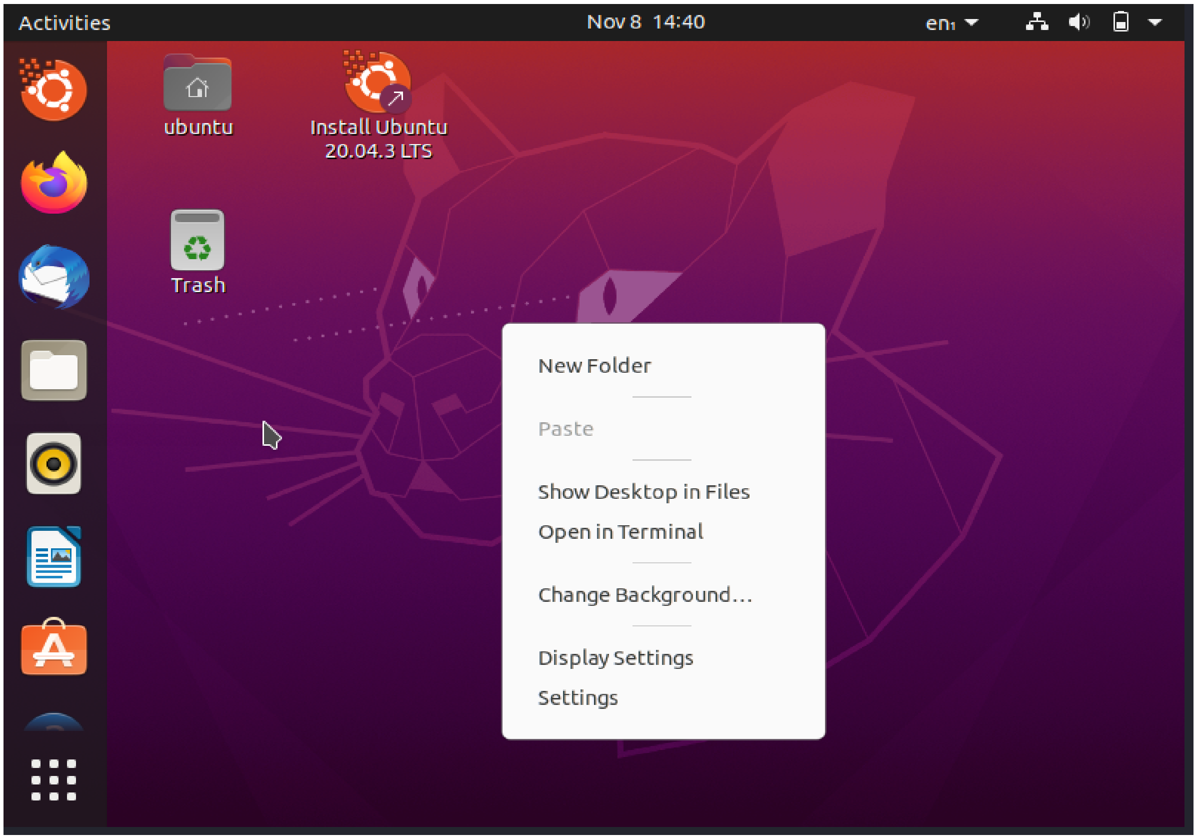
Network Programming Lab - Program Steps 4 and 5

Program 4

1. **UDP Program Execution Steps:**

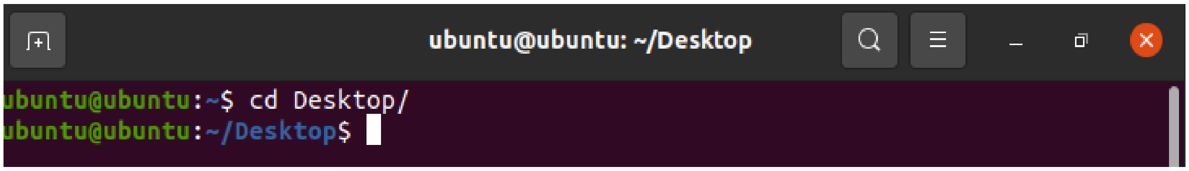
   Steps :

   - Open Terminal using **CTRL+ALT+T** or right-click and select **Open in Terminal**.



   - Once opened, type the following command in the terminal:

