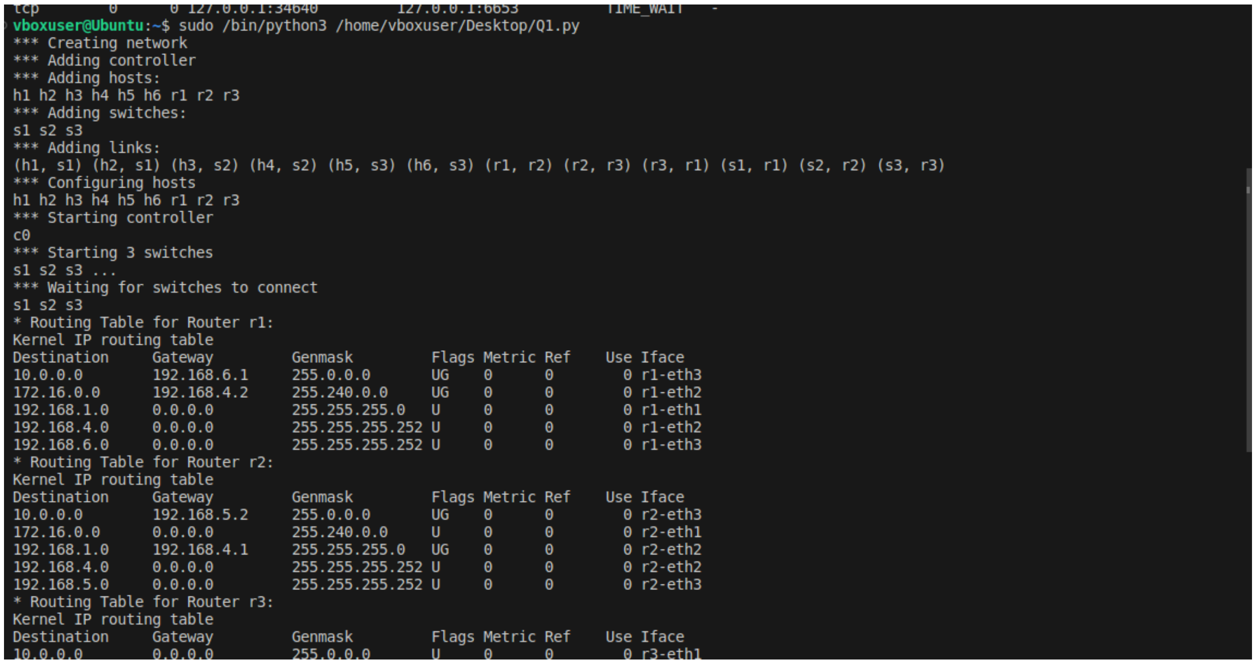
Assignment-02 **CS 433 Computer Networks (2023-24)**
Pulkit Gautam 21110169
Manav Parmar 21110120

I)

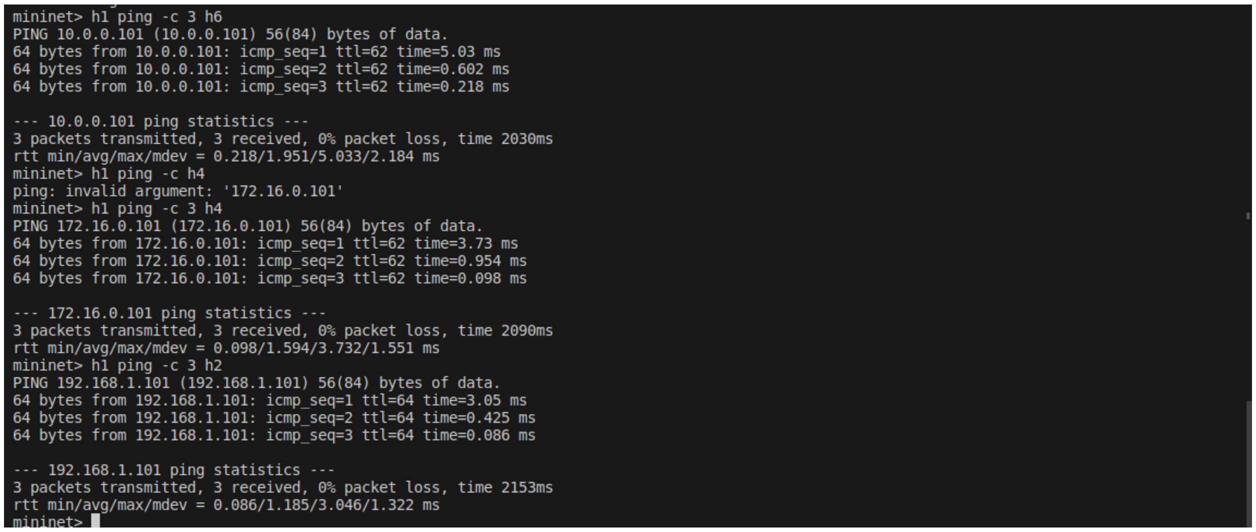
a) The code is attached with this repository. A mininet includes three inter-connected subnets each with a router and two hosts.
Here's a brief explanation of the key components and implementation:

1. **Custom Router Class (LinuxRouter):**
    - Inherits from the Mininet Node class.
    - Overrides the config method to enable IP forwarding on the router.
    - Overrides the terminate method to disable IP forwarding when terminating.
2. **Topology Class (NetworkTopo):**
    - Inherits from the Mininet Topo class.
    - Defines the network topology, including routers, switches, and links between them.
    - Configures IP addresses for router interfaces.
3. **Topology Building (build method in NetworkTopo):**
    - Adds three routers (r1, r2, r3) with specific IP addresses.
    - Adds three switches (s1, s2, s3).
    - Establishes links between routers and switches.
    - Configures point-to-point links between routers (r1-eth2 to r2-eth2, r2-eth3 to r3-eth2, r3-eth3 to r1-eth3).
    - Adds hosts (h1 to h6) and connects them to switches with assigned IP addresses and default routes.
4. **Mininet Initialization and Startup (run function):**
    - Creates an instance of the NetworkTopo topology.
    - Initializes Mininet with the specified topology.
    - Starts the Mininet network.
5. **Packet Capture and Static Routes (run function):**
    - Enables packet capture on specific interfaces of r1.
    - Configures static routes on routers to ensure proper routing between subnets.
6. **CLI Interaction (CLI(net) in run` function):**
    - Opens the Mininet Command Line Interface (CLI) for user interaction.
7. **Routing Table Display and Modification (run function):**
    - Displays the initial routing tables for routers (r1, r2, r3).
    - Modifies default routes for specific traffic.
    - Displays the updated routing tables.
8. **Packet Capture Cleanup (run function):**
    - Stops the packet capture on r1 before exiting.
9. **Mininet Cleanup (run function):**
    - Stops the Mininet network.
10. **Script Execution Check (`if name == 'main':):**
    - Ensures that the script is executed when run directly and not when imported as a

module.

