Department of Computer Science

CS2005 Networks & Operating Systems Task 1

Academic Year 2023-24

<Shofiul hasan>

<Student ID 2156039>

Table of Contents

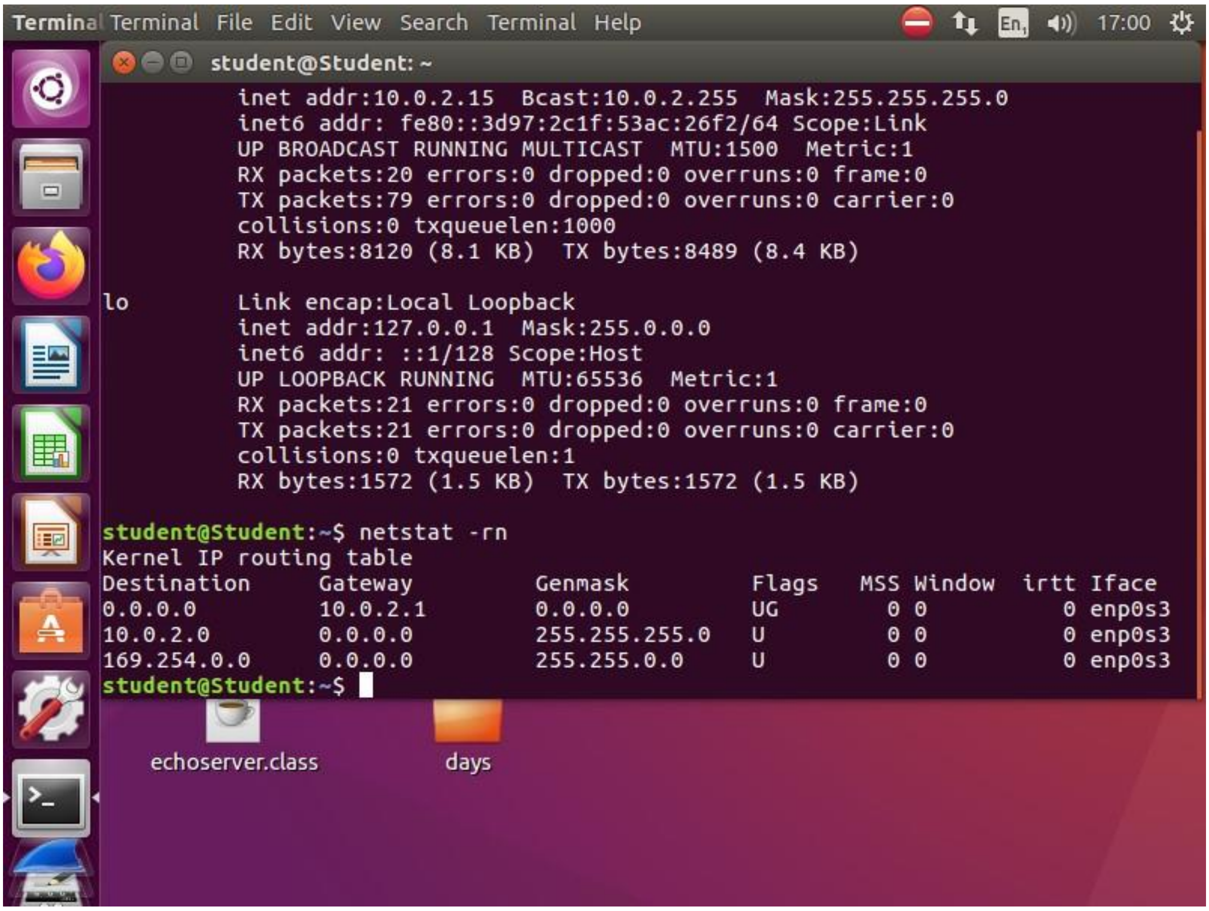
1. Introduction

In this report, I will commence by detailing various configuration tests performed on a network comprising two subnets connected through a router. This segment serves to demonstrate the proper functioning of the network through the execution of diverse commands on the hosts' terminals, such as ifconfig, netstat, ping, and Wireshark. Subsequently, I will document the protocols employed by calcclient and calcServer, utilizing a protocol table. The third section provides documentation for the protocols associated with calcClientNew and calcServerNew2, denoting the updated software version and addressing any associated issues. Lastly, a concise report will be presented to the managers at NOSsoft, outlining multiple problems identified with their updated software and urging them to address these issues promptly.

2. Test Network Docume

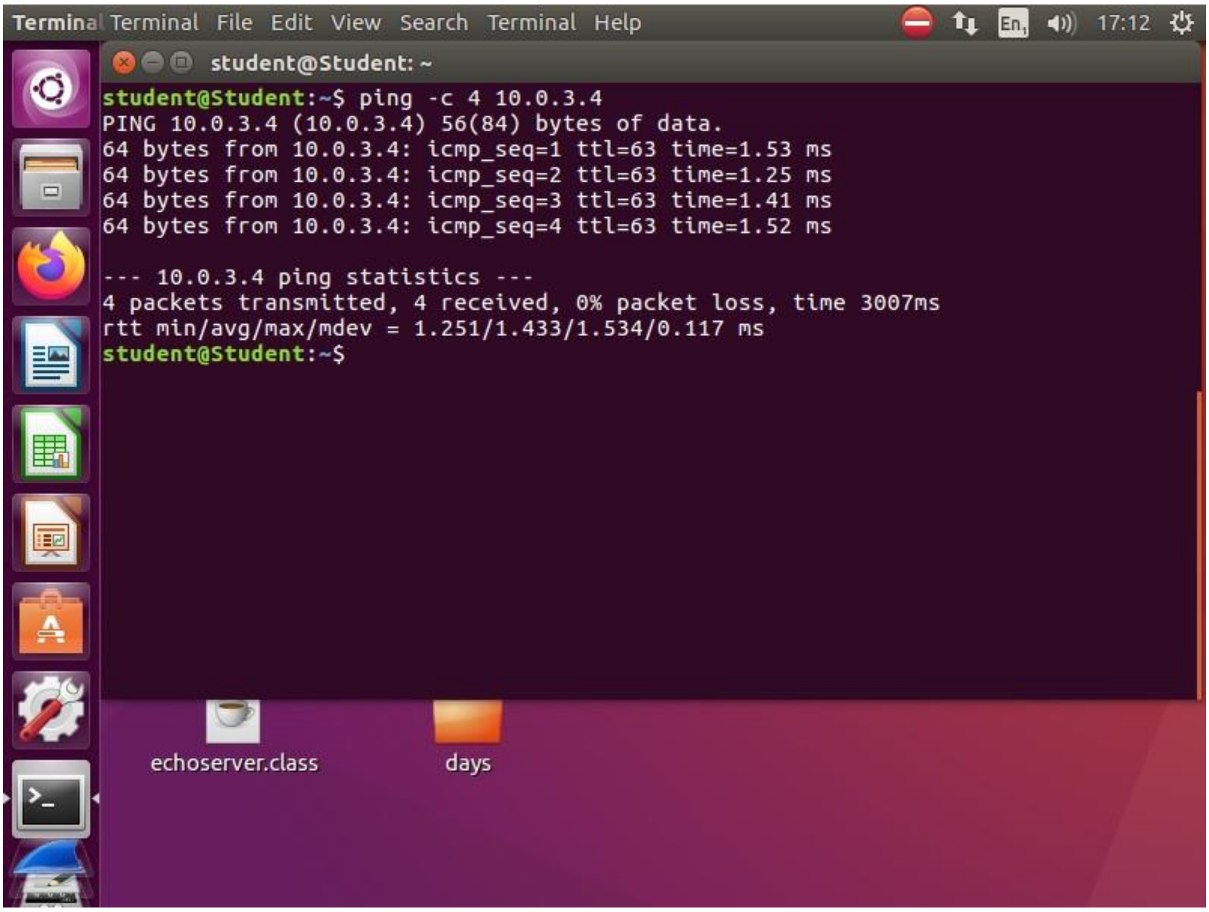
This section I am working on prove that my network is funcitioning correctly.

