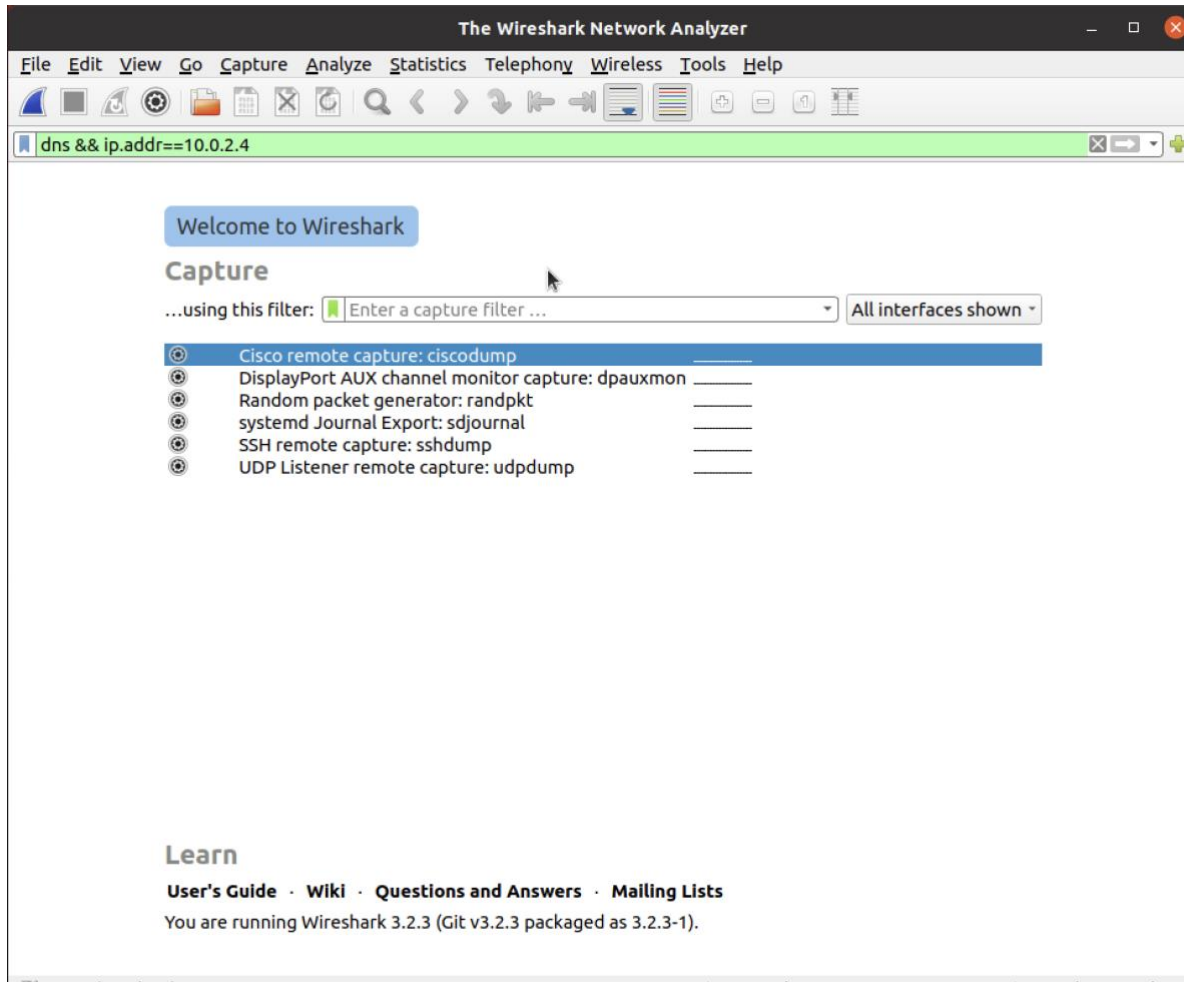


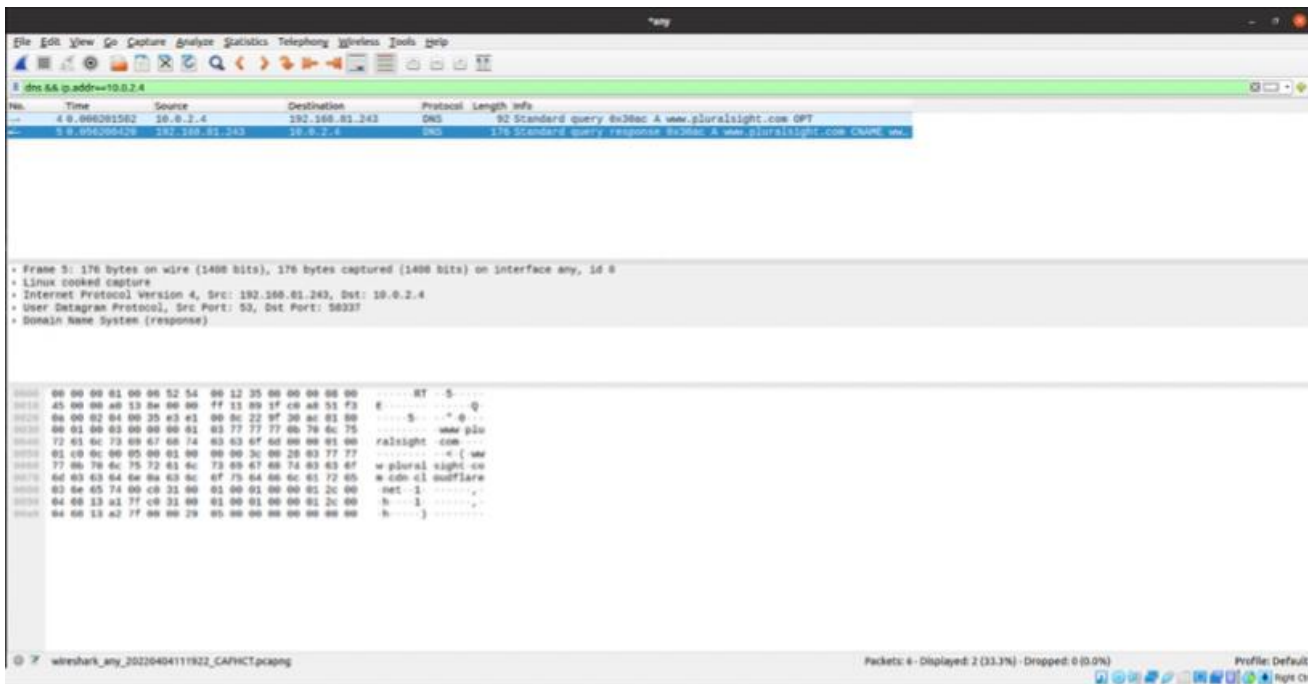
# Lab 5 – Understanding Transport and Network Layer using Wireshark