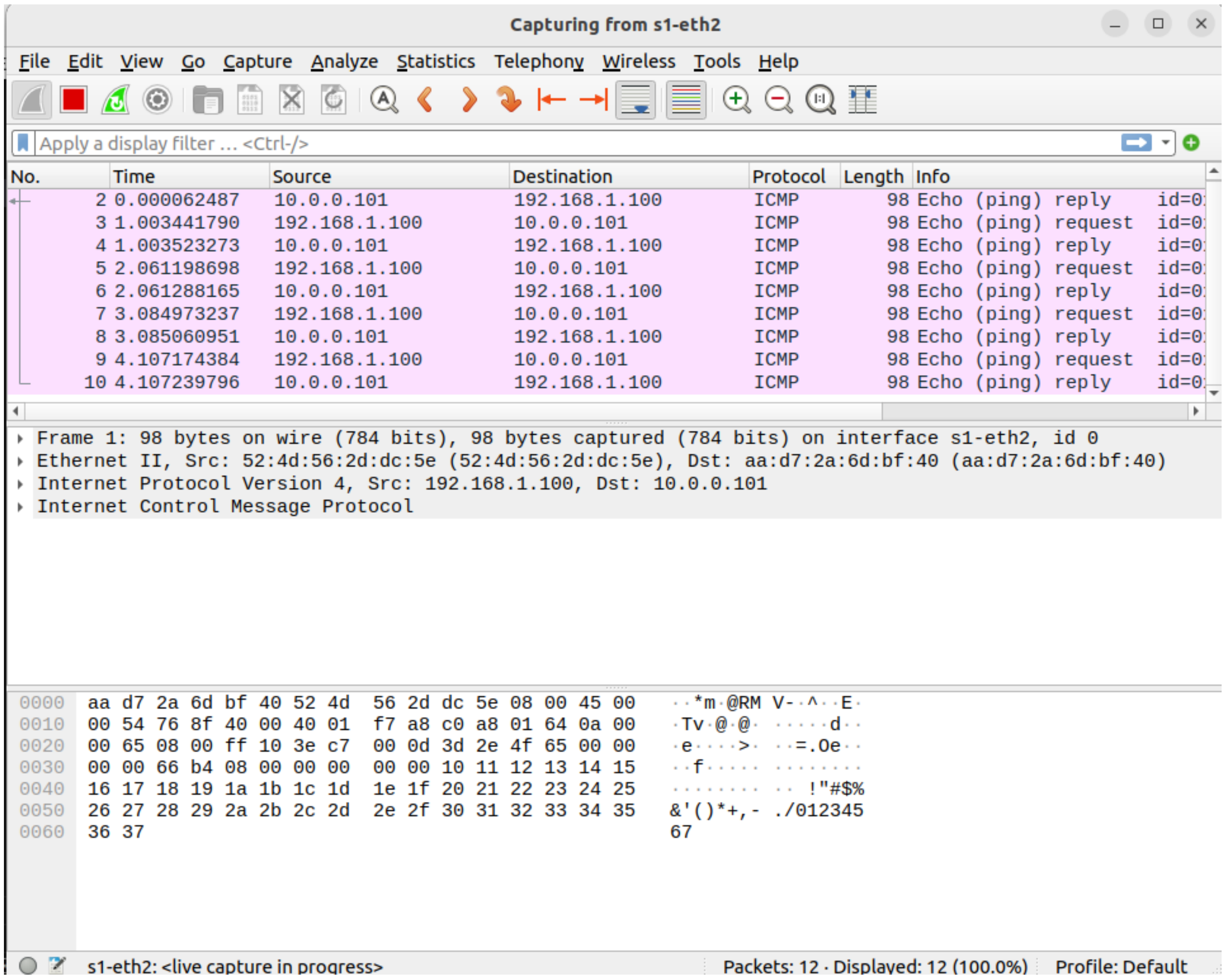


```
tcp        0      0 127.0.0.1:34640         127.0.0.1:6653          TIME_WAIT   -
vboxuser@Ubuntu:~$ sudo /bin/python3 /home/vboxuser/Desktop/Q1.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 r1 r2 r3
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (h2, s1) (h3, s2) (h4, s2) (h5, s3) (h6, s3) (r1, r2) (r2, r3) (r3, r1) (s1, r1) (s2, r2) (s3, r3)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 r1 r2 r3
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Waiting for switches to connect
s1 s2 s3
* Routing Table for Router r1:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        192.168.6.1     255.0.0.0       UG    0      0        0 r1-eth3
172.16.0.0      192.168.4.2     255.240.0.0     UG    0      0        0 r1-eth2
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 r1-eth1
192.168.4.0     0.0.0.0         255.255.255.252 U     0      0        0 r1-eth2
192.168.6.0     0.0.0.0         255.255.255.252 U     0      0        0 r1-eth3
* Routing Table for Router r2:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        192.168.5.2     255.0.0.0       UG    0      0        0 r2-eth3
172.16.0.0      0.0.0.0         255.240.0.0     U     0      0        0 r2-eth1
192.168.1.0     192.168.4.1     255.255.255.0   UG    0      0        0 r2-eth2
192.168.4.0     0.0.0.0         255.255.255.252 U     0      0        0 r2-eth2
192.168.5.0     0.0.0.0         255.255.255.252 U     0      0        0 r2-eth3
* Routing Table for Router r3:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r3-eth1
```

