Alex Shah
EN.695.741.81.SP25 Information Assurance Analysis
Mod 12 Assignment
April 22, 2025

Part 1

I used a Ubuntu 18.04 VM and installed Wireshark 2.6.10 through the Ubuntu Store and performed multiple packet captures. I had initially tried one packet capture visiting the required websites and running ping and traceroute, but the capture had too many addresses and background noise to determine which connections and addresses were visited as part of one website visit vs another. I then ran packet captures for each website visit and ping and traceroute individually while using the all in one capture to make larger observations.

Part 2

1. What IP protocols did you observe in your traffic sample? Provide a chart indicating the relative byte and packet volume for each protocol present. Note any protocols that you were not expecting to see & explain what those protocols do.

In the packet capture where I ran all tasks, I sorted the statistics view by packets and then by bytes to analyze the amounts of traffic from different protocols. I was surprised to see any IPv6 packets because they are not supported by my ISP. I also did not expect to see such a large amount of packets and bytes over UDP, since I would anticipate text articles and images to be loaded by TCP. However I know of more modern exchange methods over UDP that avoid TCP like QUIC that might have delivered the website media files, resources like fonts, and ads. More bytes were sent as TCP traffic than as UDP except when considering UDP Data in which there was more UDP data sent in bytes than TCP. Most of the TCP traffic was over SSL, indicating that the HTTP connection I initiated was upgraded to HTTPS. For navigating to a handful of websites, there were 1214 DNS packets which shows just how many domains are involved when retrieving something as simple as a homepage today.

Wireshark · Protocol Hierarchy Statistics · lab12pcap.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	19440	100.0	8658740	999 k	0	0	0
Ethernet	100.0	19440	3.1	272160	31 k	0	0	0
Internet Protocol Version 4	100.0	19434	4.5	388680	44 k	0	0	0
Transmission Control Protocol	64.8	12599	43.8	3795836	438 k	8127	1165038	134 k
Secure Sockets Layer	23.1	4497	43.3	3752767	433 k	4213	2745027	316 k
Hypertext Transfer Protocol	1.3	257	3.1	266935	30 k	8	2215	255
Online Certificate Status Protocol	1.3	244	1.0	82993	9,577	244	88382	10 k
Line-based text data	0.0	4	0.0	2355	271	4	2625	302
JPEG File Interchange Format	0.0	1	0.9	75534	8,717	1	75807	8,748
Domain Name System	0.0	2	0.0	717	82	2	717	82
User Datagram Protocol	34.8	6769	0.6	54152	6,249	0	0	0
Data	28.6	5555	46.4	4017956	463 k	5555	4017956	463 k
Domain Name System	6.2	1214	1.1	95943	11 k	1214	95943	11 k
Internet Control Message Protocol	0.3	66	0.2	13862	1,599	66	13862	1,599
Internet Protocol Version 6	0.0	6	0.0	240	27	0	0	0
Transmission Control Protocol	0.0	4	0.0	120	13	4	120	13
Internet Control Message Protocol v6	0.0	2	0.0	64	7	2	64	7

Figure 1: Statistics by packet volume

Wireshark · Protocol Hierarchy Statistics · lab12pcap.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	19440	100.0	8658740	999 k	0	0	0
Ethernet	100.0	19440	3.1	272160	31 k	0	0	0
Internet Protocol Version 4	100.0	19434	4.5	388680	44 k	0	0	0
Transmission Control Protocol	64.8	12599	43.8	3795836	438 k	8127	1165038	134 k
Secure Sockets Layer	23.1	4497	43.3	3752767	433 k	4213	2745027	316 k
Hypertext Transfer Protocol	1.3	257	3.1	266935	30 k	8	2215	255
Online Certificate Status Protocol	1.3	244	1.0	82993	9,577	244	88382	10 k
JPEG File Interchange Format	0.0	1	0.9	75534	8,717	1	75807	8,748
Line-based text data	0.0	4	0.0	2355	271	4	2625	302
Domain Name System	0.0	2	0.0	717	82	2	717	82
User Datagram Protocol	34.8	6769	0.6	54152	6,249	0	0	0
Data	28.6	5555	46.4	4017956	463 k	5555	4017956	463 k
Domain Name System	6.2	1214	1.1	95943	11 k	1214	95943	11 k
Internet Control Message Protocol	0.3	66	0.2	13862	1,599	66	13862	1,599
Internet Protocol Version 6	0.0	6	0.0	240	27	0	0	0
Transmission Control Protocol	0.0	4	0.0	120	13	4	120	13
Internet Control Message Protocol v6	0.0	2	0.0	64	7	2	64	7

Figure 2: Statistics by bytes

2. Filter your traffic capture to only TCP traffic to or from port 80 and answer the following questions for each web site you visited:

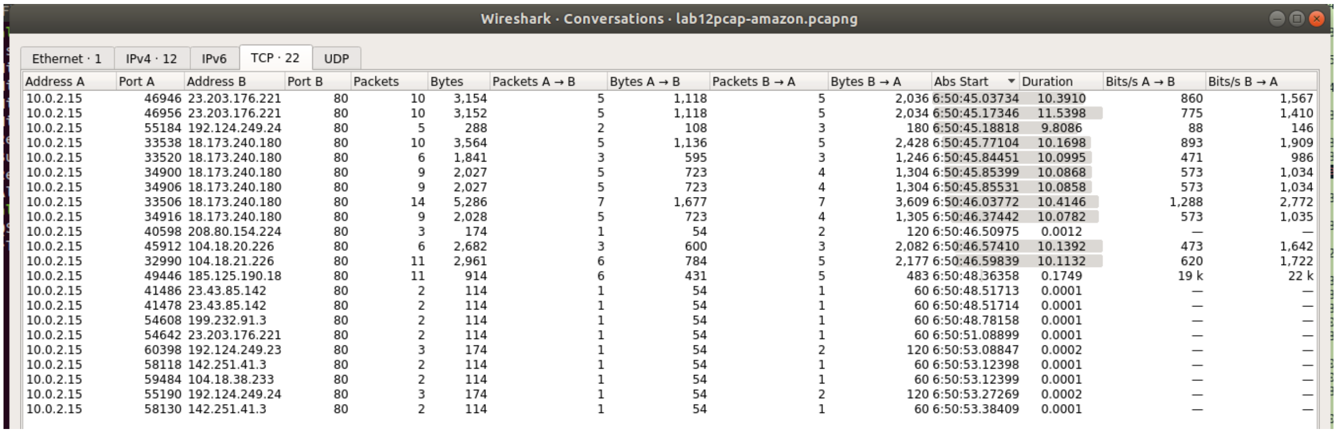
a. How many TCP/IP sessions did your computer make to connect to each site?

Each website visit connected to several domains on port 80 by TCP, and many more connections occurred over other ports like DNS on port 53, HTTPS traffic on 443, and other traffic on non standard ports. I used Wireshark's Conversations view to filter and show only tcp traffic on port 80. This shows TCP/IP sessions involved in visiting each of the webpages in the titlebar. Each webpage involved about 20 TCP/IP sessions over port 80 to several IP addresses. Some of the webpages started off with larger bursts of data in bytes, or more packets, and tapered off like aleae and amazon. And others were more consistent with fewer bytes/packets for CNN and more packets and larger transfers overall for fox news. More data was received from the webserver in these sessions than was sent from the host on port 80, which makes sense when making requests of the webserver.

Wireshark · Conversations · lab12pcap-aleae.pcapng

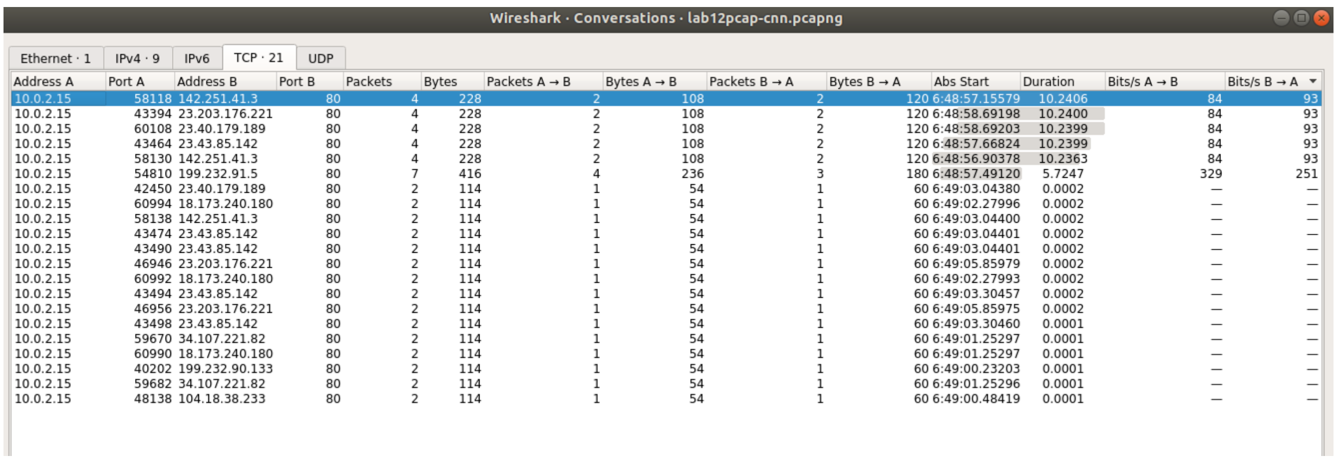
Ethernet · 1 | IPv4 · 11 | IPv6 | TCP · 20 | UDP