2.1 Ubuntu1

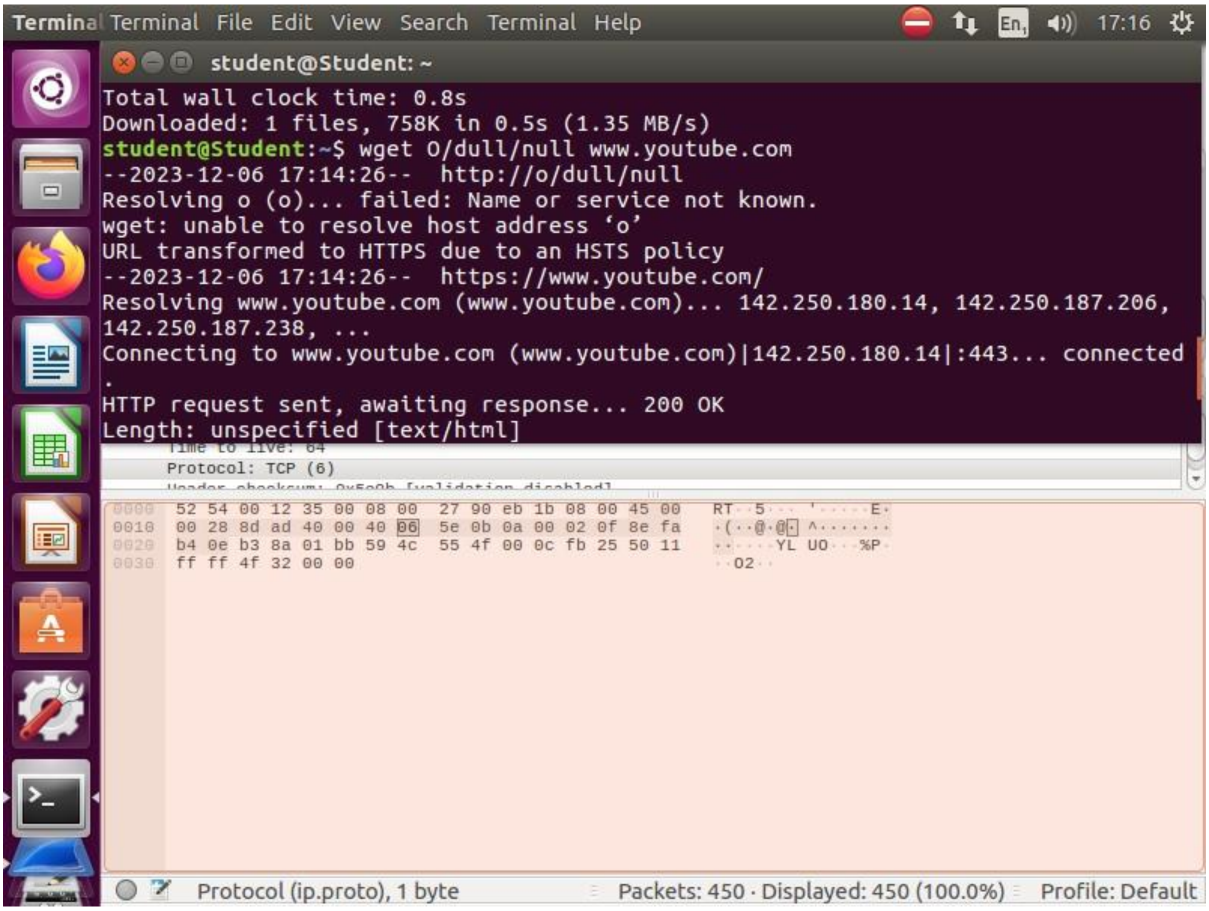


```
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::3d97:2c1f:53ac:26f2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:20 errors:0 dropped:0 overruns:0 frame:0
          TX packets:79 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8120 (8.1 KB)  TX bytes:8489 (8.4 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:21 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:1572 (1.5 KB)  TX bytes:1572 (1.5 KB)

student@Student:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.0.2.1        0.0.0.0         UG        0 0          0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 enp0s3
student@Student:~$
```

In the provided screenshots, I utilized the "ifconfig" command in the Ubuntu1 terminal to verify the IP address. Following this, I examined the connectivity between Ubuntu1 and Ubuntu2 using the "netstat -rn" command to identify the destination and gateway. Subsequently, I established a route leading to Ubuntu2, the intended destination, with the command "sudo route add -net 10.0.3.0/24 gw 10.0.2.254". To confirm the successful addition of the route, I performed a route check using the "ping -c 4 10.0.3.4" command, resulting in 4 packets transmitted, 4 received, and 0 loss. This outcome indicates that the route between Ubuntu1 and Ubuntu2 has been correctly added, and all transmitted packets were successfully received.



```
student@Student:~$ ping -c 4 10.0.3.4
PING 10.0.3.4 (10.0.3.4) 56(84) bytes of data.
64 bytes from 10.0.3.4: icmp_seq=1 ttl=63 time=1.53 ms
64 bytes from 10.0.3.4: icmp_seq=2 ttl=63 time=1.25 ms
64 bytes from 10.0.3.4: icmp_seq=3 ttl=63 time=1.41 ms
64 bytes from 10.0.3.4: icmp_seq=4 ttl=63 time=1.52 ms

--- 10.0.3.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 1.251/1.433/1.534/0.117 ms
student@Student:~$
```

