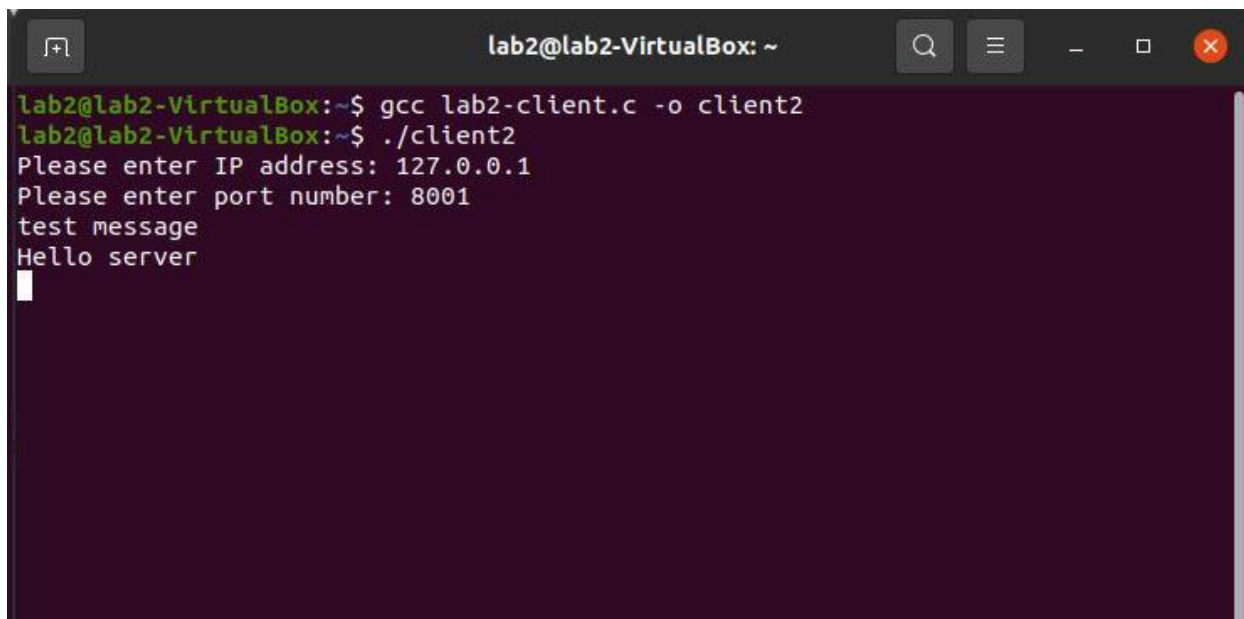
A terminal window titled 'lab2@lab2-VirtualBox: ~' with standard window controls. It shows the compilation of 'lab2-server.c' into 'server2' and its execution on port 8001. The server receives two messages: 'test message' and 'Hello server'.

```
lab2@lab2-VirtualBox:~$ gcc lab2-server.c -o server2
lab2@lab2-VirtualBox:~$ ./server2 8001
recv from client: test message
recv from client: Hello server
█
```

A terminal window titled 'lab2@lab2-VirtualBox: ~' with standard window controls. It shows the compilation of 'lab2-client.c' into 'client2' and its execution. The user is prompted for an IP address (127.0.0.1) and a port number (8001). The client sends 'test message' and 'Hello server' to the server.

```
lab2@lab2-VirtualBox:~$ gcc lab2-client.c -o client2
lab2@lab2-VirtualBox:~$ ./client2
Please enter IP address: 127.0.0.1
Please enter port number: 8001
test message
Hello server
█
```

Terminal – client & server

Please note that the file name changed after the result is correct.

The picture capture is (lab2-client & lab2-server), but the files submitted were named according to the requirement of TA. Thank you.

The image shows the Wireshark network protocol analyzer interface. The title bar reads "Capturing from Loopback: lo". The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains various icons for file operations, capture control, and analysis. The packet list pane shows five captured packets:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	UDP	54	55368 → 8001 Len=12
2	4.042689384	127.0.0.1	127.0.0.1	UDP	54	55368 → 8001 Len=12
3	39.234378537	127.0.0.1	224.0.0.251	MDNS	87	Standard query 0x0000 PTR _ipp._tcp.local, "QM" question PTR ...
4	40.467690351	127.0.0.1	127.0.0.53	DNS	100	Standard query 0x47af AAAA connectivity-check.ubuntu.com OPT
5	40.477632236	127.0.0.53	127.0.0.1	DNS	100	Standard query response 0x47af AAAA connectivity-check.ubuntu...