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Abs Start	Duration	Bits/s A → B	Bits/s B → A
10.0.2.15	42536	216.92.179.155	80	15	4,055	7	693	8	3,362	6:50:14.11394	0.1412	39 k	190 k
10.0.2.15	42538	216.92.179.155	80	7	416	4	236	3	180	6:50:14.12950	5.2868	357	272
10.0.2.15	55184	192.124.249.24	80	4	228	2	108	2	120	6:50:14.46796	10.2679	84	93
10.0.2.15	59682	34.107.221.82	80	4	228	2	108	2	120	6:50:15.38769	0.0067	129 k	143 k
10.0.2.15	59670	34.107.221.82	80	4	228	2	108	2	120	6:50:15.38772	0.0076	114 k	126 k
10.0.2.15	41486	23.43.85.142	80	2	114	1	54	1	60	6:50:17.79785	0.0001	—	—
10.0.2.15	41478	23.43.85.142	80	2	114	1	54	1	60	6:50:17.80616	0.0001	—	—
10.0.2.15	54608	199.232.91.3	80	2	114	1	54	1	60	6:50:18.06486	0.0001	—	—
10.0.2.15	54642	23.203.176.221	80	2	114	1	54	1	60	6:50:20.36117	0.0001	—	—
10.0.2.15	46946	23.203.176.221	80	2	114	1	54	1	60	6:50:22.14970	0.0001	—	—
10.0.2.15	45912	104.18.20.226	80	2	114	1	54	1	60	6:50:22.40561	0.0001	—	—
10.0.2.15	59484	104.18.38.233	80	2	114	1	54	1	60	6:50:22.40562	0.0001	—	—
10.0.2.15	46956	23.203.176.221	80	2	114	1	54	1	60	6:50:22.40563	0.0001	—	—
10.0.2.15	33538	18.173.240.180	80	2	114	1	54	1	60	6:50:22.40728	0.0001	—	—
10.0.2.15	33506	18.173.240.180	80	2	114	1	54	1	60	6:50:22.40784	0.0001	—	—
10.0.2.15	58118	142.251.41.3	80	2	114	1	54	1	60	6:50:22.40785	0.0001	—	—
10.0.2.15	33520	18.173.240.180	80	2	114	1	54	1	60	6:50:22.65984	0.0001	—	—
10.0.2.15	60398	192.124.249.23	80	2	114	1	54	1	60	6:50:22.65988	0.0001	—	—
10.0.2.15	58130	142.251.41.3	80	2	114	1	54	1	60	6:50:22.66019	0.0001	—	—
10.0.2.15	55190	192.124.249.24	80	2	114	1	54	1	60	6:50:22.92024	0.0001	—	—


Wireshark · Conversations · lab12pcap-amazon.pcapng

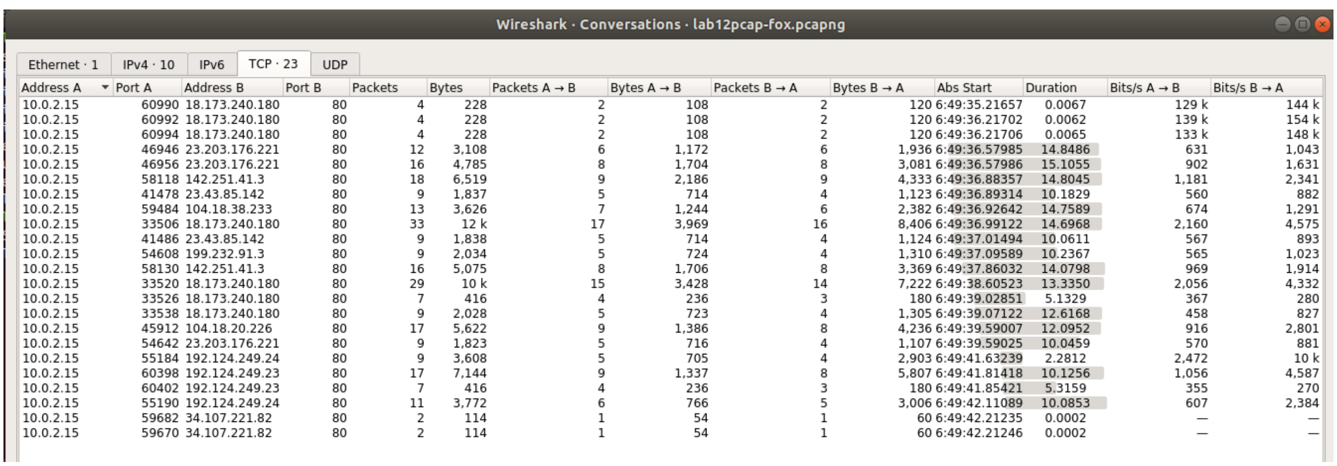
Ethernet · 1 | IPv4 · 12 | IPv6 | TCP · 22 | UDP

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Abs Start	Duration	Bits/s A → B	Bits/s B → A
10.0.2.15	46946	23.203.176.221	80	10	3,154	5	1,118	5	2,036	6:50:45.03734	10.3910	860	1,567
10.0.2.15	46956	23.203.176.221	80	10	3,152	5	1,118	5	2,034	6:50:45.17346	11.5398	775	1,410
10.0.2.15	55184	192.124.249.24	80	5	288	2	108	3	180	6:50:45.18818	9.8086	88	146
10.0.2.15	33538	18.173.240.180	80	10	3,564	5	1,136	5	2,428	6:50:45.77104	10.1698	893	1,909
10.0.2.15	33520	18.173.240.180	80	6	1,841	3	595	3	1,246	6:50:45.84451	10.0995	471	986
10.0.2.15	34900	18.173.240.180	80	9	2,027	5	723	4	1,304	6:50:45.85399	10.0868	573	1,034
10.0.2.15	34906	18.173.240.180	80	9	2,027	5	723	4	1,304	6:50:45.85531	10.0858	573	1,034
10.0.2.15	33506	18.173.240.180	80	14	5,286	7	1,677	7	3,609	6:50:46.03772	10.4146	1,288	2,772
10.0.2.15	34916	18.173.240.180	80	9	2,028	5	723	4	1,305	6:50:46.37442	10.0782	573	1,035
10.0.2.15	40598	208.80.154.224	80	3	174	1	54	2	120	6:50:46.50975	0.0012	—	—
10.0.2.15	45912	104.18.20.226	80	6	2,682	3	600	3	2,082	6:50:46.57410	10.1392	473	1,642
10.0.2.15	32990	104.18.21.226	80	11	2,961	6	784	5	2,177	6:50:46.59839	10.1132	620	1,722
10.0.2.15	49446	185.125.190.18	80	11	914	6	431	5	483	6:50:48.36358	0.1749	19 k	22 k
10.0.2.15	41486	23.43.85.142	80	2	114	1	54	1	60	6:50:48.51713	0.0001	—	—
10.0.2.15	41478	23.43.85.142	80	2	114	1	54	1	60	6:50:48.51714	0.0001	—	—
10.0.2.15	54608	199.232.91.3	80	2	114	1	54	1	60	6:50:48.78158	0.0001	—	—
10.0.2.15	54642	23.203.176.221	80	2	114	1	54	1	60	6:50:51.08899	0.0001	—	—
10.0.2.15	60398	192.124.249.23	80	3	174	1	54	2	120	6:50:53.08847	0.0002	—	—
10.0.2.15	58118	142.251.41.3	80	2	114	1	54	1	60	6:50:53.12398	0.0001	—	—
10.0.2.15	59484	104.18.38.233	80	2	114	1	54	1	60	6:50:53.12399	0.0001	—	—
10.0.2.15	55190	192.124.249.24	80	3	174	1	54	2	120	6:50:53.27269	0.0002	—	—
10.0.2.15	58130	142.251.41.3	80	2	114	1	54	1	60	6:50:53.38409	0.0001	—	—


