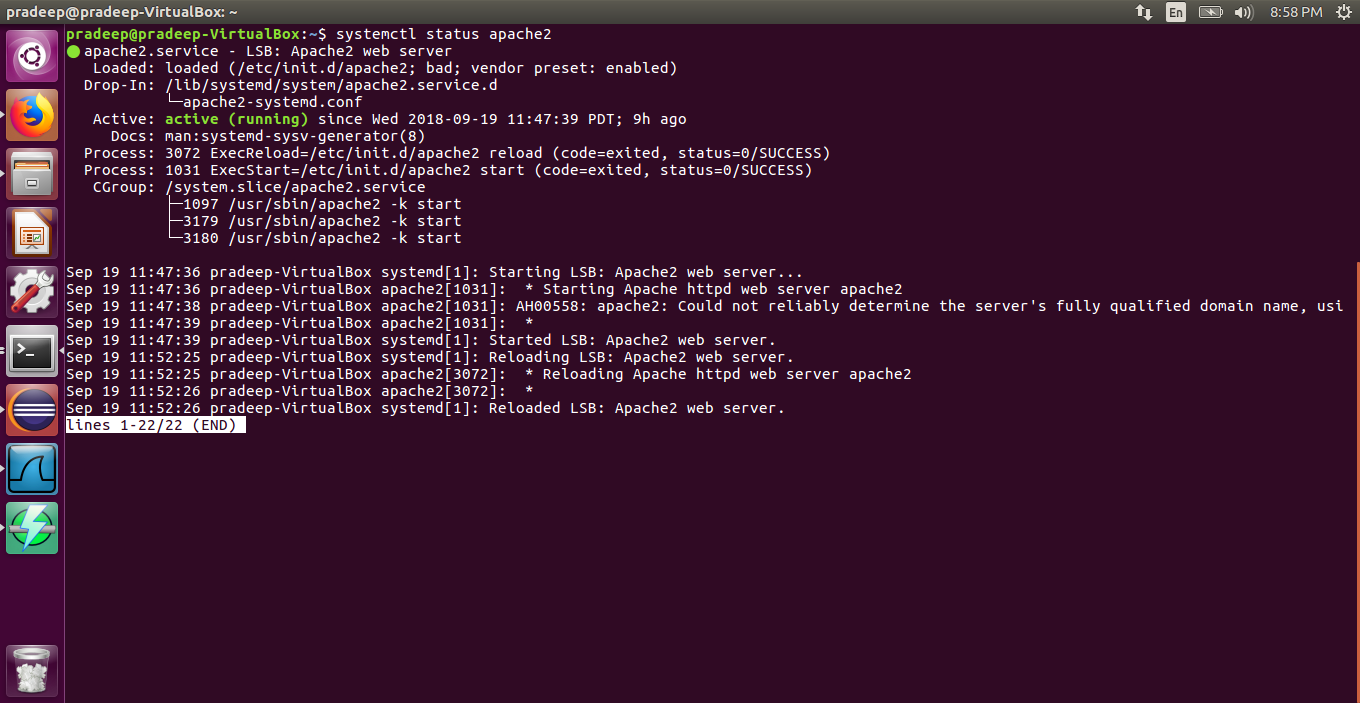
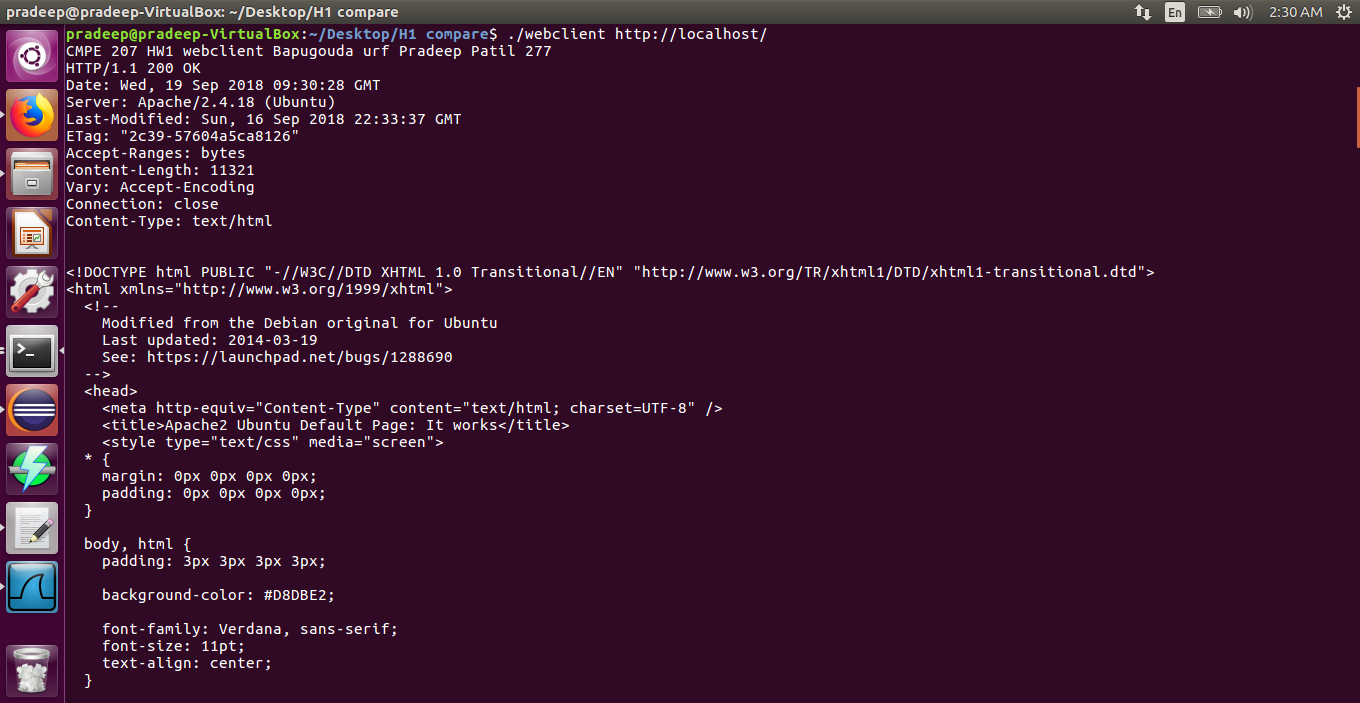
Part 1 WEBCLIENT

