Ontario Tech UNIVERSITY

Computer Networks Project Report

Group 2

Parasjeet Marwah - 100787512
Sehaj Behl - 100748987
Sabeh Khalid - 100754735

Introduction

This project aims to create a virtual network environment using virtual machines (VMs). The first step involves setting up the environment to establish a successful connection between the VMs and the host through a virtual bridge. Wireshark is used to analyze packet details and traffic between the VMs.

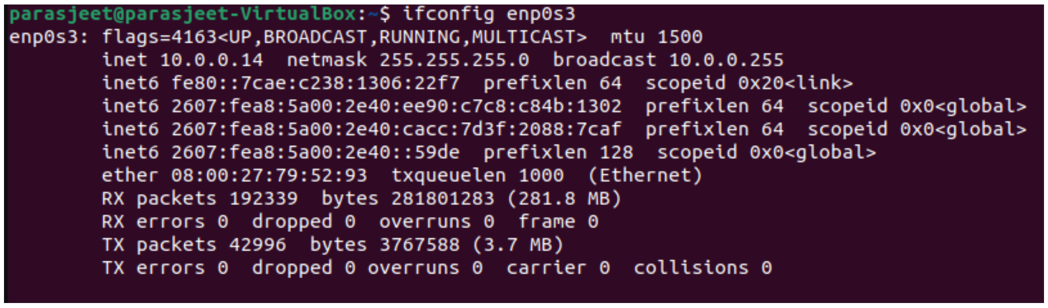
The second part of the project focuses on the TCP server and client, and their application. A "Hello" application will be written to establish a connection between the client and server. The third part involves another application task based on TCP, which requires a deep understanding of the protocol.

To conclude the project, we will learn about the UDP protocol and compare it to TCP. We will also study a Time Service and its relation to UDP.

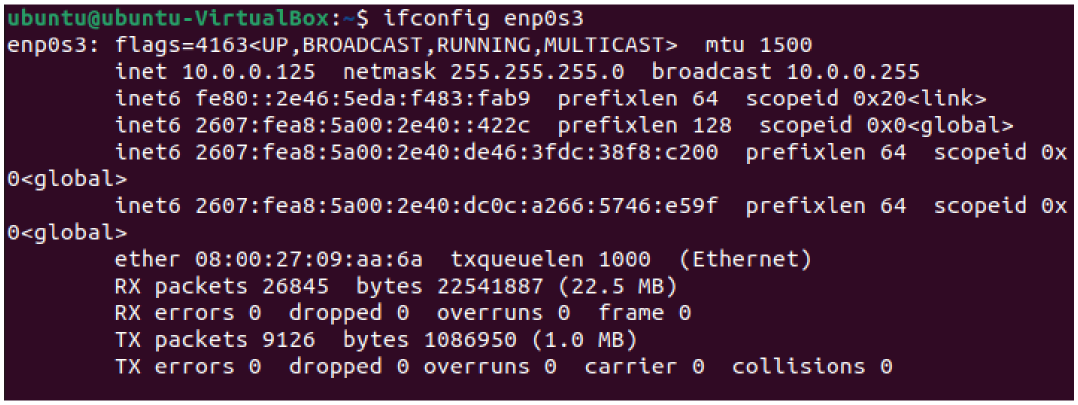
Part 1: Setup of VM's and Virtual Network Environment

1. Ifconfig commands showing the IP address of both VMs

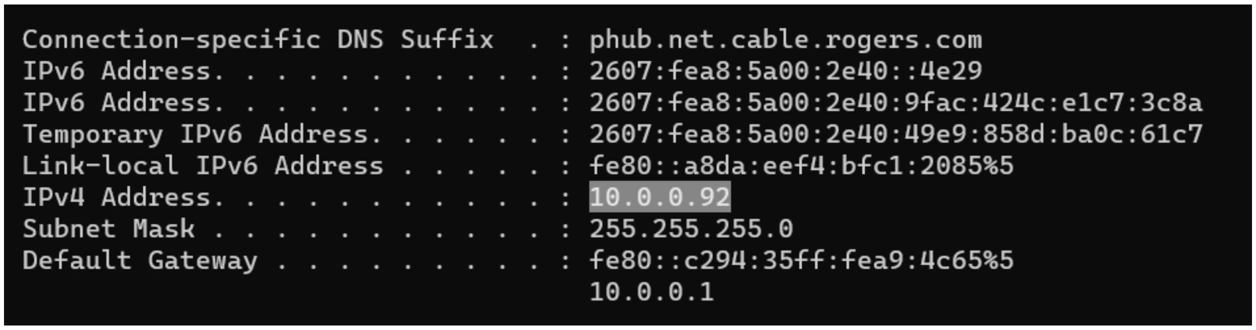
	IP Address
VM1	10.0.0.14
VM2	10.0.0.125
Host	10.0.0.92



```
parasjeet@parasjeet-VirtualBox:~$ ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.14  netmask 255.255.255.0  broadcast 10.0.0.255
        inet6 fe80::7cae:c238:1306:22f7  prefixlen 64  scopeid 0x20<link>
        inet6 2607:fea8:5a00:2e40:ee90:c7c8:c84b:1302  prefixlen 64  scopeid 0x0<global>
        inet6 2607:fea8:5a00:2e40:cacc:7d3f:2088:7caf  prefixlen 64  scopeid 0x0<global>
        inet6 2607:fea8:5a00:2e40::59de  prefixlen 128  scopeid 0x0<global>
        ether 08:00:27:79:52:93  txqueuelen 1000  (Ethernet)
        RX packets 192339  bytes 281801283 (281.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 42996  bytes 3767588 (3.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

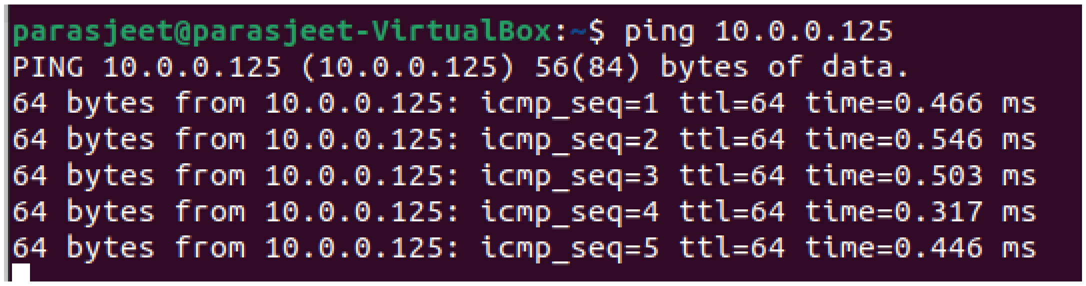


```
ubuntu@ubuntu-VirtualBox:~$ ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.125  netmask 255.255.255.0  broadcast 10.0.0.255
        inet6 fe80::2e46:5eda:f483:fab9  prefixlen 64  scopeid 0x20<link>
        inet6 2607:fea8:5a00:2e40::422c  prefixlen 128  scopeid 0x0<global>
        inet6 2607:fea8:5a00:2e40:de46:3fdc:38f8:c200  prefixlen 64  scopeid 0x
0<global>
        inet6 2607:fea8:5a00:2e40:dc0c:a266:5746:e59f  prefixlen 64  scopeid 0x
0<global>
        ether 08:00:27:09:aa:6a  txqueuelen 1000  (Ethernet)
        RX packets 26845  bytes 22541887 (22.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9126  bytes 1086950 (1.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