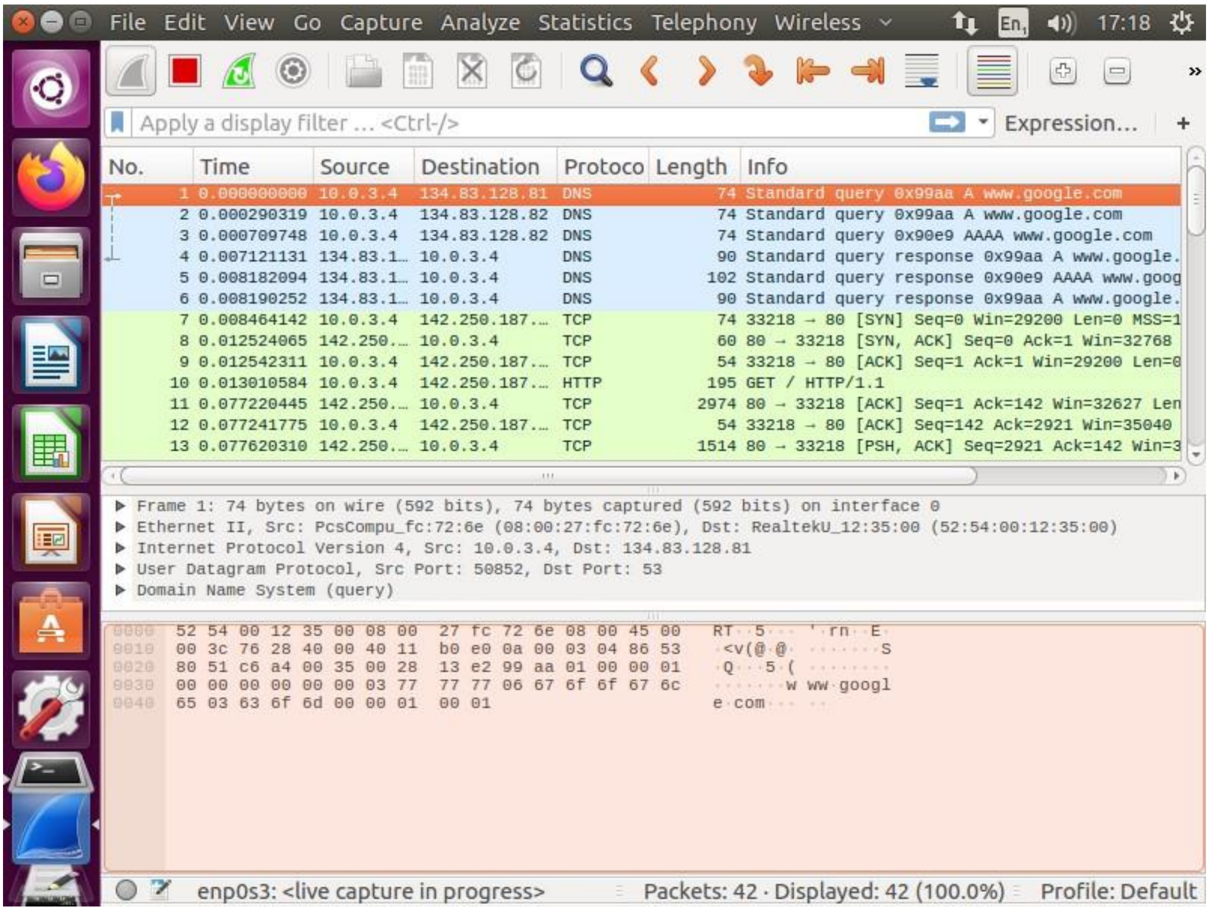


Terminal Terminal File Edit View Search Terminal Help
17:16
student@Student: ~
Protocol (ip.proto), 1 byte
Packets: 450 · Displayed: 450 (100.0%)
Profile: Default

```
Total wall clock time: 0.8s
Downloaded: 1 files, 758K in 0.5s (1.35 MB/s)
student@Student:~$ wget O/dull/null www.youtube.com
--2023-12-06 17:14:26--  http://o/dull/null
Resolving o (o)... failed: Name or service not known.
wget: unable to resolve host address ‘o’
URL transformed to HTTPS due to an HSTS policy
--2023-12-06 17:14:26--  https://www.youtube.com/
Resolving www.youtube.com (www.youtube.com)... 142.250.180.14, 142.250.187.206,
142.250.187.238, ...
Connecting to www.youtube.com (www.youtube.com)|142.250.180.14|:443... connected
.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
```

```
    Time to live: 64
    Protocol: TCP (6)
    Header checksum: 0x5e0b [validation disabled]

0000  52 54 00 12 35 00 08 00  27 90 eb 1b 08 00 45 00   RT··5···'·····E·
0010  00 28 8d ad 40 00 40 06  5e 0b 0a 00 02 0f 8e fa   ·(··@·@·^·······
0020  b4 0e b3 8a 01 bb 59 4c  55 4f 00 0c fb 25 50 11   ······YLUO···%P·
0030  ff ff 4f 32 00 00                                  ··O2··
```

In the above screenshot, I started wireshark then treminal of ubuntu called the website of youtube web-sever using the com-mand “wget -o/dev/null www.youtube.com which it sent a request to the youtube server.



File Edit View Go Capture Analyze Statistics Telephony Wireless
17:18
Apply a display filter … <Ctrl-/>
Expression…

```
No.  Time         Source        Destination    Protoco Length Info
   1 0.000000000 10.0.3.4      134.83.128.81  DNS        74 Standard query 0x99aa A www.google.com
   2 0.000290319 10.0.3.4      134.83.128.82  DNS        74 Standard query 0x99aa A www.google.com
   3 0.000709748 10.0.3.4      134.83.128.82  DNS        74 Standard query 0x90e9 AAAA www.google.com
   4 0.007121131 134.83.1…     10.0.3.4       DNS        90 Standard query response 0x99aa A www.google.
   5 0.008182094 134.83.1…     10.0.3.4       DNS       102 Standard query response 0x90e9 AAAA www.goog
   6 0.008190252 134.83.1…     10.0.3.4       DNS        90 Standard query response 0x99aa A www.google.
   7 0.008464142 10.0.3.4      142.250.187.…  TCP        74 33218 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1
   8 0.012524065 142.250.…     10.0.3.4       TCP        60 80 → 33218 [SYN, ACK] Seq=0 Ack=1 Win=32768
   9 0.012542311 10.0.3.4      142.250.187.…  TCP        54 33218 → 80 [ACK] Seq=1 Ack=1 Win=29200 Len=0
  10 0.013010584 10.0.3.4      142.250.187.…  HTTP      195 GET / HTTP/1.1
  11 0.077220445 142.250.…     10.0.3.4       TCP      2974 80 → 33218 [ACK] Seq=1 Ack=142 Win=32627 Len
  12 0.077241775 10.0.3.4      142.250.187.…  TCP        54 33218 → 80 [ACK] Seq=142 Ack=2921 Win=35040
  13 0.077620310 142.250.…     10.0.3.4       TCP      1514 80 → 33218 [PSH, ACK] Seq=2921 Ack=142 Win=3
```