Name: Vanshika Goel	SRN: PES1UG20CS484	Section: H	Roll No: 40
---------------------	--------------------	------------	-------------

## Step 1: UDP and DNS

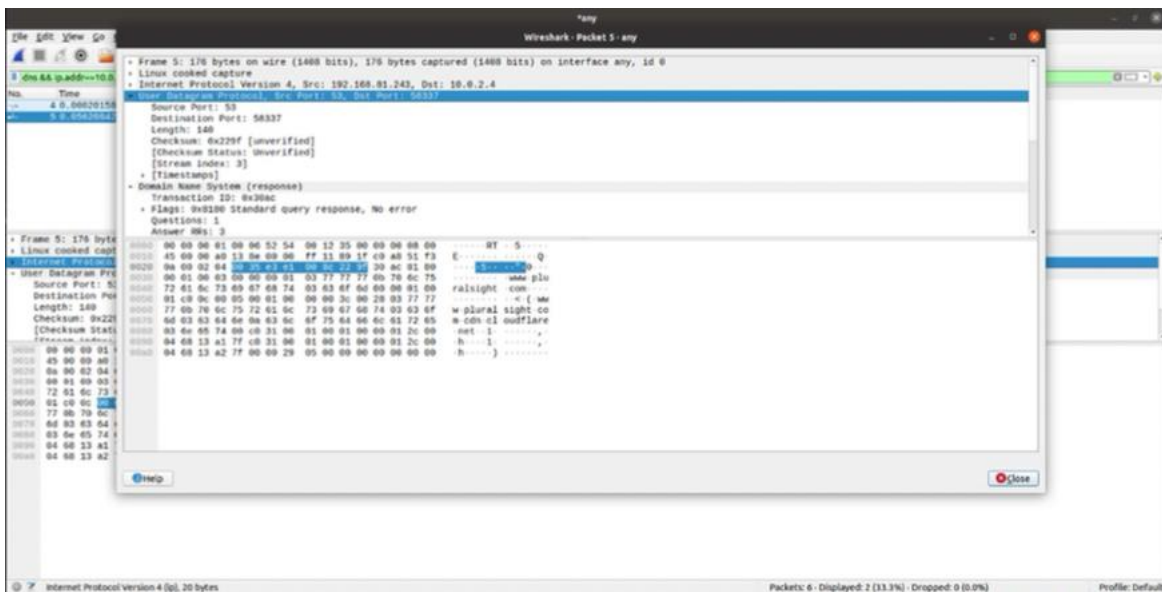


```
vans@vans: ~  
;  
;<<>> DiG 9.16.1-Ubuntu <<>> www.pluralsight.com  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33181  
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1  
;  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:; udp: 65494  
;; QUESTION SECTION:  
;www.pluralsight.com.      IN      A  
;  
;; ANSWER SECTION:  
www.pluralsight.com.      70      IN      CNAME   www.pluralsight.com.cdn.cloudflare.net.  
www.pluralsight.com.cdn.cloudflare.net. 376 IN A 104.19.161.127  
www.pluralsight.com.cdn.cloudflare.net. 376 IN A 104.19.162.127  
;  
;; Query time: 52 msec  
;; SERVER: 127.0.0.53#53(127.0.0.53)  
;; WHEN: Wed May 04 11:29:05 UTC 2022  
;; MSG SIZE rcvd: 132  
vans@vans:~$
```



Q) Before you look at the packets in Wireshark, think for a minute about what you expect to see as the UDP segment headers. What can you reasonably predict, and what could you figure out if you had some time and a calculator handy? Use your knowledge of UDP to inform your predictions.

A) Checksum can be calculated.



Q) Were your predictions correct?

A) Yes, They were correct.

Q) Continue to examine the DNS request packet. Which fields does the UDP checksum cover? Wireshark probably shows the UDP checksum as “Validation Disabled”. Why is that?

A)Checksum: 0x229f [unverified]

Yes, the reason is that Wireshark is very often used to capture the network frames of the same PC that is running Wireshark. This usually results in the checksums of outgoing frames being incorrect since they are only calculated for transmission by the network card after they were already recorded by Wireshark. To avoid constant "checksum error" messages it was decided to have the checksum validation disabled by default.

## Step 2: TCP

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp && ip.addr== 10.0.2.4

No.	Time	Source	Destination	Protocol	Length	Info
13	2.70351803	10.0.2.4	128.2.131.88	TCP	76	49280 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T...
20	2.747349384	10.0.2.4	128.2.131.88	TCP	76	49280 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T...
21	3.026563761	10.0.2.4	128.2.131.88	TCP	76	49280 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T...
22	3.037048843	128.2.131.88	10.0.2.4	TCP	62	80 → 49280 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460
23	3.037090369	10.0.2.4	128.2.131.88	TCP	56	49280 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
24	3.037651891	10.0.2.4	128.2.131.88	HTTP	415	GET /lab2/lab2a.html HTTP/1.1
25	3.045685042	128.2.131.88	10.0.2.4	TCP	62	80 → 49280 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460
26	3.045719938	10.0.2.4	128.2.131.88	TCP	56	49280 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
27	3.086466804	128.2.131.88	10.0.2.4	TCP	62	80 → 49280 [ACK] Seq=1 Ack=360 Win=32400 Len=0
28	3.341349079	128.2.131.88	10.0.2.4	TCP	62	80 → 49280 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460
29	3.341349550	128.2.131.88	10.0.2.4	HTTP	1500	HTTP/1.1 200 OK (text/html)
30	3.341405756	10.0.2.4	128.2.131.88	TCP	56	49280 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
31	3.341434535	10.0.2.4	128.2.131.88	TCP	56	49280 → 80 [ACK] Seq=360 Ack=1445 Win=63536 Len=0
36	3.409558489	10.0.2.4	34.120.208.123	TLSv1.2	201	Application Data
37	3.409598980	10.0.2.4	34.120.208.123	TLSv1.2	533	Application Data
38	3.410292991	34.120.208.123	10.0.2.4	TCP	62	443 → 52476 [ACK] Seq=1 Ack=623 Win=31445 Len=0
43	3.416008370	10.0.2.4	34.120.208.123	TLSv1.2	177	Application Data
44	3.416040448	10.0.2.4	34.120.208.123	TLSv1.2	434	Application Data
45	3.416059842	34.120.208.123	10.0.2.4	TCP	62	443 → 52476 [ACK] Seq=1 Ack=1122 Win=32768 Len=0
50	3.419247894	10.0.2.4	34.120.208.123	TLSv1.2	176	Application Data
51	3.419278812	10.0.2.4	34.120.208.123	TLSv1.2	595	Application Data
52	3.419658078	34.120.208.123	10.0.2.4	TCP	62	443 → 52476 [ACK] Seq=1 Ack=1781 Win=32109 Len=0
57	3.507837298	10.0.2.4	128.30.52.100	TCP	76	43116 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T...
64	3.678237133	34.120.208.123	10.0.2.4	TLSv1.2	140	Application Data
65	3.678255311	10.0.2.4	34.120.208.123	TCP	56	52476 → 443 [ACK] Seq=1781 Ack=85 Win=62890 Len=0
66	3.678539582	10.0.2.4	34.120.208.123	TLSv1.2	95	Application Data
67	3.707241820	34.120.208.123	10.0.2.4	TLSv1.2	140	Application Data
68	3.707265339	10.0.2.4	34.120.208.123	TCP	56	52476 → 443 [ACK] Seq=1820 Ack=169 Win=62890 Len=0