Wireshark · Conversations · lab12pcap-cnn.pcapng

Ethernet · 1 | IPv4 · 9 | IPv6 | TCP · 21 | UDP

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Abs Start	Duration	Bits/s A → B	Bits/s B → A
10.0.2.15	58118	142.251.41.3	80	4	228	2	108	2	120	6:48:57.15579	10.2406	84	93
10.0.2.15	43394	23.203.176.221	80	4	228	2	108	2	120	6:48:58.69198	10.2400	84	93
10.0.2.15	60108	23.40.179.189	80	4	228	2	108	2	120	6:48:58.69203	10.2399	84	93
10.0.2.15	43464	23.43.85.142	80	4	228	2	108	2	120	6:48:57.66824	10.2399	84	93
10.0.2.15	58130	142.251.41.3	80	4	228	2	108	2	120	6:48:56.90378	10.2363	84	93
10.0.2.15	54810	199.232.91.5	80	7	416	4	236	3	180	6:48:57.49120	5.7247	329	251
10.0.2.15	42450	23.40.179.189	80	2	114	1	54	1	60	6:49:03.04380	0.0002	—	—
10.0.2.15	60994	18.173.240.180	80	2	114	1	54	1	60	6:49:02.27996	0.0002	—	—
10.0.2.15	58138	142.251.41.3	80	2	114	1	54	1	60	6:49:03.04400	0.0002	—	—
10.0.2.15	43474	23.43.85.142	80	2	114	1	54	1	60	6:49:03.04401	0.0002	—	—
10.0.2.15	43490	23.43.85.142	80	2	114	1	54	1	60	6:49:03.04401	0.0002	—	—
10.0.2.15	46946	23.203.176.221	80	2	114	1	54	1	60	6:49:05.85979	0.0002	—	—
10.0.2.15	60992	18.173.240.180	80	2	114	1	54	1	60	6:49:02.27993	0.0002	—	—
10.0.2.15	43494	23.43.85.142	80	2	114	1	54	1	60	6:49:03.30457	0.0002	—	—
10.0.2.15	46956	23.203.176.221	80	2	114	1	54	1	60	6:49:05.85975	0.0002	—	—
10.0.2.15	43498	23.43.85.142	80	2	114	1	54	1	60	6:49:03.30460	0.0001	—	—
10.0.2.15	59670	34.107.221.82	80	2	114	1	54	1	60	6:49:01.25297	0.0001	—	—
10.0.2.15	60990	18.173.240.180	80	2	114	1	54	1	60	6:49:01.25297	0.0001	—	—
10.0.2.15	40202	199.232.90.133	80	2	114	1	54	1	60	6:49:00.23203	0.0001	—	—
10.0.2.15	59682	34.107.221.82	80	2	114	1	54	1	60	6:49:01.25296	0.0001	—	—
10.0.2.15	48138	104.18.38.233	80	2	114	1	54	1	60	6:49:00.48419	0.0001	—	—


Wireshark · Conversations · lab12pcap-fox.pcapng

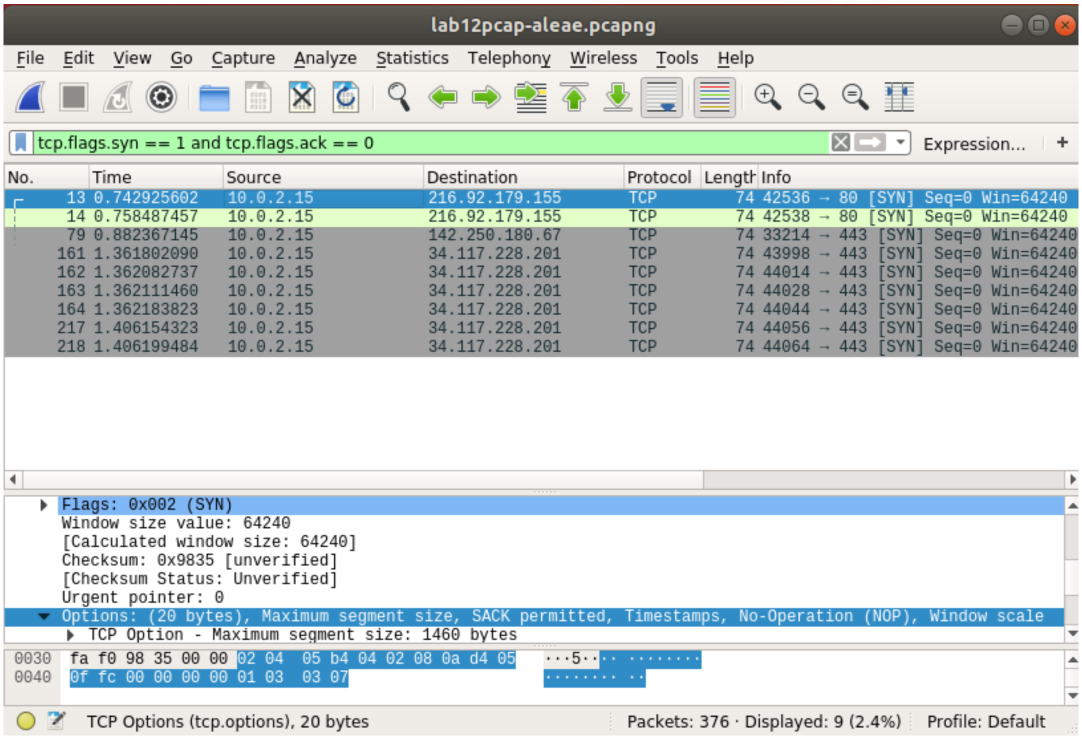
Ethernet · 1 | IPv4 · 10 | IPv6 | TCP · 23 | UDP

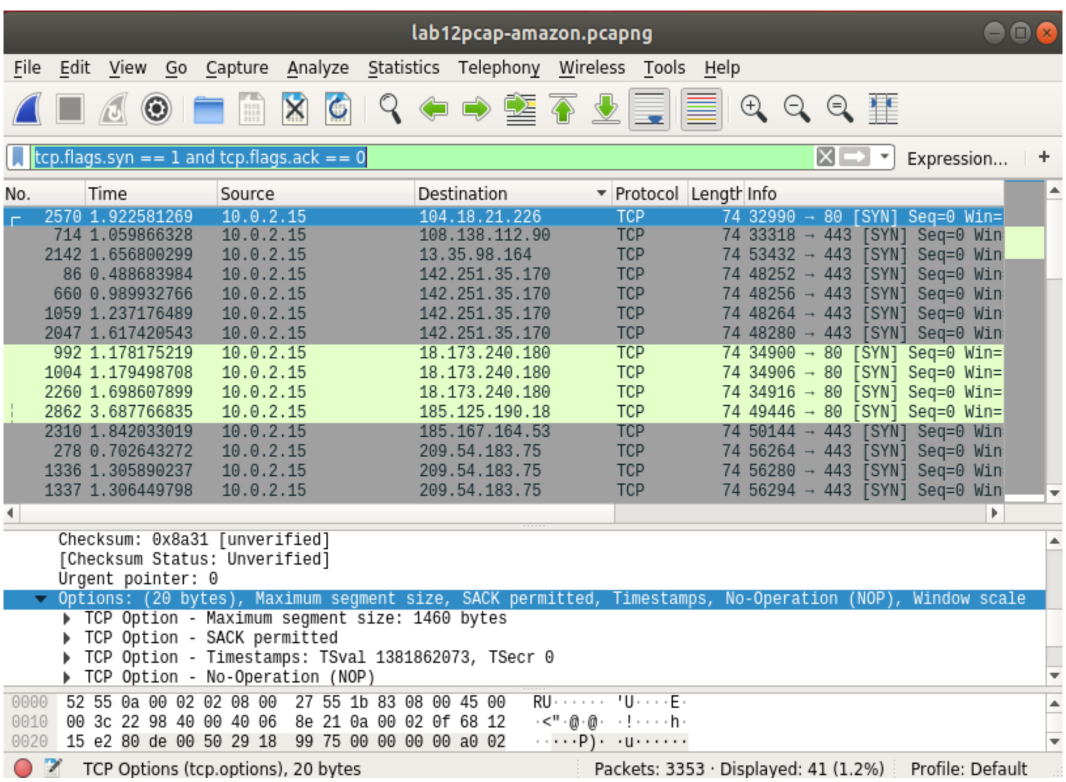
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Abs Start	Duration	Bits/s A → B	Bits/s B → A
10.0.2.15	60990	18.173.240.180	80	4	228	2	108	2	120	6:49:35.21657	0.0067	129 k	144 k
10.0.2.15	60992	18.173.240.180	80	4	228	2	108	2	120	6:49:36.21702	0.0062	139 k	154 k
10.0.2.15	60994	18.173.240.180	80	4	228	2	108	2	120	6:49:36.21706	0.0065	133 k	148 k
10.0.2.15	46946	23.203.176.221	80	12	3,108	6	1,172	6	1,936	6:49:36.57985	14.8486	631	1,043
10.0.2.15	46956	23.203.176.221	80	16	4,785	8	1,704	8	3,081	6:49:36.57986	15.1055	902	1,631
10.0.2.15	58118	142.251.41.3	80	18	6,519	9	2,186	9	4,333	6:49:36.88357	14.8045	1,181	2,341
10.0.2.15	41478	23.43.85.142	80	9	1,837	5	714	4	1,123	6:49:36.89314	10.1829	560	882
10.0.2.15	59484	104.18.38.233	80	13	3,626	7	1,244	6	2,382	6:49:36.92642	14.7589	674	1,291
10.0.2.15	33506	18.173.240.180	80	33	12 k	17	3,969	16	8,406	6:49:36.99122	14.6968	2,160	4,575
10.0.2.15	41486	23.43.85.142	80	9	1,838	5	714	4	1,124	6:49:37.01494	10.0611	567	893
10.0.2.15	54608	199.232.91.3	80	9	2,034	5	724	4	1,310	6:49:37.09589	10.2367	565	1,023
10.0.2.15	58130	142.251.41.3	80	16	5,075	8	1,706	8	3,369	6:49:37.86032	14.0798	969	1,914
10.0.2.15	33520	18.173.240.180	80	29	10 k	15	3,428	14	7,222	6:49:38.60523	13.3350	2,056	4,332
10.0.2.15	33526	18.173.240.180	80	7	416	4	236	3	180	6:49:39.02851	5.1329	367	280
10.0.2.15	33538	18.173.240.180	80	9	2,028	5	723	4	1,305	6:49:39.07122	12.6168	458	827
10.0.2.15	45912	104.18.20.226	80	17	5,622	9	1,386	8	4,236	6:49:39.59007	12.0952	916	2,801
10.0.2.15	54642	23.203.176.221	80	9	1,823	5	716	4	1,107	6:49:39.59025	10.0459	570	881
10.0.2.15	55184	192.124.249.24	80	9	3,608	5	705	4	2,903	6:49:41.63239	2.2812	2,472	10 k
10.0.2.15	60398	192.124.249.23	80	17	7,144	9	1,337	8	5,807	6:49:41.81418	10.1256	1,056	4,587
10.0.2.15	60402	192.124.249.23	80	7	416	4	236	3	180	6:49:41.85421	5.3159	355	270
10.0.2.15	55190	192.124.249.24	80	11	3,772	6	766	5	3,006	6:49:42.11089	10.0853	607	2,384
10.0.2.15	59682	34.107.221.82	80	2	114	1	54	1	60	6:49:42.21235	0.0002	—	—
10.0.2.15	59670	34.107.221.82	80	2	114	1	54	1	60	6:49:42.21246	0.0002	—	—