     cd Desktop/

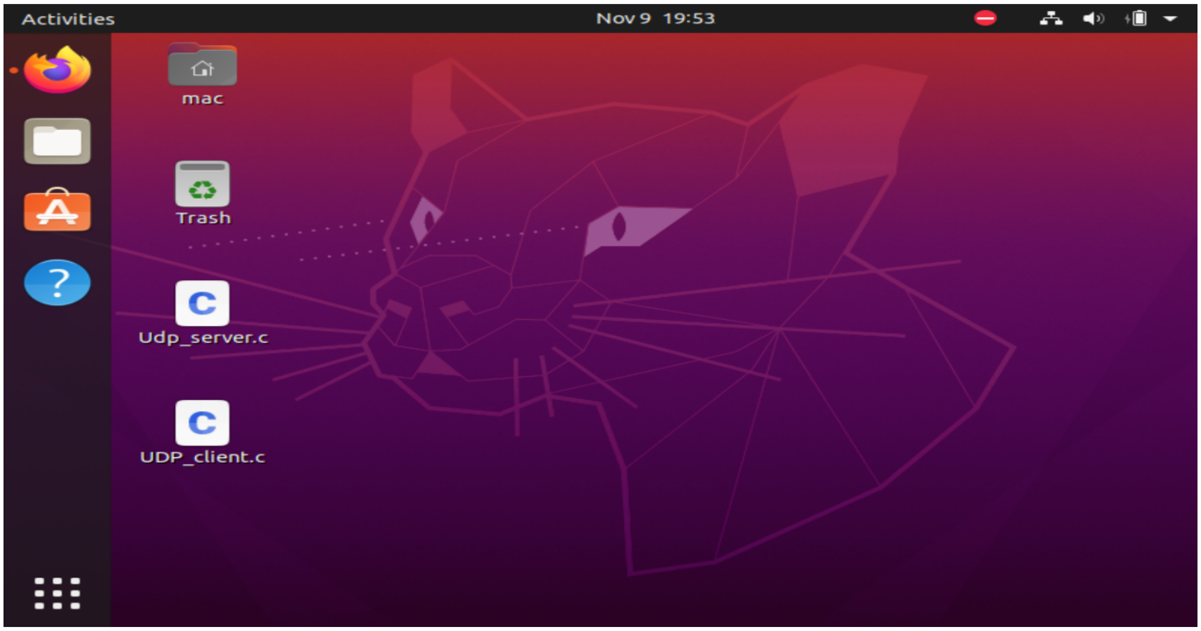


```
ubuntu@ubuntu:~$ cd Desktop/
ubuntu@ubuntu:~/Desktop$ 
```

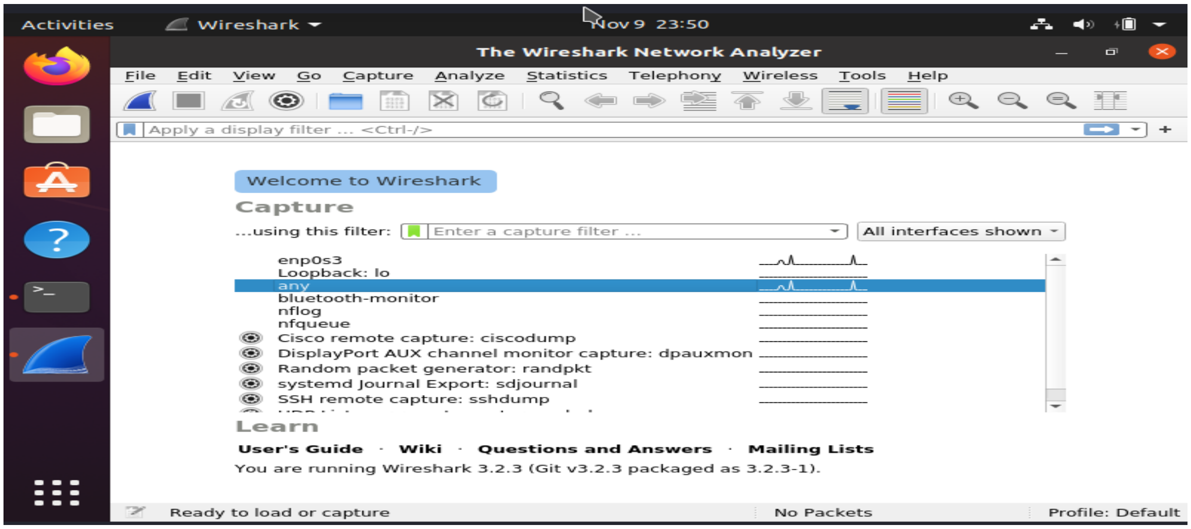
This command will change your default terminal path to desktop. **(Note: Linux is case-sensitive, so type the command as it is)**

[Before going through the next sequence of commands, make sure that both the UDP programs are located on the desktop]

Network Programming Lab - Program Steps 4 and 5



- Before executing the program, open **Wireshark** and double-click on **any-interface** for packet capture process to start.



```
enp0s3
Loopback: lo
any
bluetooth-monitor
nflog
nfqueue
Cisco remote capture: ciscodump
DisplayPort AUX channel monitor capture: dpauxmon
Random packet generator: randpkt
systemd Journal Export: sdjournal
SSH remote capture: sshdump
```

You are running Wireshark 3.2.3 (Git v3.2.3 packaged as 3.2.3-1).