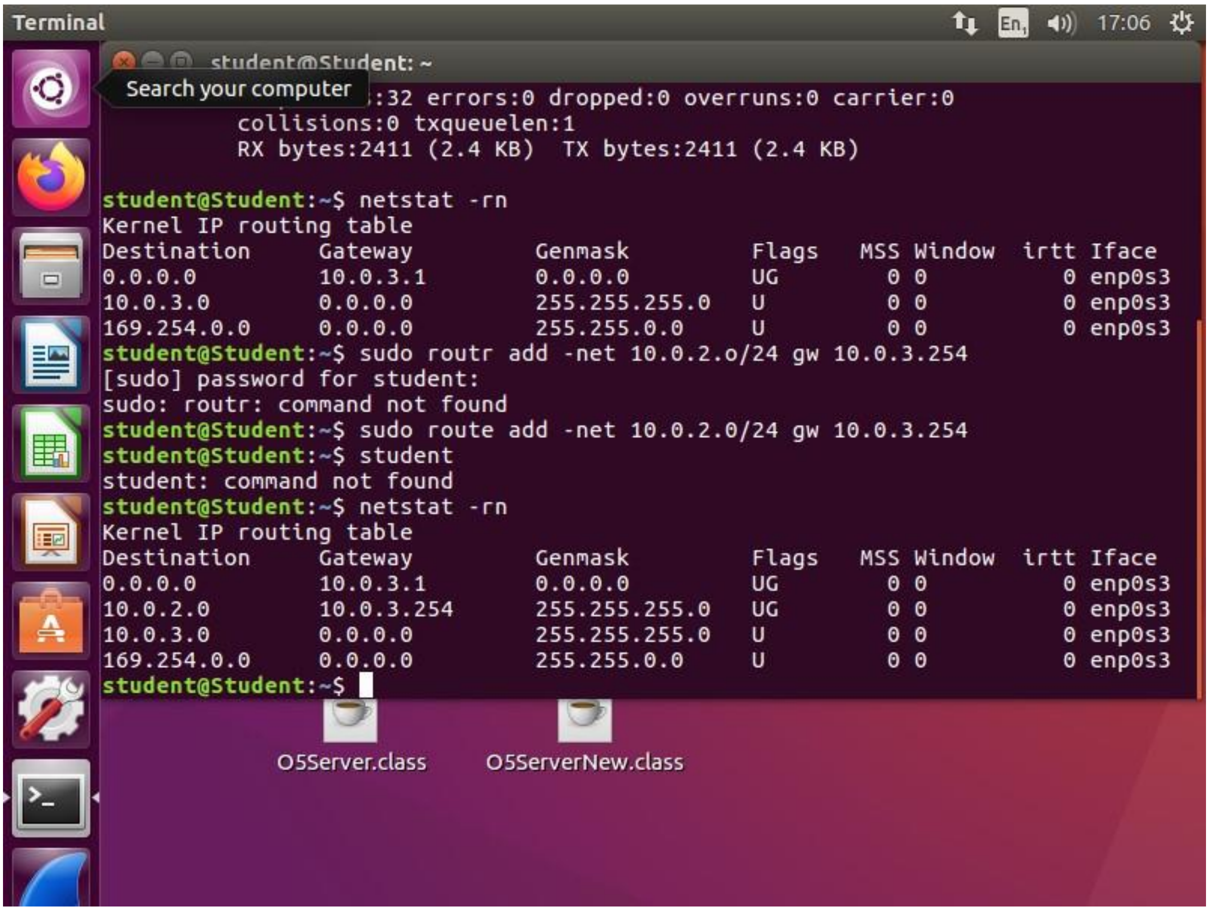
```
▶ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_fc:72:6e (08:00:27:fc:72:6e), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
▶ Internet Protocol Version 4, Src: 10.0.3.4, Dst: 134.83.128.81
▶ User Datagram Protocol, Src Port: 50852, Dst Port: 53
▶ Domain Name System (query)
```

```
0000  52 54 00 12 35 00 08 00  27 fc 72 6e 08 00 45 00   RT··5···'·rn··E·
0010  00 3c 76 28 40 00 40 11  b0 e0 0a 00 03 04 86 53   ·<v(@·@·······S
0020  80 51 c6 a4 00 35 00 28  13 e2 99 aa 01 00 00 01   ·Q···5·(········
0030  00 00 00 00 00 00 03 77  77 77 06 67 6f 6f 67 6c   ·······www·googl
0040  65 03 63 6f 6d 00 00 01  00 01                     e·com·····
```

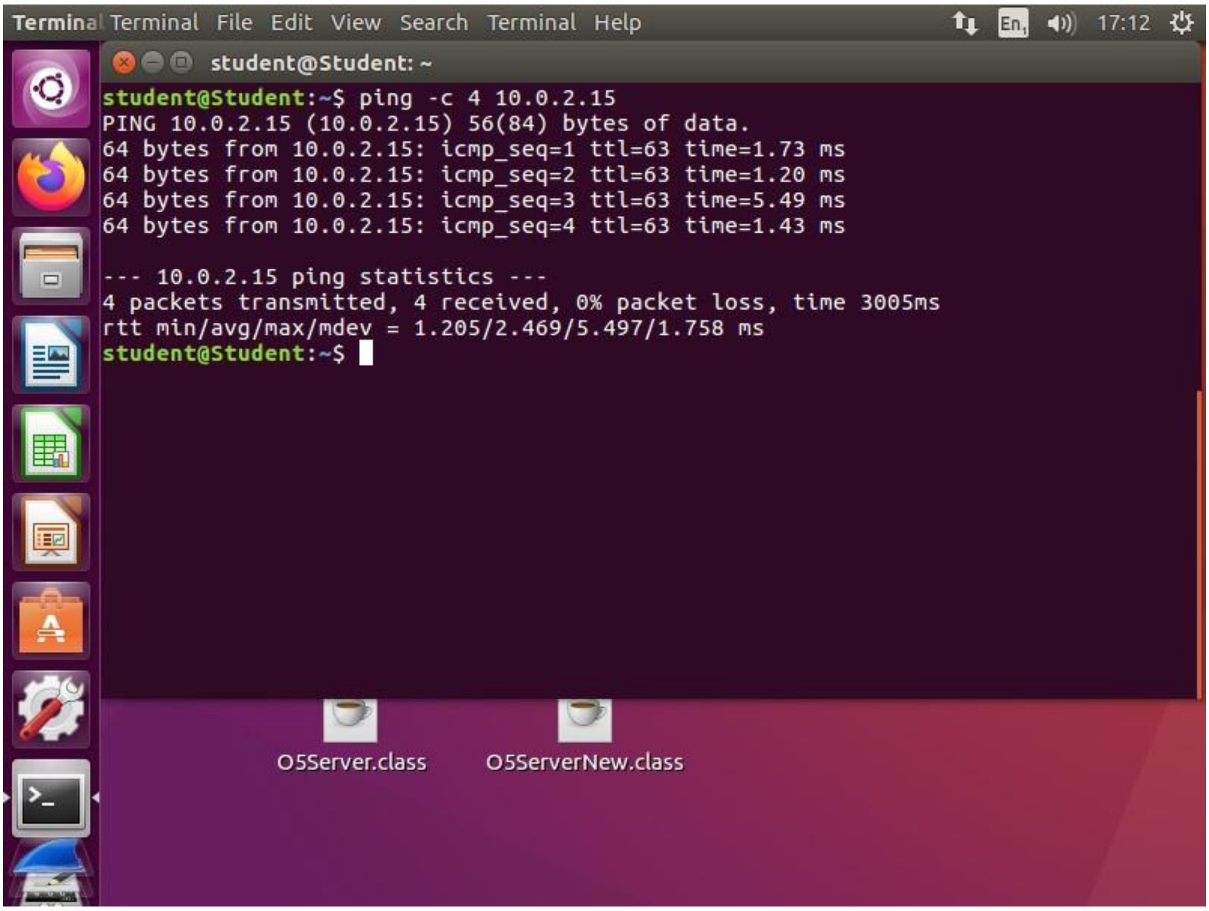
enp0s3: <live capture in progress>   Packets: 42 · Displayed: 42 (100.0%)   Profile: Default

Above screenshot shows what packets wireshark has captured after treminitaing immeiatly and capturing the prossess following the request that has been sent from browser.