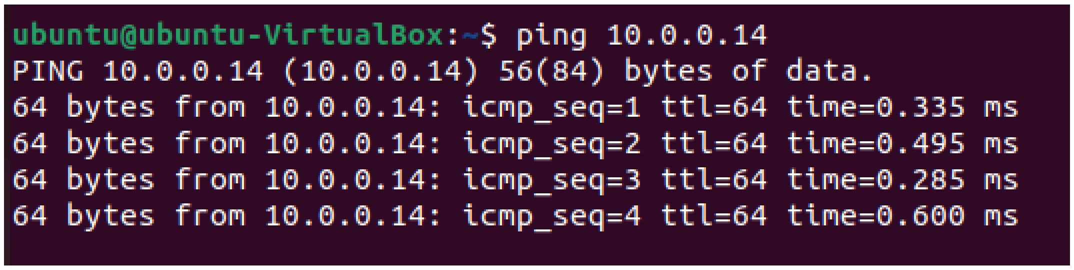


```
Connection-specific DNS Suffix  . : phub.net.cable.rogers.com
IPv6 Address. . . . . . . . . . . : 2607:fea8:5a00:2e40::4e29
IPv6 Address. . . . . . . . . . . : 2607:fea8:5a00:2e40:9fac:424c:e1c7:3c8a
Temporary IPv6 Address. . . . . . : 2607:fea8:5a00:2e40:49e9:858d:ba0c:61c7
Link-local IPv6 Address . . . . . : fe80::a8da:eef4:bfc1:2085%5
IPv4 Address. . . . . . . . . . . : 10.0.0.92
Subnet Mask . . . . . . . . . . . : 255.255.255.0
Default Gateway . . . . . . . . . : fe80::c294:35ff:fea9:4c65%5
                                    10.0.0.1
```

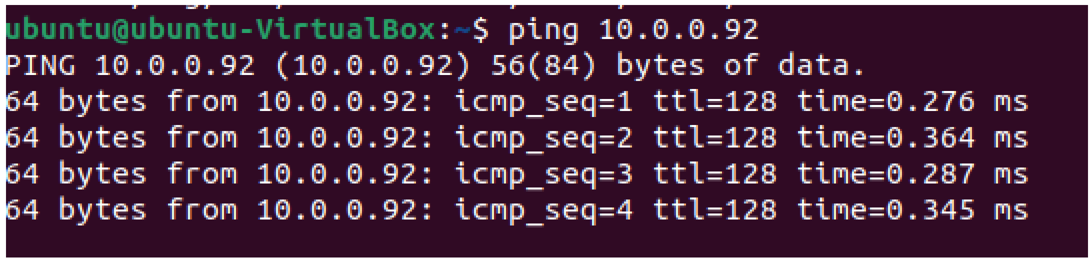
2. Pings are sent between VMs and between host and VM, and the connectivity is successful



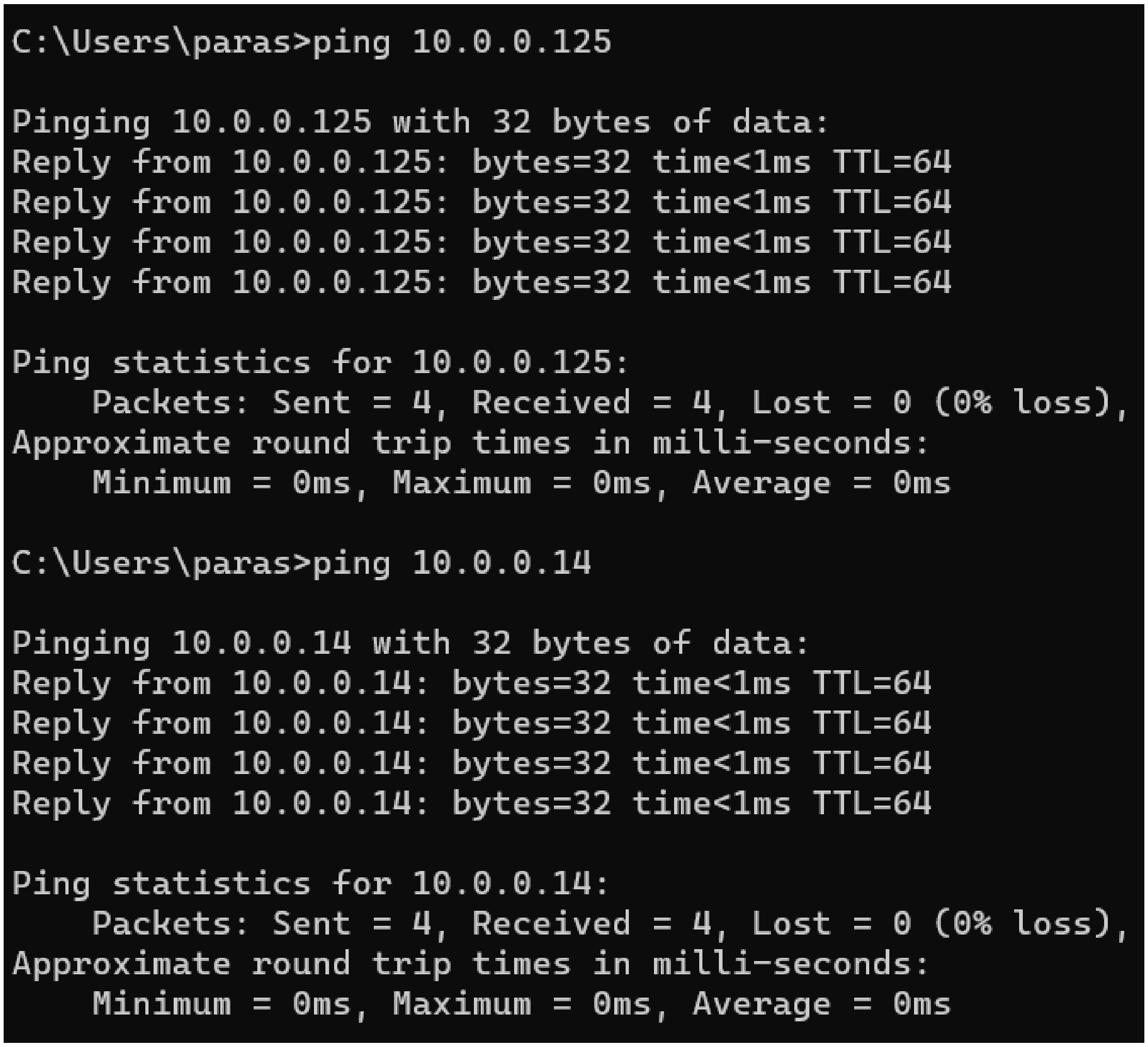
```
parasjeet@parasjeet-VirtualBox:~$ ping 10.0.0.125
PING 10.0.0.125 (10.0.0.125) 56(84) bytes of data.
64 bytes from 10.0.0.125: icmp_seq=1 ttl=64 time=0.466 ms
64 bytes from 10.0.0.125: icmp_seq=2 ttl=64 time=0.546 ms
64 bytes from 10.0.0.125: icmp_seq=3 ttl=64 time=0.503 ms
64 bytes from 10.0.0.125: icmp_seq=4 ttl=64 time=0.317 ms
64 bytes from 10.0.0.125: icmp_seq=5 ttl=64 time=0.446 ms
```



```
ubuntu@ubuntu-VirtualBox:~$ ping 10.0.0.14
PING 10.0.0.14 (10.0.0.14) 56(84) bytes of data.
64 bytes from 10.0.0.14: icmp_seq=1 ttl=64 time=0.335 ms
64 bytes from 10.0.0.14: icmp_seq=2 ttl=64 time=0.495 ms
64 bytes from 10.0.0.14: icmp_seq=3 ttl=64 time=0.285 ms
64 bytes from 10.0.0.14: icmp_seq=4 ttl=64 time=0.600 ms
```

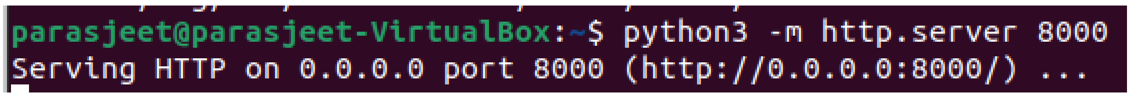


```
ubuntu@ubuntu-VirtualBox:~$ ping 10.0.0.92
PING 10.0.0.92 (10.0.0.92) 56(84) bytes of data.
64 bytes from 10.0.0.92: icmp_seq=1 ttl=128 time=0.276 ms
64 bytes from 10.0.0.92: icmp_seq=2 ttl=128 time=0.364 ms
64 bytes from 10.0.0.92: icmp_seq=3 ttl=128 time=0.287 ms
64 bytes from 10.0.0.92: icmp_seq=4 ttl=128 time=0.345 ms
```

