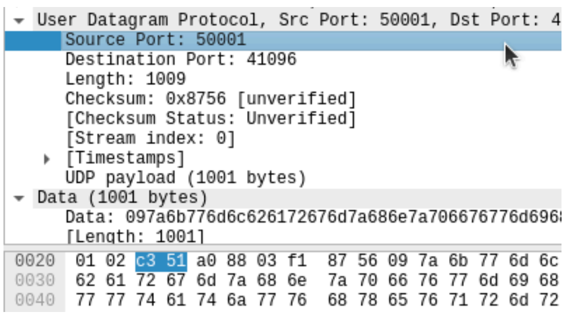
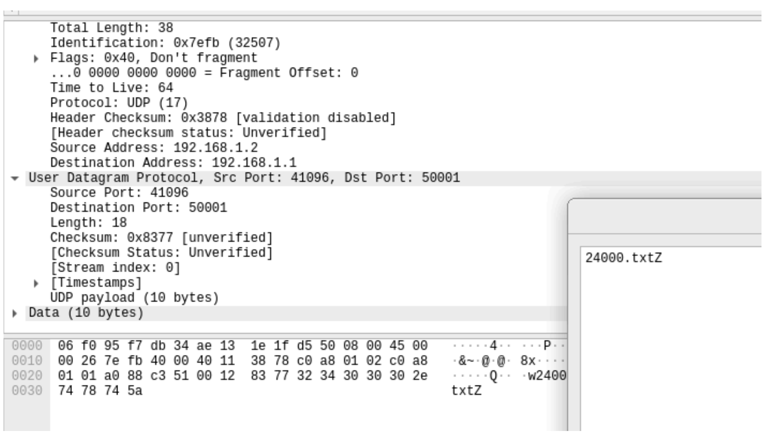
Problem 1:

We created a random text file named 24000.txt to send over 1000 byte packets.
The log file is captured on AMBER05.
The first thing that can be seen about the packets is that all packets are 42 bytes longer than our UDP payloads: 52 bytes instead of 10, 45 instead of 3, and 1043 instead of 1001. This is due to the additional headers added by the Ethernet, Ip, and UDP protocols.
The MAC address for 192.168.1.1 is 06:f0:95:f7:db:34
The MAC address for 192.168.1.2 is ae:13:1e:1f:d5:50
We can see that the ip header is 20 bytes and the ip addresses match the addresses that we used.
We can see that the UDP header is 8 bytes and port 50001 is used in the source and port 41096 is used in the client.



We can see that the payload of the first packet is equal to the file name and the trailing Z that has been added.



Problem 3:

File transfer times from the previous lab without reliable transfer were:

With 1400 byte packets:

File size	Total time in ms	bps

2560	5.398 , 8.53, 7.85 , 6.88 , 6.25	3793997, 2400937, 2605929, 2975014, 3273134
25600	165.379, 5.88, 8.03, 4.44 , 7.23	1238367, 34788517, 25507535, 46126126, 28318584
256000	71.282, 11.557 , 12.60, 13.75 , 12.44	28730955 , 177208618, 162449432 , 148902137, 164550859

With 1000 byte packets

File size	Total time in ms	bps
2560	66.30 , 3.96 , 9.29 , 4.53 , 5.78	308875, 5166498 , 2203809 , 4519973 , 3541414
25600	61.466 , 7.902 , 7.46 , 7.30 , 6.11	3331923, 25917489, 27445725 , 28027918 , 33502371
256000	10.541, 12.67 , 40.41, 85.15 , 112.96	182145906, 119318181, 50670493 , 24051673 , 16654874

Now with our reliable transfer implementation we will once again run the experiments from amber05 to amber06:

With 1400 byte packets:

File size	Total time in ms	bps
2560	3.5040 , 185.46 , 8.796 , 4.671 , 5.729	5844748 , 110425 , 2328331 , 4384500 , 3574794
25600	13.834 , 4.78 , 35.927 , 9.3160 , 8.164	14804105 , 42791475 , 5700448 , 21983683, 25085742
256000	16.059 , 12.370, 37.679, 17.31, 13.260	127529734, 165561843, 54353884, 118258459, 154449472

