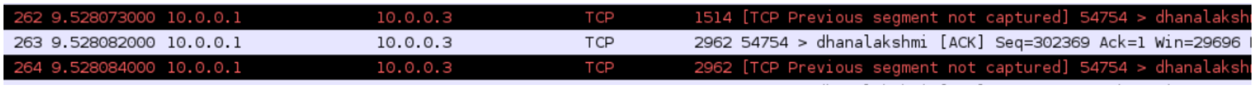
The RTT for the TCP connection was 0.45159 seconds.



```
1 0.000000000 10.0.0.1
2 0.451592000 10.0.0.3
```

Q5-6

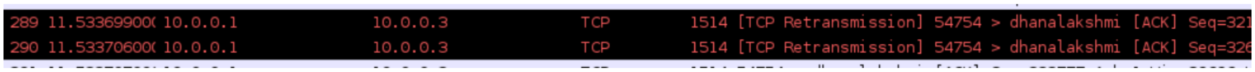
There were several cases of packet loss in the TCP connection.



```
262 9.528073000 10.0.0.1     10.0.0.3     TCP     1514 [TCP Previous segment not captured] 54754 > dhanalaksh
263 9.528082000 10.0.0.1     10.0.0.3     TCP     2962 54754 > dhanalakshmi [ACK] Seq=302369 Ack=1 Win=29696
264 9.528084000 10.0.0.1     10.0.0.3     TCP     2962 [TCP Previous segment not captured] 54754 > dhanalaksh
```

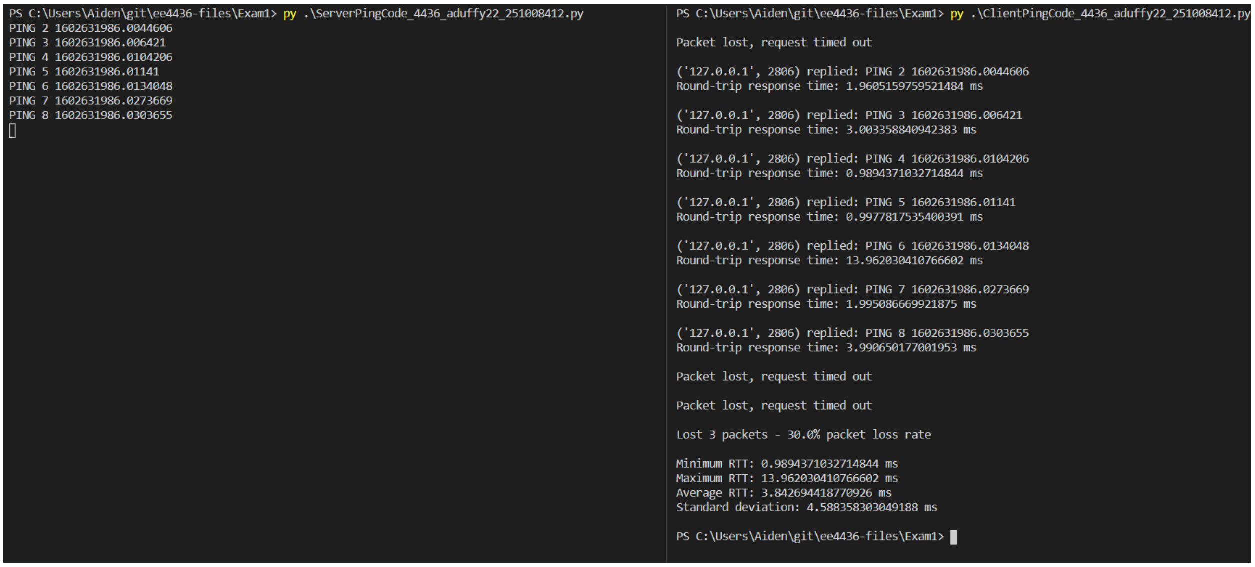
Q5-7

As with packet loss, there many packet retransmissions in the TCP connection.



```
289 11.53369900( 10.0.0.1     10.0.0.3     TCP     1514 [TCP Retransmission] 54754 > dhanalakshmi [ACK] Seq=321
290 11.53370600( 10.0.0.1     10.0.0.3     TCP     1514 [TCP Retransmission] 54754 > dhanalakshmi [ACK] Seq=326
```

Problem 4



```
PS C:\Users\Aiden\git\ee4436-files\Exam1> py .\ServerPingCode_4436_aduffy22_251008412.py
PING 2 1602631986.0044606
PING 3 1602631986.006421
PING 4 1602631986.0104206
PING 5 1602631986.01141
PING 6 1602631986.0134048
PING 7 1602631986.0273669
PING 8 1602631986.0303655
```

```
PS C:\Users\Aiden\git\ee4436-files\Exam1> py .\ClientPingCode_4436_aduffy22_251008412.py

Packet lost, request timed out

('127.0.0.1', 2806) replied: PING 2 1602631986.0044606
Round-trip response time: 1.9605159759521484 ms

('127.0.0.1', 2806) replied: PING 3 1602631986.006421
Round-trip response time: 3.003358840942383 ms

('127.0.0.1', 2806) replied: PING 4 1602631986.0104206
Round-trip response time: 0.9894371032714844 ms

('127.0.0.1', 2806) replied: PING 5 1602631986.01141
Round-trip response time: 0.9977817535400391 ms

('127.0.0.1', 2806) replied: PING 6 1602631986.0134048
Round-trip response time: 13.962030410766602 ms

('127.0.0.1', 2806) replied: PING 7 1602631986.0273669
Round-trip response time: 1.995086669921875 ms

('127.0.0.1', 2806) replied: PING 8 1602631986.0303655
Round-trip response time: 3.990650177001953 ms

Packet lost, request timed out

Packet lost, request timed out

Lost 3 packets - 30.0% packet loss rate

Minimum RTT: 0.9894371032714844 ms
Maximum RTT: 13.962030410766602 ms
Average RTT: 3.842694418770926 ms
Standard deviation: 4.588358303049188 ms

PS C:\Users\Aiden\git\ee4436-files\Exam1>
```

In the above, we can see the server running on the left, capitalizing the packets that it lets through. On the right, the client is printing the echo from the server and the RTT (if successful), or a timeout message (if unsuccessful). Once the client finishes its 10 pings, then it displays the packet loss in terms of count and percentage, followed by the minimum, maximum, average and standard deviations of the RTTs for all the successful pings.

