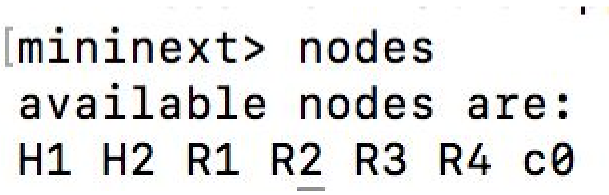
Assignment - 3
Report

Question.1) A.1 and 2)
In topo.py, I added commands to assign Eth-0 IP addresses to each router and host and also added links between routers and hosts.
Result of nodes command:



```
mininext> nodes
available nodes are:
H1 H2 R1 R2 R3 R4 c0
```

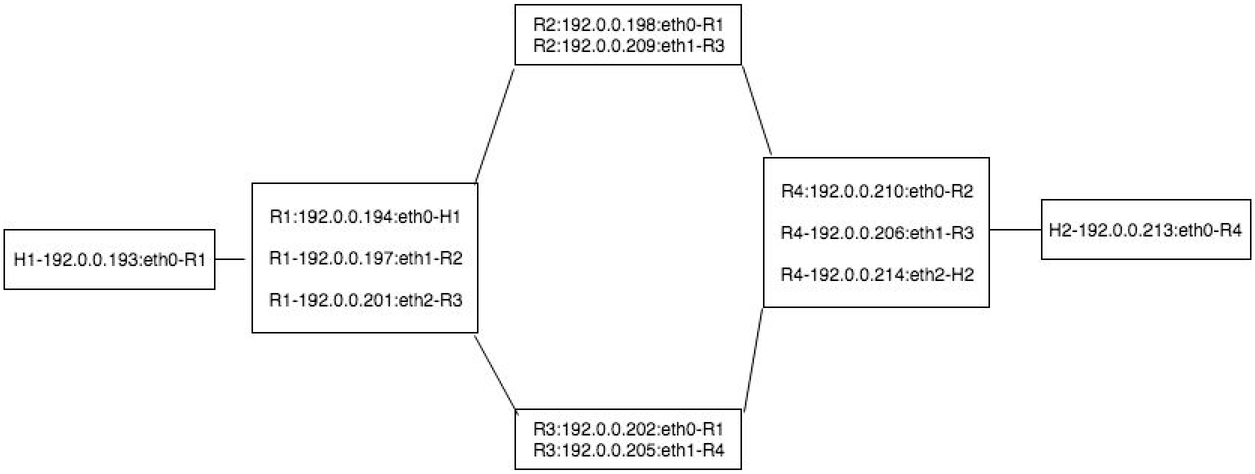
All nodes were created.
In start.py, I assigned ip addresses for other eth interfaces like R1 needs to communicate with H1, R2 and R3, so it requires 3 interfaces.
Ex. net.hosts[i].cmdPrint("ip addr add 192.0.0.201/30 dev R1-eth2")
And gateway for hosts:
net.hosts[i].cmdPrint("route add default gw 192.0.0.214")

I enabled ip forward variable to 1 for each node in start.py.