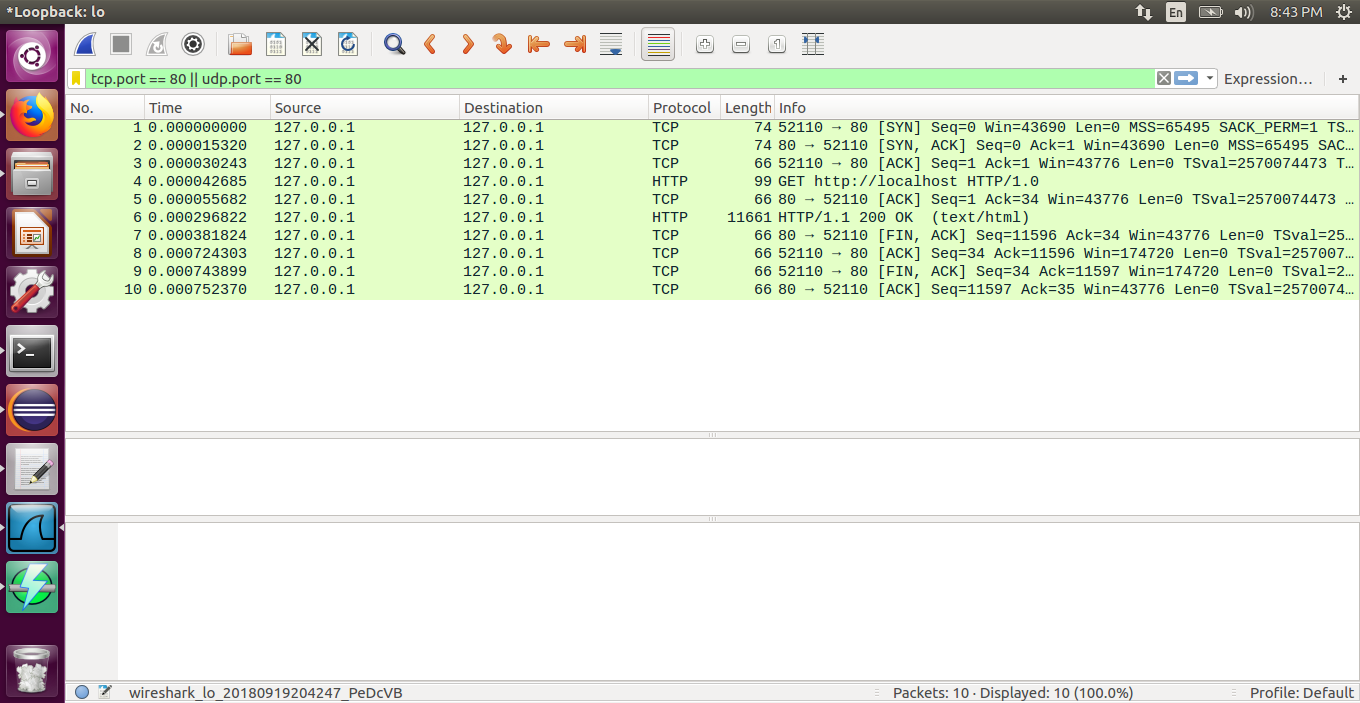
1 a. Apache2 Server Status



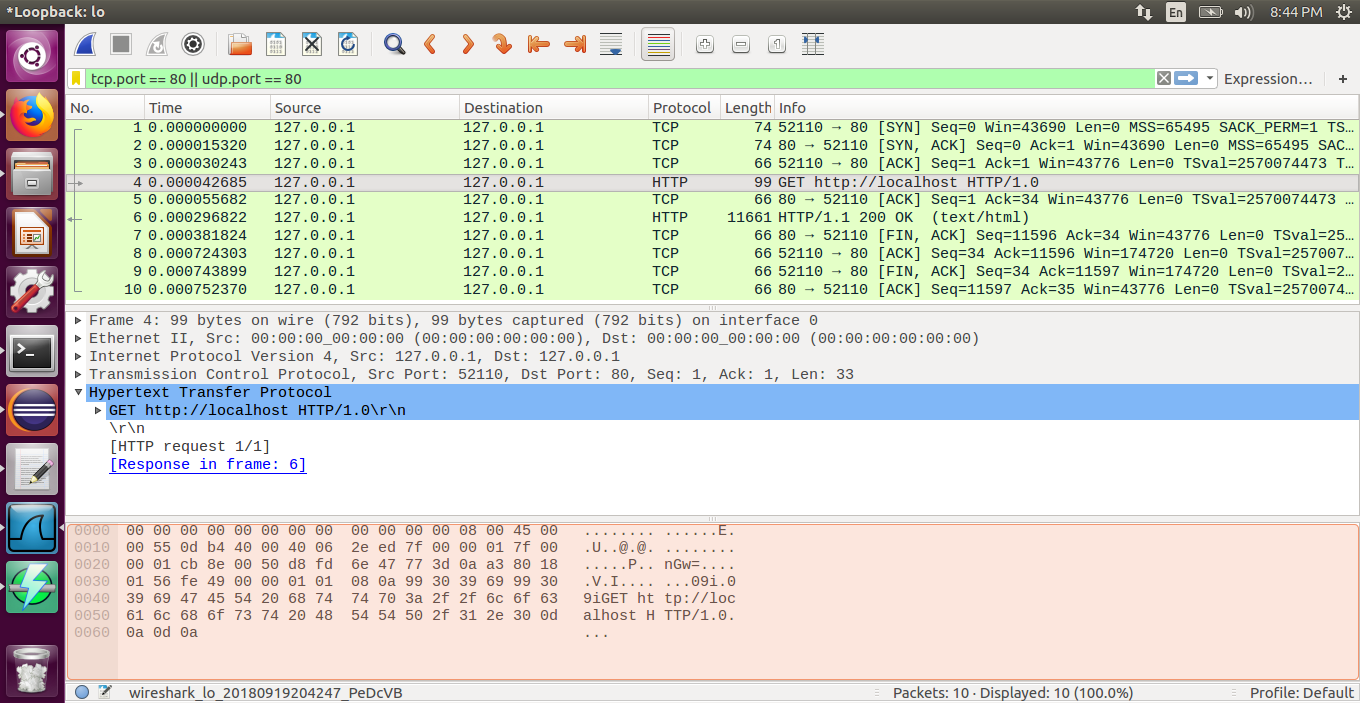