With 1000 byte packets

File size	Total time in ms	bps
2560	89.36, 8.9 , 5.744, 4.6820, 6.575000	229164 , 2295449, 3565459, 4374199, 3114828
25600	13.189000, 8.3120, 13.039 , 6.199, 136.05	15528091, 24639076 , 15706725, 33037586, 1505328

256000	29.172 , 14.64 , 311.944 , 15.467, 10.889	70204305 , 139890710 , 6565280 , 132410939 , 188079713

We can see that on average the time for reliable file transfer is not too high from the unreliable file transfer. This is due to the reliable nature of the connection between amber05 and amber06. Because these two systems are close to each other with a strong and fast connection, UDP packets are rarely lost so our functions for handling lost packets are not used. The over head for checking lost packets is too little to be seen. However, when packets are lost, the time difference is extreme. We can see this with effect with the 3rd test transmitting 256000 bytes with 1000 byte packets. The transfer process took more than 300 milliseconds which is more than 10 times the longest time previously seen. This is because we had to retransmit a number of packets.

Problem 4 (bonus):

M2_ch153:
There are 4 packets with 802.11n network types and the rest are 802.11a
SNR of these 4 packets are 40, 42, 42, 45 dB and these are all Acknowledgment frames.
All frames have frequency equal to 5765MHz. The signal to noise ratio is anywhere from 30 dB to 50dB, except for broadcast requests which are around 10-20 dB
There are different types of frame: Request-to-send, Clear-to-send, Block Ack, NDP Announcement, Probe Request.
There are different receiver addresses seen that can be seen which can be the address of the access points: a broadcast message is sent to address ff:ff:ff:ff:ff:ff - Broadcom_10:50:d6 (e0:3e:44:10:50:d6) - Broadcom_10:50:d6 (e0:3e:44:10:50:d6) - AmazonTe_af:d4:93 (a0:d2:b1:af:d4:93)

M2-ch124:
A number of frames are with 802.11a and a number of them are with 802.11n.
All the 802.11n packets are of length 72 and are Acknowledgement type. Their frequency is 5680MHz and SNR is around 13 dB. It seems like these are between different interfaces of a <Apple_> device with these addresses: Apple_39:dc:27 (a4:83:e7:39:dc:27), Apple_2a:49:1f (20:a2:e4:2a:49:1f), etc.
For the 802.11a frames, there are different types of frame: Request-to-send, Clear-to-send, Block Ack, NDP Announcement, Probe Request, beacon frame, etc.
SNR of these frames are around 10 dB

M2_ch1:
There are 802.11g, 802.11b, 802.11n , and 802.11fx frames.

802.11fx are QoS data frames with SNR of around 5 dB. All of them are from 32:0a:15:59:7f:cc (32:0a:15:59:7f:cc) to either Cisco_9f:f0:01 (00:00:0c:9f:f0:01) or IPv4mcast_fb (01:00:5e:00:00:fb).
802.11n frames are either Acknowledgements or QoS. SNR in these frames are dependent of the source and the destination. For example, those to Apple_34:d3:1a (f0:18:98:34:d3:1a) have around 30-40 dB of SNR ratio but those to Receiver IntelCor_bc:8d:fb (28:6b:35:bc:8d:fb) have less than 10 dB SNR.
Those with 802.11b type are sent to destinations such asBroadcom_00:00:00 (e0:3e:44:00:00:00) and Broadcom_00:00:00 (e0:3e:44:00:00:00). Those sent to Cisco_bb:12:82 (50:0f:80:bb:12:82) are all Null functions, but those sent to the others have different types of frame: Request-to-send, Clear-to-send, Block Ack, NDP Announcement, Probe Request, beacon frame, etc.
802.11g frames are either broadcasted or are sent to Cisco_bb:12:82 (50:0f:80:bb:12:82). Their SNR are mostly around 10 dB with some exceptions that are around 30-40 dB