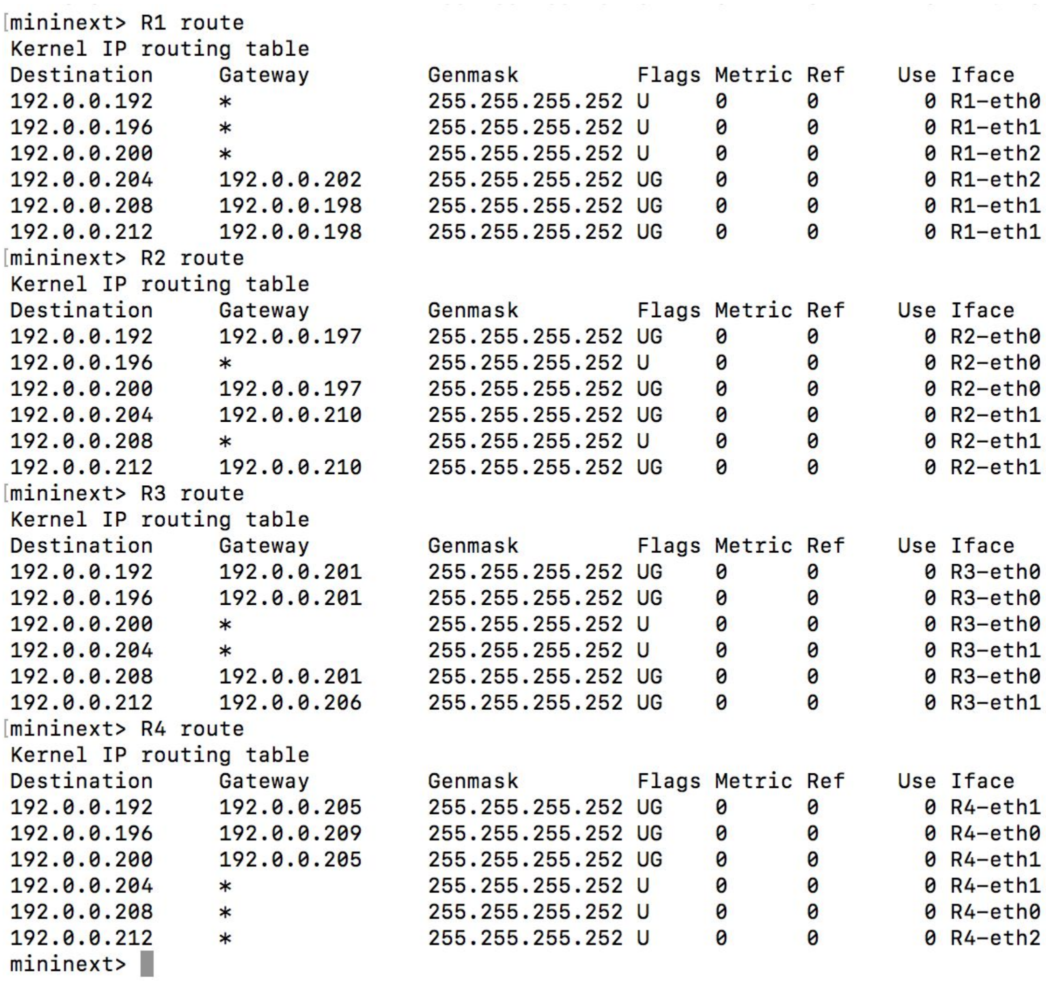
host.cmdPrint("echo 1 > /proc/sys/net/ipv4/ip_forward")

Then I enabled static routes in start.py
Example: net.hosts[i].cmdPrint("ip route add 192.0.0.204/30 via 192.0.0.202")
So for this particular host in for loop, we add to route to 192.0.0.204 via 192.0.0.202.

Image after configuring all this:

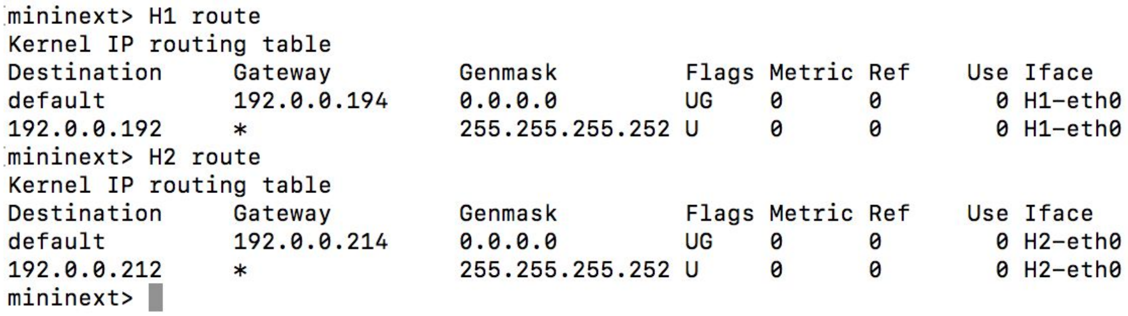


Routing Tables for Routers:



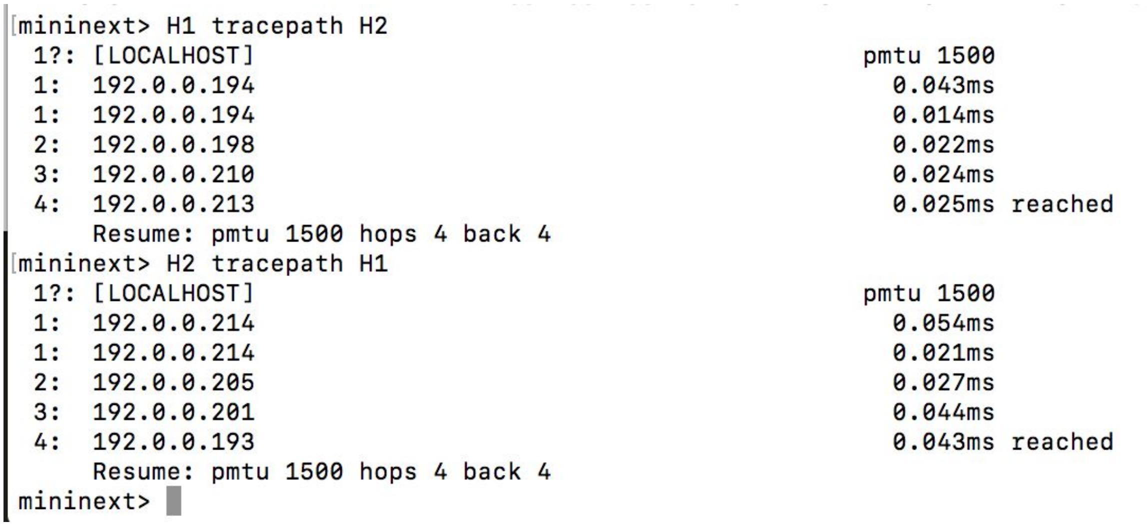
```
[mininext> R1 route
 Kernel IP routing table
 Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
 192.0.0.192     *               255.255.255.252 U     0      0        0 R1-eth0
 192.0.0.196     *               255.255.255.252 U     0      0        0 R1-eth1
 192.0.0.200     *               255.255.255.252 U     0      0        0 R1-eth2
 192.0.0.204     192.0.0.202     255.255.255.252 UG    0      0        0 R1-eth2
 192.0.0.208     192.0.0.198     255.255.255.252 UG    0      0        0 R1-eth1
 192.0.0.212     192.0.0.198     255.255.255.252 UG    0      0        0 R1-eth1
[mininext> R2 route
 Kernel IP routing table
 Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
 192.0.0.192     192.0.0.197     255.255.255.252 UG    0      0        0 R2-eth0
 192.0.0.196     *               255.255.255.252 U     0      0        0 R2-eth0
 192.0.0.200     192.0.0.197     255.255.255.252 UG    0      0        0 R2-eth0
 192.0.0.204     192.0.0.210     255.255.255.252 UG    0      0        0 R2-eth1
 192.0.0.208     *               255.255.255.252 U     0      0        0 R2-eth1
 192.0.0.212     192.0.0.210     255.255.255.252 UG    0      0        0 R2-eth1
[mininext> R3 route
 Kernel IP routing table
 Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
 192.0.0.192     192.0.0.201     255.255.255.252 UG    0      0        0 R3-eth0
 192.0.0.196     192.0.0.201     255.255.255.252 UG    0      0        0 R3-eth0
 192.0.0.200     *               255.255.255.252 U     0      0        0 R3-eth0
 192.0.0.204     *               255.255.255.252 U     0      0        0 R3-eth1
 192.0.0.208     192.0.0.201     255.255.255.252 UG    0      0        0 R3-eth0
 192.0.0.212     192.0.0.206     255.255.255.252 UG    0      0        0 R3-eth1
[mininext> R4 route
 Kernel IP routing table
 Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
 192.0.0.192     192.0.0.205     255.255.255.252 UG    0      0        0 R4-eth1
 192.0.0.196     192.0.0.209     255.255.255.252 UG    0      0        0 R4-eth0
 192.0.0.200     192.0.0.205     255.255.255.252 UG    0      0        0 R4-eth1
 192.0.0.204     *               255.255.255.252 U     0      0        0 R4-eth1
 192.0.0.208     *               255.255.255.252 U     0      0        0 R4-eth0
 192.0.0.212     *               255.255.255.252 U     0      0        0 R4-eth2
 mininext>
```

Routing Tables for hosts:



```
mininext> H1 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.0.0.194     0.0.0.0         UG    0      0        0 H1-eth0
192.0.0.192     *               255.255.255.252 U     0      0        0 H1-eth0
mininext> H2 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.0.0.214     0.0.0.0         UG    0      0        0 H2-eth0
192.0.0.212     *               255.255.255.252 U     0      0        0 H2-eth0
mininext>
```