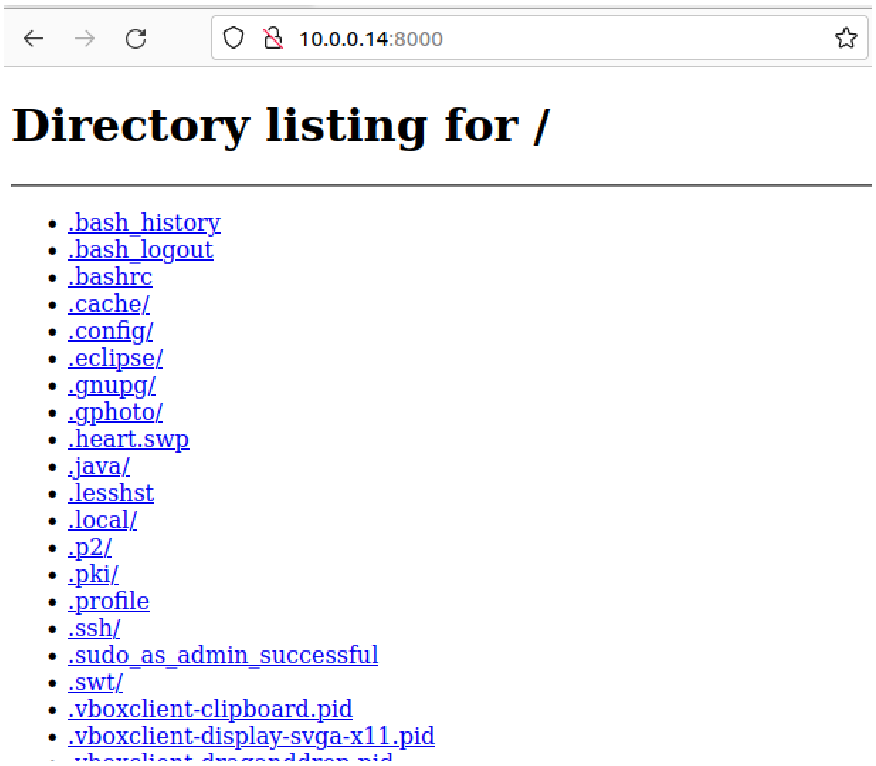


```
C:\Users\paras>ping 10.0.0.125

Pinging 10.0.0.125 with 32 bytes of data:
Reply from 10.0.0.125: bytes=32 time<1ms TTL=64
Reply from 10.0.0.125: bytes=32 time<1ms TTL=64
Reply from 10.0.0.125: bytes=32 time<1ms TTL=64
Reply from 10.0.0.125: bytes=32 time<1ms TTL=64

Ping statistics for 10.0.0.125:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\paras>ping 10.0.0.14

Pinging 10.0.0.14 with 32 bytes of data:
Reply from 10.0.0.14: bytes=32 time<1ms TTL=64
Reply from 10.0.0.14: bytes=32 time<1ms TTL=64
Reply from 10.0.0.14: bytes=32 time<1ms TTL=64
Reply from 10.0.0.14: bytes=32 time<1ms TTL=64

Ping statistics for 10.0.0.14:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

3. initialization of the web server using the ‘python3 -m http.server 8000’ command



```
parasjeet@parasjeet-VirtualBox:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

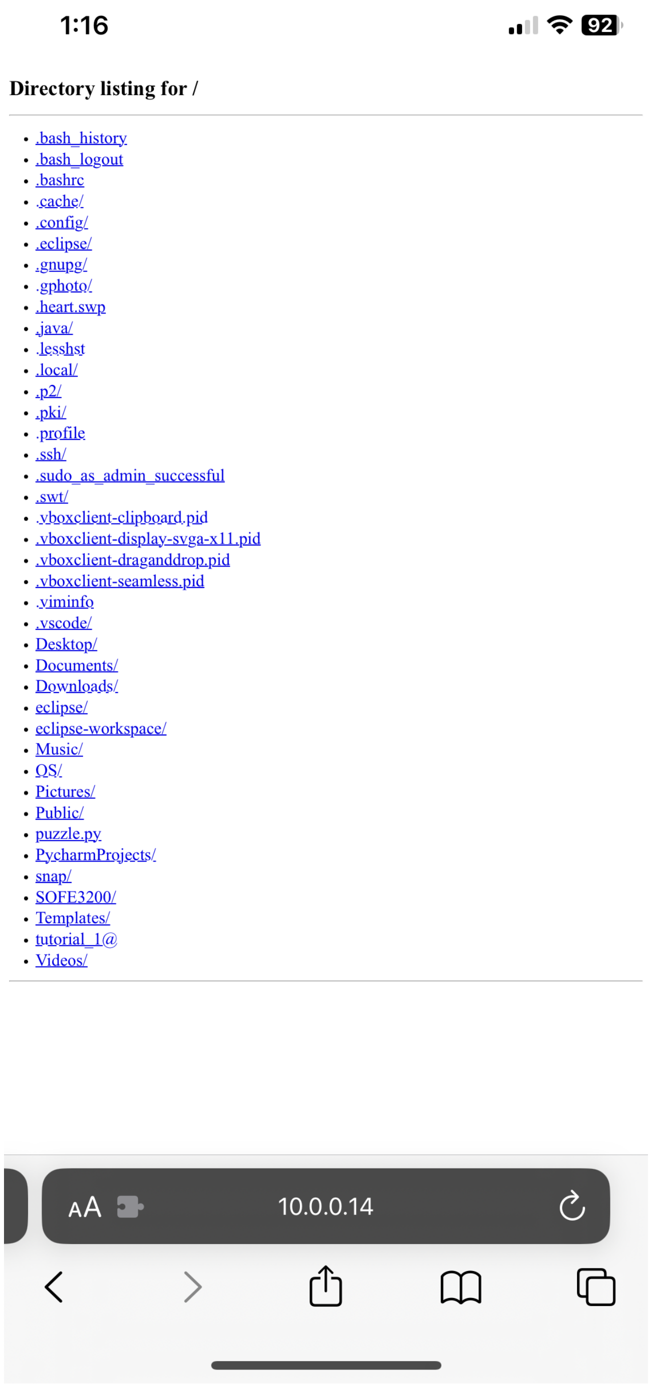


10.0.0.14:8000

Directory listing for /

- .bash_history
- .bash_logout
- .bashrc
- .cache/
- .config/
- .eclipse/
- .gnupg/
- .gphoto/
- .heart.swp
- .java/
- .lesshst
- .local/
- .p2/
- .pki/
- .profile
- .ssh/
- .sudo_as_admin_successful
- .swt/
- .vboxclient-clipboard.pid
- .vboxclient-display-svga-x11.pid