2.2 ubuntu2



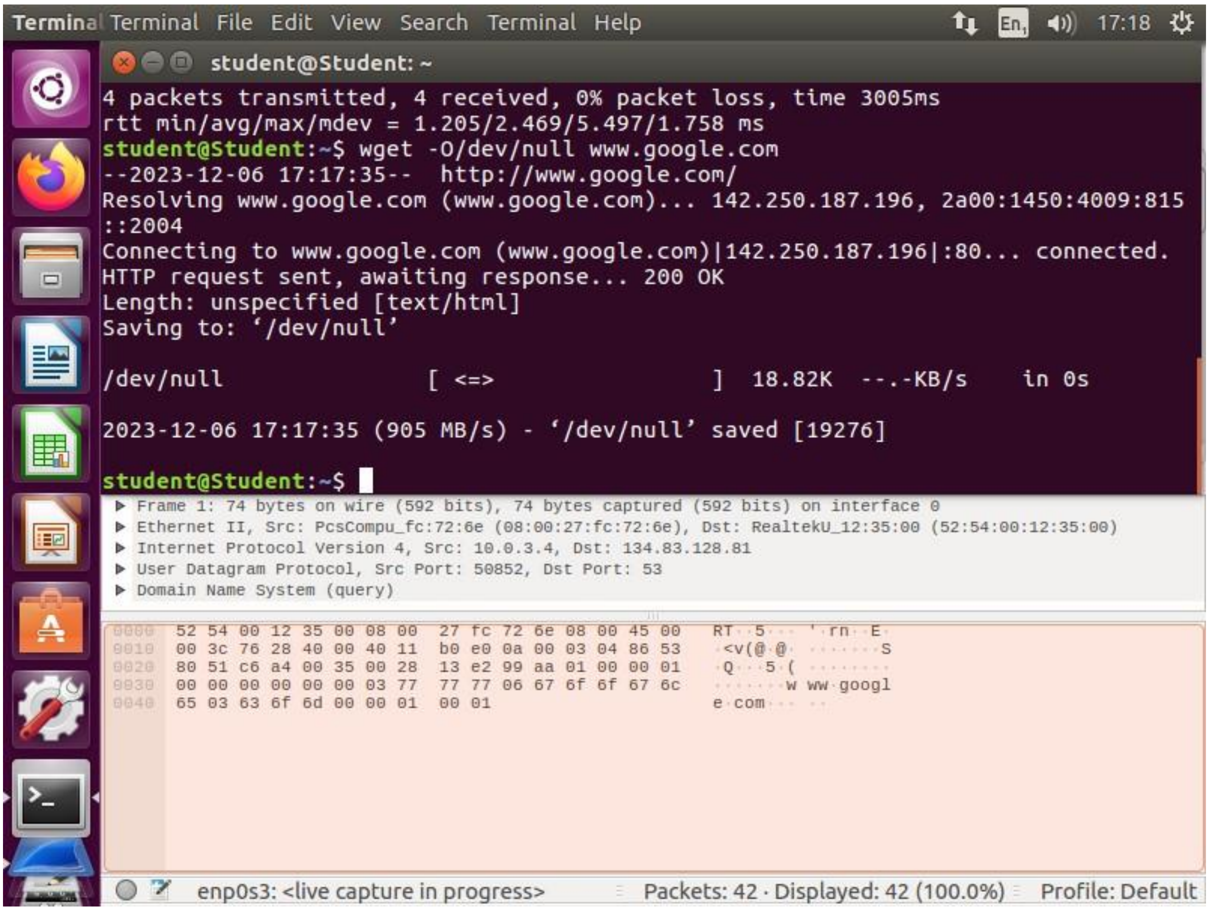
```
          :32 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:2411 (2.4 KB)  TX bytes:2411 (2.4 KB)

student@Student:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.0.3.1        0.0.0.0         UG        0 0          0 enp0s3
10.0.3.0        0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 enp0s3
student@Student:~$ sudo routr add -net 10.0.2.o/24 gw 10.0.3.254
[sudo] password for student:
sudo: routr: command not found
student@Student:~$ sudo route add -net 10.0.2.0/24 gw 10.0.3.254
student@Student:~$ student
student: command not found
student@Student:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.0.3.1        0.0.0.0         UG        0 0          0 enp0s3
10.0.2.0        10.0.3.254      255.255.255.0   UG        0 0          0 enp0s3
10.0.3.0        0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 enp0s3
student@Student:~$
```



```
student@Student:~$ ping -c 4 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=63 time=1.73 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=63 time=1.20 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=63 time=5.49 ms
64 bytes from 10.0.2.15: icmp_seq=4 ttl=63 time=1.43 ms

--- 10.0.2.15 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.205/2.469/5.497/1.758 ms
student@Student:~$
```

In the first screenshot, I utilized the "ifconfig" command in Ubuntu2's terminal to inspect its IP address. Following that, I examined the connectivity between Ubuntu1 and Ubuntu2 using the "netstat -rn" command to identify the destination and gateway. Subsequently, I established a route leading to Ubuntu1 as the destination with the command "sudo route add -net 10.0.2.0/24 gw 10.0.3.254."

In the second screenshot, I performed a route check using the "ping -c 4 10.0.2.4" command, revealing that four packets were transmitted, four were received, and there was zero packet loss. This outcome indicates the successful addition of a route between Ubuntu2 and Ubuntu1, confirming the successful transmission and reception of the transmitted packets.



```
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.205/2.469/5.497/1.758 ms
student@Student:~$ wget -O/dev/null www.google.com
--2023-12-06 17:17:35--  http://www.google.com/
Resolving www.google.com (www.google.com)... 142.250.187.196, 2a00:1450:4009:815
::2004
Connecting to www.google.com (www.google.com)|142.250.187.196|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: ‘/dev/null’

/dev/null               [ <=>                 ]  18.82K  --.-KB/s    in 0s

2023-12-06 17:17:35 (905 MB/s) - ‘/dev/null’ saved [19276]

student@Student:~$ 
```

```
▶ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_fc:72:6e (08:00:27:fc:72:6e), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
▶ Internet Protocol Version 4, Src: 10.0.3.4, Dst: 134.83.128.81
▶ User Datagram Protocol, Src Port: 50852, Dst Port: 53
▶ Domain Name System (query)
```

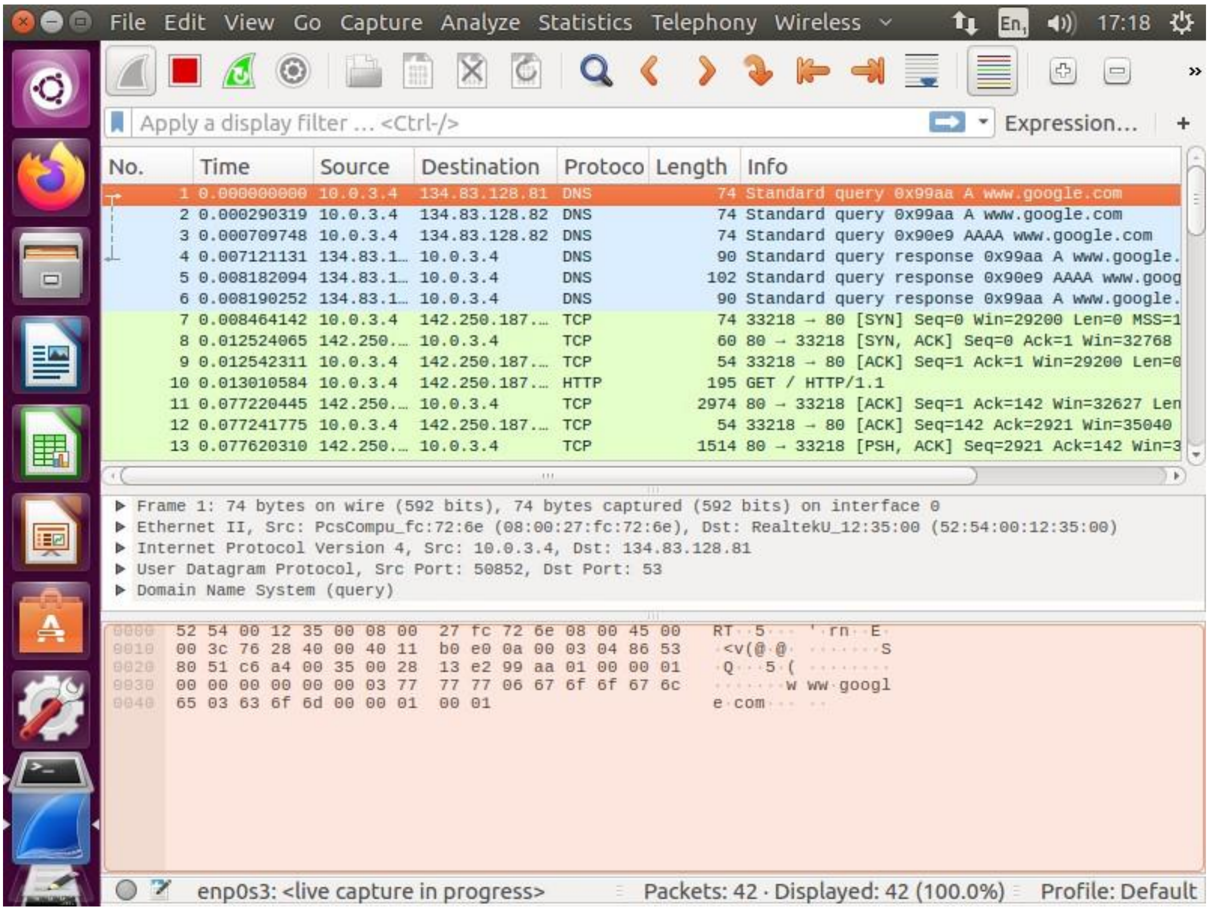
```
0000  52 54 00 12 35 00 08 00  27 fc 72 6e 08 00 45 00   RT··5···'·rn··E·
0010  00 3c 76 28 40 00 40 11  b0 e0 0a 00 03 04 86 53   ·<v(@·@·········S
0020  80 51 c6 a4 00 35 00 28  13 e2 99 aa 01 00 00 01   ·Q···5·(········
0030  00 00 00 00 00 00 03 77  77 77 06 67 6f 6f 67 6c   ·······w ww·googl
0040  65 03 63 6f 6d 00 00 01  00 01                     e·com·· ··
```

```
enp0s3: <live capture in progress>    Packets: 42 · Displayed: 42 (100.0%)    Profile: Default
```

