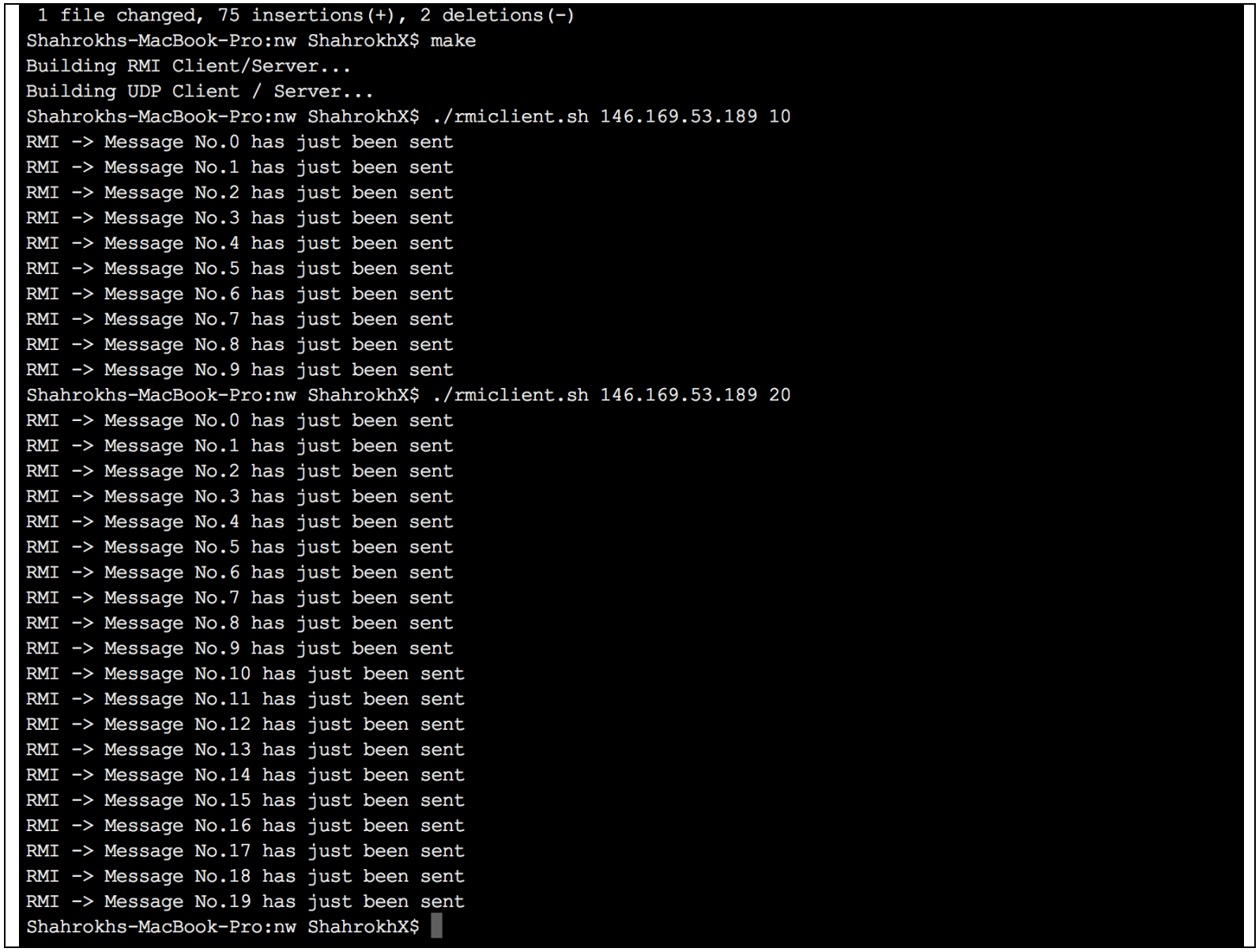
❖ A short summary of my findings

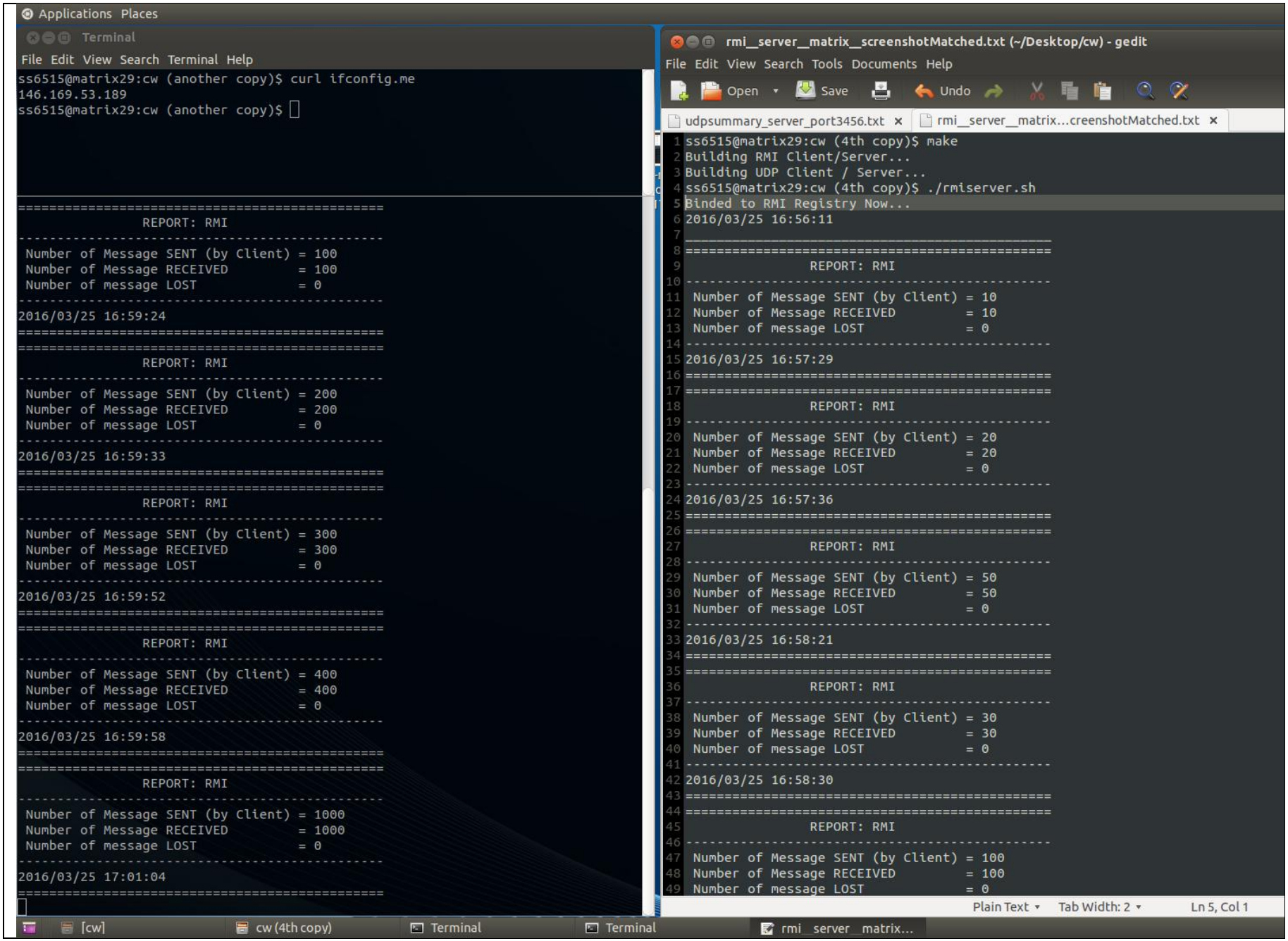
Item (Questions)			RMI	UDP		
Tests			All Tests	Test 1	Test 2	Test 3
Experiments Summaries	Number of **LOST** msg	20	0	0	0	0
		30	0	0	29	1
		40	0	0	39	23
		50	0	0	23	29
		100	0	90	0	14
		200	0	113	0	199
		300	0	0	0	154
		400	0	0	0	0
a. What are the possible causes, if any, of messages being lost?			Implementing RMI, no packet will be lost, since RMI is based on TCP. (Practically, the RMI registry runs on TCP port 1099). The packets will be sent and received in order and if a packet is missed, the sender will resend the packet until getting acknowledgment.	Packet loss is generally caused by network congestion. It can happen when a packet in a given router or network segment is greater than the possible capacity to send and consequently, there is no other option than dropping packet.		
b. Are there any patterns in the way messages are lost?			There is no packet loss in RMI.	As it can be seen in the results, there is no specific pattern in the packet loss. But generally when the number of packets increases, the packet loss will be more. It can be a result of instantaneous congestion. However, most of the time, there is no specific pattern for that. For example, in some case, no packet lost in 2000 packets, but 9 packets were lost in only 10 sent packets.		
c. What is the relative reliability of the different communication mechanisms?			RMI and in general TCP, is connection-oriented in which it requires acknowledgment of received data. If the data is not received, a retransmission will be requested. Thus, there is a absolute guarantee that the data transferred remains intact and arrives in the same order in which it sent.	UDP is unreliable. It is a connectionless protocol and it means there is no connection setup and handshake mechanism in UDP. It does not retransmit the lost packets. Consequently less delay will happen. Therefore, UDP applications must generally accept some packet loss and, as an exemplary case, it can be more functional for live conferences and transferring multi media.		
Which one is easier to program?			RMI seems easier to program, especially for large scale distributed systems. It is easy to implement the function based program and there is no need to take care of datagram details. But it seems there are some limitations in implementing RMI. For complicated programs, it requires users to implement complicated steps due to the predefined static interfaces.			