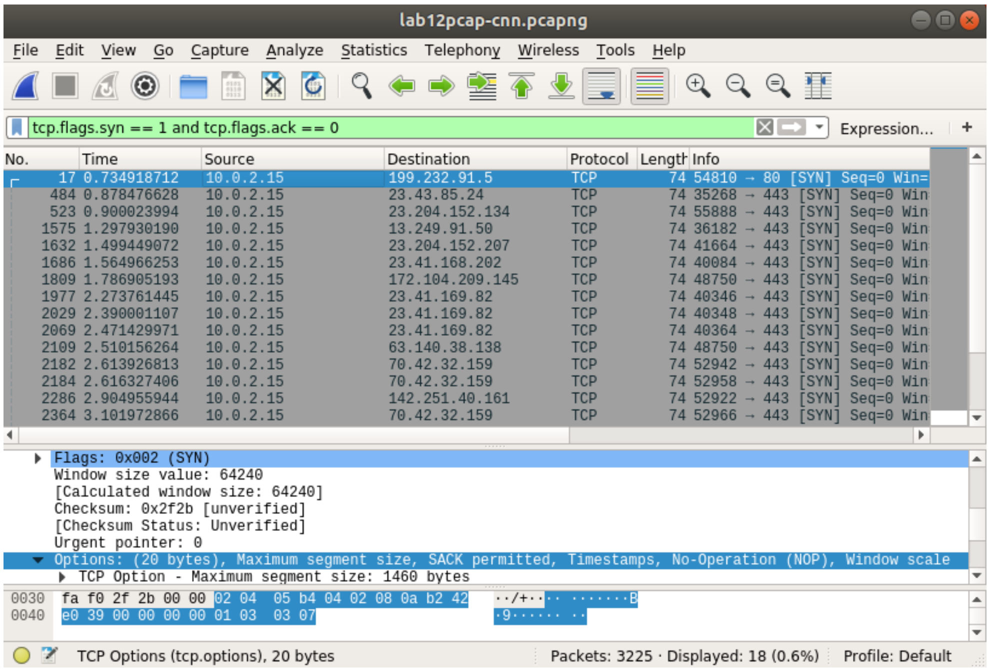
Figures 3-6: Website TCP port 80 connections

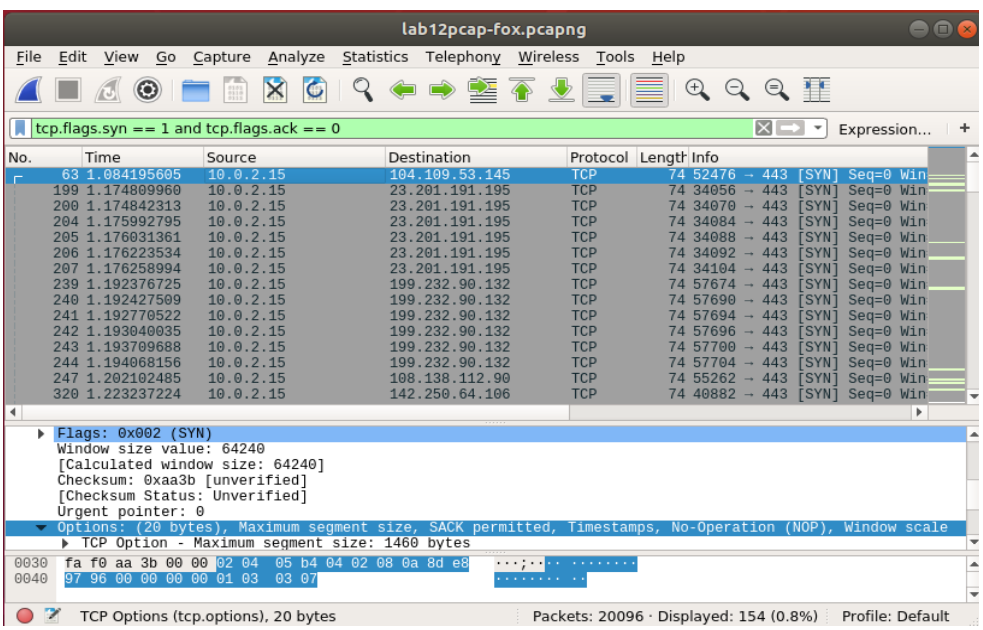
b. What TCP options (if any, i.e., MSS, NOP, Window Scale, SACK) were used for the connection to the web server (i.e. cnn, foxnews)?

I examined the SYN packet flags to look for TCP options. For each of the websites the flags were the same. They have a maximum segment size of 1460 bytes, permit SACK which allows selective acknowledgements to resend only missing portions, has timestamps, has a NOP flag for padding, and supports Window scale to enlarge the TCP window size and send more data between acknowledgements.









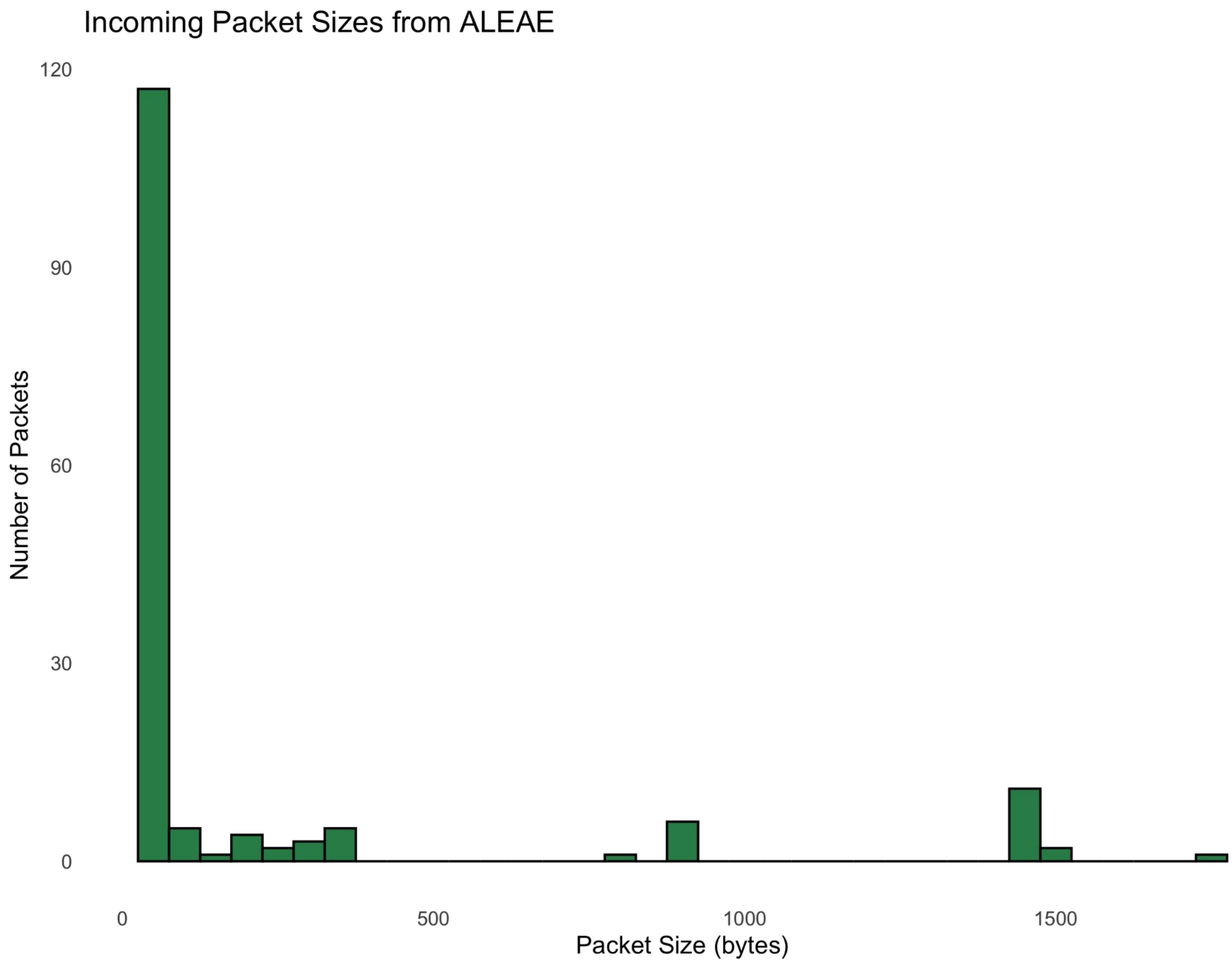
Figures 7-10: SYN packets and their TCP flags for each website