Demonstrated the Web server on host machine

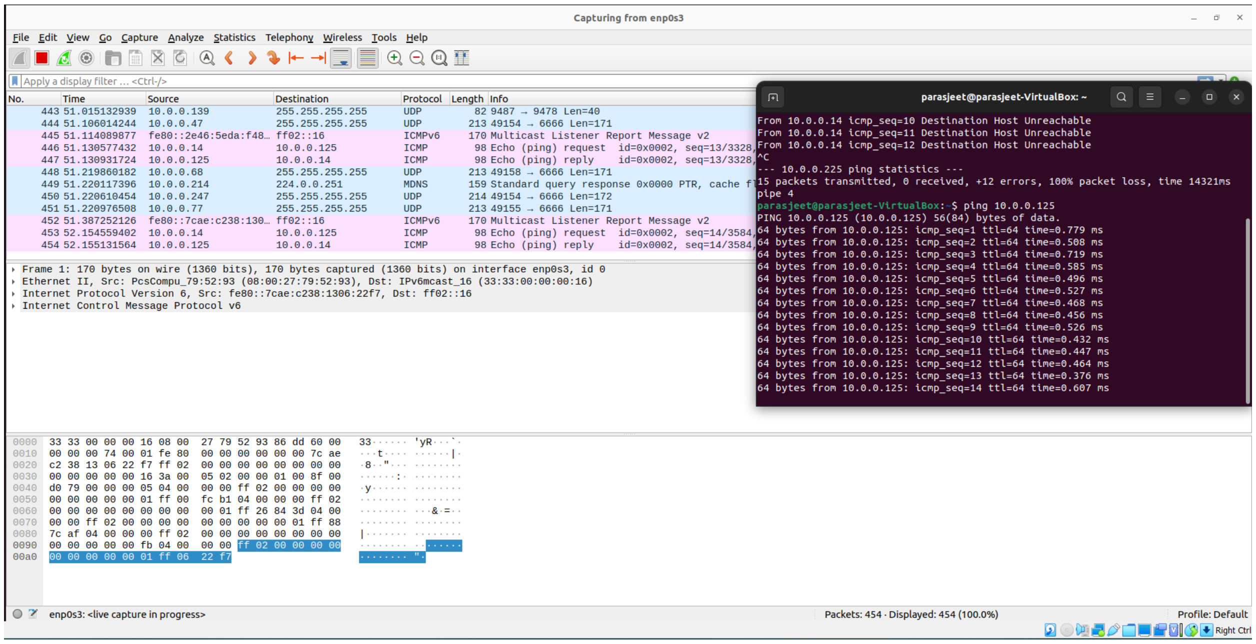

1:16

Directory listing for /

- .bash_history
- .bash_logout
- .bashrc
- .cache/
- .config/
- .eclipse/
- .gnupg/
- .gphoto/
- .heart.swp
- .java/
- .lesshst
- .local/
- .p2/
- .pki/
- .profile
- .ssh/
- .sudo_as_admin_successful
- .swt/
- .vboxclient-clipboard.pid
- .vboxclient-display-svga-x11.pid
- .vboxclient-draganddrop.pid
- .vboxclient-seamless.pid
- .viminfo
- .vscode/
- Desktop/
- Documents/
- Downloads/
- eclipse/
- eclipse-workspace/
- Music/
- OS/
- Pictures/
- Public/
- puzzle.py
- PycharmProjects/
- snap/
- SOFE3200/
- Templates/
- tutorial_1@
- Videos/

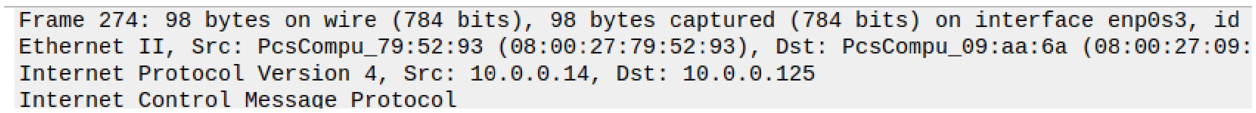
Demonstrated the Webserver on a smartphone

4. Wireshark captured data



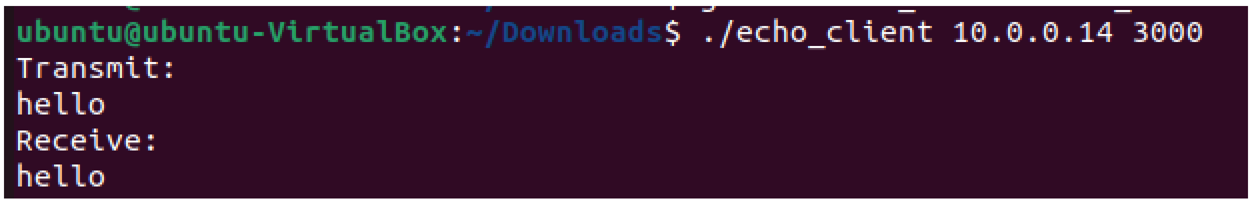
```
No.  Time          Source                   Destination      Protocol Length Info
 443 51.015132939  10.0.0.139               255.255.255.255  UDP        82 9487 → 9478 Len=40
 444 51.106014244  10.0.0.47                255.255.255.255  UDP       213 49154 → 6666 Len=171
 445 51.114089877  fe80::2e46:5eda:f48…     ff02::16         ICMPv6    170 Multicast Listener Report Message v2
 446 51.130577432  10.0.0.14                10.0.0.125       ICMP       98 Echo (ping) request  id=0x0002, seq=13/3328,
 447 51.130931724  10.0.0.125               10.0.0.14        ICMP       98 Echo (ping) reply    id=0x0002, seq=13/3328,
 448 51.219860182  10.0.0.68                255.255.255.255  UDP       213 49158 → 6666 Len=171
 449 51.220117396  10.0.0.214               224.0.0.251      MDNS      159 Standard query response 0x0000 PTR, cache fl
 450 51.220610454  10.0.0.247               255.255.255.255  UDP       214 49154 → 6666 Len=172
 451 51.220976508  10.0.0.77                255.255.255.255  UDP       213 49155 → 6666 Len=171
 452 51.387252126  fe80::7cae:c238:130…     ff02::16         ICMPv6    170 Multicast Listener Report Message v2
 453 52.154559402  10.0.0.14                10.0.0.125       ICMP       98 Echo (ping) request  id=0x0002, seq=14/3584,
 454 52.155131564  10.0.0.125               10.0.0.14        ICMP       98 Echo (ping) reply    id=0x0002, seq=14/3584,

› Frame 1: 170 bytes on wire (1360 bits), 170 bytes captured (1360 bits) on interface enp0s3, id 0
› Ethernet II, Src: PcsCompu_79:52:93 (08:00:27:79:52:93), Dst: IPv6mcast_16 (33:33:00:00:00:16)
› Internet Protocol Version 6, Src: fe80::7cae:c238:1306:22f7, Dst: ff02::16
› Internet Control Message Protocol v6

0000  33 33 00 00 00 16 08 00  27 79 52 93 86 dd 60 00   33······ 'yR···`·
0010  00 00 00 74 00 01 fe 80  00 00 00 00 00 00 7c ae   ···t···· ······|·
0020  c2 38 13 06 22 f7 ff 02  00 00 00 00 00 00 00 00   ·8··"··· ········
0030  00 00 00 00 00 16 3a 00  05 02 00 00 01 00 8f 00   ······:· ········
0040  d0 79 00 00 00 05 04 00  00 00 ff 02 00 00 00 00   ·y······ ········
0050  00 00 00 00 00 01 ff 00  fc b1 04 00 00 00 ff 02   ········ ········
0060  00 00 00 00 00 00 00 00  00 01 ff 26 84 3d 04 00   ········ ···&·=··
0070  00 00 ff 02 00 00 00 00  00 00 00 00 00 01 ff 88   ········ ········
0080  7c af 04 00 00 00 ff 02  00 00 00 00 00 00 00 00   |······· ········
0090  00 00 00 00 00 fb 04 00  00 00 ff 02 00 00 00 00   ········ ········
00a0  00 00 00 00 00 01 ff 06  22 f7                     ········ "·

enp0s3: <live capture in progress>        Packets: 454 · Displayed: 454 (100.0%)        Profile: Default
```

```
parasjeet@parasjeet-VirtualBox: ~
From 10.0.0.14 icmp_seq=10 Destination Host Unreachable
From 10.0.0.14 icmp_seq=11 Destination Host Unreachable
From 10.0.0.14 icmp_seq=12 Destination Host Unreachable
^C
--- 10.0.0.225 ping statistics ---
15 packets transmitted, 0 received, +12 errors, 100% packet loss, time 14321ms
pipe 4
parasjeet@parasjeet-VirtualBox:~$ ping 10.0.0.125
PING 10.0.0.125 (10.0.0.125) 56(84) bytes of data.
64 bytes from 10.0.0.125: icmp_seq=1 ttl=64 time=0.779 ms
64 bytes from 10.0.0.125: icmp_seq=2 ttl=64 time=0.508 ms
64 bytes from 10.0.0.125: icmp_seq=3 ttl=64 time=0.719 ms
64 bytes from 10.0.0.125: icmp_seq=4 ttl=64 time=0.585 ms
64 bytes from 10.0.0.125: icmp_seq=5 ttl=64 time=0.496 ms
64 bytes from 10.0.0.125: icmp_seq=6 ttl=64 time=0.527 ms
64 bytes from 10.0.0.125: icmp_seq=7 ttl=64 time=0.468 ms
64 bytes from 10.0.0.125: icmp_seq=8 ttl=64 time=0.456 ms
64 bytes from 10.0.0.125: icmp_seq=9 ttl=64 time=0.526 ms
64 bytes from 10.0.0.125: icmp_seq=10 ttl=64 time=0.432 ms
64 bytes from 10.0.0.125: icmp_seq=11 ttl=64 time=0.447 ms
64 bytes from 10.0.0.125: icmp_seq=12 ttl=64 time=0.464 ms
64 bytes from 10.0.0.125: icmp_seq=13 ttl=64 time=0.376 ms
64 bytes from 10.0.0.125: icmp_seq=14 ttl=64 time=0.607 ms
```