❖ Screen shots:

RMI client was my laptop (Shahrokhs-MacBook-Pro) and the messages were sent to the server on a lab's machine with IP 146.169.53.189:



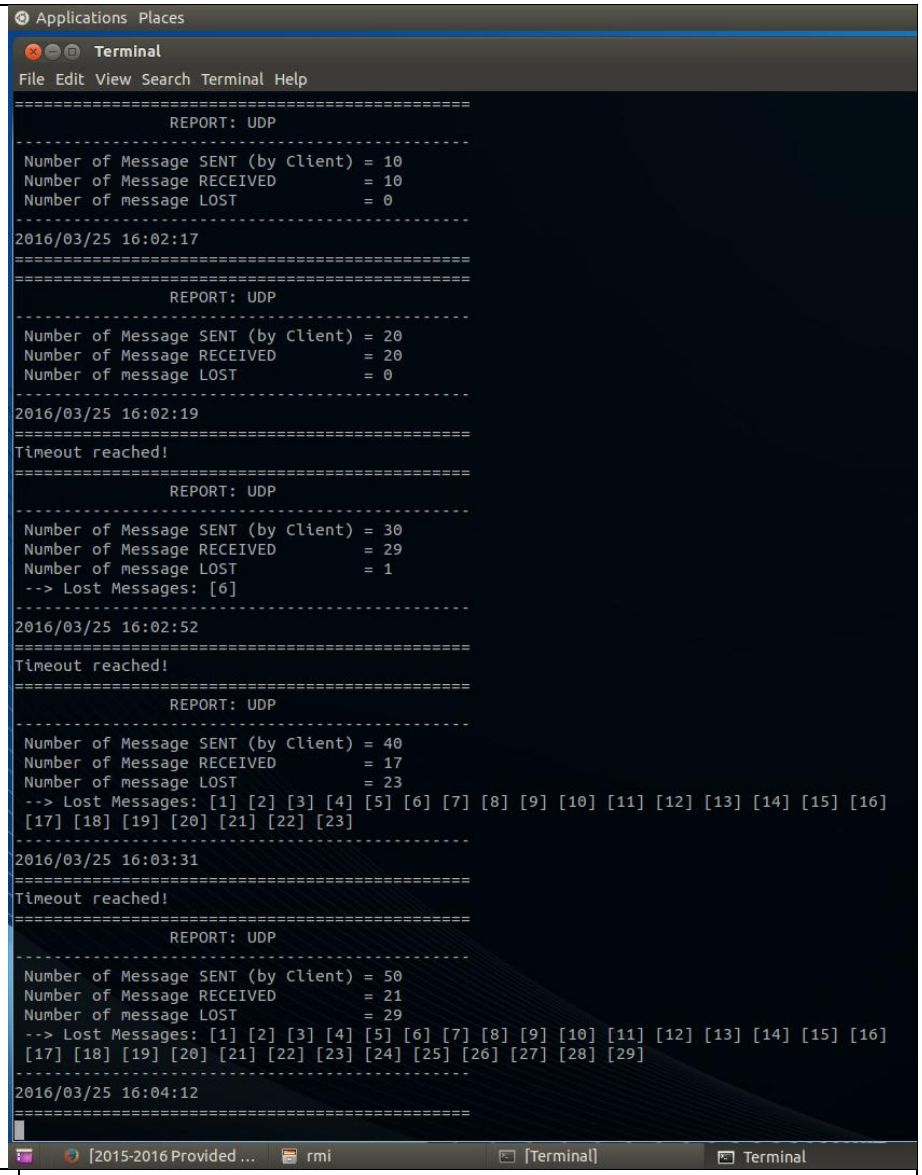
The RMI Server was a lab's machine (a Matrix series):