1 b. Program Exicution



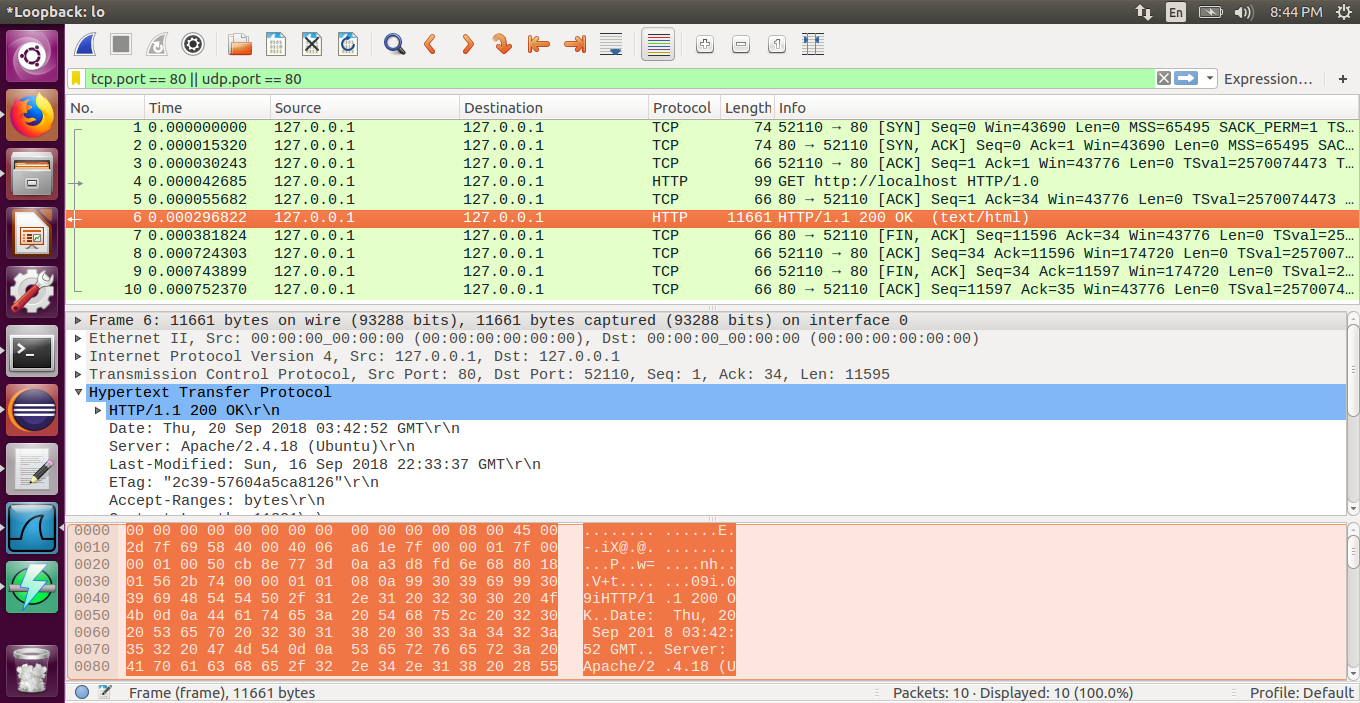
1 c. 1 List of All the Packets