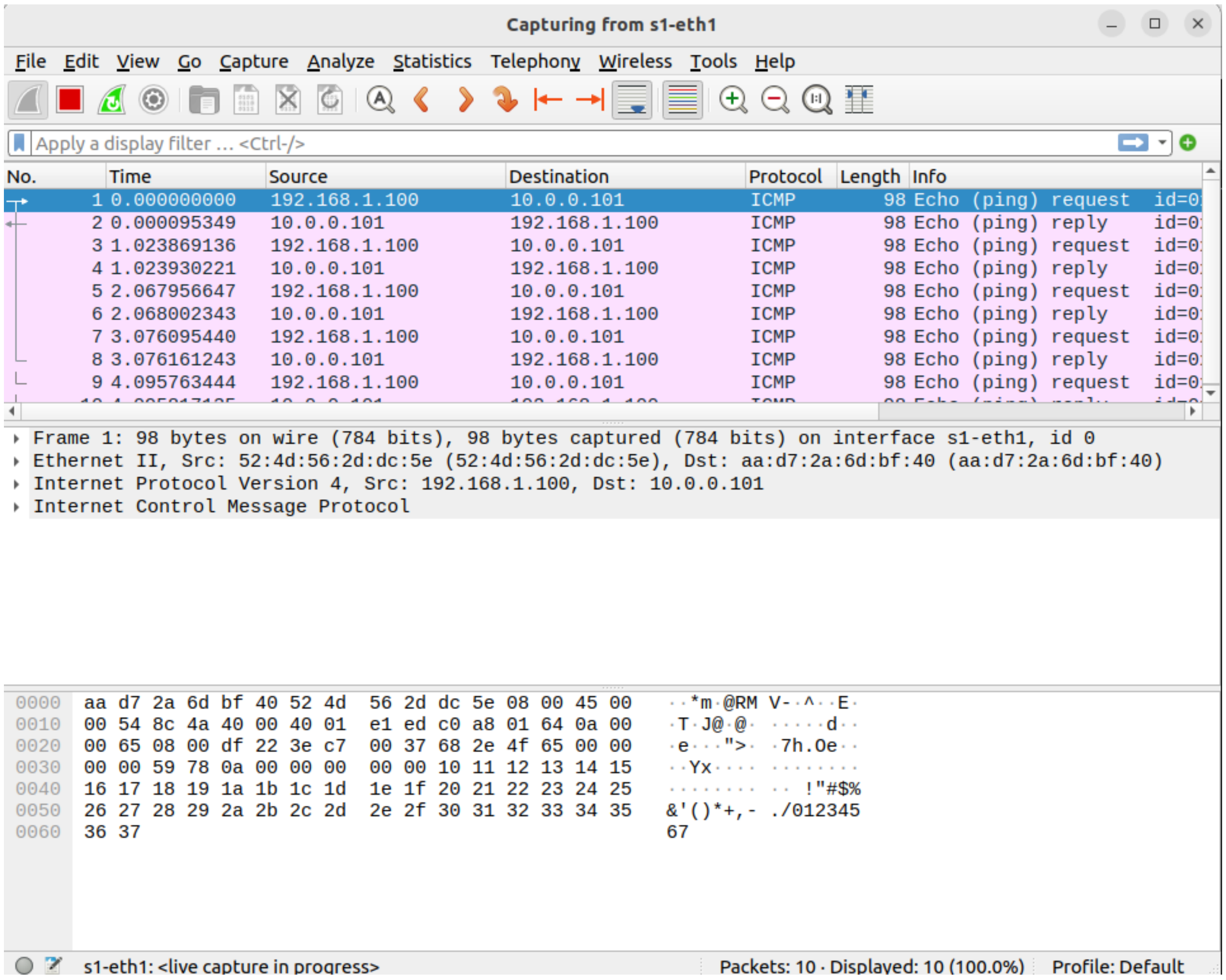


```
mininet> h1 ping -c 3 h6
PING 10.0.0.101 (10.0.0.101) 56(84) bytes of data.
64 bytes from 10.0.0.101: icmp_seq=1 ttl=62 time=5.03 ms
64 bytes from 10.0.0.101: icmp_seq=2 ttl=62 time=0.602 ms
64 bytes from 10.0.0.101: icmp_seq=3 ttl=62 time=0.218 ms

--- 10.0.0.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2030ms
rtt min/avg/max/mdev = 0.218/1.951/5.033/2.184 ms
mininet> h1 ping -c h4
ping: invalid argument: '172.16.0.101'
mininet> h1 ping -c 3 h4
PING 172.16.0.101 (172.16.0.101) 56(84) bytes of data.
64 bytes from 172.16.0.101: icmp_seq=1 ttl=62 time=3.73 ms
64 bytes from 172.16.0.101: icmp_seq=2 ttl=62 time=0.954 ms
64 bytes from 172.16.0.101: icmp_seq=3 ttl=62 time=0.098 ms

--- 172.16.0.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2090ms
rtt min/avg/max/mdev = 0.098/1.594/3.732/1.551 ms
mininet> h1 ping -c 3 h2
PING 192.168.1.101 (192.168.1.101) 56(84) bytes of data.
64 bytes from 192.168.1.101: icmp_seq=1 ttl=64 time=3.05 ms
64 bytes from 192.168.1.101: icmp_seq=2 ttl=64 time=0.425 ms
64 bytes from 192.168.1.101: icmp_seq=3 ttl=64 time=0.086 ms

--- 192.168.1.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2153ms
rtt min/avg/max/mdev = 0.086/1.185/3.046/1.322 ms
mininet> 
```

b) WireShark output for the packets captured on router 1:


Capturing from s1-eth2
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time          Source          Destination     Protocol Length Info
    2 0.000062487   10.0.0.101      192.168.1.100   ICMP        98 Echo (ping) reply    id=0
    3 1.003441790   192.168.1.100   10.0.0.101      ICMP        98 Echo (ping) request  id=0
    4 1.003523273   10.0.0.101      192.168.1.100   ICMP        98 Echo (ping) reply    id=0
    5 2.061198698   192.168.1.100   10.0.0.101      ICMP        98 Echo (ping) request  id=0
    6 2.061288165   10.0.0.101      192.168.1.100   ICMP        98 Echo (ping) reply    id=0
    7 3.084973237   192.168.1.100   10.0.0.101      ICMP        98 Echo (ping) request  id=0
    8 3.085060951   10.0.0.101      192.168.1.100   ICMP        98 Echo (ping) reply    id=0
    9 4.107174384   192.168.1.100   10.0.0.101      ICMP        98 Echo (ping) request  id=0
   10 4.107239796   10.0.0.101      192.168.1.100   ICMP        98 Echo (ping) reply    id=0

Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface s1-eth2, id 0
Ethernet II, Src: 52:4d:56:2d:dc:5e (52:4d:56:2d:dc:5e), Dst: aa:d7:2a:6d:bf:40 (aa:d7:2a:6d:bf:40)
Internet Protocol Version 4, Src: 192.168.1.100, Dst: 10.0.0.101
Internet Control Message Protocol

0000  aa d7 2a 6d bf 40 52 4d  56 2d dc 5e 08 00 45 00   ··*m·@RM V-·^··E·
0010  00 54 76 8f 40 00 40 01  f7 a8 c0 a8 01 64 0a 00   ·Tv·@·@· ·····d··
0020  00 65 08 00 ff 10 3e c7  00 0d 3d 2e 4f 65 00 00   ·e····>· ··=.Oe··
0030  00 00 66 b4 08 00 00 00  00 00 10 11 12 13 14 15   ··f····· ········
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ········ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67