Ready to load or capture   No Packets   Profile: Default

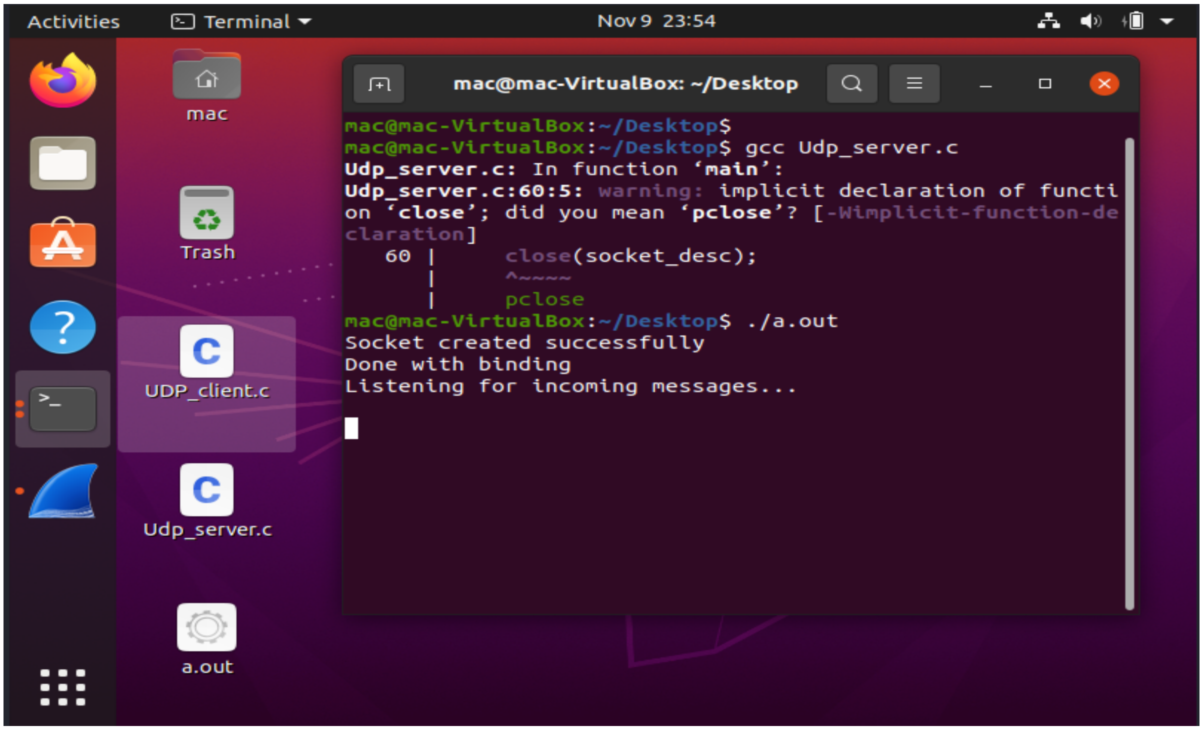
- To compile the C program, type the following command in the same order as follows in the terminal:

```
gcc  Udp_server.c
```

- To execute the program type, object code file name in the terminal as follows :

```
./a.out
```

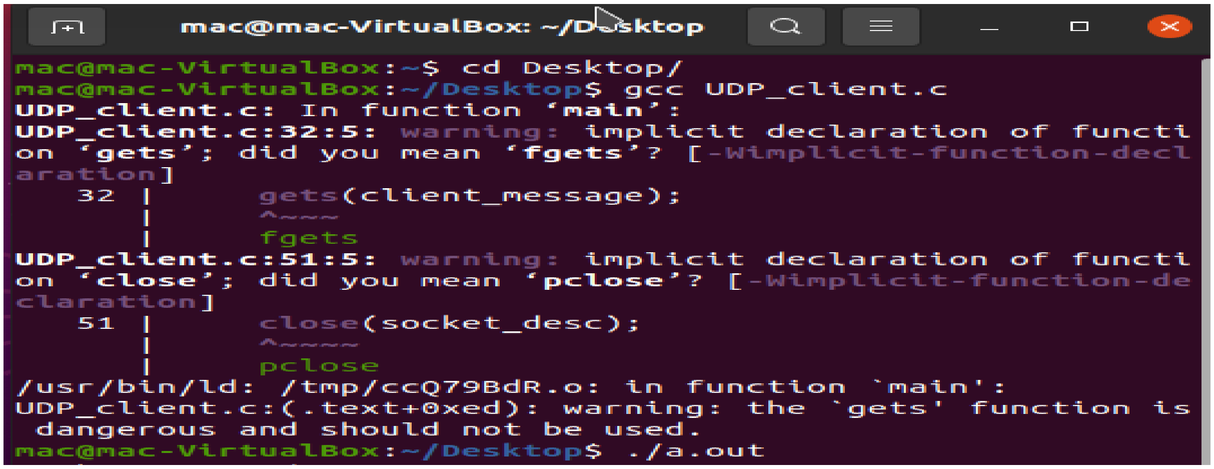
Network Programming Lab - Program Steps 4 and 5



```
mac@mac-VirtualBox:~/Desktop$
mac@mac-VirtualBox:~/Desktop$ gcc Udp_server.c
Udp_server.c: In function ‘main’:
Udp_server.c:60:5: warning: implicit declaration of functi
on ‘close’; did you mean ‘pclose’? [-Wimplicit-function-de
claration]
   60 |     close(socket_desc);
      |     ^~~~~
      |     pclose
mac@mac-VirtualBox:~/Desktop$ ./a.out
Socket created successfully
Done with binding
Listening for incoming messages...
```

- Open another terminal and change its path to Desktop using Cd command and run the c code as shown below:

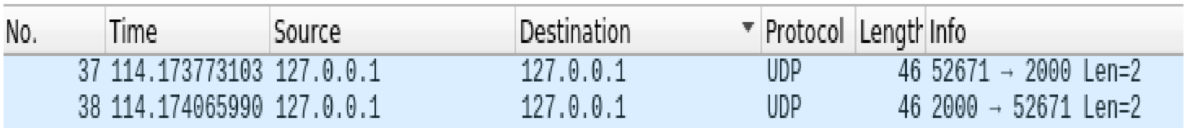
  cd Desktop/
  gcc UDP_client.c
  ./a.out



```
mac@mac-VirtualBox:~$ cd Desktop/
mac@mac-VirtualBox:~/Desktop$ gcc UDP_client.c
UDP_client.c: In function ‘main’:
UDP_client.c:32:5: warning: implicit declaration of functi
on ‘gets’; did you mean ‘fgets’? [-Wimplicit-function-decl
aration]
   32 |     gets(client_message);
      |     ^~~~
      |     fgets
UDP_client.c:51:5: warning: implicit declaration of functi
on ‘close’; did you mean ‘pclose’? [-Wimplicit-function-de
claration]
   51 |     close(socket_desc);
      |     ^~~~~
      |     pclose
/usr/bin/ld: /tmp/ccQ79BdR.o: in function `main':
UDP_client.c:(.text+0xed): warning: the `gets' function is
 dangerous and should not be used.
mac@mac-VirtualBox:~/Desktop$ ./a.out
```