The packet details pane for the selected packet (No. 1) shows the following structure:

- Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface lo, id 0
- Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
- Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
- User Datagram Protocol, Src Port: 55368, Dst Port: 8001
- Data (12 bytes)

The packet bytes pane displays the raw data in hexadecimal and ASCII:

```
0000  00 00 00 00 00 00 00 00 00 00 00 00 00 45 00  ....E.
0010  00 28 85 43 40 00 00 11 b7 7f 7f 00 00 01 7f 00  .(C@
0020  00 01 08 48 1f 41 00 14 fe 27 74 65 73 74 20 6d  .H A . 'test m
0030  65 73 73 61 67 65                                message
```

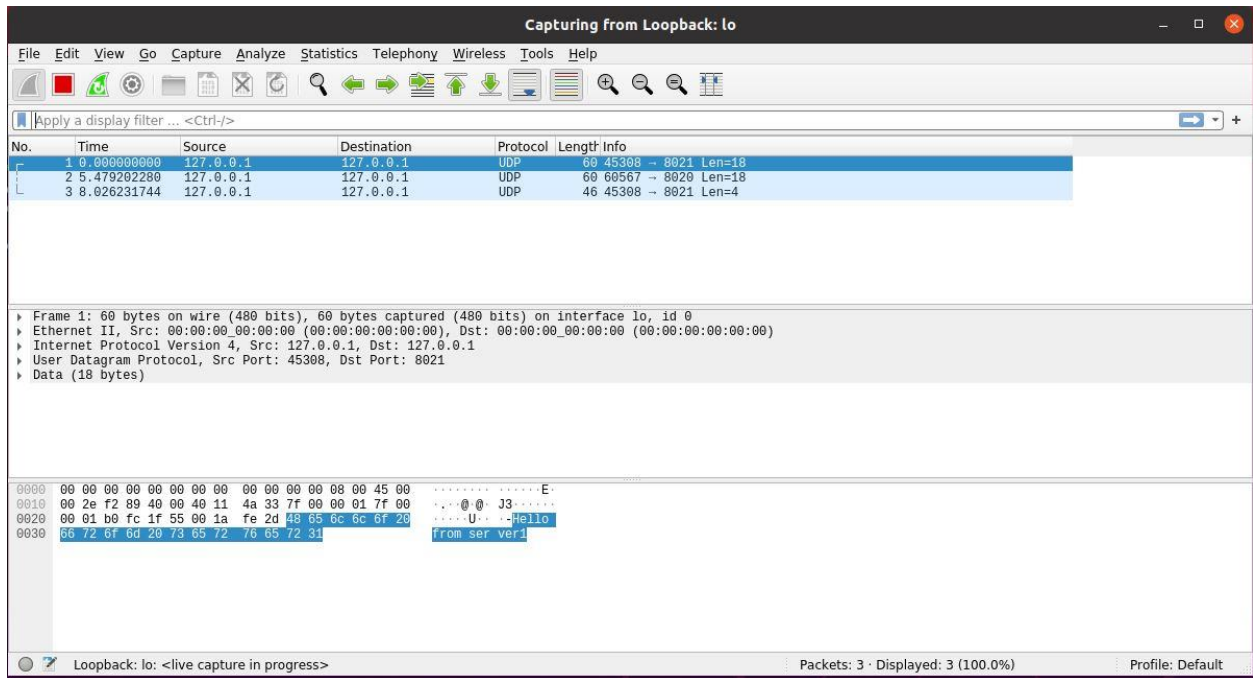
The status bar at the bottom indicates: User Datagram Protocol (udp), 8 bytes | Packets: 5 · Displayed: 5 (100.0%) | Profile: Default

Wireshark – client & server

```
lab2@lab2-VirtualBox: ~  
lab2@lab2-VirtualBox:~$ ./multithread 8020  
Please enter IP address: 127.0.0.1  
Please enter port number: 8021  
Hello from server1  
recv from client: Hello from server2  
test
```

```
lab2@lab2-VirtualBox: ~  
lab2@lab2-VirtualBox:~$ ./multithread 8021  
Please enter IP address: 127.0.0.1  
Please enter port number: 8020  
recv from client: Hello from server1  
Hello from server2  
recv from client: test  
□
```

Termial - multithread



Wireshark - multithread