Frame 13: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0

Linux cooked capture

Internet Protocol Version 4, Src: 10.0.2.4, Dst: 128.2.131.88

Transmission Control Protocol, Src Port: 49280, Dst Port: 80, Seq: 0, Len: 0

0000 00 04 00 01 00 06 08 00 27 1d 2d 3d 00 00 00 00 .....-d-....  
0010 45 00 00 3c 29 b9 40 00 40 06 01 a1 0a 00 02 01 E-4) @ 0-....  
0020 80 02 83 58 c0 80 00 50 83 9c 5b fa 00 00 00 00 -X- P -[.....  
0030 a0 02 fa f0 8f 0d 00 00 02 04 05 b4 04 02 08 0a .....  
0040 0f 14 83 62 00 00 00 00 01 03 03 07 o-b-.....

Q) What is the IP address and TCP port number used by your computer (client) to transfer the file? What is the IP address of the server? On what port number is it sending and receiving TCP segments for this transfer of the file?

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp && ip.addr== 10.0.2.4

Frame 93: 2976 bytes on wire (23808 bits), 2976 bytes captured (23808 bits) on interface any, id 0

Linux cooked capture

Internet Protocol Version 4, Src: 10.0.2.4, Dst: 128.2.131.88

Transmission Control Protocol, Src Port: 49280, Dst Port: 80, Seq: 1, Ack: 1, Len: 2976

Source Port: 49280

Destination Port: 80

[Stream index: 5]

Sequence number: 1 (relative sequence number)

Sequence number (raw): 1666000352

[Next sequence number: 2921 (relative sequence number)]

Acknowledgment number: 1 (relative ack number)

Acknowledgment number (raw): 170502

0101 ... = Header Length: 20 bytes (5)

Flags: 0x018 (PSH, ACK)

Window size value: 64240

0000 00 04 00 01 00 06 08 00 27 1d 2d 3d 00 00 00 00 .....-d-....  
0010 45 00 00 3c 29 b9 40 00 40 06 01 a1 0a 00 02 01 E-4) @ 0-....  
0020 80 02 83 58 c0 80 00 50 83 9c 5b fa 00 00 00 00 -X- P -[.....  
0030 a0 02 fa f0 8f 0d 00 00 02 04 05 b4 04 02 08 0a .....  
0040 0f 14 83 62 00 00 00 00 01 03 03 07 o-b-.....

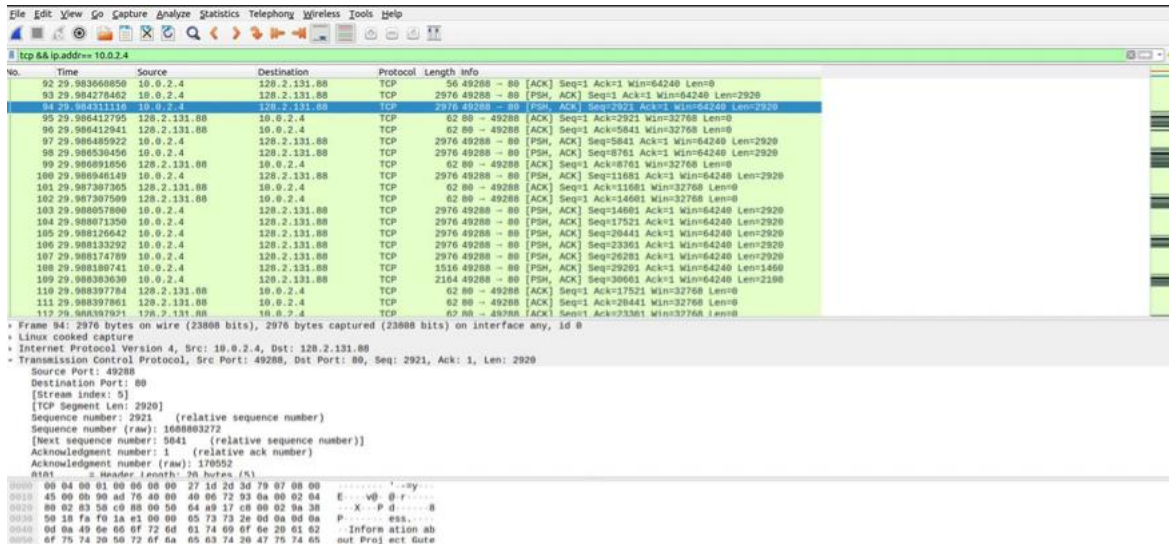
Source:

IP: 10.0.2.4 Port: 49288

Destination:

IP:

Port: 80



## Step 2b: TCP Basics

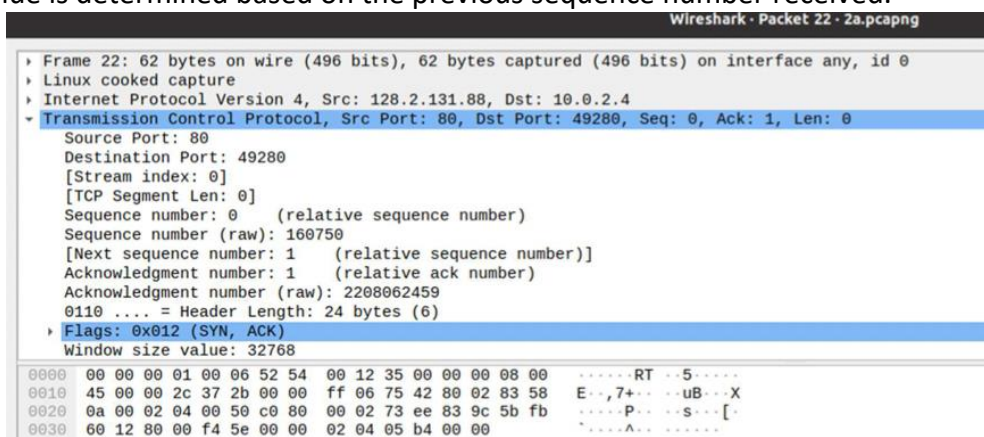
Answer the following questions for the TCP segments:

Q) What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection? What element of the segment identifies it as a SYN segment? Wireshark uses relative sequence numbers by default. Can you obtain absolute sequence numbers instead? How? You can use relative sequence numbers to answer the remaining questions.

A) The sequence number is 0 to initiate the TCP connection.

Q) What is the sequence number of the SYNACK segment sent by the server in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did the server determine that value? What element in the segment identifies it as a SYNACK segment?

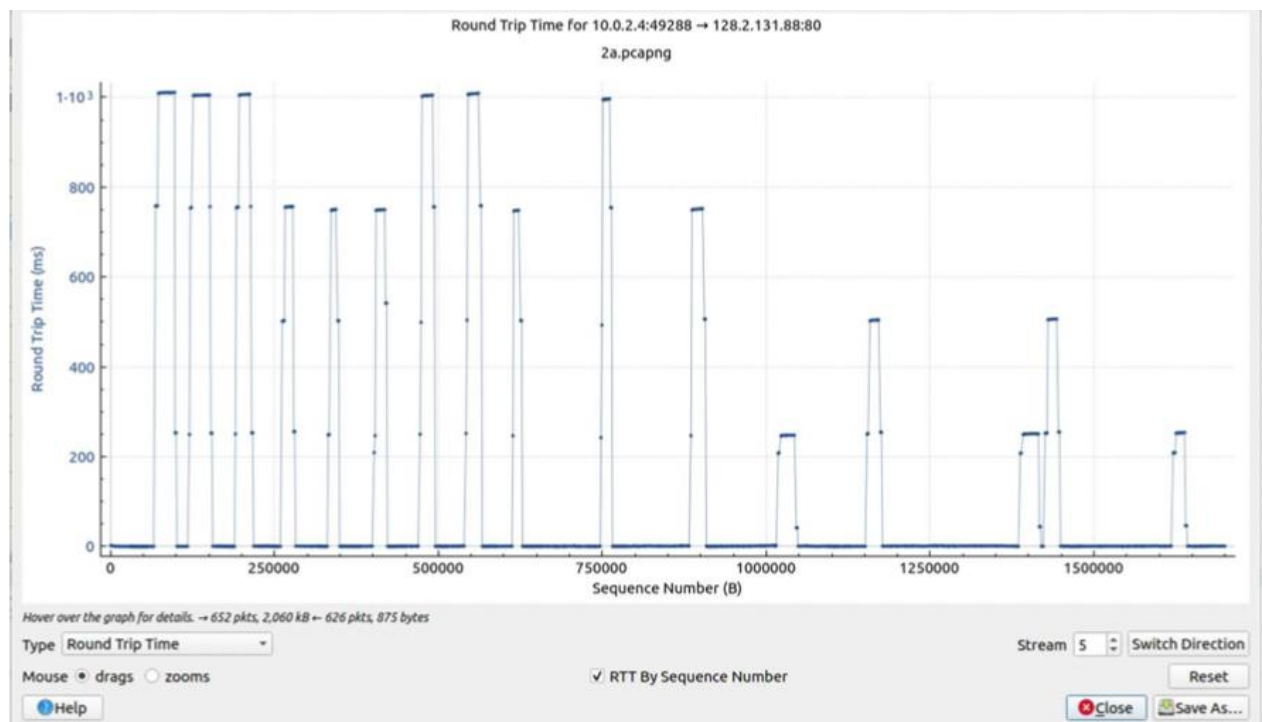
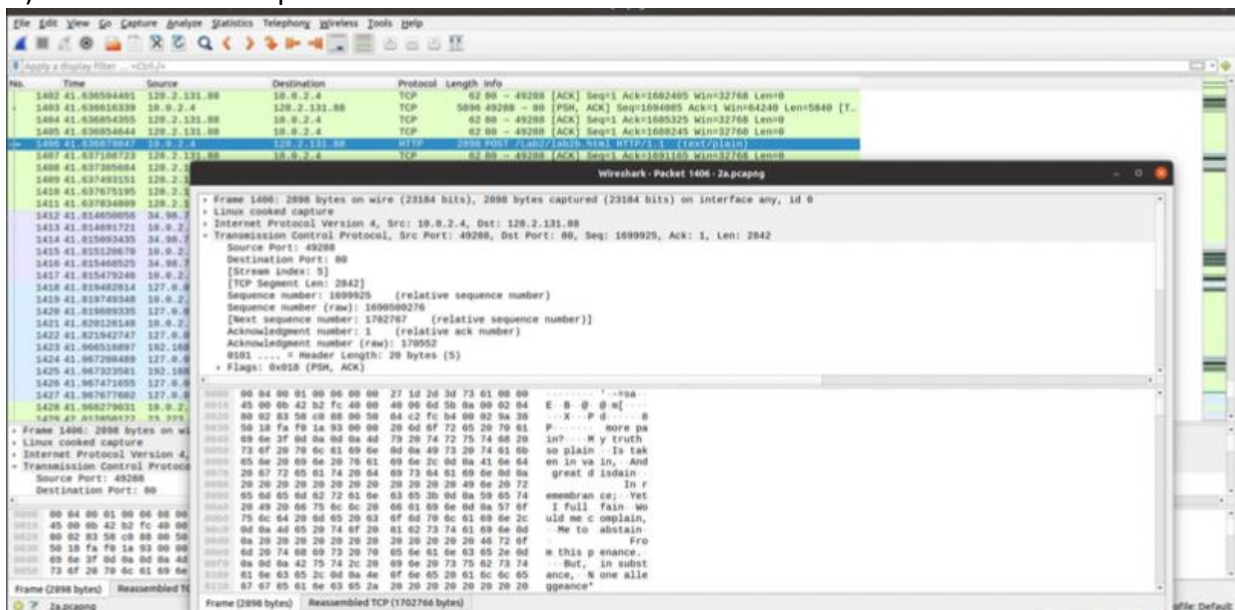
A) The sequence number is still 0. The value of the acknowledgement field in SYNACK segment is 1. This value is determined based on the previous sequence number received.