s1-eth2: <live capture in progress>     Packets: 12 · Displayed: 12 (100.0%)     Profile: Default
```

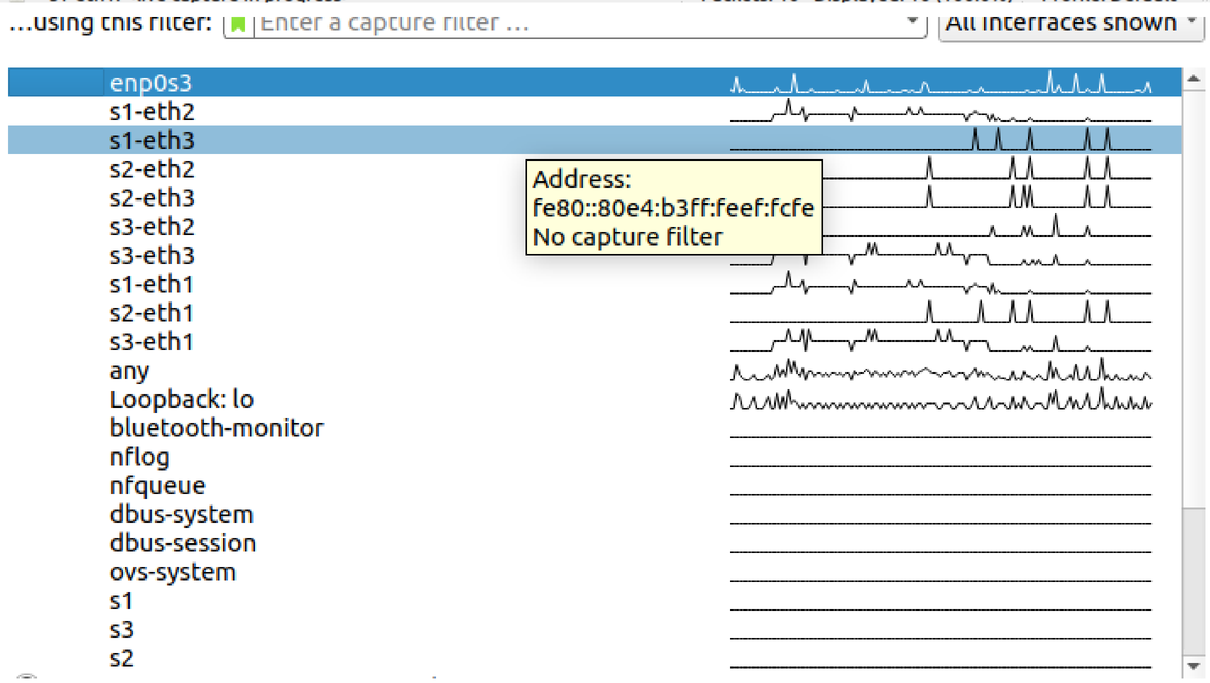

Capturing from s1-eth1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time         Source          Destination     Protocol Length Info
    1 0.000000000  192.168.1.100   10.0.0.101      ICMP        98 Echo (ping) request  id=0
    2 0.000095349  10.0.0.101      192.168.1.100   ICMP        98 Echo (ping) reply    id=0
    3 1.023869136  192.168.1.100   10.0.0.101      ICMP        98 Echo (ping) request  id=0
    4 1.023930221  10.0.0.101      192.168.1.100   ICMP        98 Echo (ping) reply    id=0
    5 2.067956647  192.168.1.100   10.0.0.101      ICMP        98 Echo (ping) request  id=0
    6 2.068002343  10.0.0.101      192.168.1.100   ICMP        98 Echo (ping) reply    id=0
    7 3.076095440  192.168.1.100   10.0.0.101      ICMP        98 Echo (ping) request  id=0
    8 3.076161243  10.0.0.101      192.168.1.100   ICMP        98 Echo (ping) reply    id=0
    9 4.095763444  192.168.1.100   10.0.0.101      ICMP        98 Echo (ping) request  id=0
```

```
Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface s1-eth1, id 0
Ethernet II, Src: 52:4d:56:2d:dc:5e (52:4d:56:2d:dc:5e), Dst: aa:d7:2a:6d:bf:40 (aa:d7:2a:6d:bf:40)
Internet Protocol Version 4, Src: 192.168.1.100, Dst: 10.0.0.101
Internet Control Message Protocol
```

```
0000  aa d7 2a 6d bf 40 52 4d  56 2d dc 5e 08 00 45 00   ··*m·@RM V-·^··E·
0010  00 54 8c 4a 40 00 40 01  e1 ed c0 a8 01 64 0a 00   ·T·J@·@· ·····d··
0020  00 65 08 00 df 22 3e c7  00 37 68 2e 4f 65 00 00   ·e···">· ·7h.Oe··
0030  00 00 59 78 0a 00 00 00  00 00 10 11 12 13 14 15   ··Yx···· ········
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ········ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

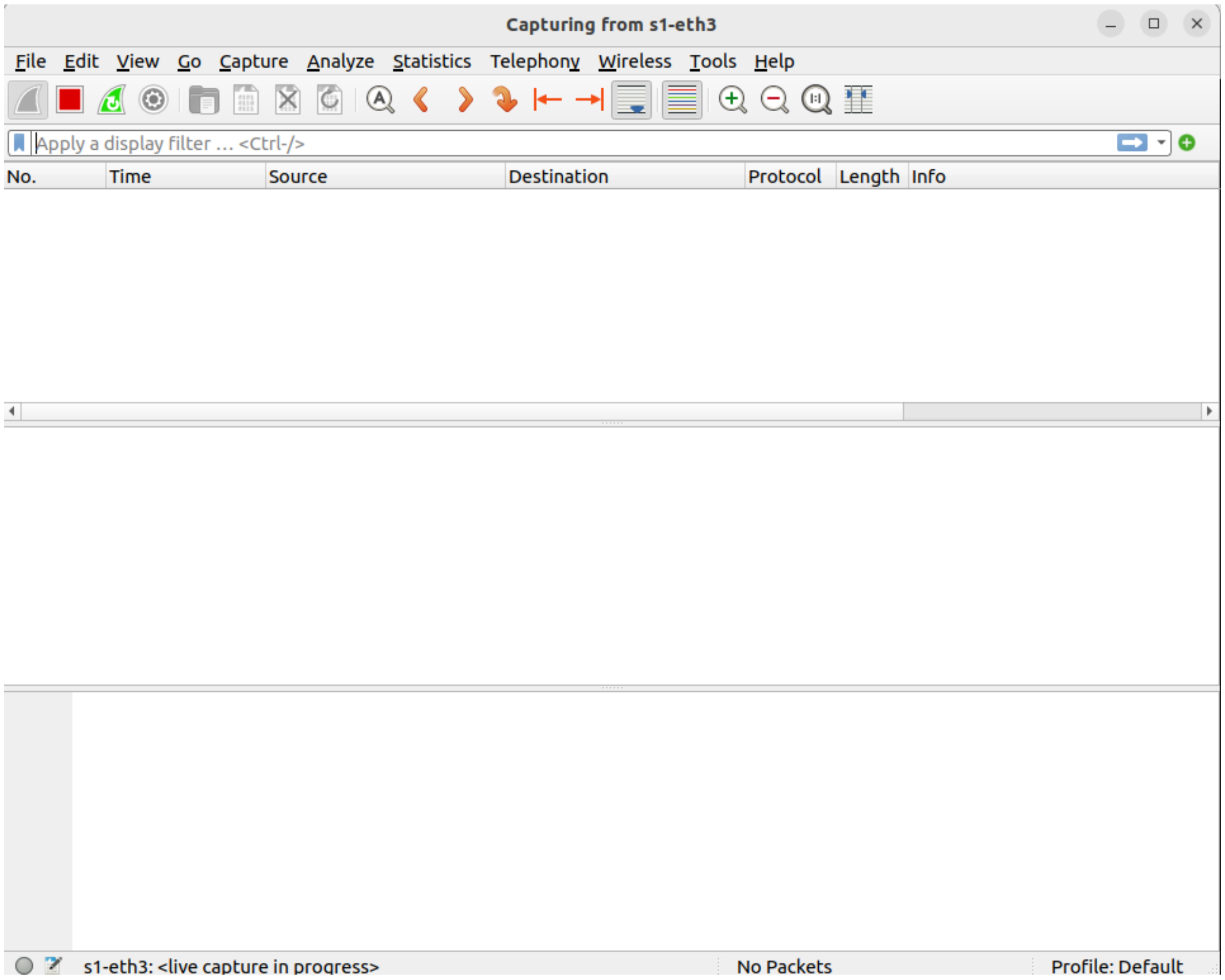
s1-eth1: <live capture in progress> Packets: 10 · Displayed: 10 (100.0%) Profile: Default

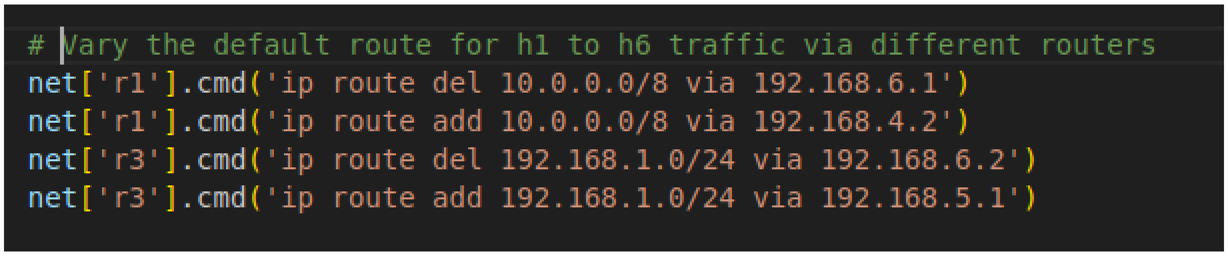


enp0s3
s1-eth2
s1-eth3
s2-eth2
s2-eth3
s3-eth2
s3-eth3
s1-eth1
s2-eth1
s3-eth1
any
Loopback: lo
bluetooth-monitor
nflog
nfqueue
dbus-system
dbus-session
ovs-system
s1
s3
s2

Address:
fe80::80e4:b3ff:feef:fcfe
No capture filter

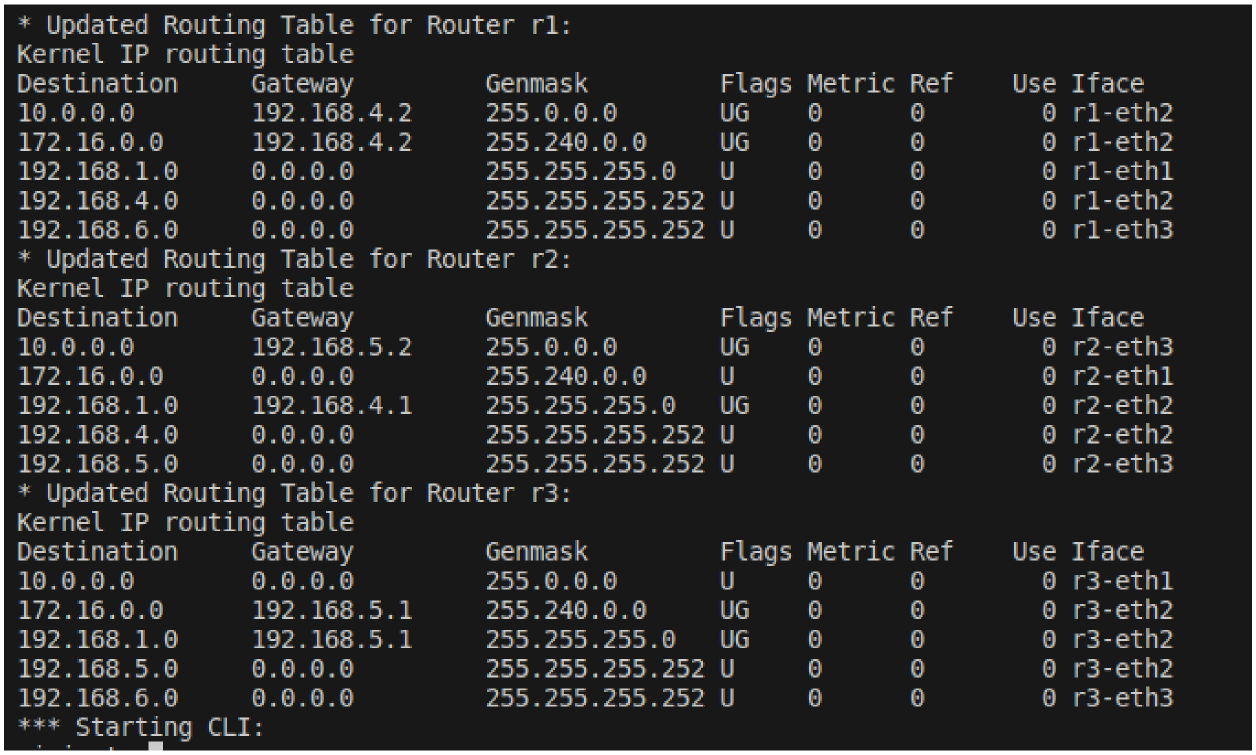
c)


Capturing from s1-eth3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
No. Time Source Destination Protocol Length Info
s1-eth3: <live capture in progress>
No Packets
Profile: Default

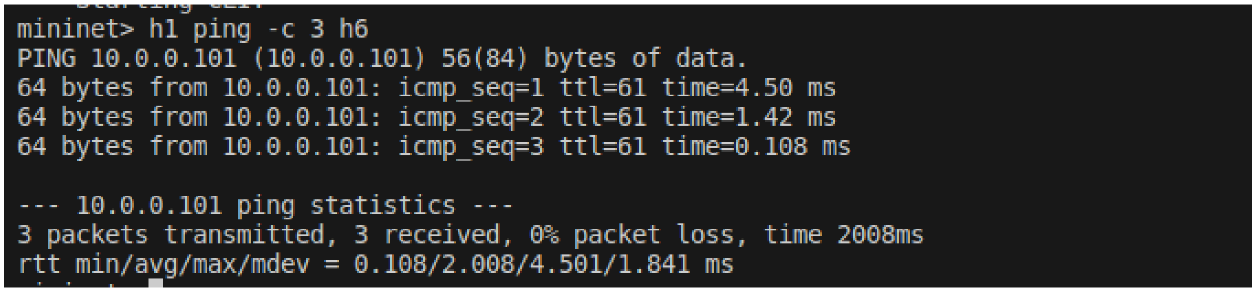