In the above screenshot I started wireshark with ubuntu2 terminal which call the google server and using the command “wget -o/dev/null www.google.com sent a request to google server.

he provided image displays the packets captured by Wireshark, specifically after halting the capturing process immediately following the transmission initiated from Ubuntu2's terminal. It is evident that a GET message originated from Ubuntu2, having the IP address 10.0.3.4, and was directed towards the destination with the IP address 185.18.139.73. This GET message possesses a length of 179 and is characterized by the information field containing "GET / HTTP/1.1."

he provided image displays the packets captured by Wireshark, specifically after halting the capturing process immediately following the transmission initiated from Ubuntu2's terminal. It is evident that a GET message originated from Ubuntu2, having the IP address 10.0.3.4, and was directed towards the destination with the IP address 185.18.139.73. This GET message possesses a length of 179 and is characterized by the information field containing "GET / HTTP/1.1."



```
No.   Time        Source        Destination    Protoco Length Info
    1 0.000000000 10.0.3.4      134.83.128.81  DNS         74 Standard query 0x99aa A www.google.com
    2 0.000290319 10.0.3.4      134.83.128.82  DNS         74 Standard query 0x99aa A www.google.com
    3 0.000709748 10.0.3.4      134.83.128.82  DNS         74 Standard query 0x90e9 AAAA www.google.com
    4 0.007121131 134.83.1…     10.0.3.4       DNS         90 Standard query response 0x99aa A www.google.
    5 0.008182094 134.83.1…     10.0.3.4       DNS        102 Standard query response 0x90e9 AAAA www.goog
    6 0.008190252 134.83.1…     10.0.3.4       DNS         90 Standard query response 0x99aa A www.google.
    7 0.008464142 10.0.3.4      142.250.187.…  TCP         74 33218 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1
    8 0.012524065 142.250.…     10.0.3.4       TCP         60 80 → 33218 [SYN, ACK] Seq=0 Ack=1 Win=32768
    9 0.012542311 10.0.3.4      142.250.187.…  TCP         54 33218 → 80 [ACK] Seq=1 Ack=1 Win=29200 Len=0
   10 0.013010584 10.0.3.4      142.250.187.…  HTTP       195 GET / HTTP/1.1
   11 0.077220445 142.250.…     10.0.3.4       TCP       2974 80 → 33218 [ACK] Seq=1 Ack=142 Win=32627 Len
   12 0.077241775 10.0.3.4      142.250.187.…  TCP         54 33218 → 80 [ACK] Seq=142 Ack=2921 Win=35040
   13 0.077620310 142.250.…     10.0.3.4       TCP       1514 80 → 33218 [PSH, ACK] Seq=2921 Ack=142 Win=3
```

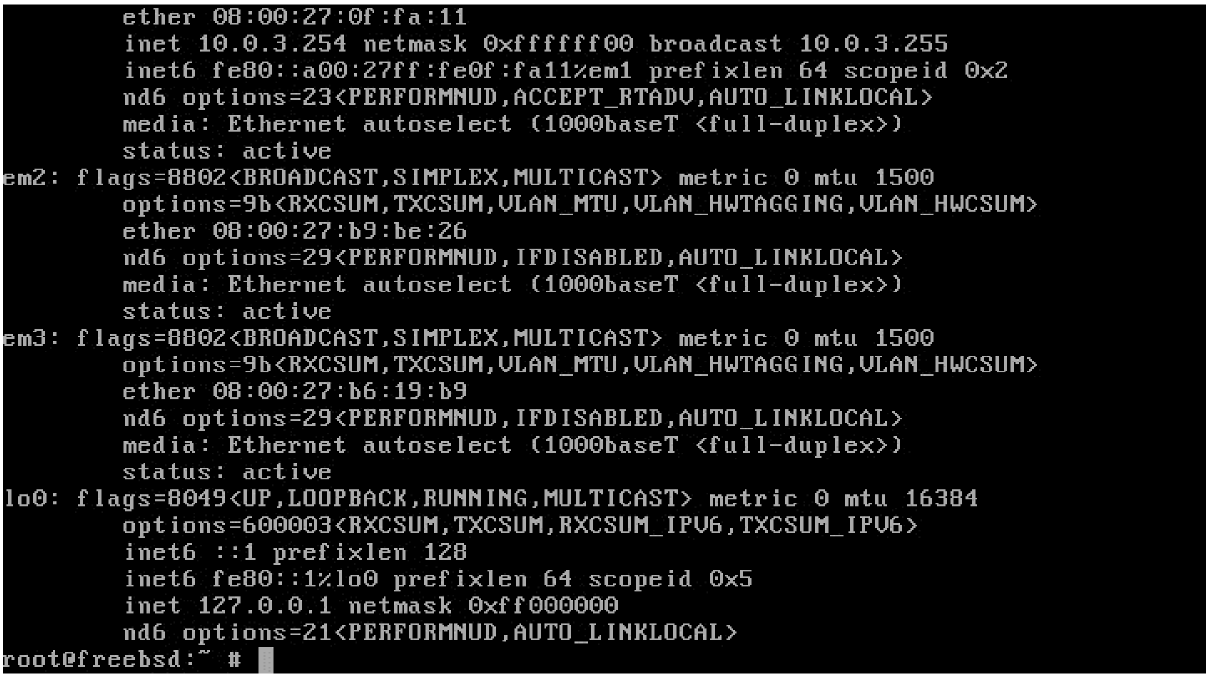
```
▶ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_fc:72:6e (08:00:27:fc:72:6e), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
▶ Internet Protocol Version 4, Src: 10.0.3.4, Dst: 134.83.128.81
▶ User Datagram Protocol, Src Port: 50852, Dst Port: 53
▶ Domain Name System (query)
```

```
0000  52 54 00 12 35 00 08 00  27 fc 72 6e 08 00 45 00   RT··5···'·rn··E·
0010  00 3c 76 28 40 00 40 11  b0 e0 0a 00 03 04 86 53   ·<v(@·@·········S
0020  80 51 c6 a4 00 35 00 28  13 e2 99 aa 01 00 00 01   ·Q···5·(········
0030  00 00 00 00 00 00 03 77  77 77 06 67 6f 6f 67 6c   ·······www·googl
0040  65 03 63 6f 6d 00 00 01  00 01                     e·com·····
```

```
enp0s3: <live capture in progress>    Packets: 42 · Displayed: 42 (100.0%)    Profile: Default
```

The provided image displays the packets captured by Wireshark, specifically after halting the capturing process immediately following the transmission initiated from Ubuntu2's terminal. It is evident that a GET message originated from Ubuntu2, having the IP address 10.0.3.4, and was directed towards the destination with the IP address 185.18.139.73. This GET message possesses a length of 179 and is characterized by the information field containing "GET / HTTP/1.1."