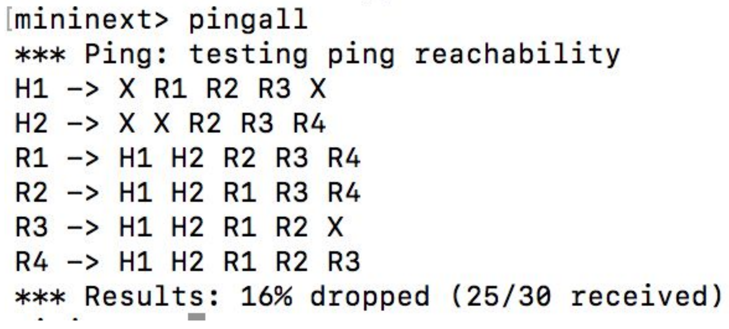
Tracepath between H1 to H2:



```
mininext> H1 tracepath H2
 1?: [LOCALHOST]                                         pmtu 1500
 1:  192.0.0.194                                           0.043ms
 1:  192.0.0.194                                           0.014ms
 2:  192.0.0.198                                           0.022ms
 3:  192.0.0.210                                           0.024ms
 4:  192.0.0.213                                           0.025ms reached
     Resume: pmtu 1500 hops 4 back 4
mininext> H2 tracepath H1
 1?: [LOCALHOST]                                         pmtu 1500
 1:  192.0.0.214                                           0.054ms
 1:  192.0.0.214                                           0.021ms
 2:  192.0.0.205                                           0.027ms
 3:  192.0.0.201                                           0.044ms
 4:  192.0.0.193                                           0.043ms reached
     Resume: pmtu 1500 hops 4 back 4
mininext>
```

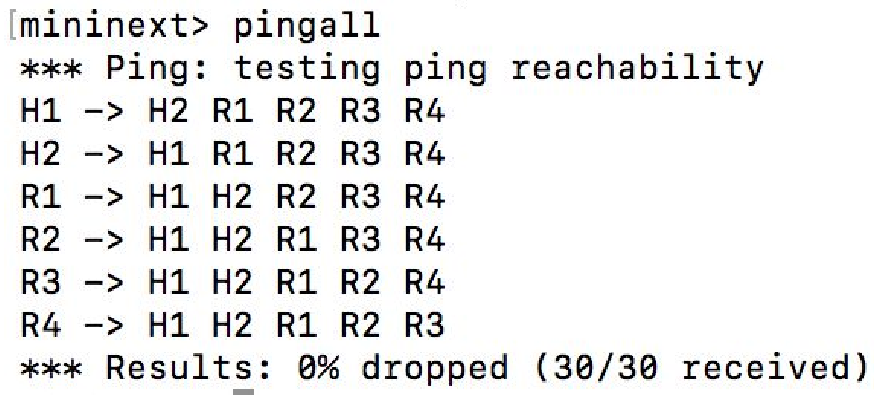
So 1 problem that I faced was that even after configuring the routes and ip forward variable, some packets were not going to other side.
This is output before enabling masquerading as the firewall was not allowing them to go. So had to run a script by **source masq inside mininext CLI** for certain interfaces

which allowed them to pass through interface.



```
mininext> pingall
*** Ping: testing ping reachability
H1 -> X R1 R2 R3 X
H2 -> X X R2 R3 R4
R1 -> H1 H2 R2 R3 R4
R2 -> H1 H2 R1 R3 R4
R3 -> H1 H2 R1 R2 X
R4 -> H1 H2 R1 R2 R3
*** Results: 16% dropped (25/30 received)
```

Output after enabling masquerading:



```
mininext> pingall
*** Ping: testing ping reachability
H1 -> H2 R1 R2 R3 R4
H2 -> H1 R1 R2 R3 R4
R1 -> H1 H2 R2 R3 R4
R2 -> H1 H2 R1 R3 R4
R3 -> H1 H2 R1 R2 R4
R4 -> H1 H2 R1 R2 R3
*** Results: 0% dropped (30/30 received)
```

As we see the packets dropped were 0.
Commands inside the masq script.