```
# Vary the default route for h1 to h6 traffic via different routers
net['r1'].cmd('ip route del 10.0.0.0/8 via 192.168.6.1')
net['r1'].cmd('ip route add 10.0.0.0/8 via 192.168.4.2')
net['r3'].cmd('ip route del 192.168.1.0/24 via 192.168.6.2')
net['r3'].cmd('ip route add 192.168.1.0/24 via 192.168.5.1')
```

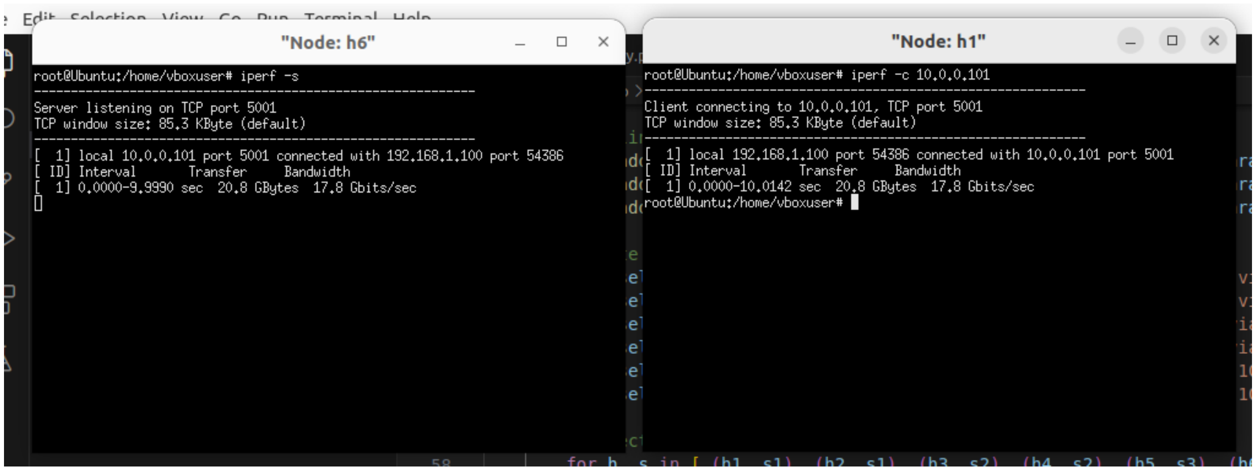
Code snippet for varying the default route