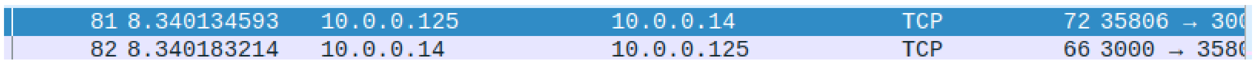


```
Frame 274: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp0s3, id
Ethernet II, Src: PcsCompu_79:52:93 (08:00:27:79:52:93), Dst: PcsCompu_09:aa:6a (08:00:27:09:
Internet Protocol Version 4, Src: 10.0.0.14, Dst: 10.0.0.125
Internet Control Message Protocol
```

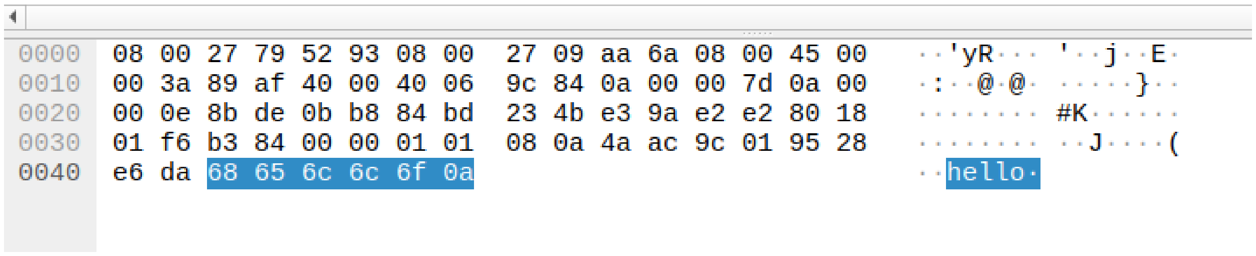
Part 2: TCP Server



```
ubuntu@ubuntu-VirtualBox:~/Downloads$ ./echo_client 10.0.0.14 3000
Transmit:
hello
Receive:
hello
```

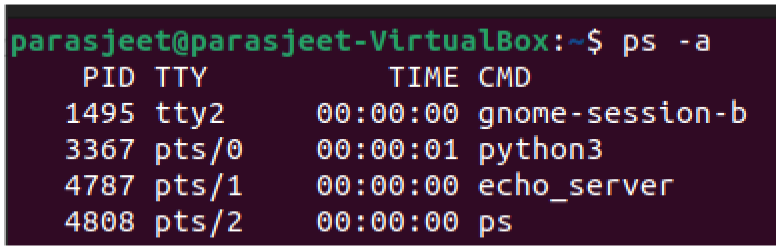


```
   81 8.340134593   10.0.0.125        10.0.0.14         TCP        72 35806 → 30
   82 8.340183214   10.0.0.14         10.0.0.125        TCP        66 3000 → 358
```



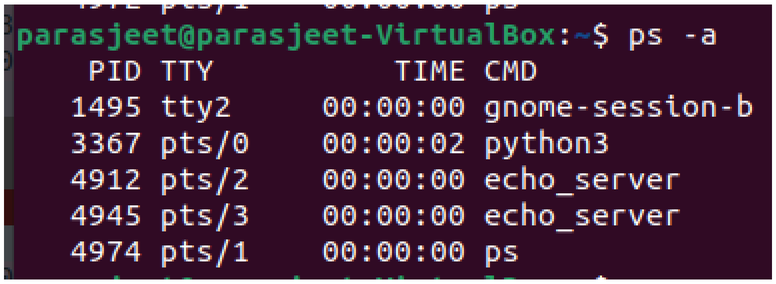
```
0000  08 00 27 79 52 93 08 00  27 09 aa 6a 08 00 45 00   ··'yR··· '··j··E·
0010  00 3a 89 af 40 00 40 06  9c 84 0a 00 00 7d 0a 00   ·:··@·@· ·····}··
0020  00 0e 8b de 0b b8 84 bd  23 4b e3 9a e2 e2 80 18   ········ #K······
0030  01 f6 b3 84 00 00 01 01  08 0a 4a ac 9c 01 95 28   ········ ··J····(
0040  e6 da 68 65 6c 6c 6f 0a                            ··hello·
```

6. One echo server process



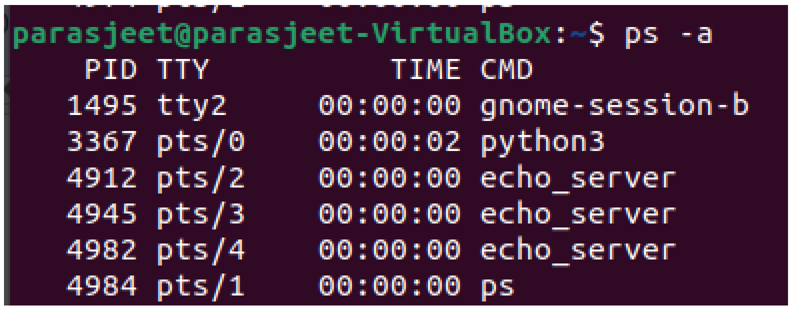
```
parasjeet@parasjeet-VirtualBox:~$ ps -a
    PID TTY          TIME CMD
   1495 tty2     00:00:00 gnome-session-b
   3367 pts/0    00:00:01 python3
   4787 pts/1    00:00:00 echo_server
   4808 pts/2    00:00:00 ps
```

8. Two echo servers processes



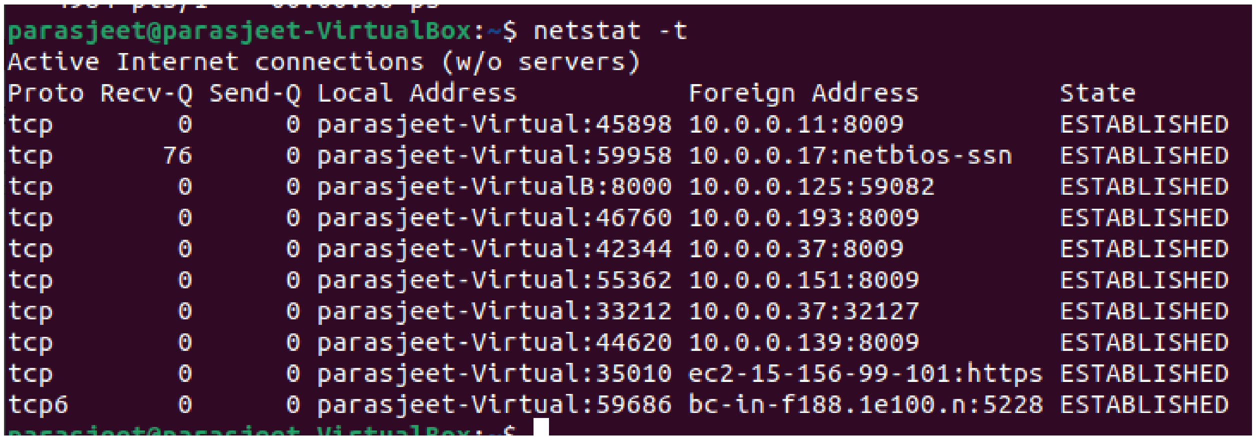
```
parasjeet@parasjeet-VirtualBox:~$ ps -a
    PID TTY          TIME CMD
   1495 tty2     00:00:00 gnome-session-b
   3367 pts/0    00:00:02 python3
   4912 pts/2    00:00:00 echo_server
   4945 pts/3    00:00:00 echo_server
   4974 pts/1    00:00:00 ps
```