2.3freebsdnat

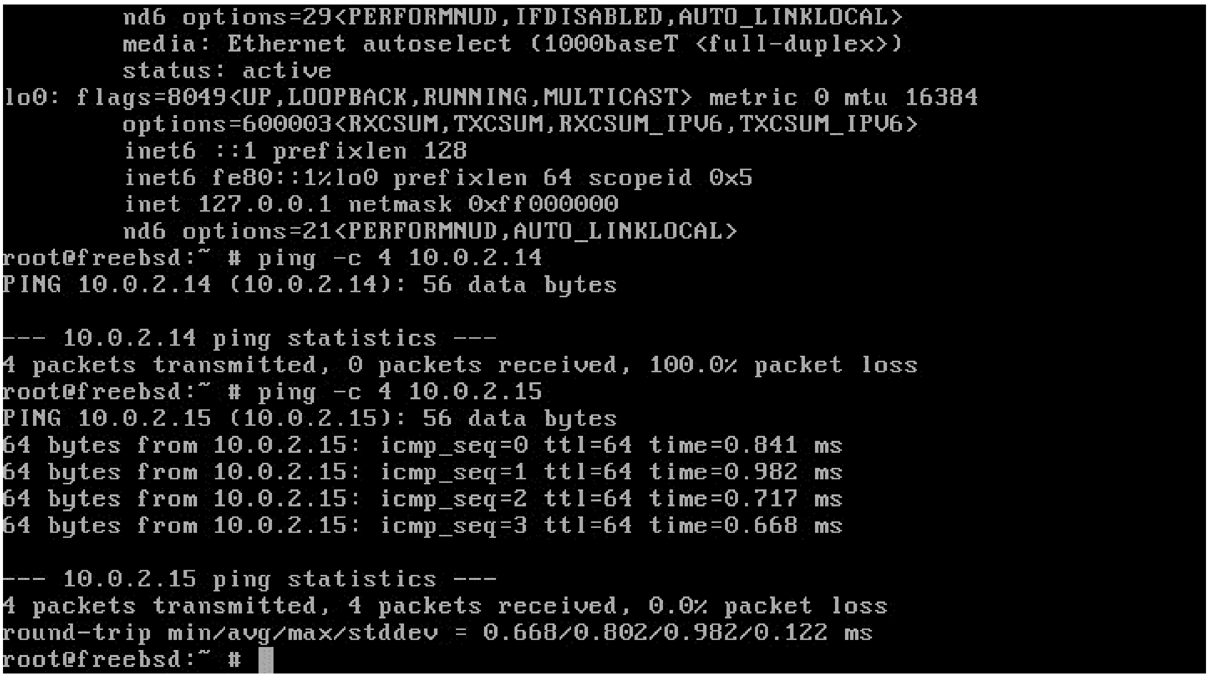


```
        ether 08:00:27:0f:fa:11
        inet 10.0.3.254 netmask 0xffffff00 broadcast 10.0.3.255
        inet6 fe80::a00:27ff:fe0f:fa11%em1 prefixlen 64 scopeid 0x2
        nd6 options=23<PERFORMNUD,ACCEPT_RTADV,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
em2: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
        ether 08:00:27:b9:be:26
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
em3: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
        ether 08:00:27:b6:19:b9
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=600003<RXCSUM,TXCSUM,RXCSUM_IPV6,TXCSUM_IPV6>
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x5
        inet 127.0.0.1 netmask 0xff000000
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
root@freebsd:~ #
```

In this screenshot,the command “inconfig “within freebsd to check the Ip address that connects to the nat route1 and nat route 2

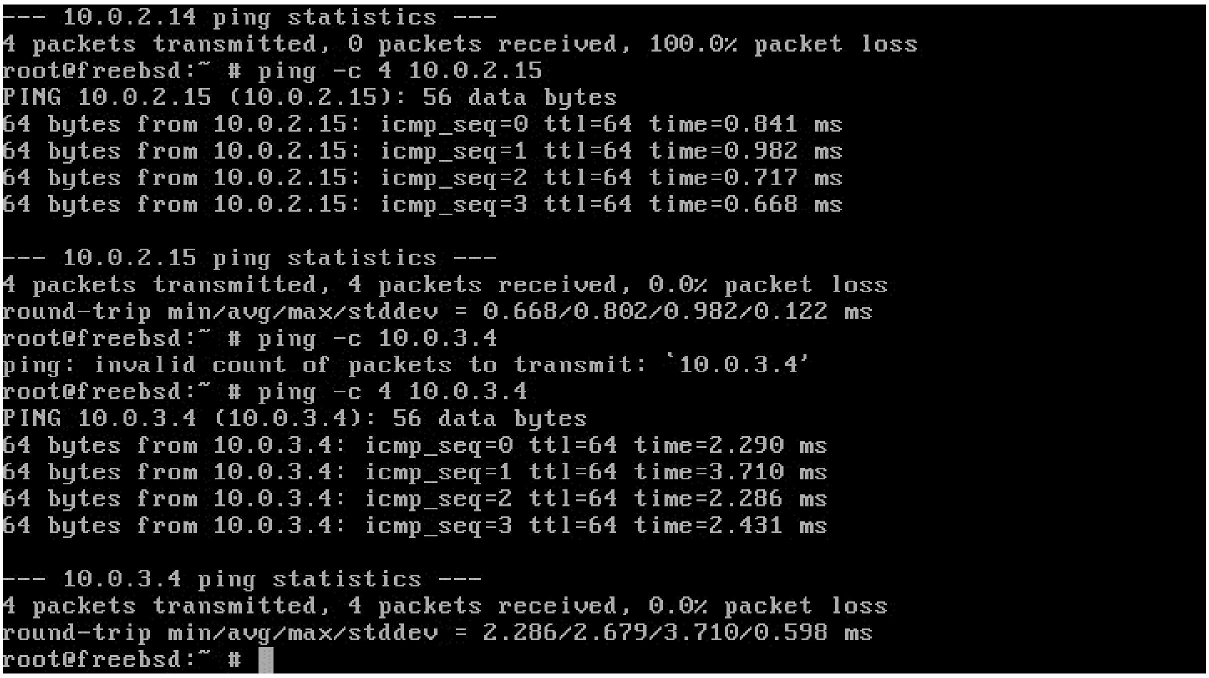
10.0.2.254 meansfreesbd connected nat route 1