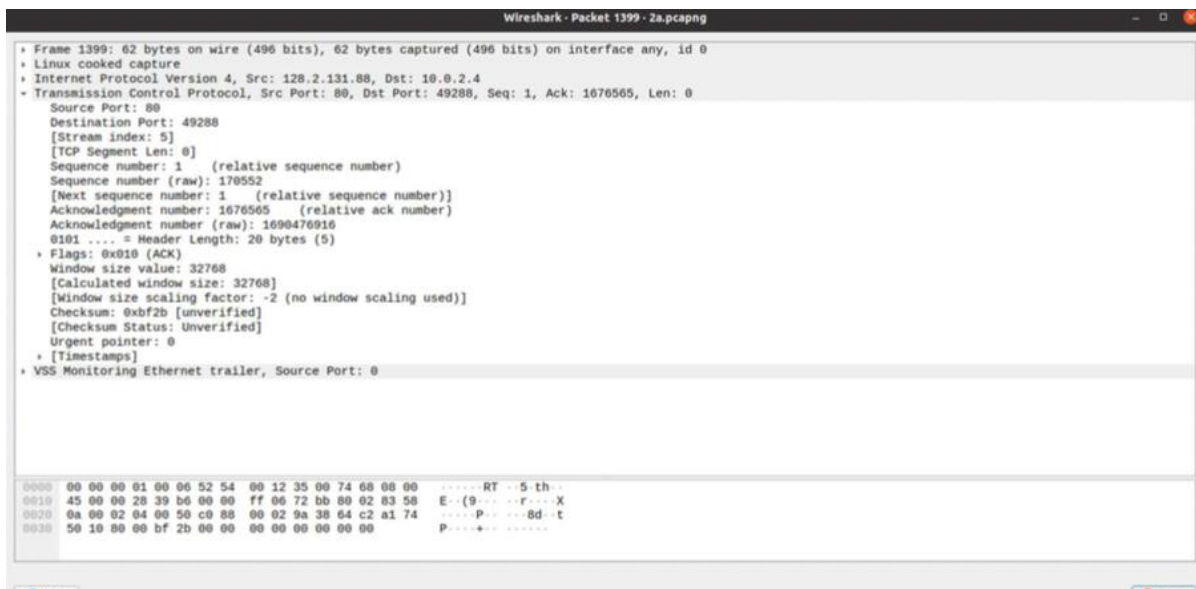


Q) What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.

A) 1699925 is the sequence number.



Q) What is the minimum amount of available buffer space advertised at the receiver for the entire trace? Does the lack of receiver buffer space ever throttle the sender?



A) Buffer space available: 32768 Yes, it does throttle the sender.

Q) Are there any retransmitted segments? What did you check for (in the trace) to answer this question?

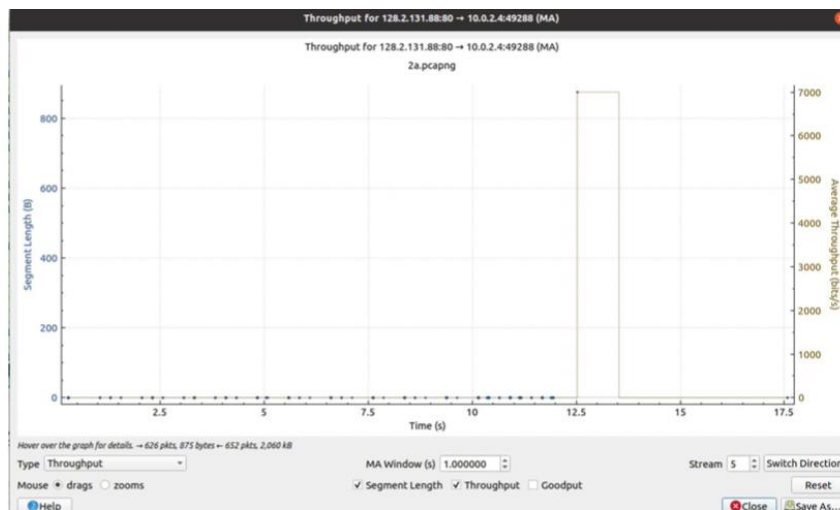
No.	Time	Source	Destination	Protocol	Length	Info
467	34.535342863	10.0.2.4	128.2.131.88	TCP	7356	49288 → 80 [PSH, ACK] Seq=410745 Ack=1 Win=64240 Len=7300 [TC...
468	34.535496450	128.2.131.88	10.0.2.4	TCP	62	80 → 49288 [ACK] Seq=1 Ack=401985 Win=31308 Len=0
469	34.535512633	10.0.2.4	128.2.131.88	TCP	1516	49288 → 80 [ACK] Seq=418045 Ack=1 Win=64240 Len=1460 [TCP seg...
470	34.743227088	10.0.2.4	128.2.131.88	TCP	1516	[TCP Out-Of-Order] 49288 → 80 [ACK] Seq=401985 Ack=1 Win=6424...
471	34.743889445	128.2.131.88	10.0.2.4	TCP	62	80 → 49288 [ACK] Seq=1 Ack=403445 Win=31308 Len=0
472	34.743919038	10.0.2.4	128.2.131.88	TCP	1516	49288 → 80 [ACK] Seq=419505 Ack=1 Win=64240 Len=1460 [TCP seg...
473	34.743942658	10.0.2.4	128.2.131.88	TCP	1516	49288 → 80 [ACK] Seq=420965 Ack=1 Win=64240 Len=1460 [TCP seg...
474	34.744209448	128.2.131.88	10.0.2.4	TCP	62	[TCP Dup ACK 471#1] 80 → 49288 [ACK] Seq=1 Ack=403445 Win=313...
475	34.744269715	128.2.131.88	10.0.2.4	TCP	62	[TCP Dup ACK 471#2] 80 → 49288 [ACK] Seq=1 Ack=403445 Win=313...
476	34.744294316	10.0.2.4	128.2.131.88	TCP	1516	[TCP Fast Retransmission] 49288 → 80 [ACK] Seq=403445 Ack=1 W...
477	34.744317157	10.0.2.4	128.2.131.88	TCP	1516	[TCP Out-Of-Order] 49288 → 80 [ACK] Seq=404905 Ack=1 Win=6424...
478	34.781685671	128.2.131.88	10.0.2.4	TCP	62	80 → 49288 [ACK] Seq=1 Ack=404905 Win=31308 Len=0
479	35.283168569	10.0.2.4	128.2.131.88	TCP	1516	[TCP Retransmission] 49288 → 80 [ACK] Seq=404905 Ack=1 Win=64...
480	35.283460511	128.2.131.88	10.0.2.4	TCP	62	80 → 49288 [ACK] Seq=1 Ack=406365 Win=32768 Len=0
481	35.283537853	10.0.2.4	128.2.131.88	TCP	2976	[TCP Retransmission] 49288 → 80 [PSH, ACK] Seq=412205 Ack=1 W...
482	35.284050239	128.2.131.88	10.0.2.4	TCP	62	80 → 49288 [ACK] Seq=1 Ack=409285 Win=32768 Len=0
483	35.284069738	10.0.2.4	128.2.131.88	TCP	1516	[TCP Retransmission] 49288 → 80 [ACK] Seq=409285 Ack=1 Win=64...
484	35.284089838	10.0.2.4	128.2.131.88	TCP	1516	[TCP Retransmission] 49288 → 80 [ACK] Seq=410745 Ack=1 Win=64...
485	35.284446663	128.2.131.88	10.0.2.4	TCP	62	80 → 49288 [ACK] Seq=1 Ack=412205 Win=32768 Len=0
486	35.284482766	10.0.2.4	128.2.131.88	TCP	2976	[TCP Retransmission] 49288 → 80 [PSH, ACK] Seq=412205 Ack=1 W...
487	35.284924288	128.2.131.88	10.0.2.4	TCP	62	80 → 49288 [ACK] Seq=1 Ack=415125 Win=32768 Len=0
488	35.284941858	10.0.2.4	128.2.131.88	TCP	2976	[TCP Retransmission] 49288 → 80 [PSH, ACK] Seq=415125 Ack=1 W...
489	35.285366960	128.2.131.88	10.0.2.4	TCP	62	80 → 49288 [ACK] Seq=1 Ack=418045 Win=32768 Len=0
490	35.285381562	10.0.2.4	128.2.131.88	TCP	1516	[TCP Retransmission] 49288 → 80 [ACK] Seq=418045 Ack=1 Win=64...
491	35.285397462	10.0.2.4	128.2.131.88	TCP	1516	[TCP Retransmission] 49288 → 80 [ACK] Seq=419505 Ack=1 Win=64...
492	35.285791493	128.2.131.88	10.0.2.4	TCP	62	80 → 49288 [ACK] Seq=1 Ack=422425 Win=32768 Len=0
493	35.285808725	10.0.2.4	128.2.131.88	TCP	1516	49288 → 80 [ACK] Seq=422425 Ack=1 Win=64240 Len=1460 [TCP seg...
494	35.285834027	10.0.2.4	128.2.131.88	TCP	1516	49288 → 80 [ACK] Seq=422425 Ack=1 Win=64240 Len=1460 [TCP seg...