9. After starting another client on VM2, there are three echo_server processes



```
parasjeet@parasjeet-VirtualBox:~$ ps -a
    PID TTY          TIME CMD
   1495 tty2     00:00:00 gnome-session-b
   3367 pts/0    00:00:02 python3
   4912 pts/2    00:00:00 echo_server
   4945 pts/3    00:00:00 echo_server
   4982 pts/4    00:00:00 echo_server
   4984 pts/1    00:00:00 ps
```

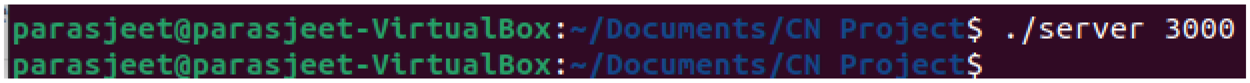
10. TCP connections using netstat -t command



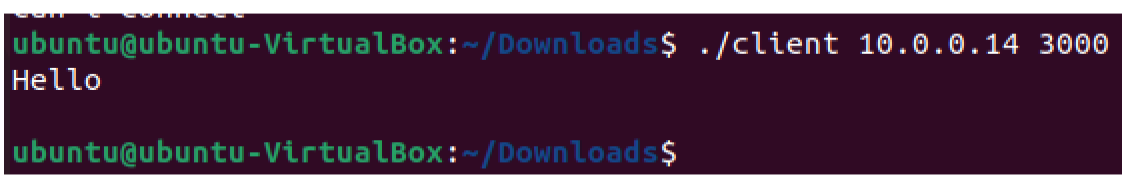
```
parasjeet@parasjeet-VirtualBox:~$ netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 parasjeet-Virtual:45898 10.0.0.11:8009          ESTABLISHED
tcp       76      0 parasjeet-Virtual:59958 10.0.0.17:netbios-ssn   ESTABLISHED
tcp        0      0 parasjeet-VirtualB:8000 10.0.0.125:59082        ESTABLISHED
tcp        0      0 parasjeet-Virtual:46760 10.0.0.193:8009         ESTABLISHED
tcp        0      0 parasjeet-Virtual:42344 10.0.0.37:8009          ESTABLISHED
tcp        0      0 parasjeet-Virtual:55362 10.0.0.151:8009         ESTABLISHED
tcp        0      0 parasjeet-Virtual:33212 10.0.0.37:32127         ESTABLISHED
tcp        0      0 parasjeet-Virtual:44620 10.0.0.139:8009         ESTABLISHED
tcp        0      0 parasjeet-Virtual:35010 ec2-15-156-99-101:https ESTABLISHED
tcp6       0      0 parasjeet-Virtual:59686 bc-in-f188.1e100.n:5228 ESTABLISHED
```

Part 3 : File Download Application based on TCP

Both the client and server successfully running, server sends a Hello message, client waits and displays that message

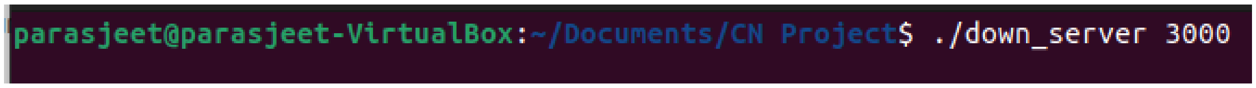


```
parasjeet@parasjeet-VirtualBox:~/Documents/CN Project$ ./server 3000
parasjeet@parasjeet-VirtualBox:~/Documents/CN Project$
```



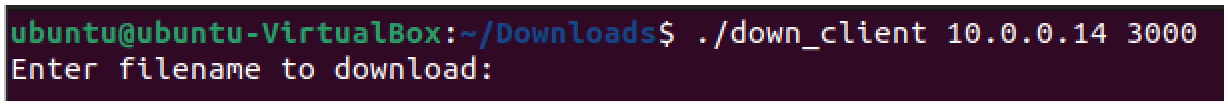
```
ubuntu@ubuntu-VirtualBox:~/Downloads$ ./client 10.0.0.14 3000
Hello

ubuntu@ubuntu-VirtualBox:~/Downloads$
```

TCP Download Application



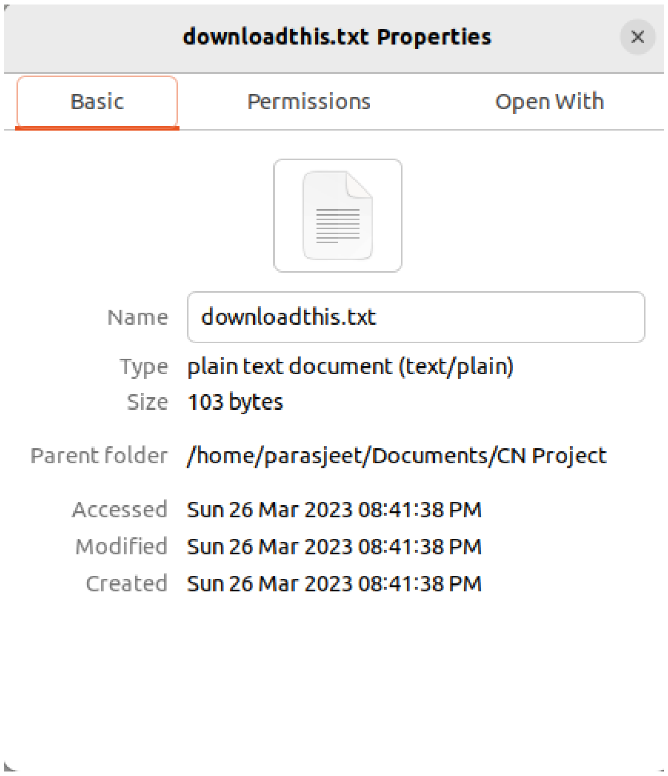
```
parasjeet@parasjeet-VirtualBox:~/Documents/CN Project$ ./down_server 3000
```

Starting the server for file downloading applications



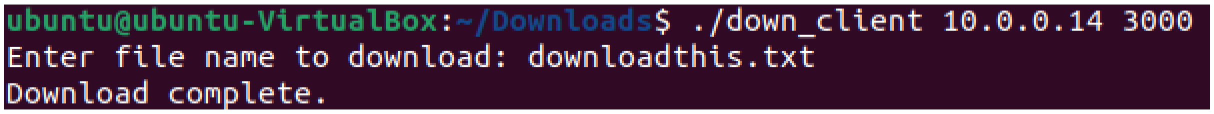
```
ubuntu@ubuntu-VirtualBox:~/Downloads$ ./down_client 10.0.0.14 3000
Enter filename to download:
```

Start of client-side with a prompt of which file the user would like to download



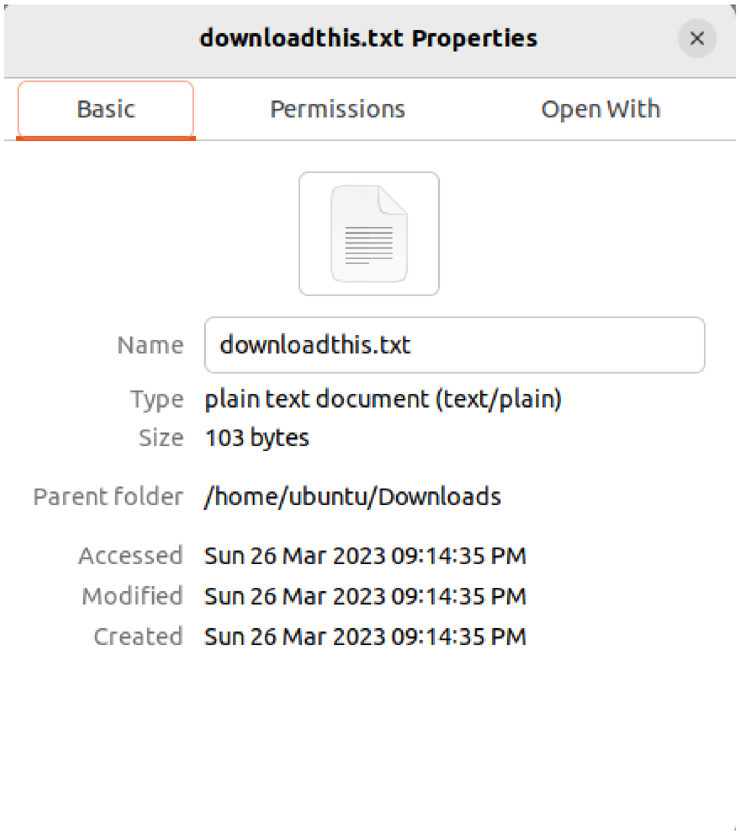
Name	downloadthis.txt
Type	plain text document (text/plain)
Size	103 bytes
Parent folder	/home/parasjeet/Documents/CN Project
Accessed	Sun 26 Mar 2023 08:41:38 PM
Modified	Sun 26 Mar 2023 08:41:38 PM
Created	Sun 26 Mar 2023 08:41:38 PM