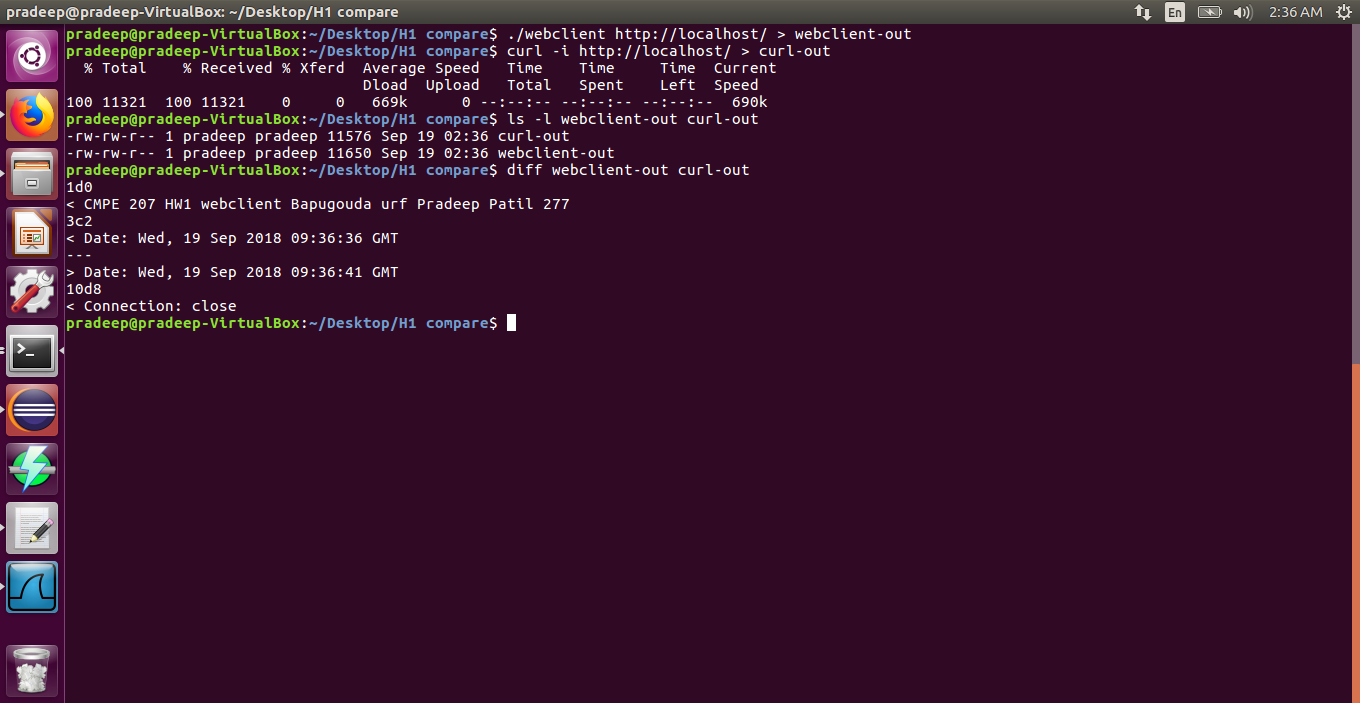
1 c. 2 Capture of Request Packet Expanded