There is a Fast TCP re-transmission and TCP Dup ACK.

Q) How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is delayed ACKing segments? Explain how or why not.

A) There is a TCP payload of 4380 bytes and window size is 64240. The 'keep-alive' status helps identify delayed ACKing segments.

Q) What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.



## Step 2c: Statistics

Q)What is the most common TCP packet length range? What is the second most common TCP packet length range? Why is the ratio of TCP packets of length < 40 bytes equal to zero? Describe what actions you took to get answers to these questions from Wireshark.

A) Average packet length is 1467.09

Q)What average throughput did you use in Mbps? How many packets were captured in the packet capture session? How many bytes in total? Explain your methods.

Details

File

Name: /home/vk27/Desktop/PES1UG20CSS00/2a.pcapng  
Length: 2,239 kB  
Hash (SHA256): aad3e6b5dc85e959a9c18d0d56908b3e9b91a3f2730ddefc752fa55d64df5a92  
Hash (RIPEMD160): 5f35b19caf1a34f2bcc29461a8eb320120e91490  
Hash (SHA1): 5fae27d1bb444daf6bda4b84facf37d2d15baee9  
Format: Wireshark/... - pcapng  
Encapsulation: Linux cooked-mode capture

Time

First packet: 2022-04-04 11:42:04  
Last packet: 2022-04-04 11:42:56  
Elapsed: 00:00:52

Capture

Hardware: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz (with SSE4.2)  
OS: Linux 5.13.0-30-generic  
Application: Dumpcap (Wireshark) 3.2.3 (Git v3.2.3 packaged as 3.2.3-1)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
any	0 (0.0%)	none	Linux cooked-mode capture	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	1493	1419 (95.0%)	—
Time span, s	52.228	49.489	—
Average pps	28.6	28.7	—
Average packet size, B	1467	1536	—
Bytes	2190372	2180037 (99.5%)	0
Average bytes/s	41 k	44 k	—
Average bits/s	335 k	352 k	—

