Data Communication and Computer Networks (CS536)

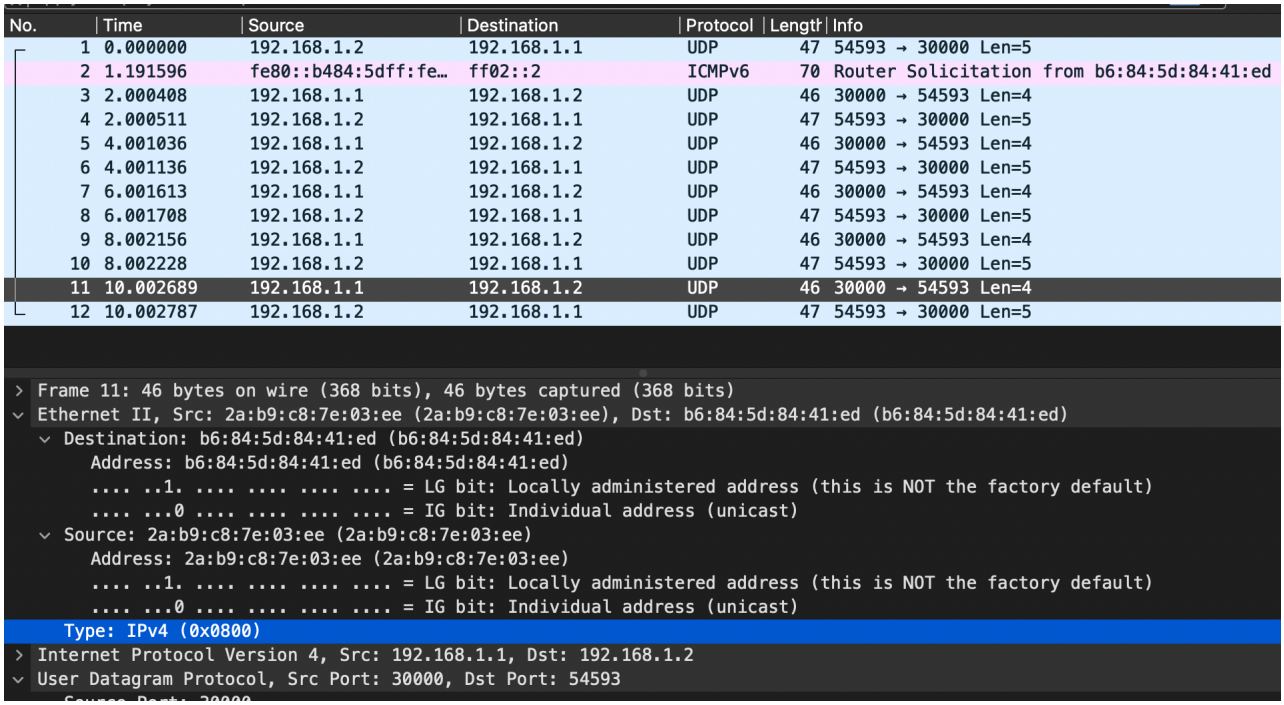
Lab 3

Student Name: Chandrika Mukherjee
Student ID: 0032808289
Email: cmukherj@purdue.edu

Problem 1:

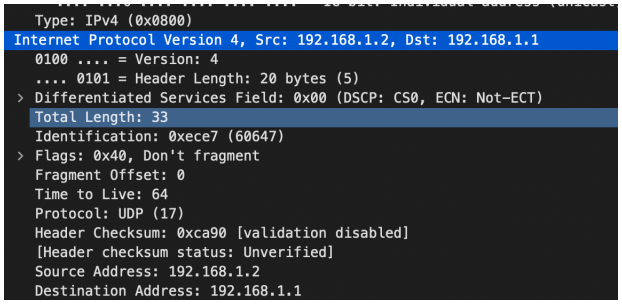
MAC address associated with server (192.168.1.1) - 2a:b9:c8:7e:03:ee (checked using "ifconfig -a")

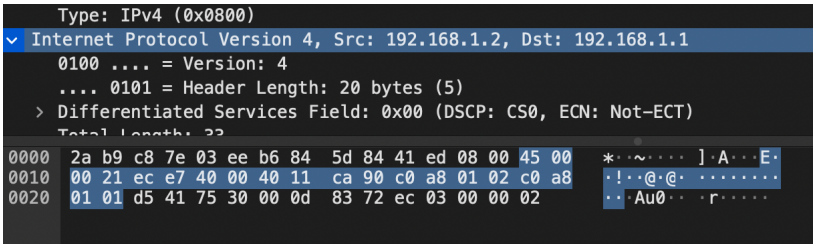
MAC address associated with client (192.168.1.2) - b6:84:5d:84:41:ed (checked using veth "ifconfig -a")



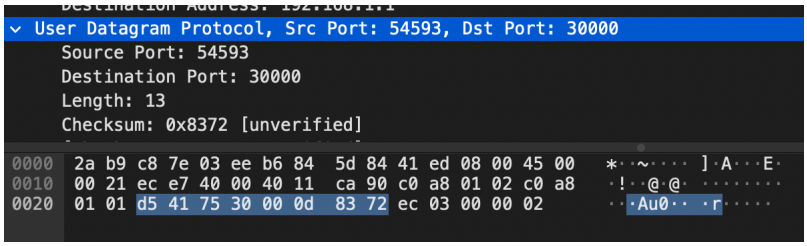
As per the above image, 12 ethernet frames are detected.
From the highlighted Type field, we can detect the payload as IPv4.
Also, The Protocol associated is UDP.
The length of payload for 11th frame in the picture is 46 bytes. For a DIX frame, payload is minimum 46 bytes and maximum 1500 bytes. Therefore, we can conclude that the captured

ethernet frame is DIX frame. For all 12 captured ethernet frames, payload is either 46 or 47 bytes.