File to be downloaded from VM1



```
ubuntu@ubuntu-VirtualBox:~/Downloads$ ./down_client 10.0.0.14 3000
Enter file name to download: downloadthis.txt
Download complete.
```

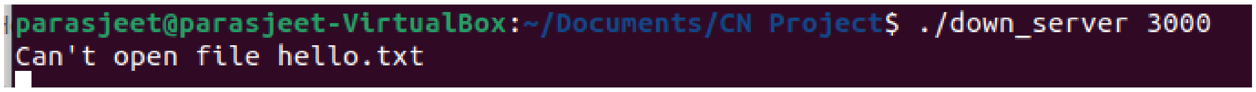
Download successful from VM2


downloadthis.txt Properties

Basic | Permissions | Open With

Name	downloadthis.txt
Type	plain text document (text/plain)
Size	103 bytes
Parent folder	/home/ubuntu/Downloads
Accessed	Sun 26 Mar 2023 09:14:35 PM
Modified	Sun 26 Mar 2023 09:14:35 PM
Created	Sun 26 Mar 2023 09:14:35 PM

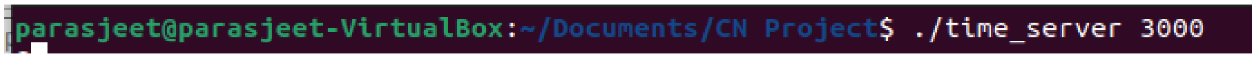
Downloaded file from VM2



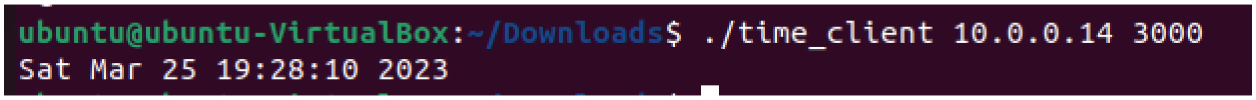
```
parasjeet@parasjeet-VirtualBox:~/Documents/CN Project$ ./down_server 3000
Can't open file hello.txt
```

Error message if file cannot be found

Part 4: UDP Server Implementation

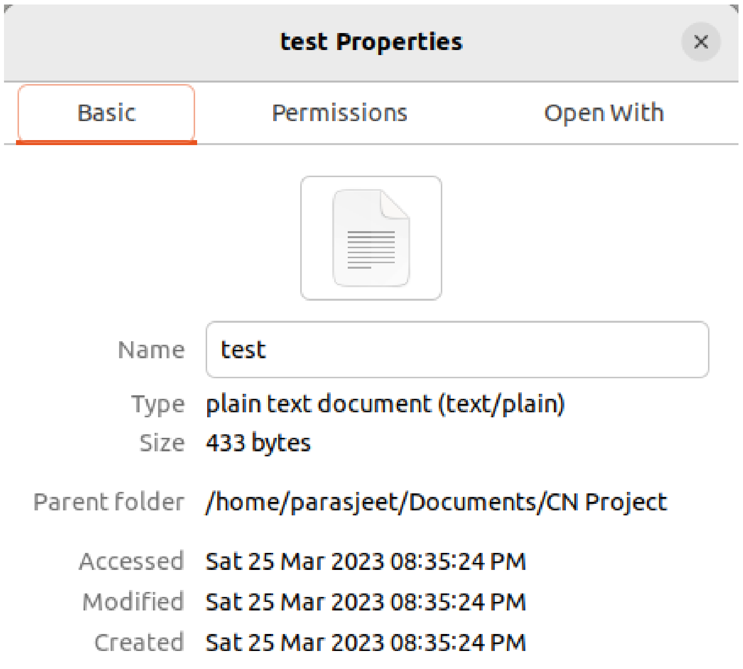


```
parasjeet@parasjeet-VirtualBox:~/Documents/CN Project$ ./time_server 3000
```



```
ubuntu@ubuntu-VirtualBox:~/Downloads$ ./time_client 10.0.0.14 3000
Sat Mar 25 19:28:10 2023
```

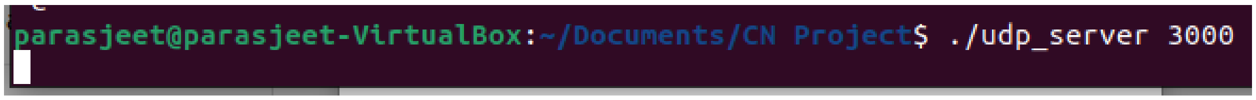
Start up of time server and client on both VM's. As shown above, on the client side the current date and time is being displayed


test Properties

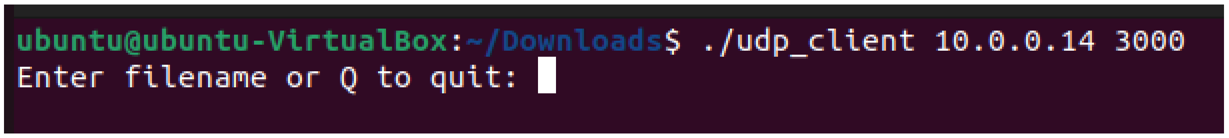
Basic | Permissions | Open With

Name	test
Type	plain text document (text/plain)
Size	433 bytes
Parent folder	/home/parasjeet/Documents/CN Project
Accessed	Sat 25 Mar 2023 08:35:24 PM
Modified	Sat 25 Mar 2023 08:35:24 PM
Created	Sat 25 Mar 2023 08:35:24 PM

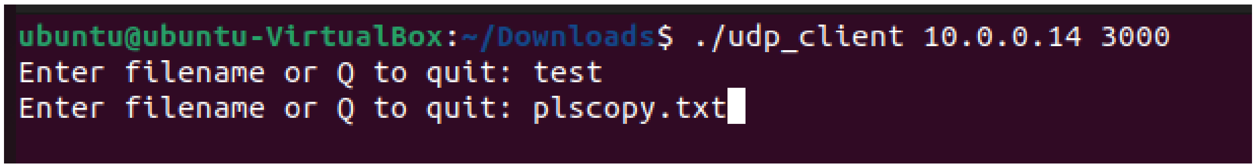
File to be downloaded from VM1


```
parasjeet@parasjeet-VirtualBox:~/Documents/CN Project$ ./udp_server 3000
```