- Once the client program is executed, it will ask you to enter the appropriate message that will be sent to the server.
- After the execution of the program, go to Wireshark and stop the capturing process and check for the packet that has the source and the destination address as **127.0.0.1.**

Network Programming Lab - Program Steps 4 and 5



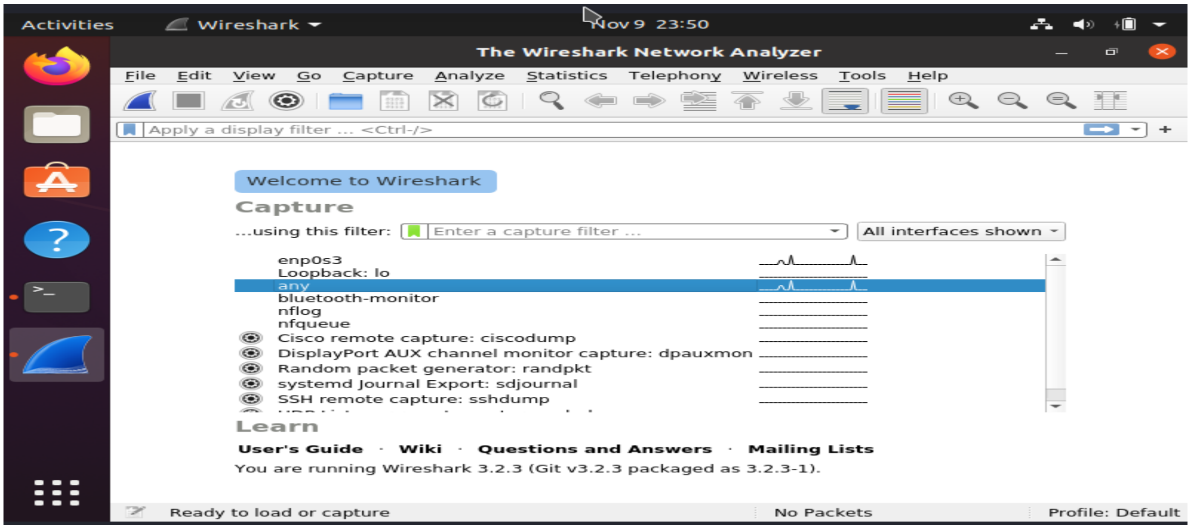
No.	Time	Source	Destination	Protocol	Length	Info
37	114.173773103	127.0.0.1	127.0.0.1	UDP	46	52671 → 2000 Len=2
38	114.174065990	127.0.0.1	127.0.0.1	UDP	46	2000 → 52671 Len=2

- Analyze the UDP Packets.

2. Capturing UDP Packets with browser :

Steps:

- Open Wireshark and double-click on **any-interface** to start the packet capture process.

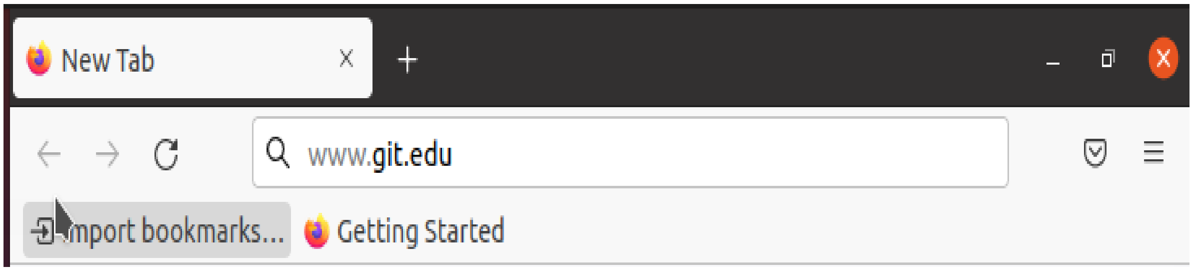


Activities
Wireshark
Nov 9 23:50
The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Welcome to Wireshark
Capture
…using this filter:
Enter a capture filter …
All interfaces shown
enp0s3
Loopback: lo
any
bluetooth-monitor
nflog
nfqueue
Cisco remote capture: ciscodump
DisplayPort AUX channel monitor capture: dpauxmon
Random packet generator: randpkt
systemd Journal Export: sdjournal
SSH remote capture: sshdump
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists

You are running Wireshark 3.2.3 (Git v3.2.3 packaged as 3.2.3-1).