```
* Updated Routing Table for Router r1:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        192.168.4.2     255.0.0.0       UG    0      0        0 r1-eth2
172.16.0.0      192.168.4.2     255.240.0.0     UG    0      0        0 r1-eth2
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 r1-eth1
192.168.4.0     0.0.0.0         255.255.255.252 U     0      0        0 r1-eth2
192.168.6.0     0.0.0.0         255.255.255.252 U     0      0        0 r1-eth3
* Updated Routing Table for Router r2:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        192.168.5.2     255.0.0.0       UG    0      0        0 r2-eth3
172.16.0.0      0.0.0.0         255.240.0.0     U     0      0        0 r2-eth1
192.168.1.0     192.168.4.1     255.255.255.0   UG    0      0        0 r2-eth2
192.168.4.0     0.0.0.0         255.255.255.252 U     0      0        0 r2-eth2
192.168.5.0     0.0.0.0         255.255.255.252 U     0      0        0 r2-eth3
* Updated Routing Table for Router r3:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r3-eth1
172.16.0.0      192.168.5.1     255.240.0.0     UG    0      0        0 r3-eth2
192.168.1.0     192.168.5.1     255.255.255.0   UG    0      0        0 r3-eth2
192.168.5.0     0.0.0.0         255.255.255.252 U     0      0        0 r3-eth2
192.168.6.0     0.0.0.0         255.255.255.252 U     0      0        0 r3-eth3
*** Starting CLI:
```



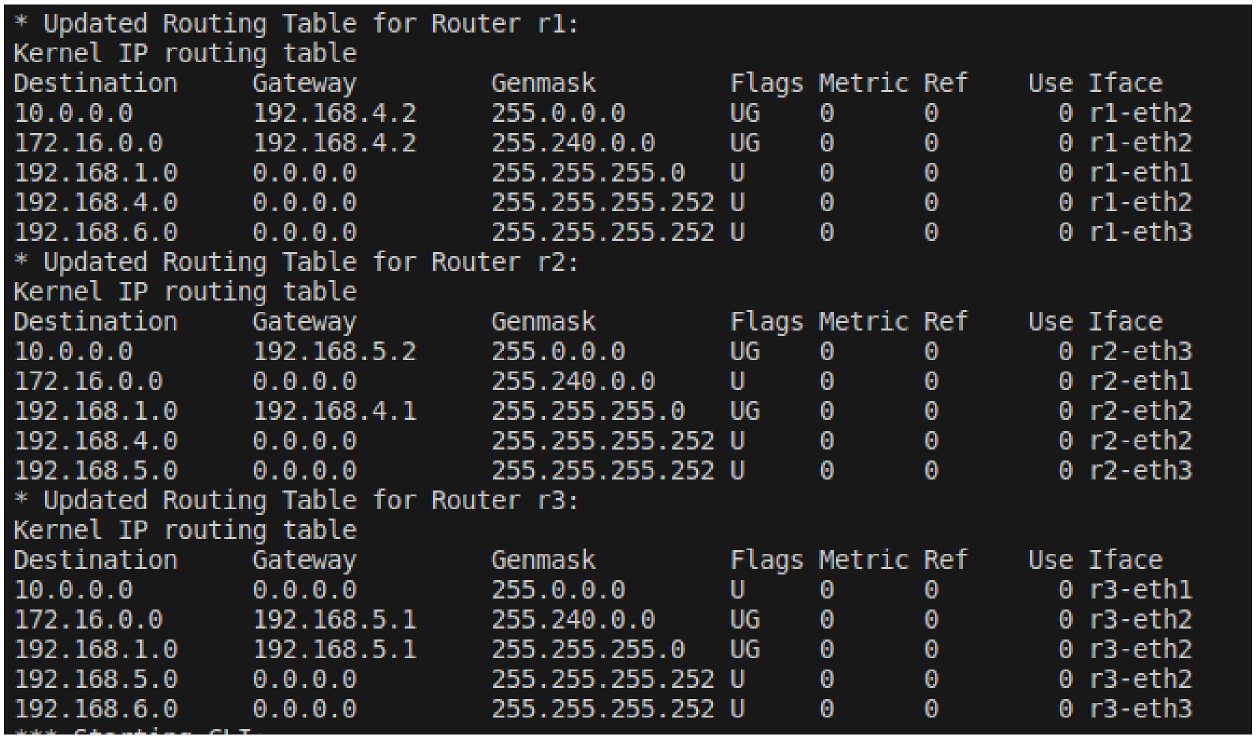
```
mininet> h1 ping -c 3 h6
PING 10.0.0.101 (10.0.0.101) 56(84) bytes of data.
64 bytes from 10.0.0.101: icmp_seq=1 ttl=61 time=4.50 ms
64 bytes from 10.0.0.101: icmp_seq=2 ttl=61 time=1.42 ms
64 bytes from 10.0.0.101: icmp_seq=3 ttl=61 time=0.108 ms