R1 iptables -t nat -A POSTROUTING -o R1-eth0 -j MASQUERADE
R2 iptables -t nat -A POSTROUTING -o R2-eth1 -j MASQUERADE
R4 iptables -t nat -A POSTROUTING -o R4-eth0 -j MASQUERADE
R1 iptables -t nat -A POSTROUTING -o R1-eth2 -j MASQUERADE
R3 iptables -t nat -A POSTROUTING -o R3-eth0 -j MASQUERADE
R4 iptables -t nat -A POSTROUTING -o R4-eth2 -j MASQUERADE

Also I created 6 subnets in this as there were 6 connections to be made. So for each subnet 2 ips were required and 1 for broadcast and subnet start address. So total 4 ips were required per subnet. So I took the subnet mask as 30.
First subnet IP is starting from 192.0.0.192-192.0.0.195, for second it is 192.0.0.196-192.0.0.199 and so on.

Question-2.) B-1,2,3)

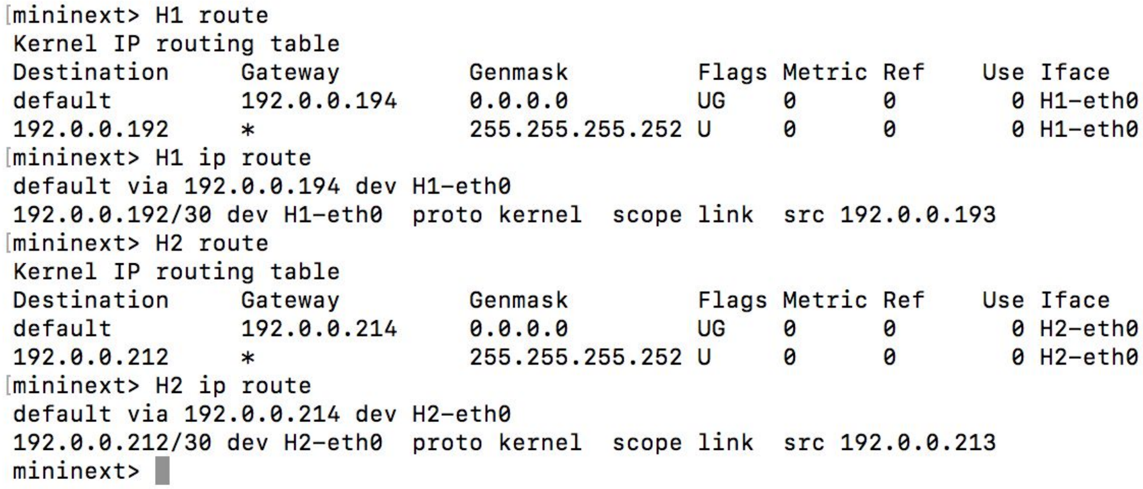
So first I put daemons file inside each of the router, to configure them as RIP daemon and enabled **zebra=yes, bgpd=no and ripd=yes** .
This enabled zebra and ripd daemons.
Took these files from /etc/quagga/. The file debian.conf is also taken from /etc/quagga and copied in each router.

In ripd.conf of R1, I added
router rip
network 192.0.0.192/30
network 192.0.0.196/30
network 192.0.0.200/30

These network are the starting addresses of the subnets to which this router belongs to. So R1 can listen to these 3 subnets.