Aiden Duffy 251008412

Problem 5

Givens

F$_{max}$ = 24 KHz
Encoding = 16 bits
L$_1$ = 2,000 m
L$_2$ = 6,000,000 m
L$_3$ = 6,000 m

B$_1$ = 20 Mbps
B$_2$ = 300 Gbps
B$_3$ = 20 Mbps
Delay$_q$ = e$^{(6x/x^{\wedge}3 + x + 2)}$
Delay$_{Total}$ = 0.15 seconds

Find One-Way Bandwidth

$$bandwidth = f_{max} * 2 \ * encoding$$
$$bandwidth = 24{,}000 \ * 2 \ * 16$$
$$bandwidth = 768{,}000 \ bits/s$$

Therefore, the one-way bandwidth is 768 Kbps

Find Delay$_{Total}$

Delay$_{Total}$ = Delay$_{Trans1}$ + Delay$_{Trans2}$ + Delay$_{Trans3}$ + Delay$_{Prop1}$ + Delay$_{Prop2}$ + Delay$_{Prop3}$ + 2 * Delay$_q$

$$0.15 = \frac{768 * 10^3}{20 * 10^6} + \frac{768 * 10^3}{300 * 10^9} + \frac{768 * 10^3}{20 * 10^6} + \frac{2 * 10^3}{2.8 * 10^8} + \frac{6 * 10^6}{2.8 * 10^8} + \frac{6 * 10^3}{2.8 * 10^8} + 2\left(e^{\frac{6x}{x^3+x+2}}\right)$$

$$0.15 - \frac{24}{625} - \frac{1}{390625} - \frac{24}{625} - \frac{1}{140000} - \frac{3}{140} - \frac{3}{140000} = 2\left(e^{\frac{6x}{x^3+x+2}}\right)$$

$$\frac{0.15 - \frac{48}{625} - \frac{1}{390625} - \frac{4}{140000} - \frac{3}{140}}{2} = \left(e^{\frac{6x}{x^3+x+2}}\right)$$

Take the left side of the above equation, and set the expression equal to a variable to simplify the following algebra

$$\therefore \ let \ a = \frac{0.15 - \frac{48}{625} - \frac{1}{390625} - \frac{4}{140000} - \frac{3}{140}}{2}$$

$$a = e^{\frac{6x}{x^3+x+2}}$$

$$\ln(a) = \left(\frac{6x}{x^3 + x + 2}\right)$$

$$\ln(a) * (x^3 + x + 2) = 6x$$

$$\ln(a)\, x^3 + [\ln(a) - 6]x + 2\ln(a) = 0$$

Use cubic formula to solve for x

$$x\ =\sqrt[3]{\left(\frac{0}{27\ln(a)^3}+\frac{0*(\ln(a)-6)}{6\ln(a)^2}-\frac{2\ln(a)}{2\ln(a)}\right)+\sqrt{\left(\frac{0}{27\ln(a)^3}+\frac{0*(\ln(a)-6)}{6\ln(a)^2}-\frac{2\ln(a)}{2\ln(a)}\right)^2+\left(\frac{\ln(a)-6}{3\ln(a)}-\frac{0^2}{9\ln(a)^2}\right)^3}}$$
$$+\sqrt[3]{\left(\frac{0}{27\ln(a)^3}+\frac{0*(\ln(a)-6)}{6\ln(a)^2}-\frac{2\ln(a)}{2\ln(a)}\right)-\sqrt{\left(\frac{0}{27\ln(a)^3}+\frac{0*(\ln(a)-6)}{6\ln(a)^2}-\frac{2\ln(a)}{2\ln(a)}\right)^2+\left(\frac{\ln(a)-6}{3\ln(a)}-\frac{0^2}{9\ln(a)^2}\right)^3}}$$
$$-\frac{0}{3\ln(a)}$$

$$x=\sqrt[3]{(-1)+\sqrt{(-1)^2+\left(\frac{\ln(a)-6}{3\ln(a)}\right)^3}}+\sqrt[3]{(-1\ )-\sqrt{(-1)^2+\left(\frac{\ln(a)-6}{3\ln(a)}\right)^3}}$$

$$x=\sqrt[3]{-1+\sqrt{1+\left(\frac{\ln(a)-6}{3\ln(a)}\right)^3}}+\sqrt[3]{-1-\sqrt{1+\left(\frac{\ln(a)-6}{3\ln(a)}\right)^3}}$$

Now substitute in the value of a

$$a=\frac{0.15-\frac{48}{625}-\frac{1}{390625}-\frac{4}{140000}-\frac{3}{140}}{2}$$

$$a\cong 0.02587014857$$

$$x=\sqrt[3]{-1+\sqrt{1+\left(\frac{\ln(0.02587014857)-6}{3\ln(0.02587014857)}\right)^3}}+\sqrt[3]{-1-\sqrt{1+\left(\frac{\ln(0.02587014857)-6}{3\ln(0.02587014857)}\right)^3}}$$

$$\therefore x\cong -0.6521$$

Therefore, the x in the equation for the queuing delays is x = ~ -0.6251