Ready to load or capture | No Packets | Profile: Default

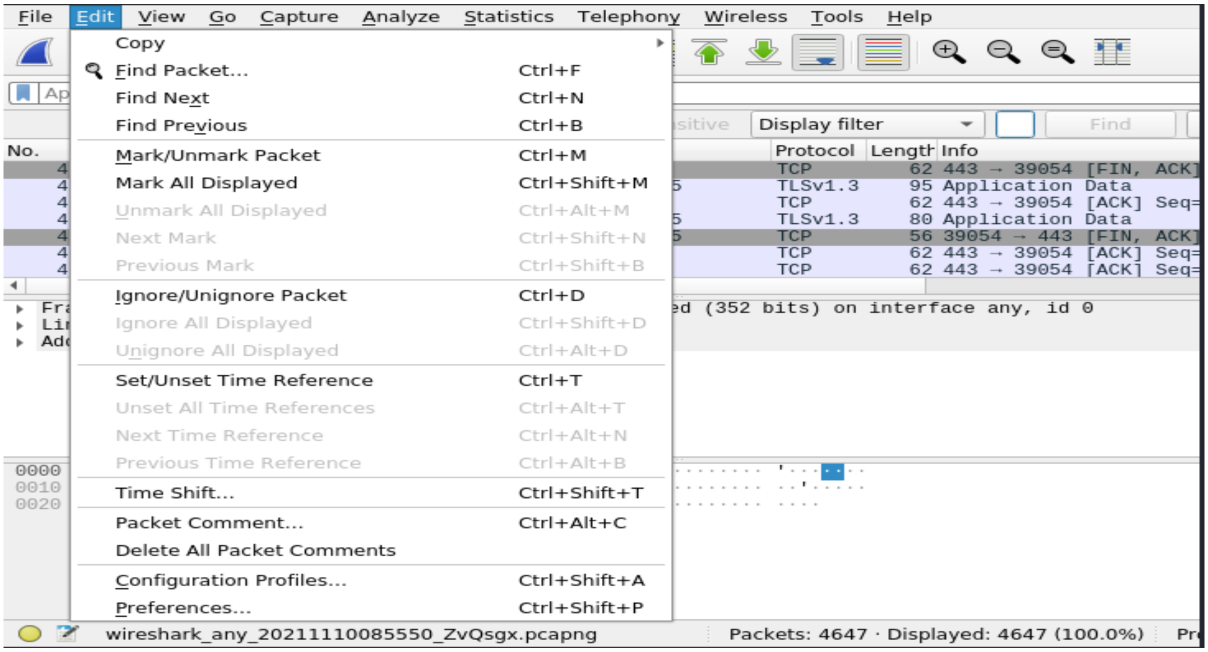
- Open the browser and enter any website's fully qualified domain name in the browser address bar and hit enter.



www.git.edu

- After the site is fully loaded, stop the capturing process in Wireshark go to edit in the menu bar and select find packet option or just press CTRL+F.