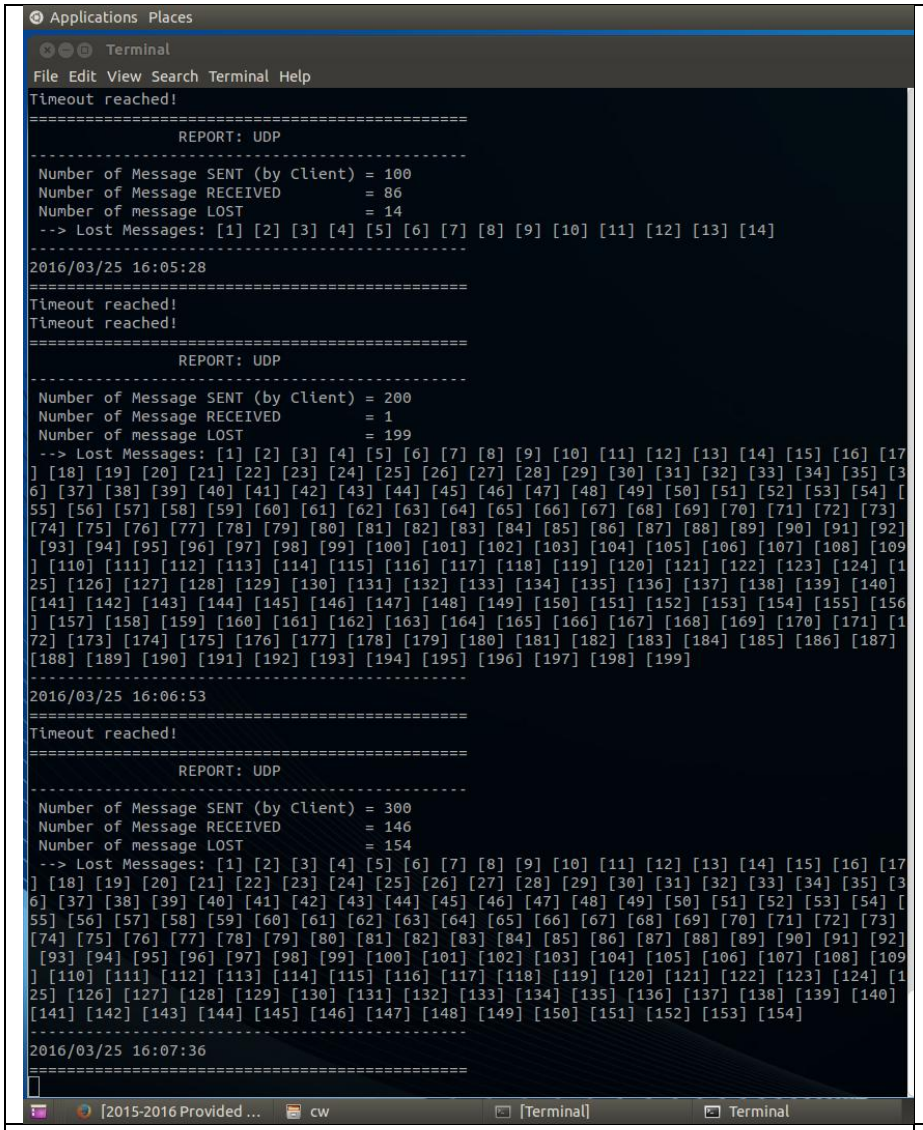
For the UDP test, different machines with different ports around the lab and also my laptop have been tried:

- UDP Server: Corona-20 IP: 146.169.53.100 PORT: 3456

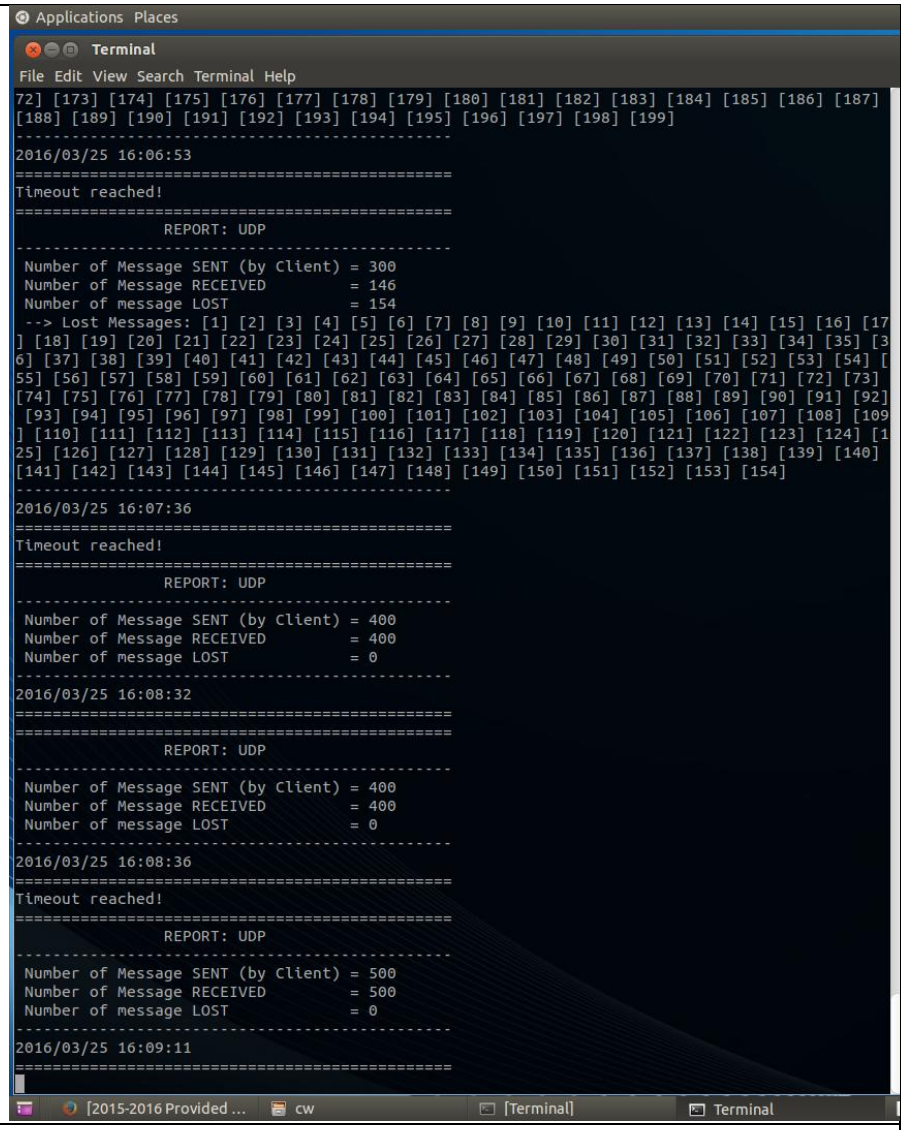
Receiving some packets from first client (It was a Corona lab's machine)