--- 10.0.0.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2008ms
rtt min/avg/max/mdev = 0.108/2.008/4.501/1.841 ms
```



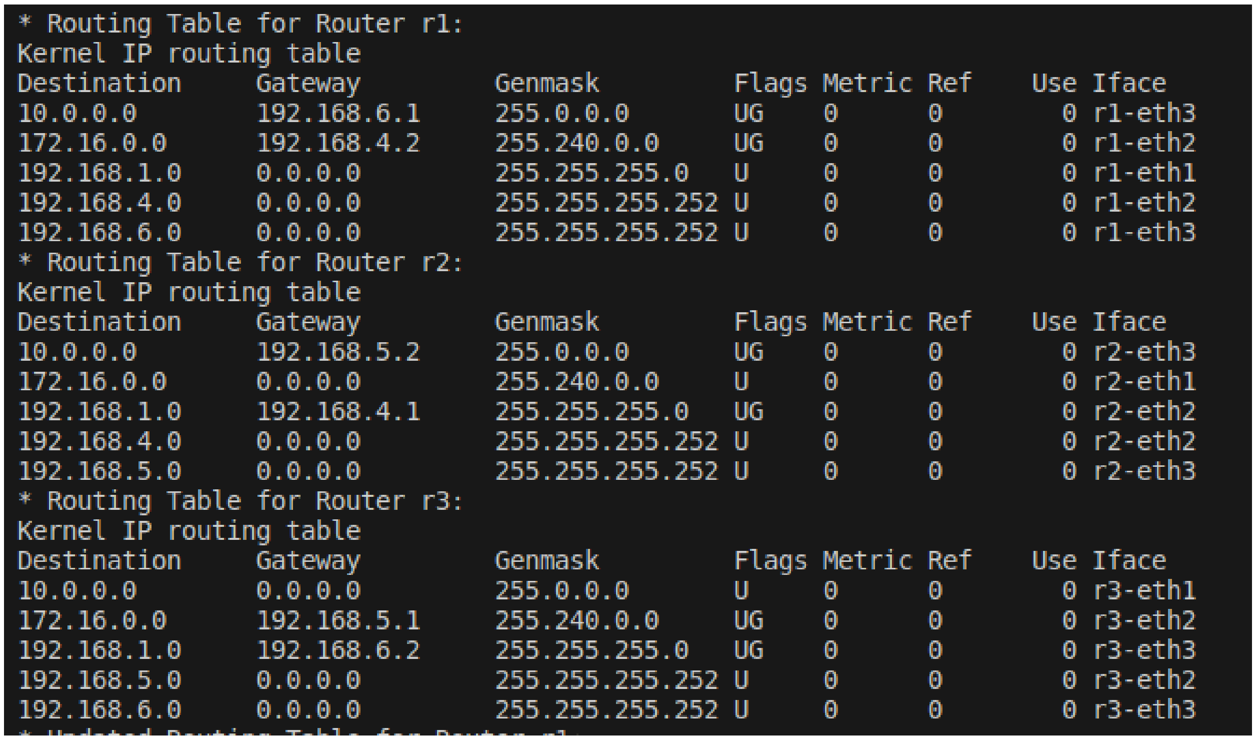
```
root@Ubuntu:/home/vboxuser# iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  1] local 10.0.0.101 port 5001 connected with 192.168.1.100 port 54386
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-9.9990 sec  20.8 GBytes  17.8 Gbits/sec
```

```
root@Ubuntu:/home/vboxuser# iperf -c 10.0.0.101
------------------------------------------------------------
Client connecting to 10.0.0.101, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  1] local 192.168.1.100 port 54386 connected with 10.0.0.101 port 5001
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.0142 sec  20.8 GBytes  17.8 Gbits/sec
root@Ubuntu:/home/vboxuser#
```

d)



```
* Updated Routing Table for Router r1:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        192.168.4.2     255.0.0.0       UG    0      0        0 r1-eth2
172.16.0.0      192.168.4.2     255.240.0.0     UG    0      0        0 r1-eth2
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 r1-eth1
192.168.4.0     0.0.0.0         255.255.255.252 U     0      0        0 r1-eth2
192.168.6.0     0.0.0.0         255.255.255.252 U     0      0        0 r1-eth3
* Updated Routing Table for Router r2:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        192.168.5.2     255.0.0.0       UG    0      0        0 r2-eth3
172.16.0.0      0.0.0.0         255.240.0.0     U     0      0        0 r2-eth1
192.168.1.0     192.168.4.1     255.255.255.0   UG    0      0        0 r2-eth2
192.168.4.0     0.0.0.0         255.255.255.252 U     0      0        0 r2-eth2
192.168.5.0     0.0.0.0         255.255.255.252 U     0      0        0 r2-eth3
* Updated Routing Table for Router r3:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r3-eth1
172.16.0.0      192.168.5.1     255.240.0.0     UG    0      0        0 r3-eth2
192.168.1.0     192.168.5.1     255.255.255.0   UG    0      0        0 r3-eth2
192.168.5.0     0.0.0.0         255.255.255.252 U     0      0        0 r3-eth2
192.168.6.0     0.0.0.0         255.255.255.252 U     0      0        0 r3-eth3
```



```
* Routing Table for Router r1:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        192.168.6.1     255.0.0.0       UG    0      0        0 r1-eth3
172.16.0.0      192.168.4.2     255.240.0.0     UG    0      0        0 r1-eth2
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 r1-eth1
192.168.4.0     0.0.0.0         255.255.255.252 U     0      0        0 r1-eth2
192.168.6.0     0.0.0.0         255.255.255.252 U     0      0        0 r1-eth3
* Routing Table for Router r2:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        192.168.5.2     255.0.0.0       UG    0      0        0 r2-eth3
172.16.0.0      0.0.0.0         255.240.0.0     U     0      0        0 r2-eth1
192.168.1.0     192.168.4.1     255.255.255.0   UG    0      0        0 r2-eth2
192.168.4.0     0.0.0.0         255.255.255.252 U     0      0        0 r2-eth2
192.168.5.0     0.0.0.0         255.255.255.252 U     0      0        0 r2-eth3
* Routing Table for Router r3:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 r3-eth1
172.16.0.0      192.168.5.1     255.240.0.0     UG    0      0        0 r3-eth2
192.168.1.0     192.168.6.2     255.255.255.0   UG    0      0        0 r3-eth3
192.168.5.0     0.0.0.0         255.255.255.252 U     0      0        0 r3-eth2
192.168.6.0     0.0.0.0         255.255.255.252 U     0      0        0 r3-eth3
```

2.

a) The provided Python script uses Mininet to create a network topology with two switches (S1, S2) and four hosts (H1 to H4). It employs a custom TCP client-server program using the iperf tool. The program runs experiments for two configurations: (b) a single client (H1) connecting to a server (H4), and (c) multiple clients (H1, H2, H3) connecting to the same server (H4). The script allows specifying congestion control algorithms (Reno, Vegas, Cubic, BBR) and link loss for each experiment configuration. Throughput analysis, focusing on the mentioned congestion control schemes, is conducted by measuring data transfer rates between clients and the server using iperf. The results are logged for further analysis. The script also facilitates interactive exploration of the network through the Mininet Command Line Interface (CLI).