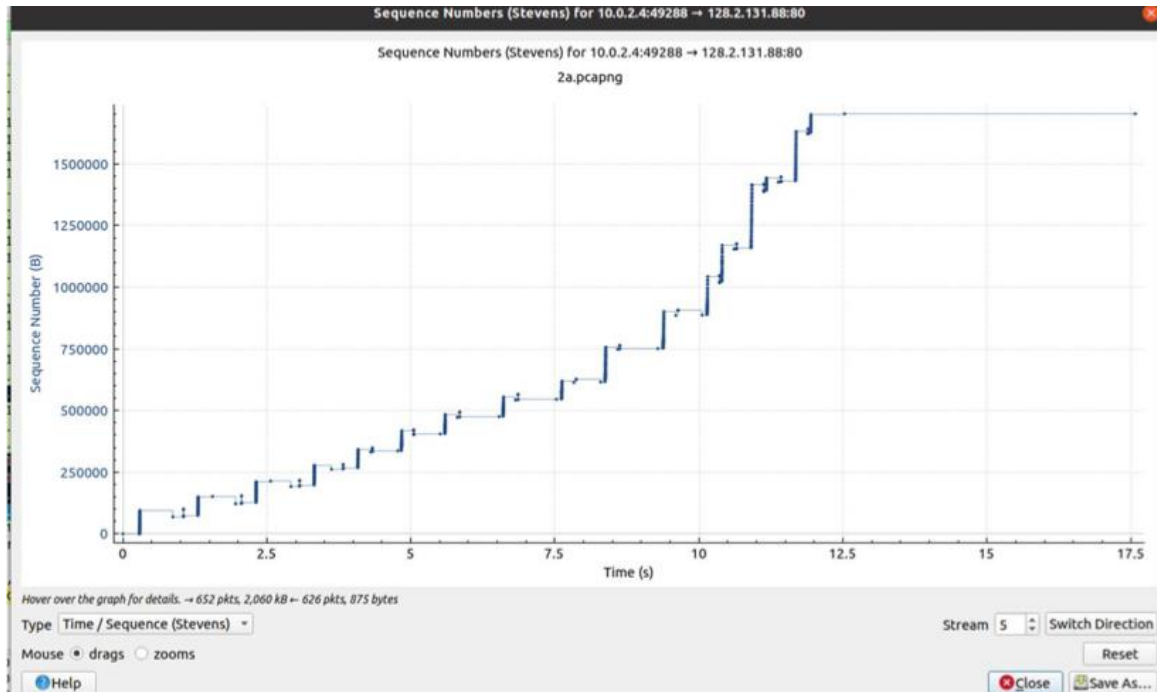
Capture file comments

Help Refresh Copy To Clipboard Close Save Comments

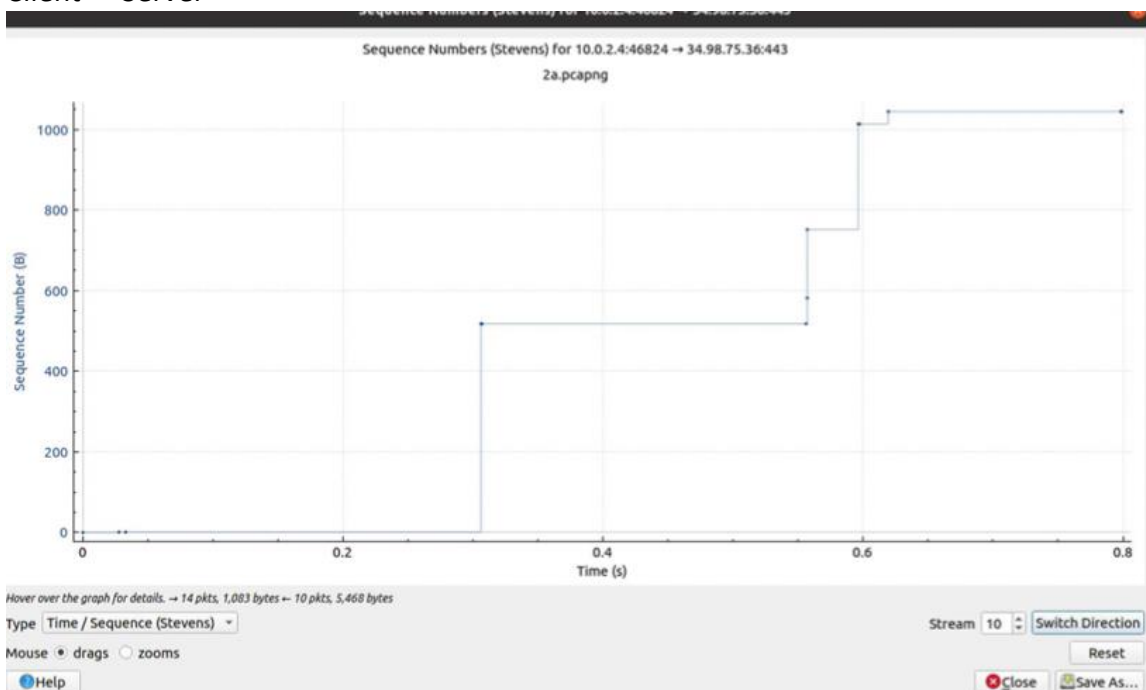
Q) A conversation represents a traffic between two hosts. With which remote host did your local host converse the most (in bytes)? How many packets were sent from your host? How many packets were sent from the remote host?

We converse the most with port 443 of server with IP: 128.2.131.88 (1309 packets = 2140K bytes)

### Step 3: Congestion Control



Client -> Server



Server -> Client



## Step 4: The Network Layer

TTL is set to 64 by OS (default).

## Step 5: ICMP

The screenshot shows a terminal window with the following commands and output:

```
vans@vans:~$ traceroute www.cmu.jp
Command 'traceroute' not found, but can be installed with:
sudo apt install inetutils-traceroute # version 2:1.9.4-11ubuntu0.1, or
sudo apt install traceroute          # version 1:2.1.0-2
vans@vans:~$
```

Below the terminal is a Wireshark packet capture window. The top pane shows a list of packets. The bottom pane shows the details of the selected packet (Frame 2: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface any, id 0).

No.	Time	Source	Destination	Protocol	Length	Info
26	1.125805642	10.0.2.4	122.17.163.205	UDP	76	58328 → 33446 Len=32
27	1.125978079	10.0.2.4	122.17.163.205	UDP	76	35190 → 33447 Len=32
28	1.126034294	10.0.2.4	122.17.163.205	UDP	76	37649 → 33448 Len=32
29	1.126068521	10.0.2.4	122.17.163.205	UDP	76	43381 → 33449 Len=32
31	1.126095198	10.0.2.4	192.168.1.1	DNS	94	Standard query 0xfe1a PTR 1.2.0.10.in-addr.arpa OPT
32	2.559951493	192.168.1.1	10.0.2.4	DNS	171	Standard query response 0xfe1a No such name PTR 1.2.0.10.in-addr.arpa
33	2.560170246	10.0.2.4	192.168.1.1	DNS	83	Standard query 0xfe1a PTR 1.2.0.10.in-addr.arpa
34	2.948918106	192.168.1.1	10.0.2.4	DNS	160	Standard query response 0xfe1a No such name PTR 1.2.0.10.in-addr.arpa
36	2.950077979	10.0.2.4	122.17.163.205	UDP	76	54161 → 33450 Len=32
37	2.950135794	10.0.2.4	122.17.163.205	UDP	76	50637 → 33451 Len=32
38	2.950255596	10.0.2.4	122.17.163.205	UDP	76	42912 → 33452 Len=32
39	6.128911570	10.0.2.4	122.17.163.205	UDP	76	58770 → 33453 Len=32
40	6.128960889	10.0.2.4	122.17.163.205	UDP	76	59052 → 33454 Len=32
41	6.129126865	10.0.2.4	122.17.163.205	UDP	76	46658 → 33455 Len=32
42	6.129170889	10.0.2.4	122.17.163.205	UDP	76	33164 → 33456 Len=32
43	6.129214420	10.0.2.4	122.17.163.205	UDP	76	49093 → 33457 Len=32
44	6.129247778	10.0.2.4	122.17.163.205	UDP	76	47787 → 33458 Len=32
45	6.129286272	10.0.2.4	122.17.163.205	UDP	76	53961 → 33459 Len=32
46	6.129505686	10.0.2.4	122.17.163.205	UDP	76	37223 → 33460 Len=32
47	6.129546387	10.0.2.4	122.17.163.205	UDP	76	59054 → 33461 Len=32
48	6.129579155	10.0.2.4	122.17.163.205	UDP	76	51228 → 33462 Len=32
49	6.129744552	10.0.2.4	122.17.163.205	UDP	76	37788 → 33463 Len=32
50	6.129792647	10.0.2.4	122.17.163.205	UDP	76	47844 → 33464 Len=32
51	6.129827775	10.0.2.4	122.17.163.205	UDP	76	60523 → 33465 Len=32
52	7.952860921	10.0.2.4	122.17.163.205	UDP	76	48868 → 33466 Len=32
53	7.953069729	10.0.2.4	122.17.163.205	UDP	76	49775 → 33467 Len=32
54	7.953181130	10.0.2.4	122.17.163.205	UDP	76	50958 → 33468 Len=32
55	11.130943373	10.0.2.4	122.17.163.205	UDP	76	48008 → 33469 Len=32