1 c. 2 Capture of Response Packet Expanded

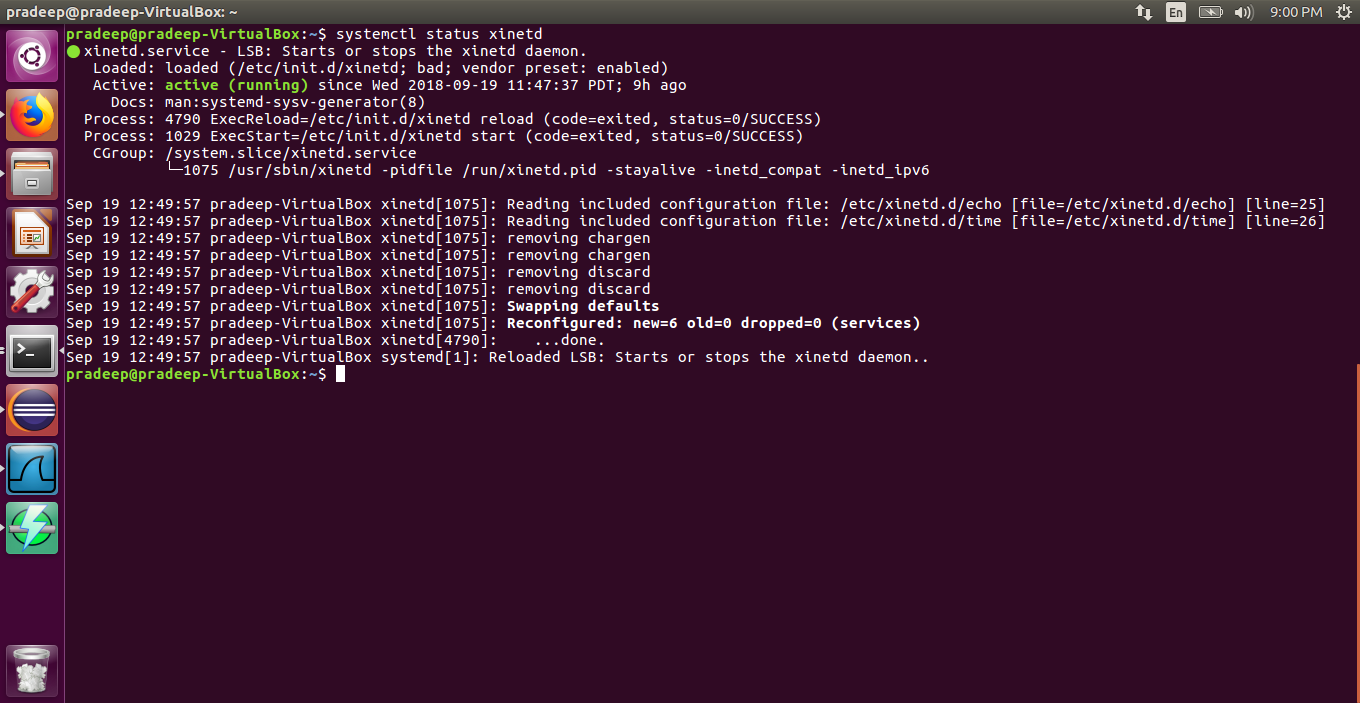


1 d. Comparison between the Webclient output and Curl Output.