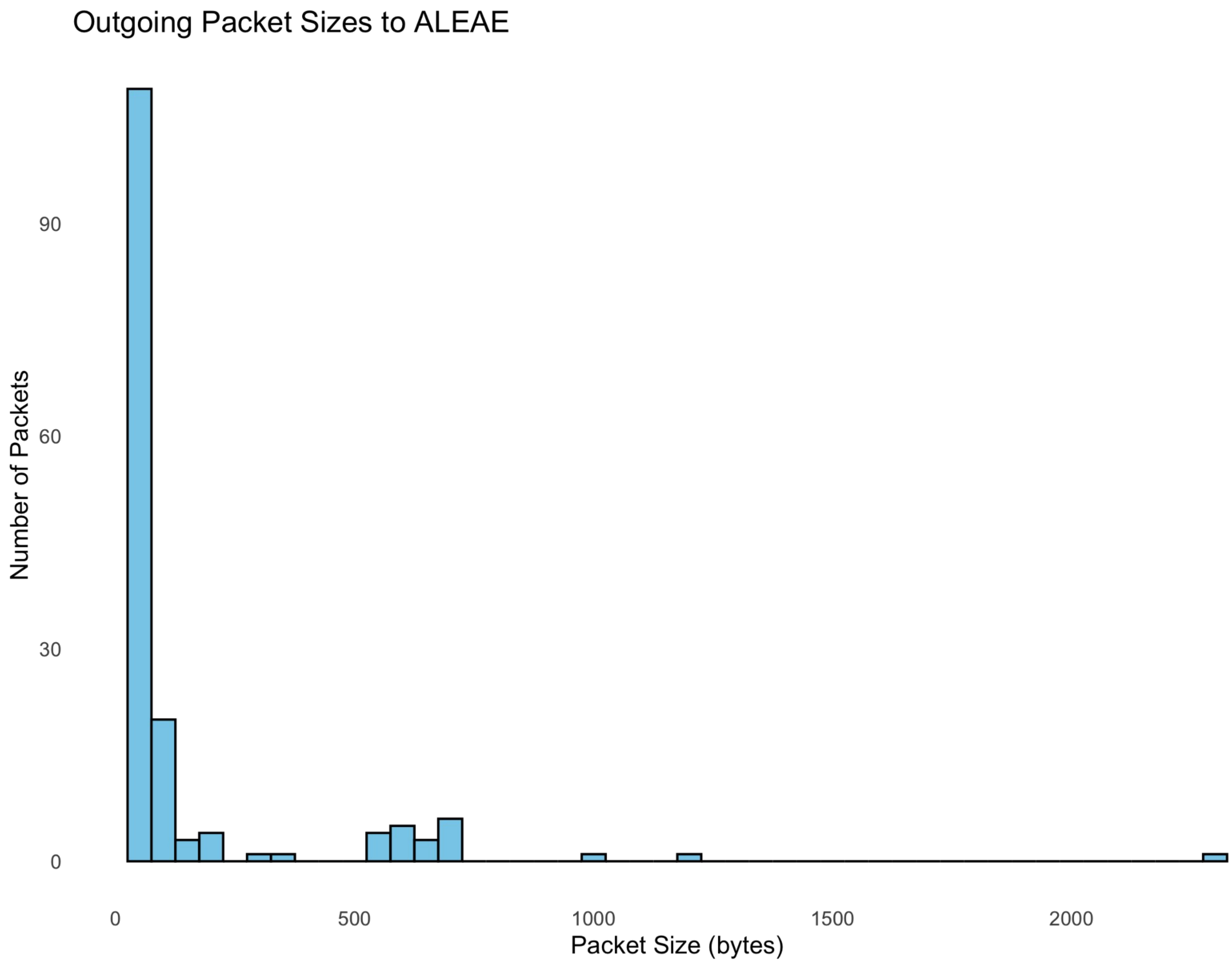
c. How many IP addresses did your system connect to after making a request to the website?

For each of the websites, there was a lot of traffic to CDNs, Google, AWS, advertisers, and other 3rd party domains. For each of the websites, I looked at the list of tcp port 80 traffic and I counted the number of unique IP addresses in each capture. Aleae had 11, Amazon had 11, CNN had 9, and Fox news had 10 IP addresses connected to from the host when making requests to the website.

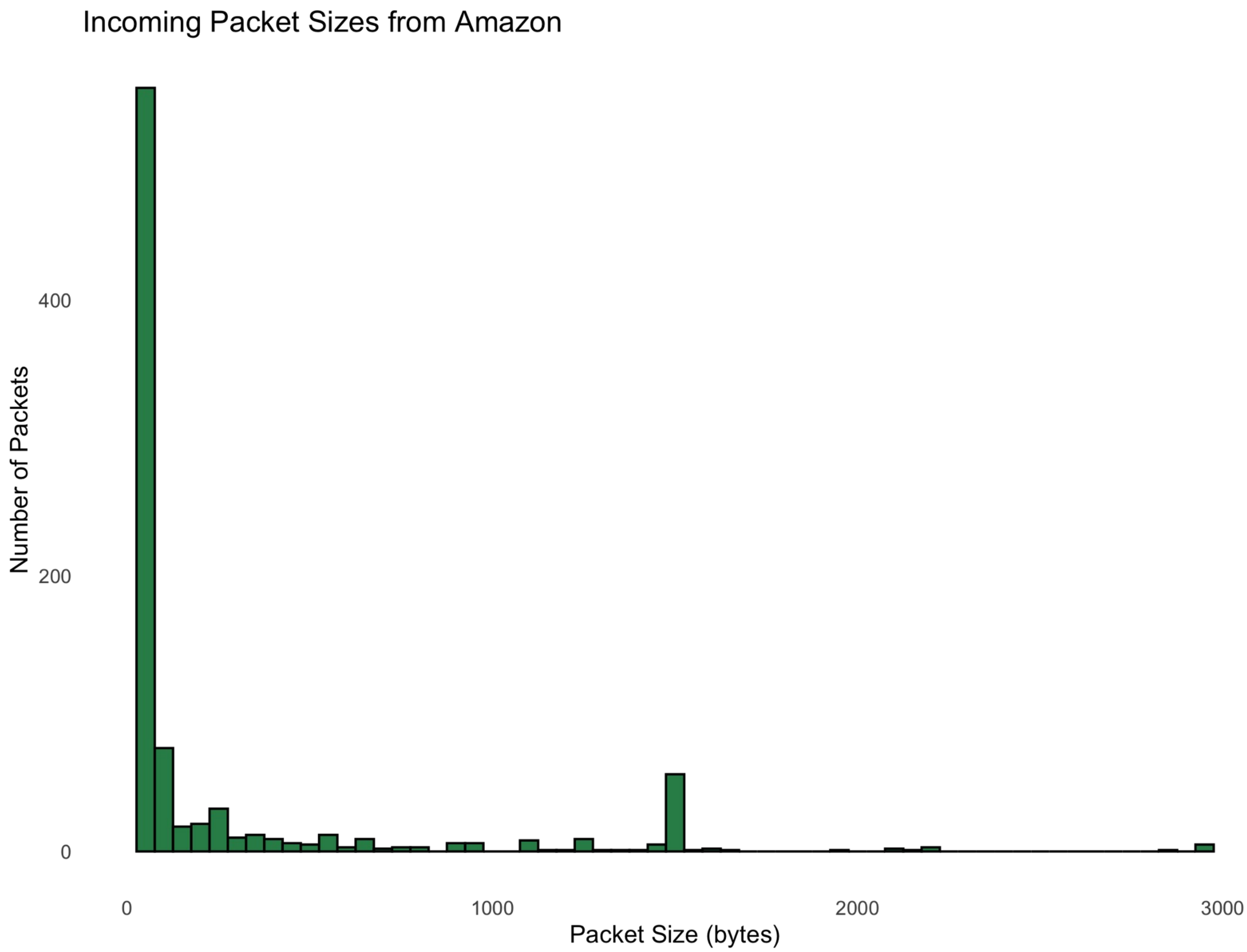
3. Using Excel, GNUPlot, or R develop two histograms per website visited: a.) the bytes per packet for traffic going to the webservers you visited and b.) the bytes per packet for traffic coming back from the webservers you visited. In these histograms, the x- axis should be the size of the packets and the y- axis should be the number of packets. Your images should have descriptive axis labels, a title, and NO grid lines. Compare and contrast the results. What can be said about the; size of your web request and the size of the webservers response? How do the response sizes from each of the sites compare?