Network Programming Lab - Program Steps 4 and 5

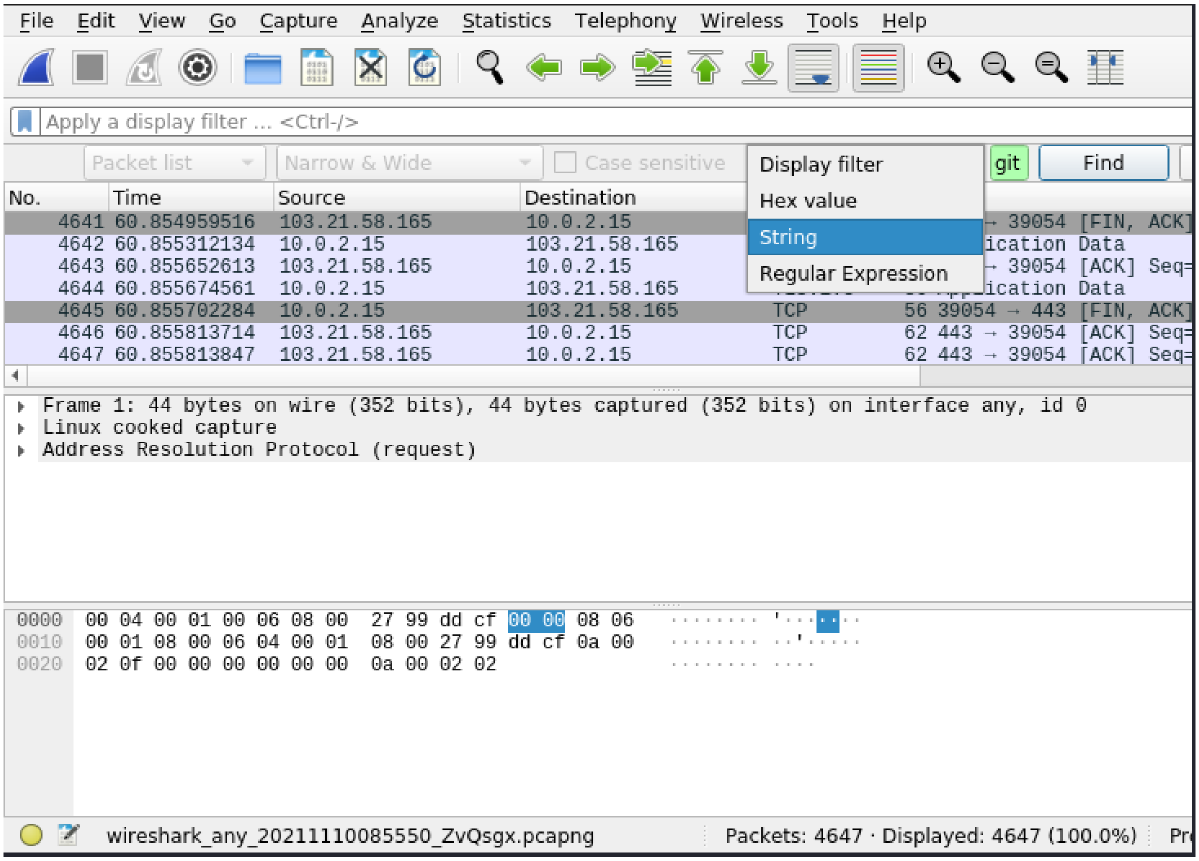

File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Copy
Find Packet... Ctrl+F
Find Next Ctrl+N
Find Previous Ctrl+B
Mark/Unmark Packet Ctrl+M
Mark All Displayed Ctrl+Shift+M
Unmark All Displayed Ctrl+Alt+M
Next Mark Ctrl+Shift+N
Previous Mark Ctrl+Shift+B
Ignore/Unignore Packet Ctrl+D
Ignore All Displayed Ctrl+Shift+D
Unignore All Displayed Ctrl+Alt+D
Set/Unset Time Reference Ctrl+T
Unset All Time References Ctrl+Alt+T
Next Time Reference Ctrl+Alt+N
Previous Time Reference Ctrl+Alt+B
Time Shift... Ctrl+Shift+T
Packet Comment... Ctrl+Alt+C
Delete All Packet Comments
Configuration Profiles... Ctrl+Shift+A
Preferences... Ctrl+Shift+P
sitive
Display filter
Find
No.
Protocol
Length
Info

```
TCP       62 443 → 39054 [FIN, ACK]
TLSv1.3   95 Application Data
TCP       62 443 → 39054 [ACK] Seq=
TLSv1.3   80 Application Data
TCP       56 39054 → 443 [FIN, ACK]
TCP       62 443 → 39054 [ACK] Seq=
TCP       62 443 → 39054 [ACK] Seq=
```

```
ed (352 bits) on interface any, id 0
```

wireshark_any_20211110085550_ZvQsgx.pcapng
Packets: 4647 · Displayed: 4647 (100.0%)

- In Find Packet menu bar, select the **String** option in the **display filter drop-down** menu and enter the name of the website in the next box and click on find.



```
No.   Time            Source           Destination
 4641 60.854959516    103.21.58.165    10.0.2.15                    → 39054 [FIN, ACK]
 4642 60.855312134    10.0.2.15        103.21.58.165                ication Data
 4643 60.855652613    103.21.58.165    10.0.2.15                    → 39054 [ACK] Seq=
 4644 60.855674561    10.0.2.15        103.21.58.165                ication Data
 4645 60.855702284    10.0.2.15        103.21.58.165    TCP      56 39054 → 443 [FIN, ACK]
 4646 60.855813714    103.21.58.165    10.0.2.15        TCP      62 443 → 39054 [ACK] Seq=
 4647 60.855813847    103.21.58.165    10.0.2.15        TCP      62 443 → 39054 [ACK] Seq=
```

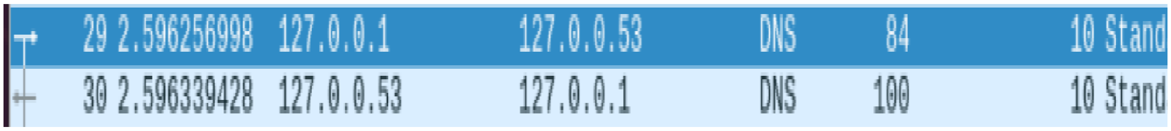
```
Frame 1: 44 bytes on wire (352 bits), 44 bytes captured (352 bits) on interface any, id 0
Linux cooked capture
Address Resolution Protocol (request)
```

```
0000  00 04 00 01 00 06 08 00  27 99 dd cf 00 00 08 06
0010  00 01 08 00 06 04 00 01  08 00 27 99 dd cf 0a 00
0020  02 0f 00 00 00 00 00 00  0a 00 02 02
```

wireshark_any_20211110085550_ZvQsgx.pcapng
Packets: 4647 · Displayed: 4647 (100.0%)

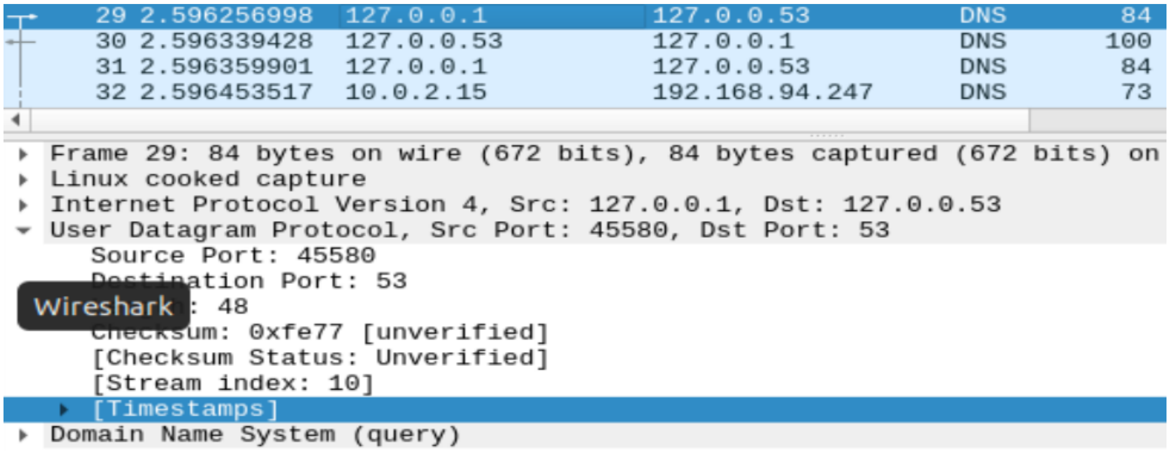
Network Programming Lab - Program Steps 4 and 5

- The arrow indicating towards the packet is the **request packet**, and the arrow coming out from the packet is the **response packet**.



```
29 2.596256998  127.0.0.1    127.0.0.53   DNS   84   10 Stand
30 2.596339428  127.0.0.53   127.0.0.1    DNS   100  10 Stand
```