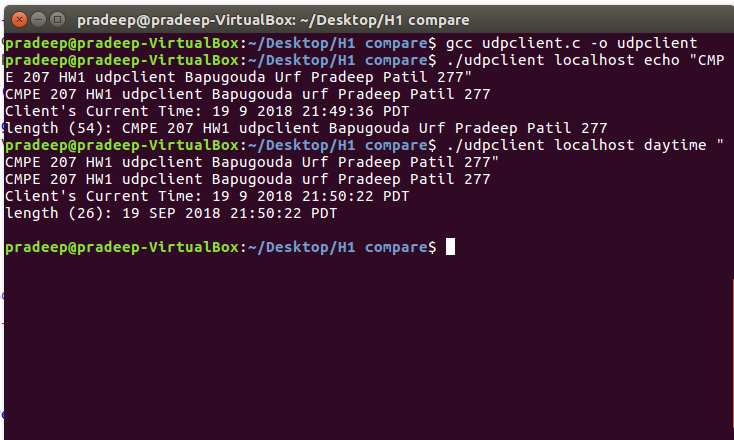


Part 2 UDPCLIENT

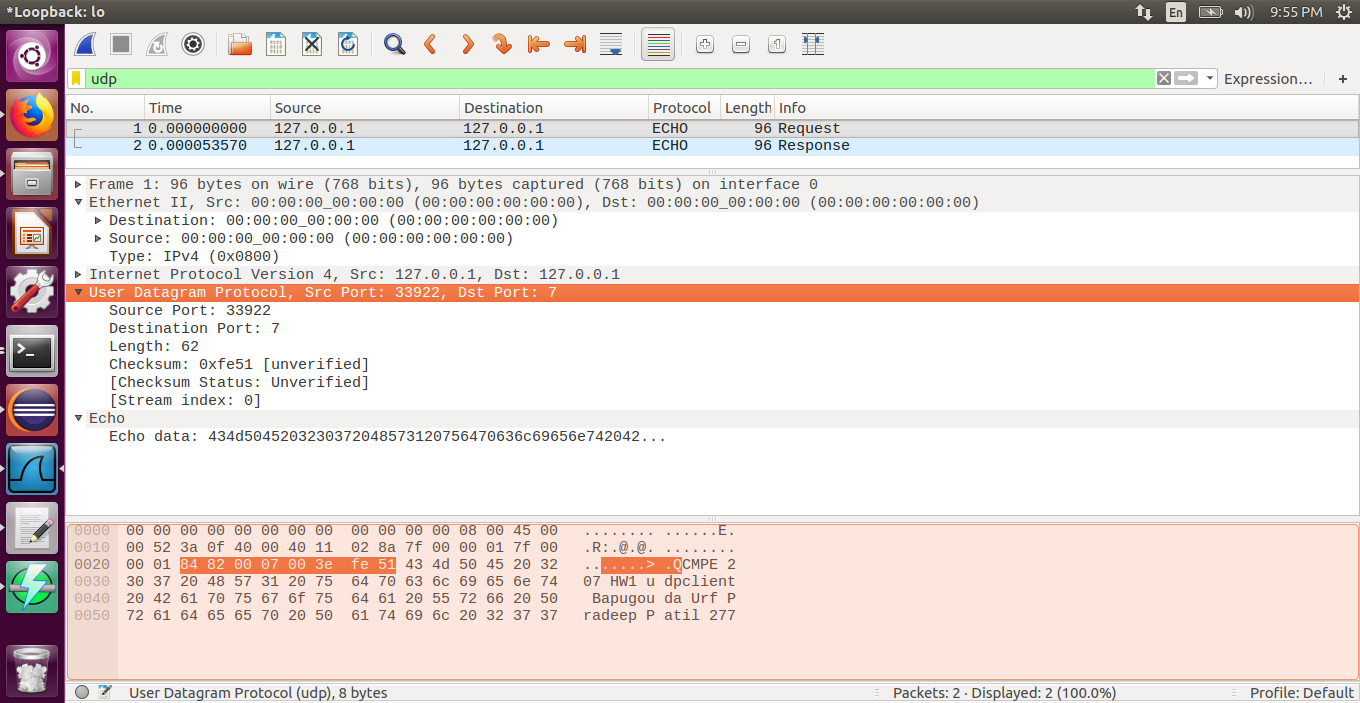
2 a. XINETD Server Status



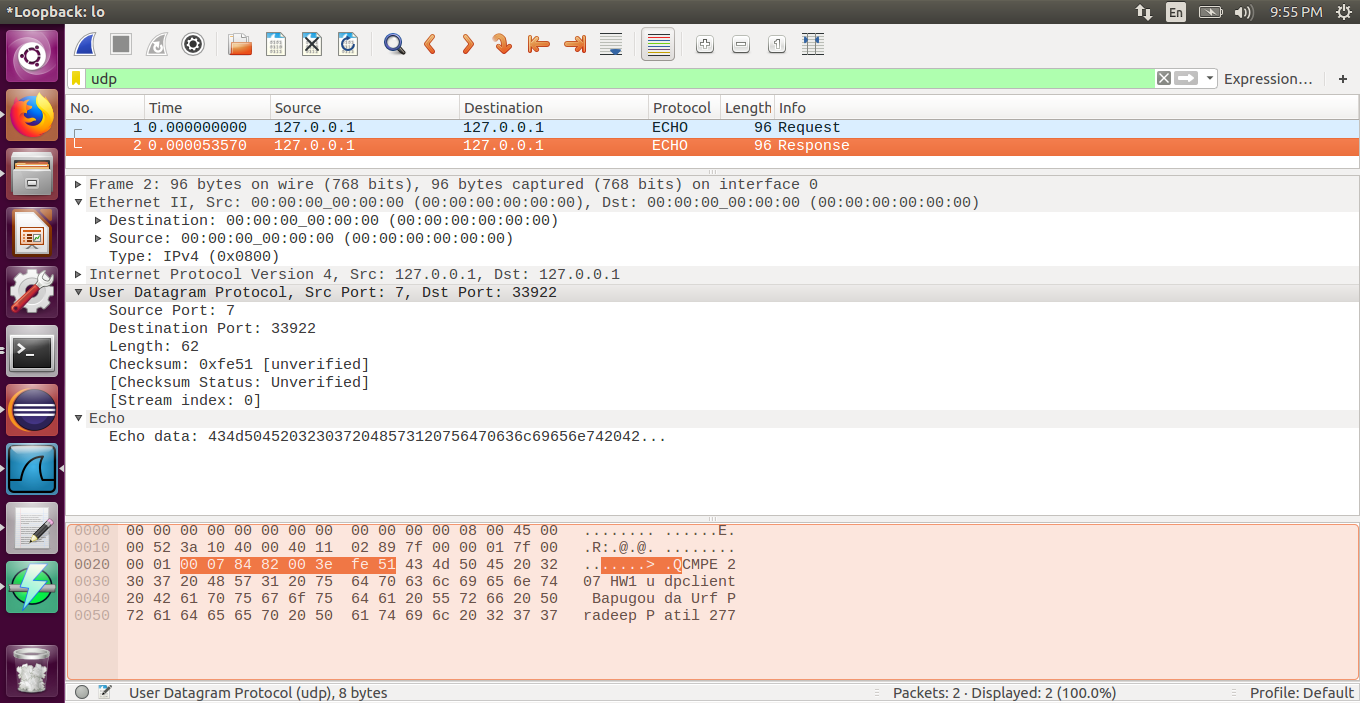
2 b. Program Exicution udpclient



2 c. WireShark Capture of Request



2 c. Wireshark Capture of Response



Part 3 Capture Compare

3. (4 pts each, 20 pts)Use your wireshark screenshots for TCP and UDP, and compare the differences of client-server packet exchange between TCP and UDP. In particular, other than the request and response packets, explain the following “additional” packets of TCP (when compared with UDP):

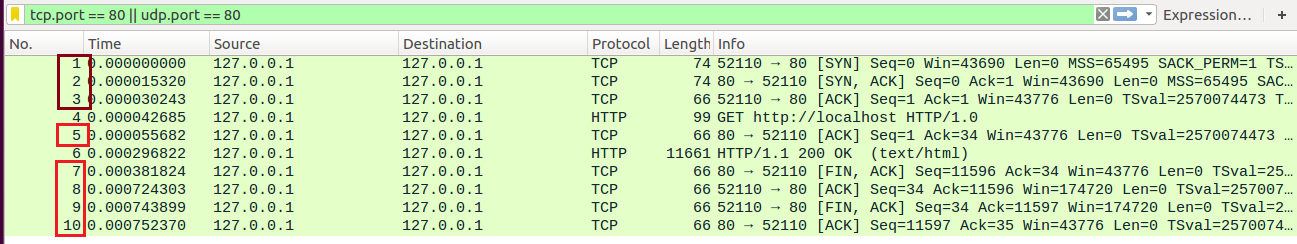
a. Initiate connection

b. Destroy connection

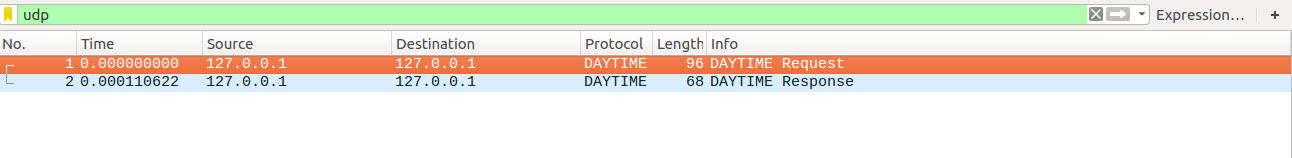
c. After the client sends a request to the server but before server sends the response

d. After the server sends a response to the client

TCP Packets



UDP Packets



1. **Initiate connection**