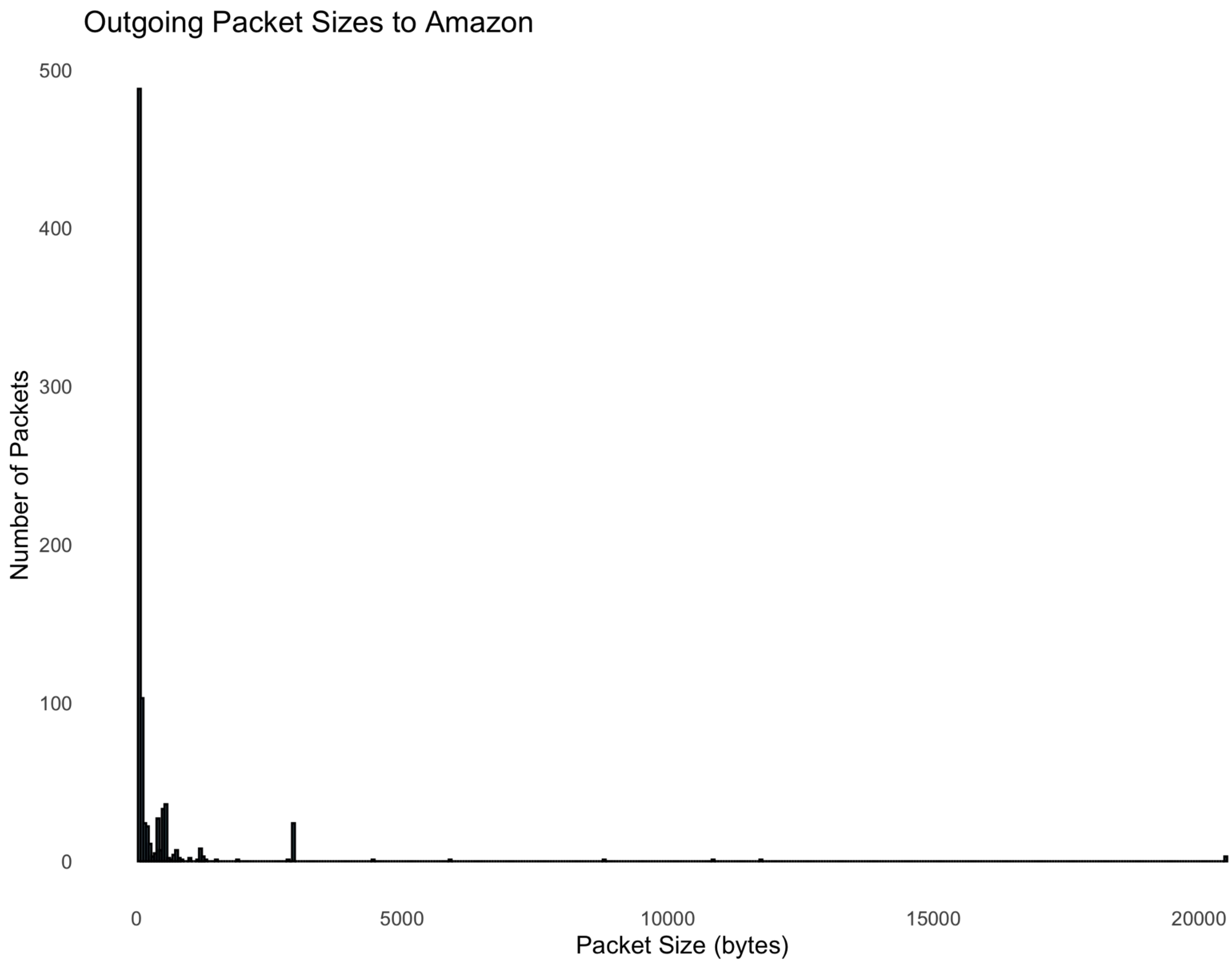
I filtered the results in wireshark to include tcp port 80 and 443 traffic. I then exported the results as a csv that I brought into R and plotted and saved with ggplot2. I looked at traffic coming to or going from the host as incoming and outgoing traffic for the histograms. Looking at the packet count by size for each website, there were some interesting patterns. Almost all websites had the highest packet counts for the smallest packet sizes, both incoming and outgoing. Except for Fox which sent and received vastly more packets than the other websites, and the highest count of packets in the incoming traffic were not the smallest packets received, but somewhere near the smaller end. The website aleae had the smallest counts and sizes overall, and the website fox had the largest counts and sizes. For most of the websites the largest outgoing packets were larger then the largest incoming packet sizes, such as Amazon which had a peak incoming packet size around 3000 bytes, but the largest outgoing packet was over 20,000 bytes. So a few large packets are sent out from the host when connecting to each domain, with some websites sent more data than others. While most incoming packet sizes peaked around a few thousand bytes, the largest packet received from Fox was over 30,000 bytes.

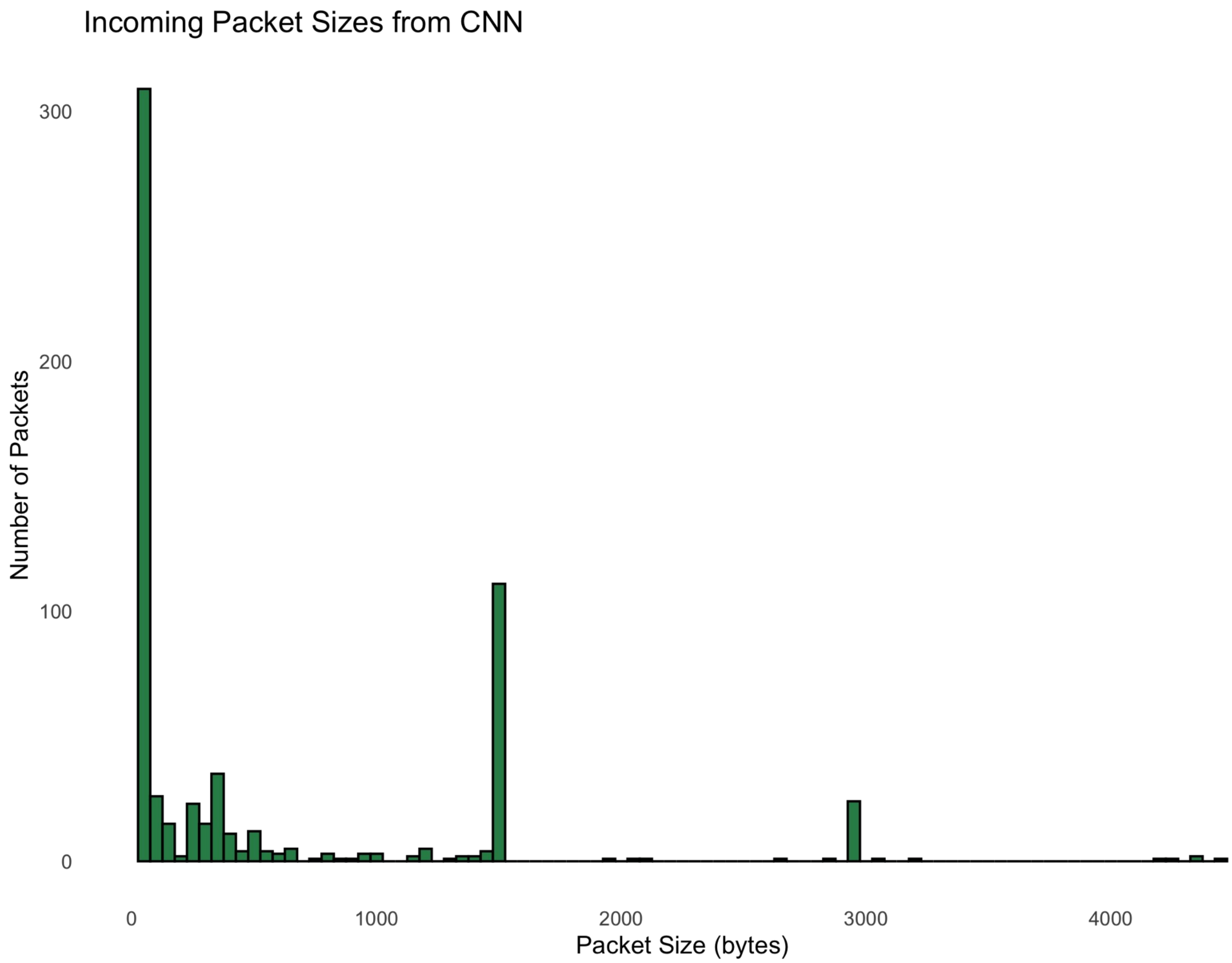

Incoming Packet Sizes from ALEAE
Number of Packets
120
90
60
30
0
0
500
1000
1500
Packet Size (bytes)