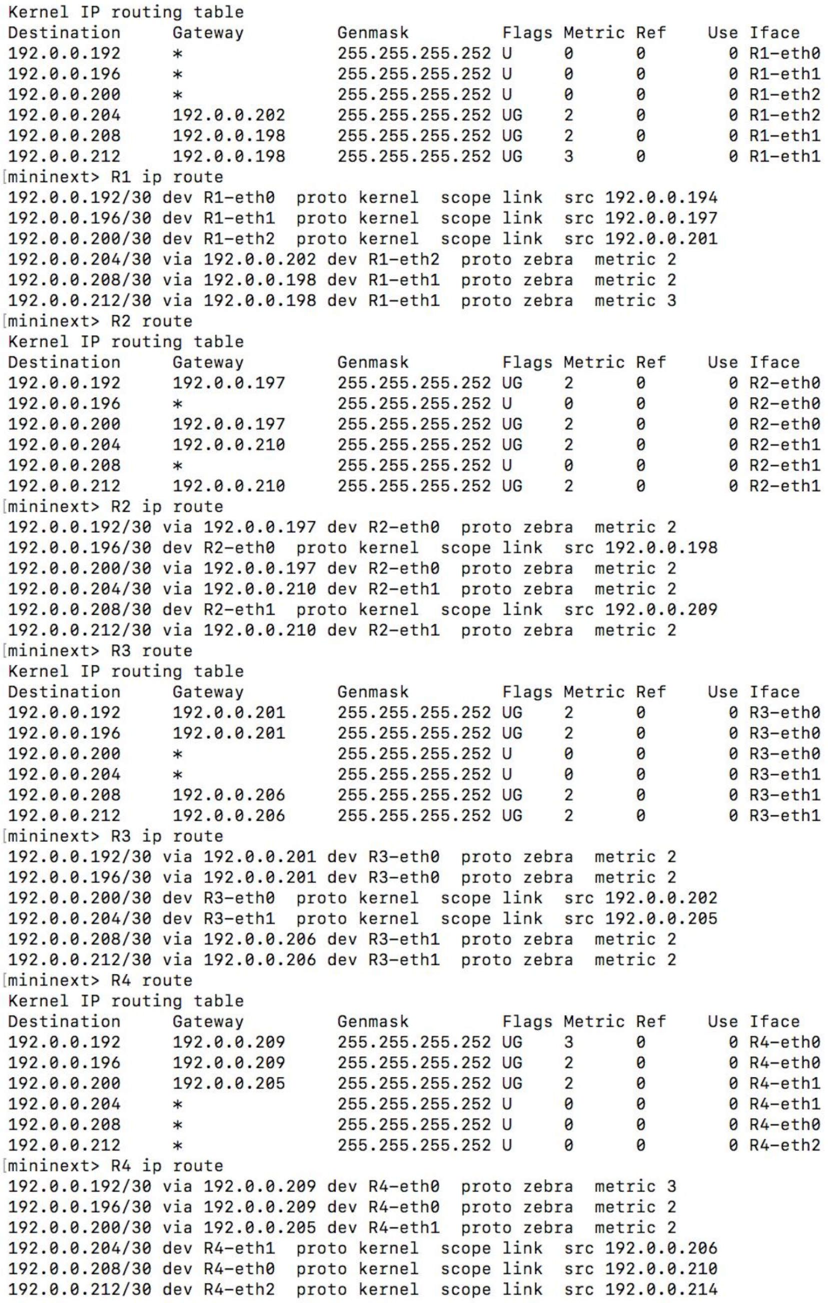
Starting **sudo service quagga start**, started the quagga service on each node after which I was able to ping other nodes as the routes were dynamically set up.

The routing tables were generated and H1 was able to ping H2.
Host Routing Tables:



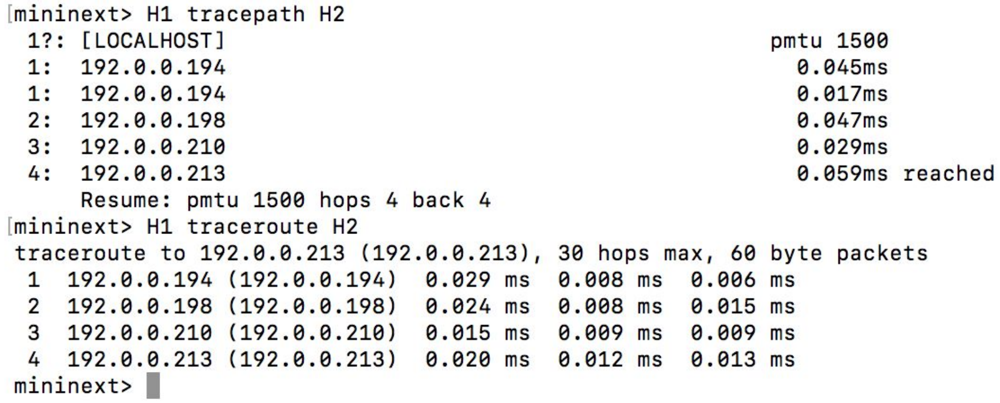
```
[mininext> H1 route
 Kernel IP routing table
 Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
 default         192.0.0.194     0.0.0.0         UG    0      0        0 H1-eth0
 192.0.0.192     *               255.255.255.252 U     0      0        0 H1-eth0
[mininext> H1 ip route
 default via 192.0.0.194 dev H1-eth0
 192.0.0.192/30 dev H1-eth0  proto kernel  scope link  src 192.0.0.193
[mininext> H2 route
 Kernel IP routing table
 Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
 default         192.0.0.214     0.0.0.0         UG    0      0        0 H2-eth0
 192.0.0.212     *               255.255.255.252 U     0      0        0 H2-eth0
[mininext> H2 ip route
 default via 192.0.0.214 dev H2-eth0
 192.0.0.212/30 dev H2-eth0  proto kernel  scope link  src 192.0.0.213
 mininext> 
```