- Click on any request or response DNS packet and examine UDP packet.



```
29 2.596256998  127.0.0.1    127.0.0.53      DNS   84
30 2.596339428  127.0.0.53   127.0.0.1       DNS  100
31 2.596359901  127.0.0.1    127.0.0.53      DNS   84
32 2.596453517  10.0.2.15    192.168.94.247  DNS   73

Frame 29: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on
Linux cooked capture
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.53
User Datagram Protocol, Src Port: 45580, Dst Port: 53
   Source Port: 45580
   Destination Port: 53
   Wireshark : 48
   Checksum: 0xfe77 [unverified]
   [Checksum Status: Unverified]
   [Stream index: 10]
   [Timestamps]
Domain Name System (query)
```

Program 5

1. **TCP Program Execution Steps:**

   Steps :

   - Open Terminal using **CTRL+ALT+T** or right-click and select **Open in Terminal**.
   - Once opened, type the following command in the terminal:

     ```
     cd  Desktop/
     ```

   [Before going through the next sequence of commands, make sure that both the TCP programs are located on the desktop]

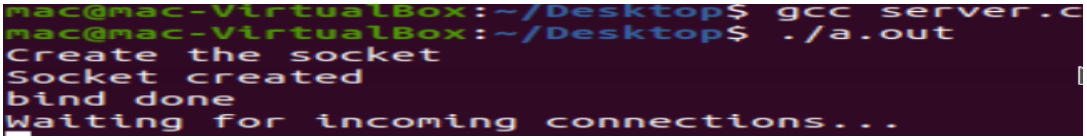
   - Before executing the program, open **Wireshark** and double-click on **any-interface** for packet capture process to start.
   - To compile the C program, type the following command in the same order as follows in the terminal:

     ```
     gcc  server.c
     ```

   - To execute the program type, object code file name in the terminal as follows :

     ```
     ./a.out
     ```

Network Programming Lab - Program Steps 4 and 5

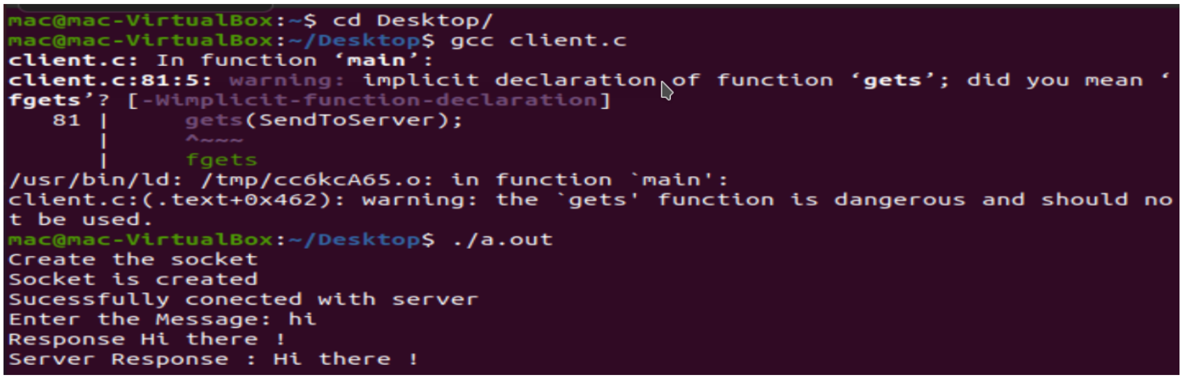


```
mac@mac-VirtualBox:~/Desktop$ gcc server.c
mac@mac-VirtualBox:~/Desktop$ ./a.out
Create the socket
Socket created
bind done
Waiting for incoming connections...
```

- Open another terminal and change its path to Desktop using Cd command and run the c code as shown below:

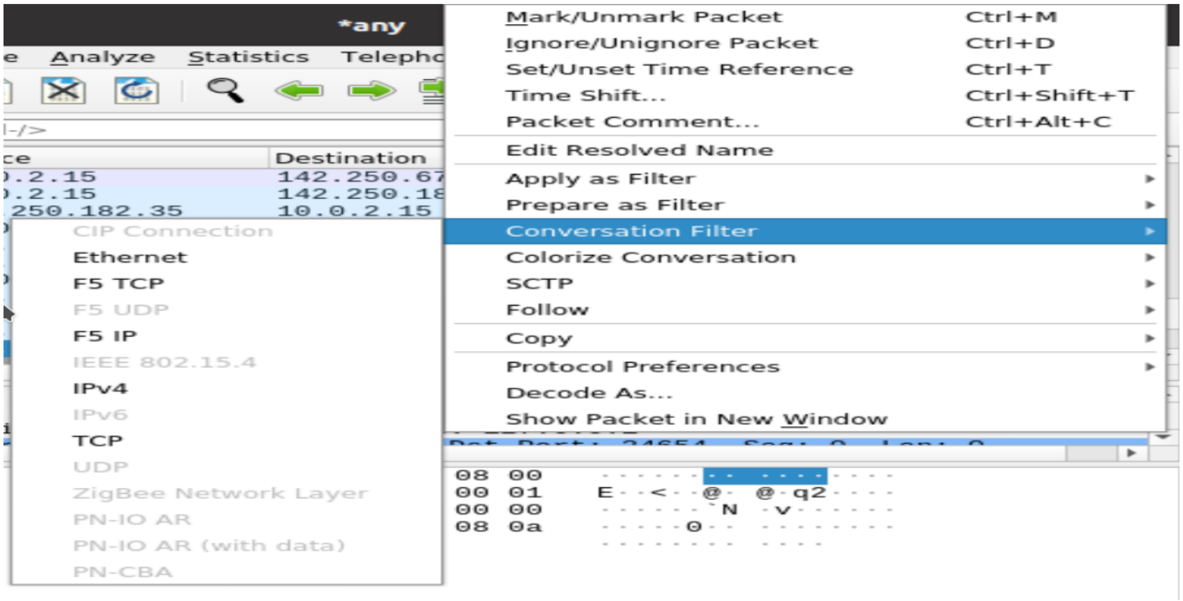
  cd Desktop/
  gcc client.c
  ./a.out

- Once the client program is executed, it will ask you to enter a message type **hi without any space** and hit enter.
- The server will replay back with **hi there!** Message.



```
mac@mac-VirtualBox:~$ cd Desktop/
mac@mac-VirtualBox:~/Desktop$ gcc client.c
client.c: In function ‘main’:
client.c:81:5: warning: implicit declaration of function ‘gets’; did you mean ‘
fgets’? [-Wimplicit-function-declaration]
   81 |     gets(SendToServer);
      |     ^~~~
      |     fgets
/usr/bin/ld: /tmp/cc6kcA65.o: in function `main':
client.c:(.text+0x462): warning: the `gets' function is dangerous and should no
t be used.
mac@mac-VirtualBox:~/Desktop$ ./a.out
Create the socket
Socket is created
Sucessfully conected with server
Enter the Message: hi
Response Hi there !
Server Response : Hi there !
```

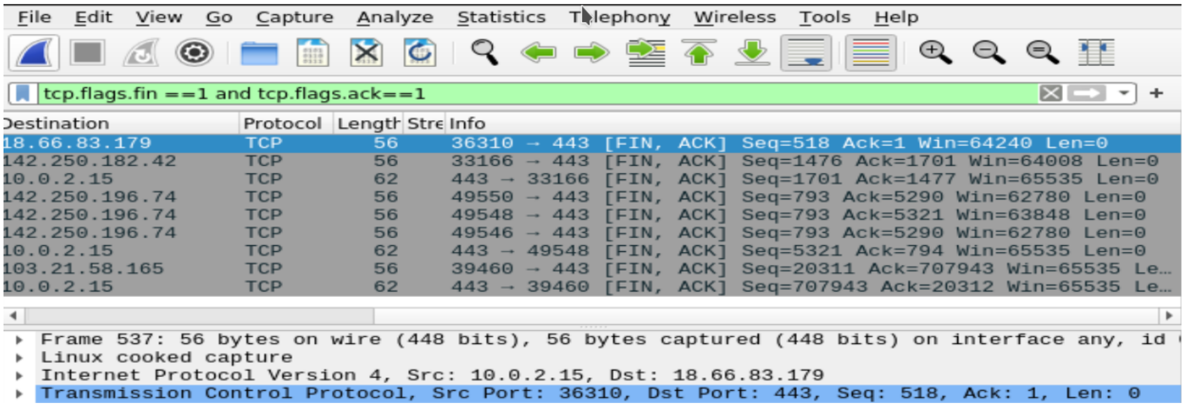
- Once the program execution is completed, stop the capturing process in Wireshark.
- In Wireshark, check for the packets that has the source and destination address as **127.0.0.1** and select any one of these packets, right-click and hover on conversation filter and select TCP.



- Once done analyze the TCP Packets.

2. **Capturing TCP Packets with browser :**
   Steps:
   - Open Wireshark and double-click on **any-interface** to start the packet capture process.
   - Open the browser and enter any website's fully qualified domain name in the browser address bar and hit enter.
   - After the site is fully loaded, stop the capturing process, in Wireshark.
   - Type the following in, apply a filter column and hit-enter :

     tcp.flags.fin==1 and tcp.flags.ack ==1


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp.flags.fin ==1 and tcp.flags.ack==1

Destination       Protocol Length Stre Info
18.66.83.179      TCP      56     36310 → 443 [FIN, ACK] Seq=518 Ack=1 Win=64240 Len=0
142.250.182.42    TCP      56     33166 → 443 [FIN, ACK] Seq=1476 Ack=1701 Win=64008 Len=0
10.0.2.15         TCP      62     443 → 33166 [FIN, ACK] Seq=1701 Ack=1477 Win=65535 Len=0
142.250.196.74    TCP      56     49550 → 443 [FIN, ACK] Seq=793 Ack=5290 Win=62780 Len=0
142.250.196.74    TCP      56     49548 → 443 [FIN, ACK] Seq=793 Ack=5321 Win=63848 Len=0
142.250.196.74    TCP      56     49546 → 443 [FIN, ACK] Seq=793 Ack=5290 Win=62780 Len=0
10.0.2.15         TCP      62     443 → 49548 [FIN, ACK] Seq=5321 Ack=794 Win=65535 Len=0
103.21.58.165     TCP      56     39460 → 443 [FIN, ACK] Seq=20311 Ack=707943 Win=65535 Le…
10.0.2.15         TCP      62     443 → 39460 [FIN, ACK] Seq=707943 Ack=20312 Win=65535 Le…

▸ Frame 537: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface any, id
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 18.66.83.179
▸ Transmission Control Protocol, Src Port: 36310, Dst Port: 443, Seq: 518, Ack: 1, Len: 0
```

   - Select any one of these listed packets, right-click and hover on conversation filter and select TCP.
   - Once done analyze the TCP Packets.