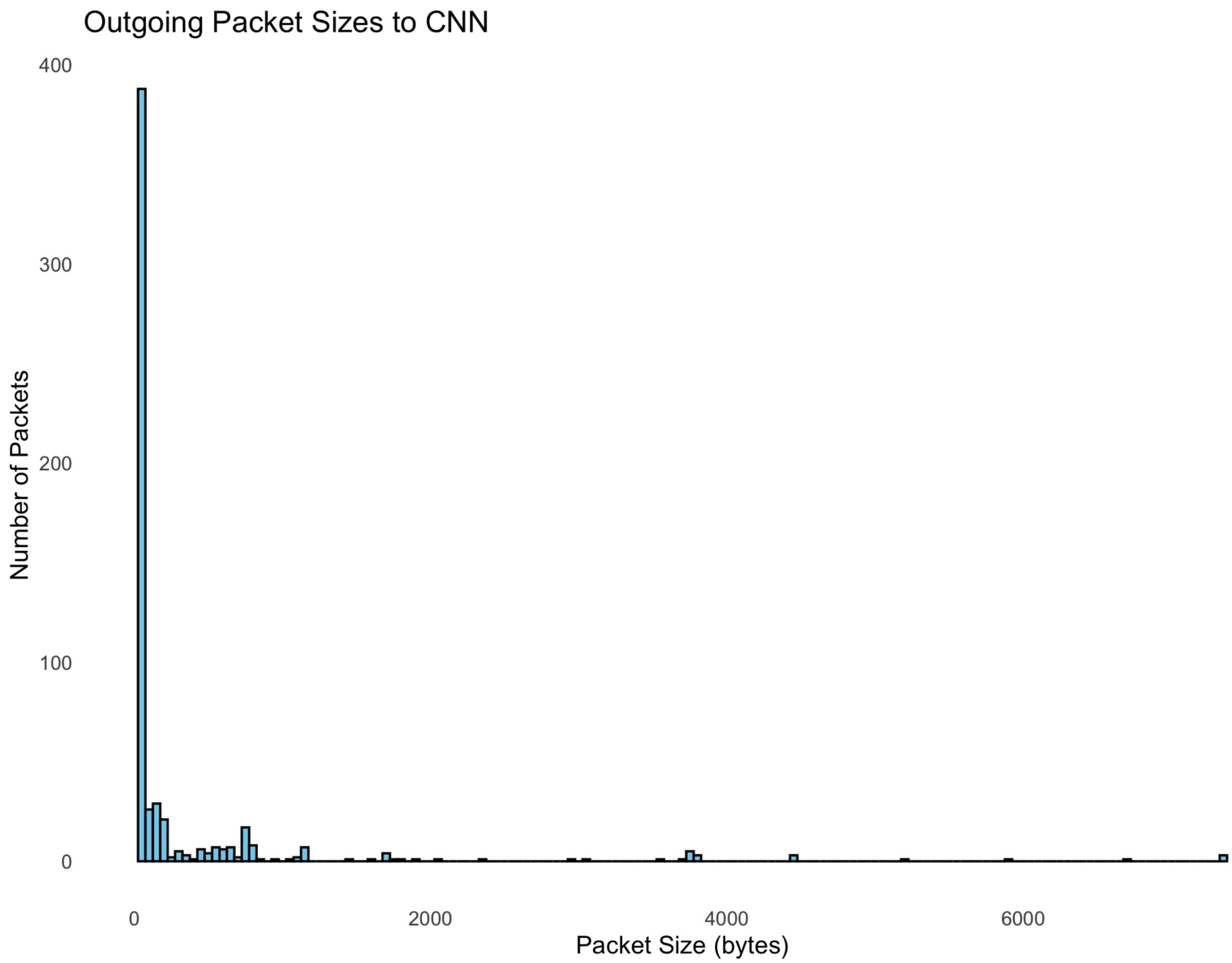


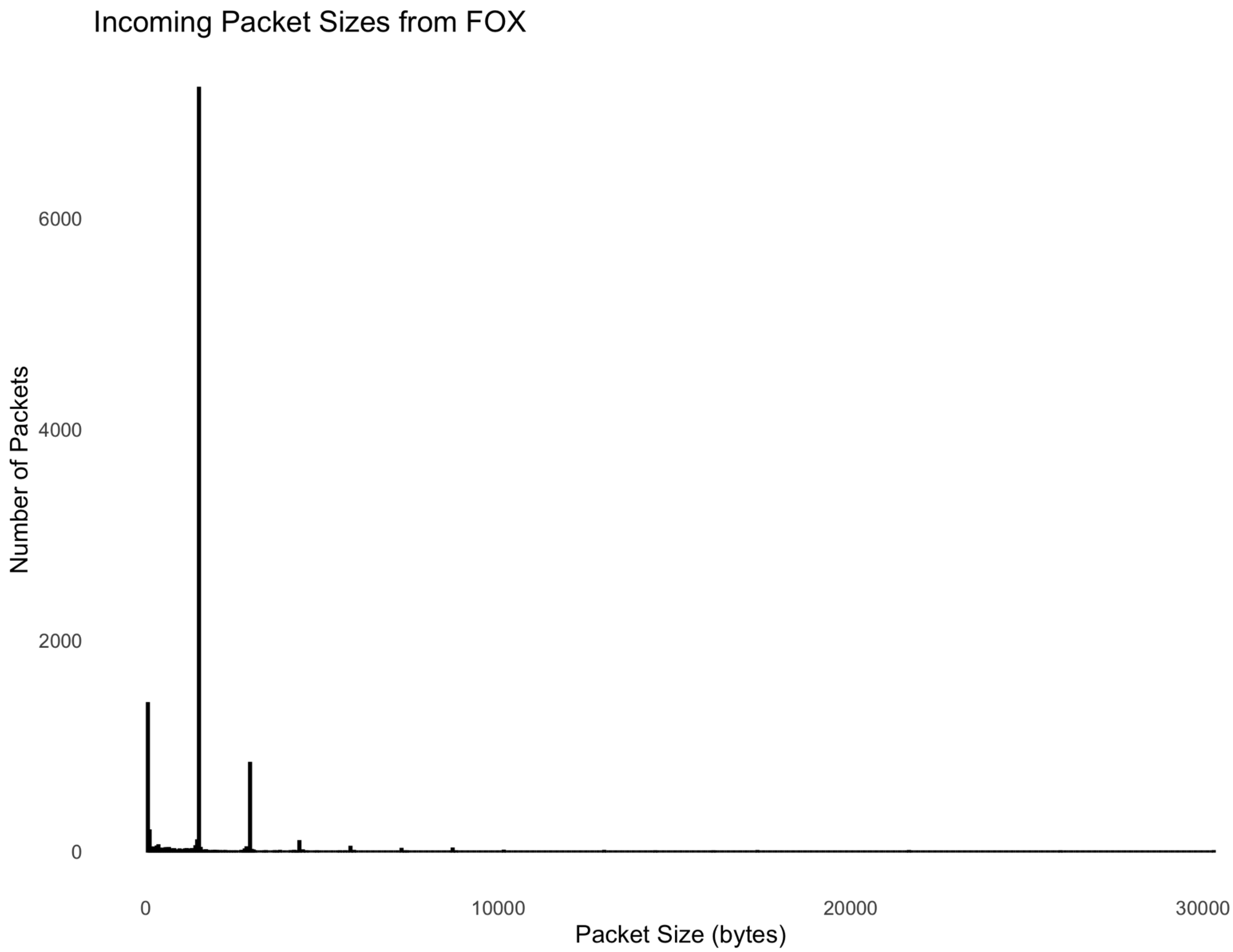
Outgoing Packet Sizes to ALEAE
Number of Packets
90
60
30
0
0
500
1000
1500
2000
Packet Size (bytes)