Receiving some packets from the second client (My Laptop)
The lost packets can be seen in the picture

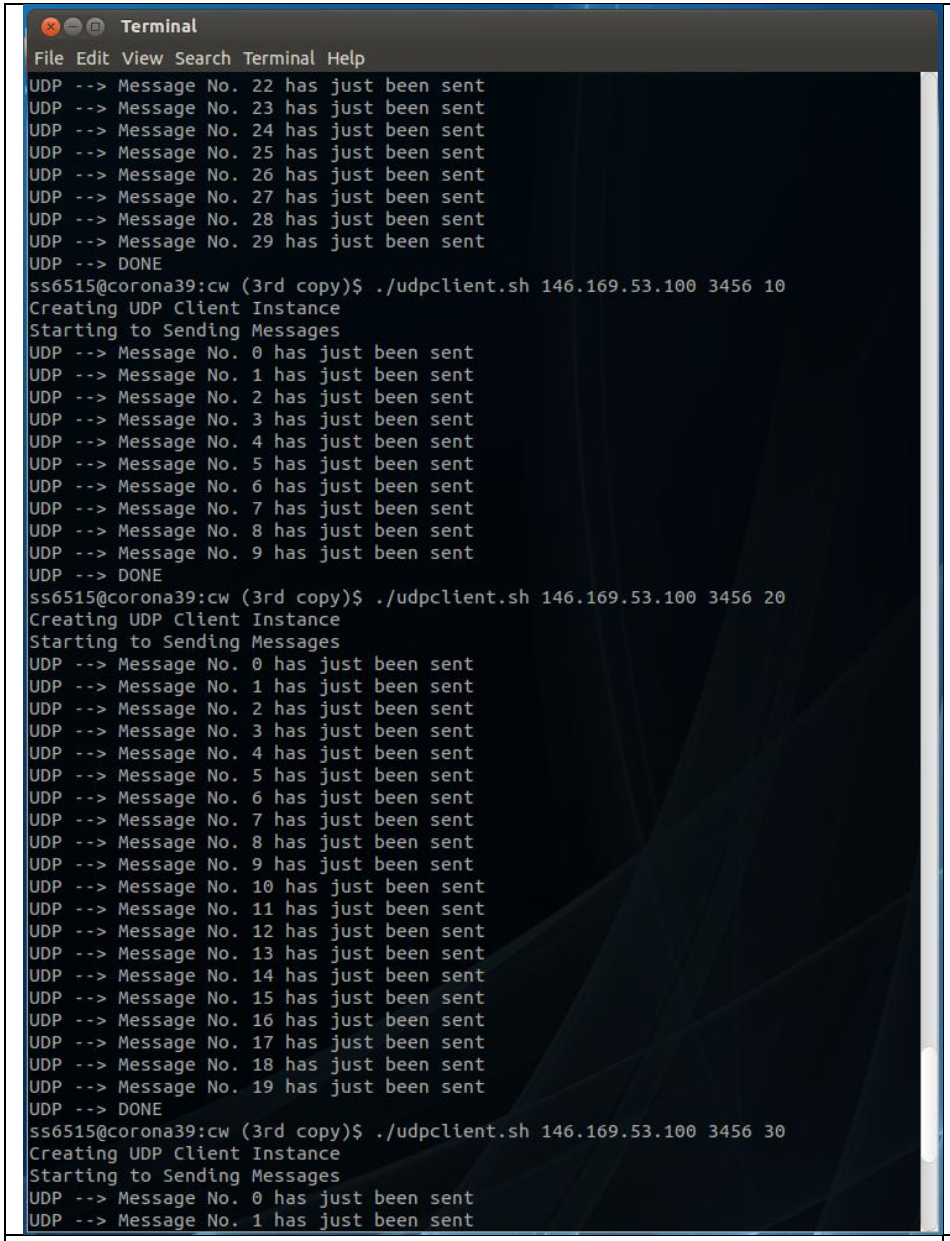


A lot of packets are lost in 200 and 300 sent messages but there is no pattern for that (199 out of 200, 154 out of 300)

