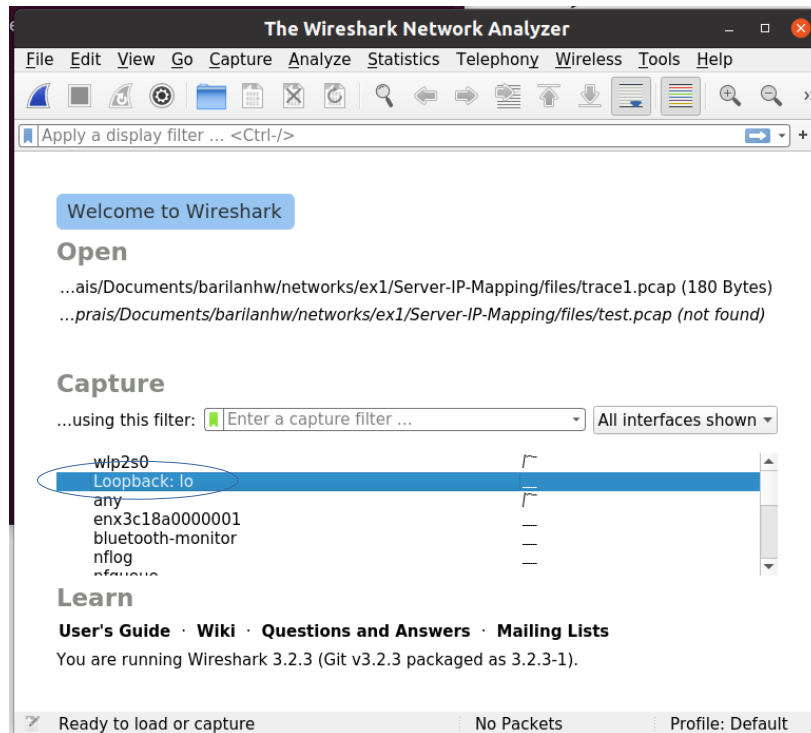


Networks - Assignment 1 Report

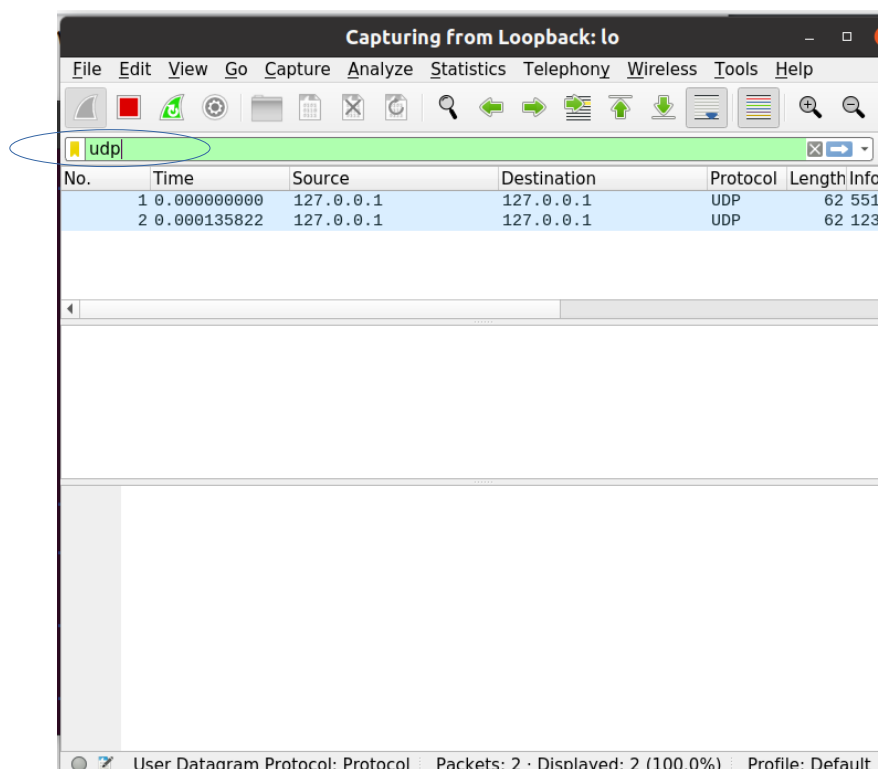
Part 1

How we Filtered the Packets

First, we opened the Wireshark and got into loopback to listen to the transport in the local machine.



We filtered the packets according to their specific protocol, in purpose to find the UDP packets, where our packets are exist. We found two packets – the packet that the client sent to the server, and the packet that the server sent to the client.



About the Port Numbers

The ports are numbers that specify the actual resource in the computer the message is sent to or received from. For example, in client-server architecture, the server is bound to a specific port which the client knows, so when the client sends a message to the server, he sends it to the server ip address, on the server port.

In the Wireshark, we can see two port numbers in the packet header - the first port is the sender port on his local machine, and the second port is the receiver port on his local machine (for example, in the packet that the client sent to the server, the first port is the client port, and the second port is the server port). The port logic is in the Transport layer.