10.0.3.254 means freebsd connected to nat route 2



```
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=600003<RXCSUM,TXCSUM,RXCSUM_IPV6,TXCSUM_IPV6>
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x5
        inet 127.0.0.1 netmask 0xff000000
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
root@freebsd:~ # ping -c 4 10.0.2.14
PING 10.0.2.14 (10.0.2.14): 56 data bytes

--- 10.0.2.14 ping statistics ---
4 packets transmitted, 0 packets received, 100.0% packet loss
root@freebsd:~ # ping -c 4 10.0.2.15
PING 10.0.2.15 (10.0.2.15): 56 data bytes
64 bytes from 10.0.2.15: icmp_seq=0 ttl=64 time=0.841 ms
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.982 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.717 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.668 ms

--- 10.0.2.15 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.668/0.802/0.982/0.122 ms
root@freebsd:~ #
```

In this screenshot,I used the command ping-c 4 10.0.2.4 with freebsd to check connection between freebds and ubuntu1,the result shows that 4packet has been sent as well as receive that’s means route exiting.



```
--- 10.0.2.14 ping statistics ---
4 packets transmitted, 0 packets received, 100.0% packet loss
root@freebsd:~ # ping -c 4 10.0.2.15
PING 10.0.2.15 (10.0.2.15): 56 data bytes
64 bytes from 10.0.2.15: icmp_seq=0 ttl=64 time=0.841 ms
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.982 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.717 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.668 ms

--- 10.0.2.15 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.668/0.802/0.982/0.122 ms
root@freebsd:~ # ping -c 10.0.3.4
ping: invalid count of packets to transmit: `10.0.3.4'
root@freebsd:~ # ping -c 4 10.0.3.4
PING 10.0.3.4 (10.0.3.4): 56 data bytes
64 bytes from 10.0.3.4: icmp_seq=0 ttl=64 time=2.290 ms
64 bytes from 10.0.3.4: icmp_seq=1 ttl=64 time=3.710 ms
64 bytes from 10.0.3.4: icmp_seq=2 ttl=64 time=2.286 ms
64 bytes from 10.0.3.4: icmp_seq=3 ttl=64 time=2.431 ms

--- 10.0.3.4 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 2.286/2.679/3.710/0.598 ms
root@freebsd:~ #
```

In the above screenshot,I used the command “ping c- 4 10.0.3.4with freebsd to check the connect to freebsd and ubuntu2,the result shows that 4packet has been sent and receive that’s means route is exciting.

3 calclient and calserver documentation:

calcclient	calcserver
	Run clacserver

runcalclient	
	[Accept calcclient connection]
WHILE NOT TERMINATED	WHILE NOT TERMINATED
	SEND”calculator server ready and waiting”
Receive” calculator sever ready and waiting”	
SEND”1”	
	RECIVE”1”
SEND “add operands”	
“R	RECEIVE “add operands”
	SEND “operands to add”
Receive”send operands to add”	
“SEND “7 and 2”	

	Receive “7 and 2”
“PRINT”12”	SEND”9”
SEND”NEXT OPERATION PLEASE”	
	RECEIVE “nexrt operation please”
	SEND”CALCULATOR ready and waiting”
RECEIVE “calculator server ready and waiting”	
SEND 2	
	RECEIVE 2
SEND “sub operands”	
	RECEIVE “sub operands”
	SEND “send operands to subtracr”
RECEIVE”send operands to subtract”	

SEND “13 and 3”	
	RECEIVE “13and3”
PRINT “10”	
SEND” NEXT OPERATION please”	
	RECEIVE “next operation please”
	SEND”calculator server ready and waiting”
RECEIVE “calculator server ready and waiting”	
SEND” 3”	
	RECEIVE “3”SEND “
SEND” multi operands”	
	RECEIVE “MULTI OPERANDS”
	SEND” send operands to multiply”

RECEIVE””send operands to multiply”	
SEND”12 AND 2”	RECEIVE “12AND 2
	SEND”24”
PRINT”24”	
SEND “next to operation please”	
	RECEIVE”next operation please”
	SEND” calculator server ready and waiting
Receive”calculator server ready and waiting”	
SEND”4”	
	Receive”4”
SEND” div operands “	
	RECEIVE “div operands”

	SEND “send operands to divide”
Receive send operands to divide	
SEND”24and2”	
	RECIVE”24and2
	Send “12”
PRINT “12”	
SEND “next operation please”	
	RECEIVE “next operation please
	SEND” calculator server ready and waiting”
RECEIVE”calculator server ready and waiting”	
SEND”0”	
	RECIVE”0”

	SEND”endcom ms”
	[TERMINATE]

4.CalcClientupdate and calcserverupdate documentation

calcclientupdate	calcserverupdate
[RUNcalcclient]	[RUNcalcserverupdate]
	SEND” calculator server ready and waiting
Receive “calculator server ready and waiting”	
Send”1”	
	RECEIVE”1”
SEND”add operands”	

	Receive “add operands”
	SEND”send operands to add”
RECEIVE”send operands to add”	
SEND “12 And 6”	
	RECEIVE “12and 6”
	SEND”6”
PRINT”6”	
SEND”next to operation please”	
	RECEIVE”next operation please”
	SDEND”calculator server ready and waiting”
RECEIVE “calculator server ready and waiting”	
SEND”2”	

	RECEIVE”2”
SENDE”sub operands”	
	RECEIVE “sub operands”
	SEND”operands to sub tract”
RECEIVE”send operands to subtract”	
SEND”12 and 6”	
	RECIVE”12and6”
	SEND”6”
PRINT”6”	
SEND”NEXT TO operation please”	
	RECEIVE”next operation please
	SEND”CALCUTOR SERVER READY AND WAITING”

RECIEVEE”calculator server ready and waiting”	
SEND”3”	
	RECEIVE”3”
SEND”multi operands”	
	RECEIVE “multi operands”
	SEND”send OPERANDS TO multiply”
RECEIVE”sends operands to multiply”	
SEND”4and3”	RECEIVE “4AND3”
	SEND”16”
PRINT”16”	
SEND”next operation please”	

	RECEIVE”next operation please”
	SEND”calculator server ready and waitying”
RECEIVE “calculator server ready and waiting”	
SEND”4”	
	RECEIVE”4”
	SEND”endcom ms”
RECEIVE”endcom ms”	
	[TERMINATE]
[RUNcalcclientupdate]	[RUNcalcserverupdate]
	SEND”calculator server ready and waiting”
RECEIVE”CALCULATOR SERVER ready and waiting”	

SEND”0”	
	RECEIVE”0”
	SEND”operands to divide”
RECEIVE”send operands to divide”	
SEND”30and5”	
	RECEIVE”30and5”
	SEND”6”
PRINT”6”	

5REPORT TO THE NOSsoft Management

In this section I would like to write a report to NOSsoft management their update softwear outlinig problem and its proved.

Problem1:client sent wrong messege to server

Problem2:client sending wrong messeage to server

Problem3:server sends wrong message to client.

3. Conclusions

In this report, I commenced by establishing a network comprising two subnets, namely Ubuntu1 and Ubuntu2, interconnected by the Freebsd router. Subsequently, I conducted various configuration tests to validate the network's functionality. These tests involved the utilization of commands such as ifconfig, netstat -rn, and ping -c, alongside employing the Firefox browser to access the Brunel server, while capturing transactions using Wireshark. The comprehensive testing instilled confidence in the network's connectivity.

Transitioning to the subsequent section, I delved into the initial version of NOSsoft managment. I executed calcClient on Ubuntu1 and calcServer on Ubuntu1, monitoring the transactions between them through Wireshark. The recorded transactions were meticulously documented in a table. Upon advancing to the updated version of O5AppShop, I repeated the process, but this time I discerned several issues during the transaction observation. I diligently identified and documented these problems.

Concluding the report, I compiled a detailed summary of the identified issues and presented it to the Nossoft managers. The purpose of this presentation was to persuade them of the imperative need to address and rectify these problems with the latest update.