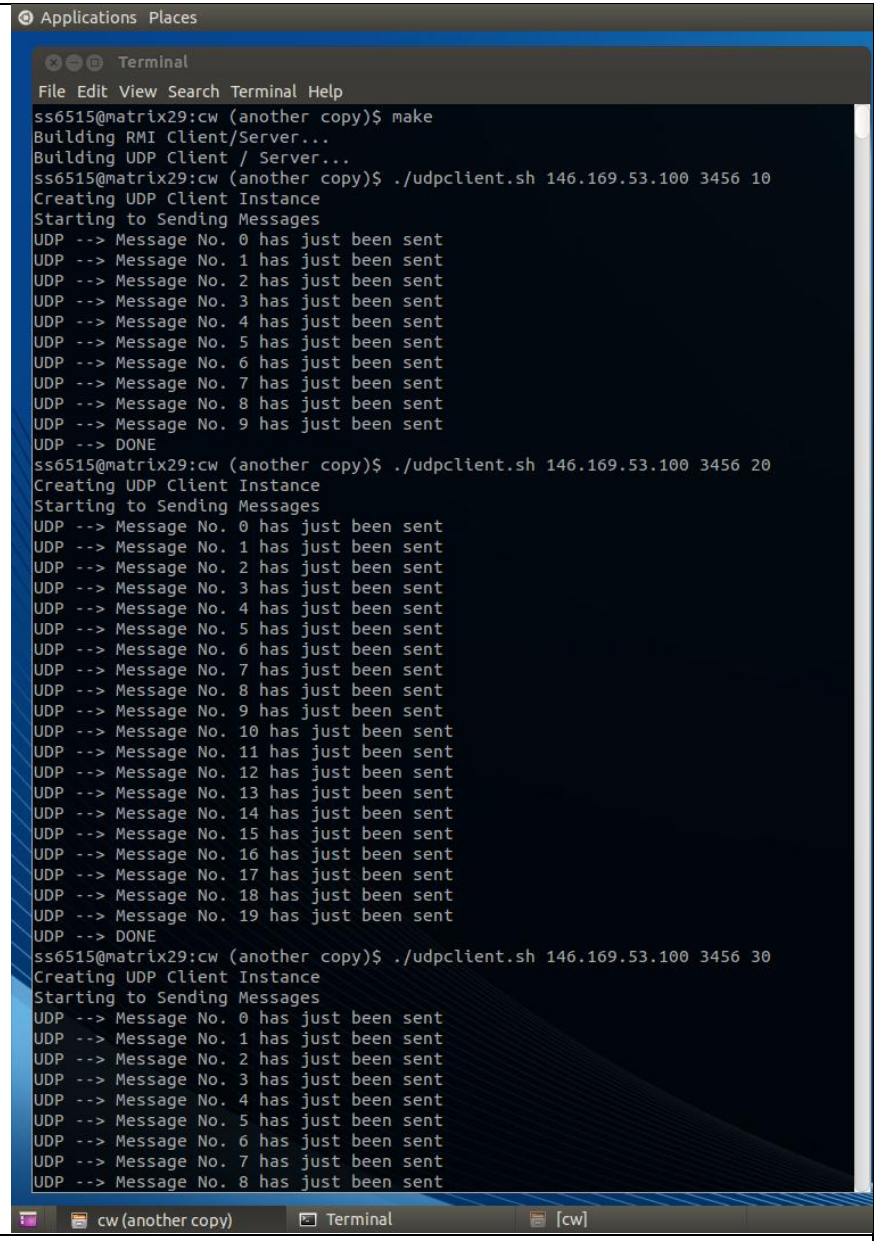


Again, all packets are received!

- UDP Client: Corona-39 and Matrix-29



Corona-39 as UDP Client



Matrix-29 as UDP client

❖ **All the results are saved in text files attached to submitted zip file.**