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 a list of captured packets, with packet 1 selected. The packet details pane shows the structure of the selected packet, with the User Datagram Protocol (UDP) section expanded. The packet bytes pane shows the raw data of the packet, with the first 30 bytes highlighted. The status bar at the bottom indicates "User Datagram Protocol: Packets: 32 · Displayed: 32 (100.0%) Profile: Default".

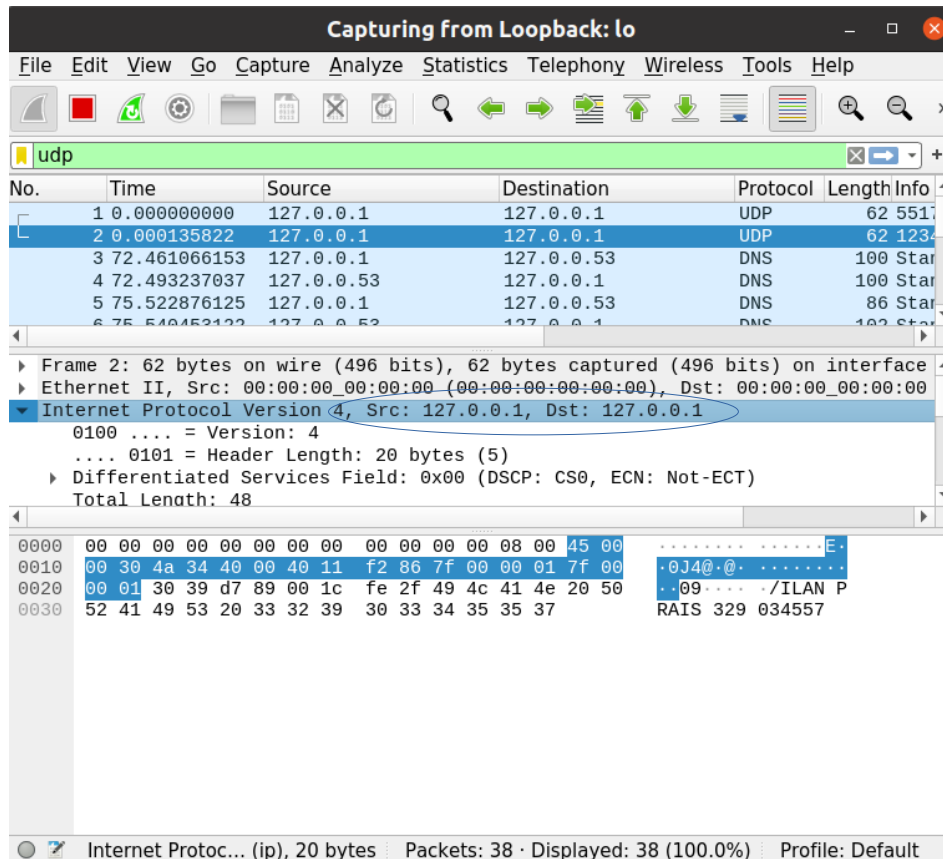
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	UDP	62	55177 → 12345
2	0.000135822	127.0.0.1	127.0.0.1	UDP	62	12345 → 55177
3	72.461066153	127.0.0.1	127.0.0.53	DNS	100	Standard query
4	72.493237037	127.0.0.53	127.0.0.1	DNS	100	Standard query response
5	75.522876125	127.0.0.1	127.0.0.53	DNS	86	Standard query
6	75.540453122	127.0.0.53	127.0.0.1	DNS	102	Standard query response

Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 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: 55177, Dst Port: 12345
Source Port: 55177
Destination Port: 12345
Length: 28

```
0000  00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00  .....E.
0010  00 30 4a 33 40 00 40 11 f2 87 7f 00 00 01 7f 00  .0J3@.@. ....
0020  00 01 d7 89 30 39 00 1c fe 2f 49 6c 61 6e 20 50  ....09.. ./Ilan P
0030  72 61 69 73 20 33 32 39 30 33 34 35 35 37      rais 329 034557
```

About the IP Addresses

The first ip address in the packet header is the ip address of the sender, and the second ip address is the ip address of the receiver.



As we can see, the ip addresses in the packet header are similar to the ip addresses of the local machine, that we found using the `ifconfig` command.

```
ilando vprais@ilando vprais-linux: ~/Documents/barilanhw/networks/...
enx3c18a0000001: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
ether 3c:18:a0:00:00:01 txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 30536 bytes 3122523 (3.1 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 30536 bytes 3122523 (3.1 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

vlp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.68.107 netmask 255.255.255.0 broadcast 192.168.68.255
inet6 fe80::a64c:df1d:31d1:d4af prefixlen 64 scopeid 0x20<link>
ether f8:94:c2:68:e6:83 txqueuelen 1000 (Ethernet)
RX packets 3626784 bytes 3694989141 (3.6 GB)
RX errors 0 dropped 42 overruns 0 frame 0
TX packets 1767863 bytes 552973870 (552.9 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```