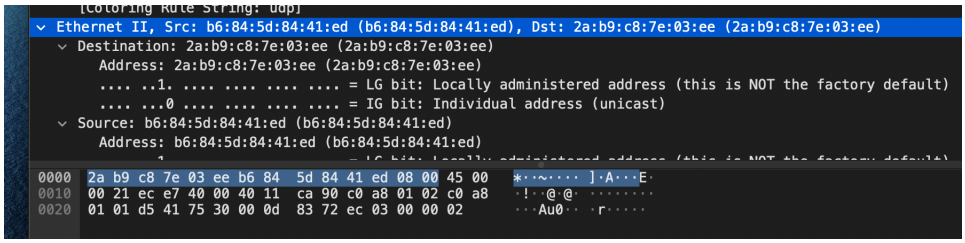




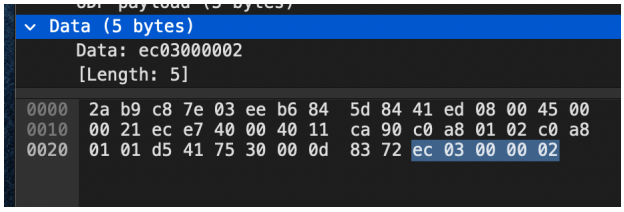
The image in the left side shows that the IPv4 header length is 20 bytes. The image in the right side highlights the associated 20 bytes.

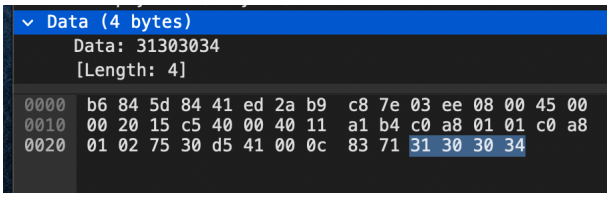


The above image shows the User Datagram Protocol (UDP). The highlighted 8 bytes is the UDP header. First 4 bytes is source port (54593 - client chose this port to communicate with server) and last 4 bytes is destination port (30000 - selected port of server).



Last 8 bytes in IPv4 packet denotes the source and destination addresses. This frame suggests that the communication was done from client to server. So, source MAC address (b6:84:5d:84:41:ed [client's]) and destination MAC address (2a:b9:c8:7e:03:ee [server's]) are correctly captured.





The left image shows the 5 bytes data sent from client to server, the highlighted portion in the IPv4 packets shows the data associated. (The data is 5 bytes because client sends 4 bytes MID and 1 byte control command).

The right image shows 4 bytes data sent from server to client, the highlighted portion in the IPv4 packets shows the data associated. (The data is 4 bytes because server sends 4 bytes MID back to client).
At the left client sends MID and control field together as 5 bytes (ec03000002), 02 is the command field, rest is the MID. At the right server sends back the MID (1004) as (hex - 31303034).

Problem 2:

For the experiment, I have used a file of 10mb.

- With blocksize = 512, throughput calculated between same lab machines is 875810.125000 bytes/ms, completion time is 11.418000 ms.
  With blocksize = 512, throughput calculated between different lab machines is 768639.508 bytes/ms, completion time is 13.01 ms.

- With blocksize = 1024, throughput calculated between same lab machines is 988142.292 bytes/ms, completion time is 10.12 ms.
  With blocksize = 1024, throughput calculated between different lab machines is 890860.986 bytes/ms, completion time is 11.15 ms.

- With blocksize = 2048, throughput calculated between same lab machines is 1042861.625000 bytes/ms , completion time is 9.589000 ms.
  With blocksize = 2048, throughput calculated between different lab machines is is 966744.006 bytes/ms , completion time is 10.344 ms.

- With blocksize = 4096, throughput calculated between same lab machines is 1396843.1250005 bytes/ms , completion time is 7.159000 ms.
  With blocksize = 4096, throughput calculated between different lab machines is 1227445.68 bytes/ms , completion time is 8.147 ms

Observation: If we increase the block-size, number of system calls will be reduced. Therefore, the completion time should decrease, as we are using same file for experiment. So, throughput should increase.

For same lab experiment, while increasing block size from 512 to 1024 to 2048 to 4096, the throughput is increasing and completion time is decreasing. throughput range [875810.125000, 1396843.1250005]bytes/ms.

For different lab experiment, while increasing block size from 512 to 1024 to 2048 to 4096, the throughput is increasing and completion time is decreasing. throughput range [768639.508 , 1227445.68]bytes/ms.

Bonus Problem:

For the experiment, I have used a file of 10 mb.
I obtained the best performance with 4096 bytes as blocksize, therefore, using this for performing experiments in \tmp folder.

- With blocksize = 4096, throughput calculated between same lab machines is 1536098.250000 bytes/ms , completion time is 6.510000 ms.
- With blocksize = 4096, throughput calculated between different lab machines is 1383125.86 bytes/ms , completion time is 7.23 ms

  Observation: Due to caching in \tmp folder, the completion time has decreased from that of normal experiment.