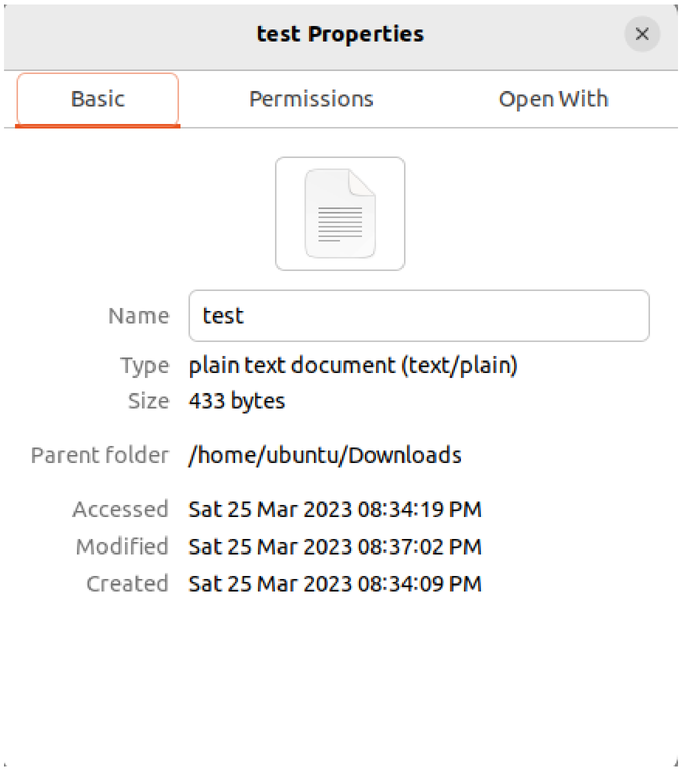
Start of UDP connection on server side


```
ubuntu@ubuntu-VirtualBox:~/Downloads$ ./udp_client 10.0.0.14 3000
Enter filename or Q to quit: 
```

Start of the UDP client, the user is prompted to enter in a file name or exit the program


```
ubuntu@ubuntu-VirtualBox:~/Downloads$ ./udp_client 10.0.0.14 3000
Enter filename or Q to quit: test
Enter filename or Q to quit: plscopy.txt
```

The user is able to enter in multiple files to be downloaded and are stored onto the local storage of VM2

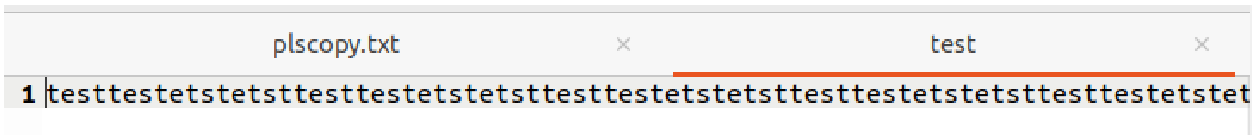


test Properties

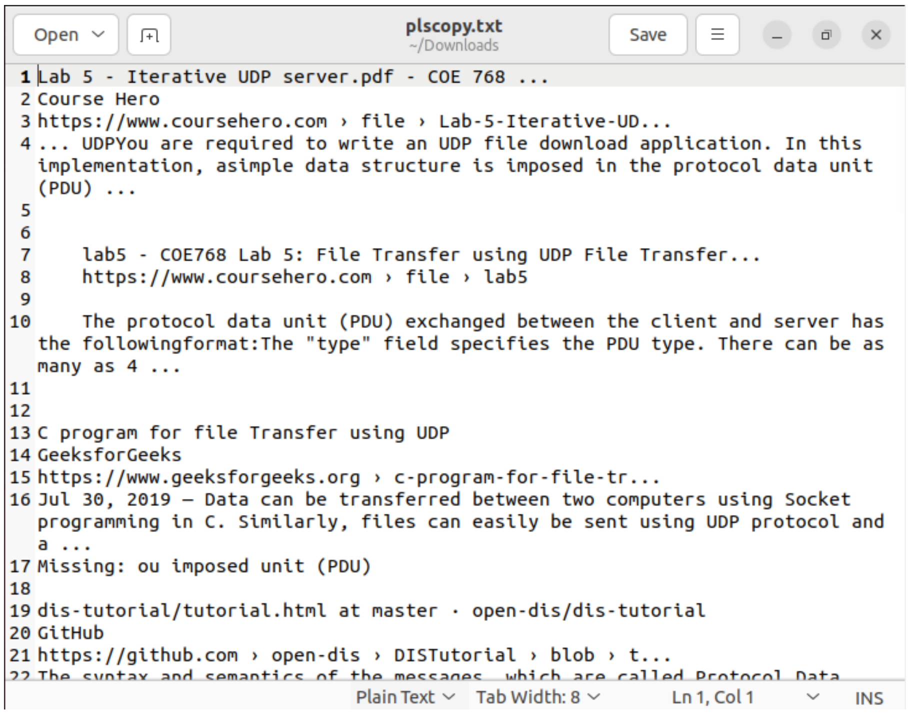
Basic | Permissions | Open With

Name: test
Type: plain text document (text/plain)
Size: 433 bytes
Parent folder: /home/ubuntu/Downloads
Accessed: Sat 25 Mar 2023 08:34:19 PM
Modified: Sat 25 Mar 2023 08:37:02 PM
Created: Sat 25 Mar 2023 08:34:09 PM

File download onto VM2



```
testtestetstetsttesttestetstetsttesttestetstetsttesttestetstetsttesttestetstet
```



plscopy.txt ~/Downloads

```
Lab 5 - Iterative UDP server.pdf - COE 768 ...
Course Hero
https://www.coursehero.com › file › Lab-5-Iterative-UD...
... UDPYou are required to write an UDP file download application. In this
implementation, asimple data structure is imposed in the protocol data unit
(PDU) ...


    lab5 - COE768 Lab 5: File Transfer using UDP File Transfer...
    https://www.coursehero.com › file › lab5

    The protocol data unit (PDU) exchanged between the client and server has
the followingformat:The "type" field specifies the PDU type. There can be as
many as 4 ...


C program for file Transfer using UDP
GeeksforGeeks
https://www.geeksforgeeks.org › c-program-for-file-tr...
Jul 30, 2019 — Data can be transferred between two computers using Socket
programming in C. Similarly, files can easily be sent using UDP protocol and
a ...
Missing: ou imposed unit (PDU)

dis-tutorial/tutorial.html at master · open-dis/dis-tutorial
GitHub
https://github.com › open-dis › DISTutorial › blob › t...
The syntax and semantics of the messages, which are called Protocol Data
```

Plain Text · Tab Width: 8 · Ln 1, Col 1 · INS

Contents of downloaded files from VM1

3. **The Time service can run on both UDP and TCP transport protocols. Discuss which transport protocol is more appropriate?**

Even though time service can run on UDP and TCP transport protocols, one of them is more efficient. Specifically, TCP is more efficient since it makes sure that the process of delivering packets is much more stable. This is because TCP provides much higher data reliability and integrity. Also, TCP makes sure that the data packets are being delivered in the same order in which they were received.

On the other hand, UDP is a connectionless protocol which means that the transmission of data packets is not consistent. Also, the order of the packets being delivered in UDP is not consistent. However, UDP is a better option in time served as the service only needs to send the current time to the client. This is because even if some packets are lost the client can just request for them again. Also, UDP is better as it sends packets with lower latency. UDP is better in this situation because we know that accuracy is not as important so we don't have to go with TCP.

4. **Discuss if the Time server should be designed as a concurrent server or non-concurrent server**

A concurrent server is able to handle multiple client requests simultaneously, while a non-concurrent is only able to handle one client request at a time. Therefore, the time server is designed a concurrent server or non-concurrent is dependent on the number of requests received by clients, if many are received, concurrent seems appropriate. Otherwise, it can be implemented using a non-concurrent. In general, using a concurrent server may be beneficial due to improved performance and faster response time when dealing with lots of requests.

7. **By examining the captured data, you probably find that the first message was sent by the client. This message contained some random message that was ignored by the server. What is the function of this message?**

After analyzing the wireshark data, we can see that the first message was indeed sent by the client. This message is essentially a placeholder to start the connection between the server and client. This message is actually not considered by the server and is disregarded. Instead, the server responds to the client with the current time. Basically, the function of this message is only to create a channel between the client and server where they can talk to each other, without having any setup earlier.

Conclusion

To conclude, we built and used two virtual machines in order to conduct a series of operations in a virtual setting. This network allowed us to utilize socket programming fundamentals to create and operate UDP and TCP server connections. We were able to first set up the virtual machines and ensure that they were functioning correctly with the use of TCP/IP. Furthermore, we installed the network analyzer wireshark and used it to capture packets. Also, we used wireshark to analyze the creation procedures and termination procedures of the TCP connection. The file download application based on TCP was also explained and the process of data transfer from the file and client was also demonstrated. The construction of the UDP server was also shown as well as the file download application based on UDP. Overall, we demonstrated a thorough understanding of building TCP and UDP servers as well as applications based on them in a virtual network setting.