Frame 2: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface any, id 0

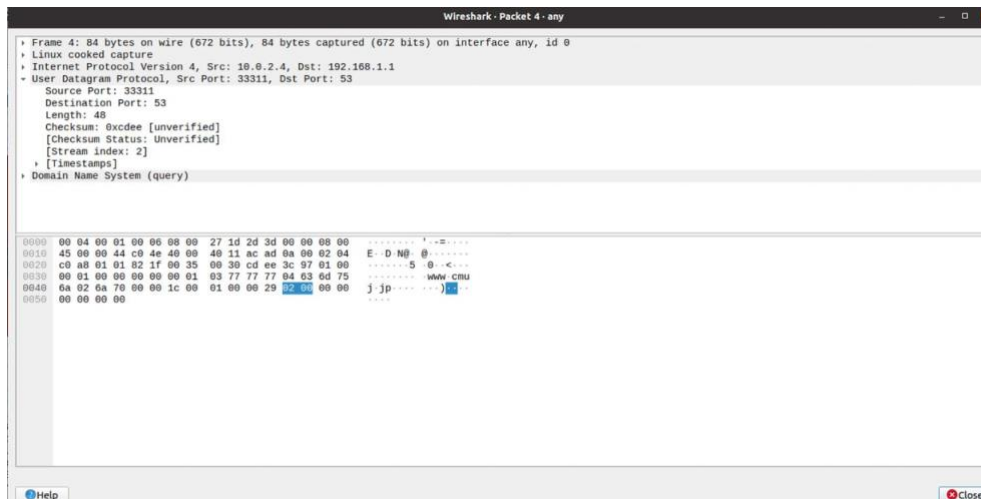
- Linux cooked capture
- Internet Protocol Version 4, Src: 10.0.2.4, Dst: 192.168.1.1
- User Datagram Protocol, Src Port: 52633, Dst Port: 53
- Domain Name System (query)

0000 00 04 00 01 00 06 00 00 27 1d 2d 3d 00 00 00 00 .....E.D.MD. @.....  
0010 45 00 00 44 c0 4d 40 00 40 11 ac ae 0a 00 02 04 .....S 0 1.....  
0020 c0 a8 01 01 cd 99 00 35 00 30 cd ee 09 a1 01 00 .....www.cmu.jp.....  
0030 00 01 00 00 00 00 01 03 77 77 04 03 0d 75 .....  
0040 0a 02 6a 70 00 00 01 00 01 00 00 29 02 00 00 00 .....  
0050 00 00 00 00

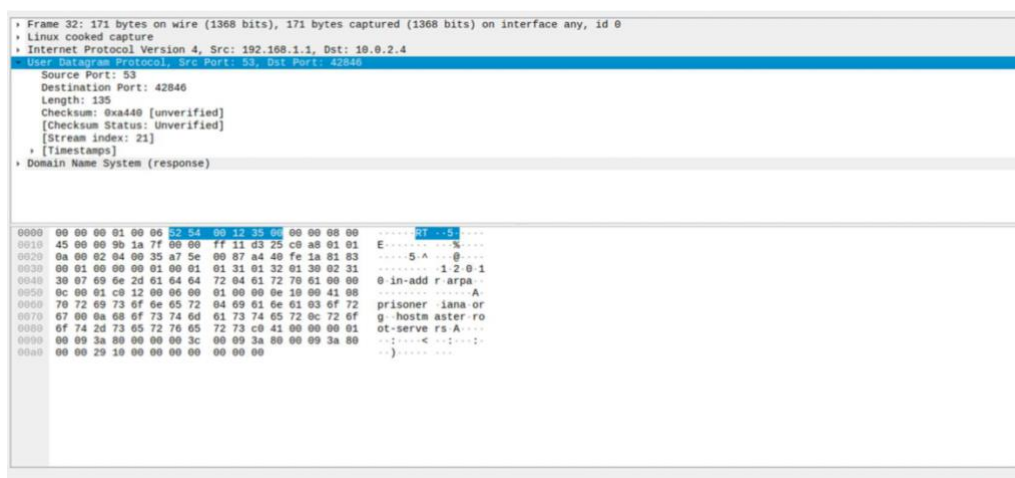
Wireshark: wireshark\_any\_20220404224641\_BuMKYe.pcapng

Q) What are the transmitted segments like? Describe the important features of the segments you observe. In particular, examine the destination port field. What characteristics do you observe about this port number and why would it be chosen so?

A) The destination port number chosen is 53, as it can transmit DNS queries in UDP segment.



Q) What about the return packets? What are the values of the various header fields?



Q) The ICMP packets carry some interesting data. What is it? Can you show the relationship to the sent packets?



R) S) Lab1 asserted that ping operates in a similar fashion to traceroute. Use Wireshark to show the degree to which this is true. What differences and similarities are there between the network traffic of ping versus traceroute?

A) Both 'ping' & 'traceroute' are used to check the network connectivity issues.