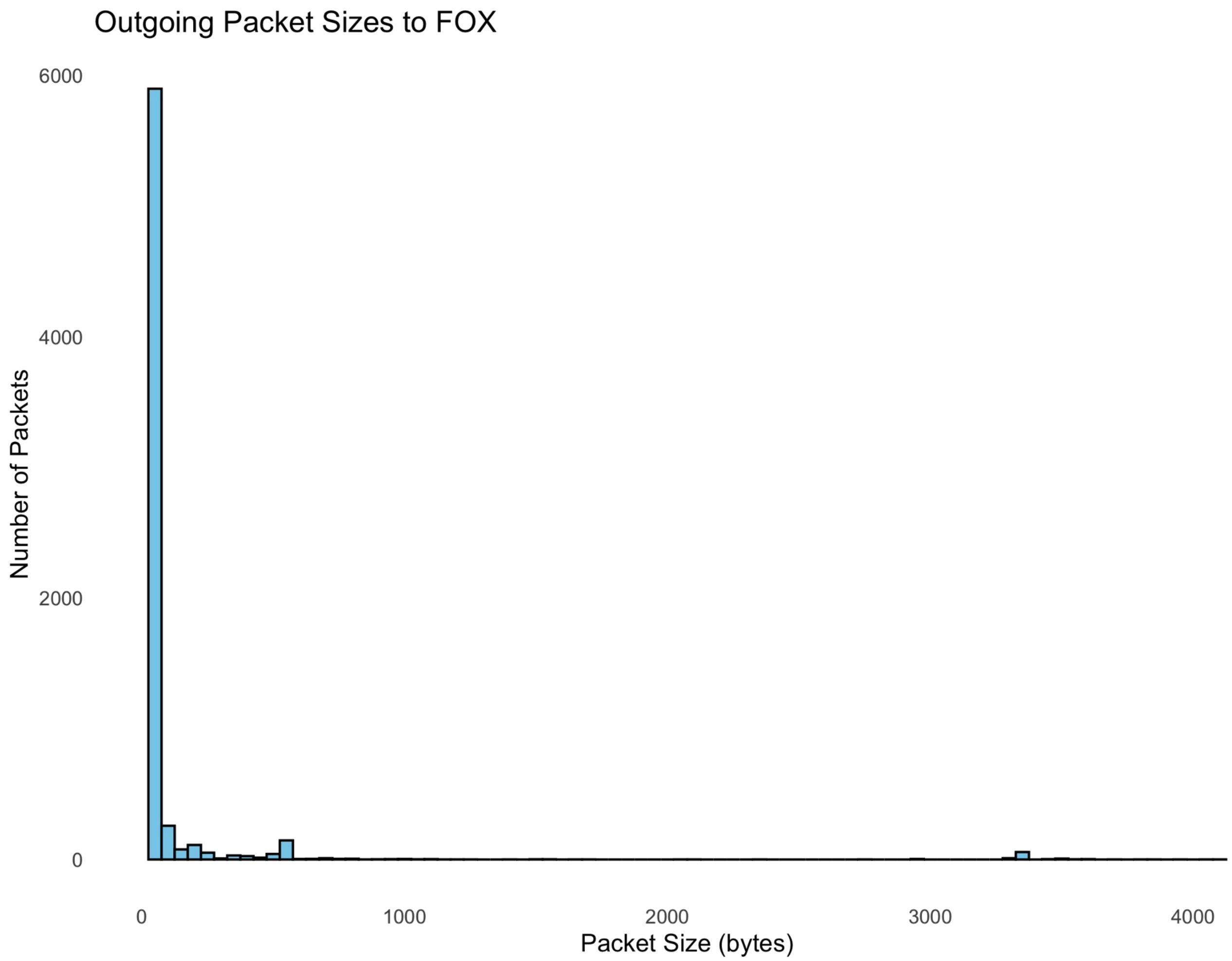

Incoming Packet Sizes from Amazon
Number of Packets
400
200
0
0
1000
2000
3000
Packet Size (bytes)


Outgoing Packet Sizes to Amazon
Number of Packets
500
400
300
200
100
0
0
5000
10000
15000
20000
Packet Size (bytes)


Incoming Packet Sizes from CNN
Number of Packets
0
100
200
300
Packet Size (bytes)
0
1000
2000
3000
4000


Outgoing Packet Sizes to CNN
400
300
200
100
0
Number of Packets
0
2000
4000
6000
Packet Size (bytes)


Incoming Packet Sizes from FOX
Number of Packets
6000
4000
2000
0
0
10000
20000
30000
Packet Size (bytes)



4. Isolate the packets in your capture that came from the ping and traceroute conducted in Part 1, Step 3 and answer the following:

a. What is the round trip time from the Google server you connected with and you? Support your answer.

Looking at the ICMP packets, there are request and response packets with the response times listed around 6ms, which is what the ping command also showed and represents the round trip time.

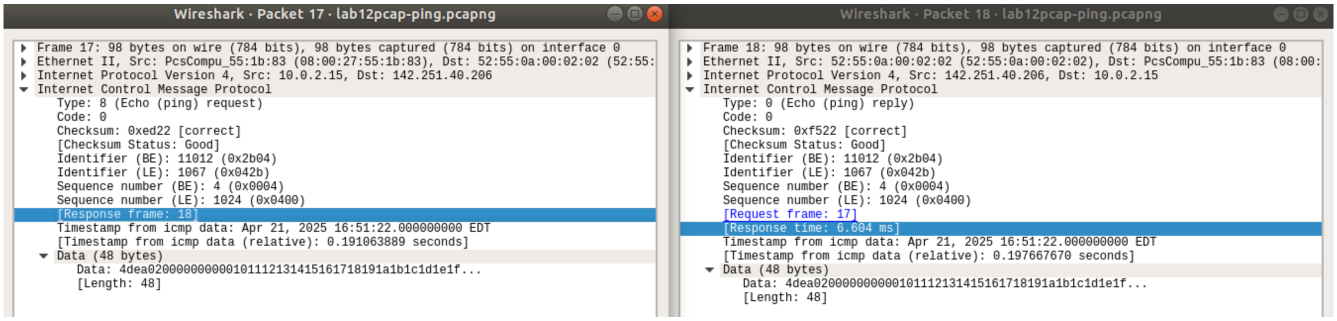


Figure 8: ICMP ping request and response packets and the round trip time

b. How many routers are between you and Google site? Support your answer.

There are an unknown number of routers between my host and Google because the traceroute showed that the maximum number of hops was exceeded at 30. I then ran an online traceroute that showed a successful route.

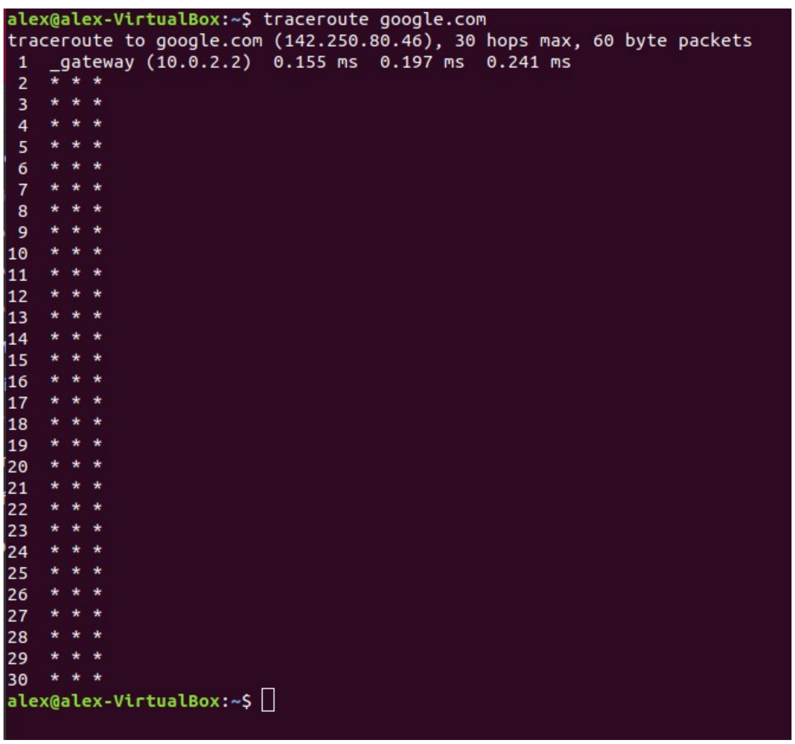


Figure 9: Traceroute between my host and Google, which fails to resolve after 30 hops

```
Start: 2025-04-25T01:44:31+0500
HOST: DNSChecker.org                              Loss%   Snt   Last   Avg  Best  Wrst StDev
  1.|-- ???                                       100.0     3    0.0   0.0   0.0   0.0   0.0
  2.|-- 10.74.132.195                              0.0%     3    0.5   3.0   0.5   6.9   3.5
  3.|-- 138.197.248.252                           33.3%     3    7.1  25.3   7.1  43.5  25.7
  4.|-- 143.244.192.172                            0.0%     3    0.4   0.7   0.4   1.3   0.5
  5.|-- 143.244.225.96                             0.0%     3    0.9   0.9   0.9   0.9   0.0
  6.|-- 143.244.225.25                             0.0%     3    0.7   0.7   0.7   0.8   0.1
  7.|-- 146.190.180.25                             0.0%     3    0.7   1.6   0.7   3.2   1.4
  8.|-- 192.178.106.111                            0.0%     3    1.5   2.1   1.5   3.1   0.9
  9.|-- 108.170.236.89                             0.0%     3    0.8   1.0   0.8   1.2   0.2
 10.|-- lga34s39-in-f14.1e100.net (142.251.40.238)  0.0%     3    0.6   0.7   0.6   0.8   0.1
```

Figure 10: Traceroute on DNSchecker.org that shows hops between the host and Google

c. List the routers between you and the Google site?

I am unable to trace the route between my host and Google, but in Figure 10 DNSchecker shows a total of 10 hops, where the first hop is obscured, then 8 more IP addresses are listed between the DNSchecker host and Google's response server at the end of the route.

Sources

Borman, David; Braden, Bob; Jacobson, Van (September 2014). Scheffenegger, Richard (ed.). TCP Extensions for High Performance. doi:10.17487/RFC7323. RFC 7323.

DNS check Propagation Tool. (n.d.). Retrieved from https://www.dnschecker.org/

Mathis, Matt; Mahdavi, Jamshid; Floyd, Sally; Romanow, Allyn (October 1996). TCP Selective Acknowledgment Options. doi:10.17487/RFC2018. RFC 2018.