The first 3 packets are responsible the TCP 3 way handshake to create a connection between the client and server. To establish a connection, TCP uses a three-way [handshake](https://en.wikipedia.org/wiki/Handshaking). Before a client attempts to connect with a server, the server must first bind to and listen at a port to open it up for connections: this is called a passive open. Once the passive open is established, a client may initiate an active open. Resulting 3 packets 1. [SYN] 2. [SYN,ACK] 3. [ACK]

1. **Destroy connection**

Once the transfer is completed the active connection between client and server is closed and all the associated resources are released. The connection termination phase uses a four-way handshake, with each side of the connection terminating independently. When an endpoint wishes to stop its half of the connection, it transmits a FIN packet, which the other end acknowledges with an ACK. Therefore, a typical tear-down requires a pair of FIN and ACK segments from each TCP endpoint. After the side that sent the first FIN has responded with the final ACK, it waits for a timeout before finally closing the connection, during which time the local port is unavailable for new connections.

1. **After the client sends a request to the server but before server sends the response**

This packet is an acknowledgement sent from server to client to assert that the request was successfully received. As TCP is connection oriented all the incoming packets are acknowledged by the receiving end.

1. **After the server sends a response to the client**

After the response is sent we have two cases, I. if all the data requested by the client is sent, the server sends a signal to indicate the requested. II. If data is sent partially the server sends the number of bytes sent as a parameter and then sends the remaining data in parts and continues. All the packets from the server are acknowledged by the client. Once the transfer is complete one of the end points initiate the close of the connection and the connection is closed as explained in b.