Routers routing Tables:



```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.0.0.192     *               255.255.255.252 U     0      0        0 R1-eth0
192.0.0.196     *               255.255.255.252 U     0      0        0 R1-eth1
192.0.0.200     *               255.255.255.252 U     0      0        0 R1-eth2
192.0.0.204     192.0.0.202     255.255.255.252 UG    2      0        0 R1-eth2
192.0.0.208     192.0.0.198     255.255.255.252 UG    2      0        0 R1-eth1
192.0.0.212     192.0.0.198     255.255.255.252 UG    3      0        0 R1-eth1
mininext> R1 ip route
192.0.0.192/30 dev R1-eth0  proto kernel  scope link  src 192.0.0.194
192.0.0.196/30 dev R1-eth1  proto kernel  scope link  src 192.0.0.197
192.0.0.200/30 dev R1-eth2  proto kernel  scope link  src 192.0.0.201
192.0.0.204/30 via 192.0.0.202 dev R1-eth2  proto zebra  metric 2
192.0.0.208/30 via 192.0.0.198 dev R1-eth1  proto zebra  metric 2
192.0.0.212/30 via 192.0.0.198 dev R1-eth1  proto zebra  metric 3
mininext> R2 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.0.0.192     192.0.0.197     255.255.255.252 UG    2      0        0 R2-eth0
192.0.0.196     *               255.255.255.252 U     0      0        0 R2-eth0
192.0.0.200     192.0.0.197     255.255.255.252 UG    2      0        0 R2-eth0
192.0.0.204     192.0.0.210     255.255.255.252 UG    2      0        0 R2-eth1
192.0.0.208     *               255.255.255.252 U     0      0        0 R2-eth1
192.0.0.212     192.0.0.210     255.255.255.252 UG    2      0        0 R2-eth1
mininext> R2 ip route
192.0.0.192/30 via 192.0.0.197 dev R2-eth0  proto zebra  metric 2
192.0.0.196/30 dev R2-eth0  proto kernel  scope link  src 192.0.0.198
192.0.0.200/30 via 192.0.0.197 dev R2-eth0  proto zebra  metric 2
192.0.0.204/30 via 192.0.0.210 dev R2-eth1  proto zebra  metric 2
192.0.0.208/30 dev R2-eth1  proto kernel  scope link  src 192.0.0.209
192.0.0.212/30 via 192.0.0.210 dev R2-eth1  proto zebra  metric 2
mininext> R3 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.0.0.192     192.0.0.201     255.255.255.252 UG    2      0        0 R3-eth0
192.0.0.196     192.0.0.201     255.255.255.252 UG    2      0        0 R3-eth0
192.0.0.200     *               255.255.255.252 U     0      0        0 R3-eth0
192.0.0.204     *               255.255.255.252 U     0      0        0 R3-eth1
192.0.0.208     192.0.0.206     255.255.255.252 UG    2      0        0 R3-eth1
192.0.0.212     192.0.0.206     255.255.255.252 UG    2      0        0 R3-eth1
mininext> R3 ip route
192.0.0.192/30 via 192.0.0.201 dev R3-eth0  proto zebra  metric 2
192.0.0.196/30 via 192.0.0.201 dev R3-eth0  proto zebra  metric 2
192.0.0.200/30 dev R3-eth0  proto kernel  scope link  src 192.0.0.202
192.0.0.204/30 dev R3-eth1  proto kernel  scope link  src 192.0.0.205
192.0.0.208/30 via 192.0.0.206 dev R3-eth1  proto zebra  metric 2
192.0.0.212/30 via 192.0.0.206 dev R3-eth1  proto zebra  metric 2
mininext> R4 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.0.0.192     192.0.0.209     255.255.255.252 UG    3      0        0 R4-eth0
192.0.0.196     192.0.0.209     255.255.255.252 UG    2      0        0 R4-eth0
192.0.0.200     192.0.0.205     255.255.255.252 UG    2      0        0 R4-eth1
192.0.0.204     *               255.255.255.252 U     0      0        0 R4-eth1
192.0.0.208     *               255.255.255.252 U     0      0        0 R4-eth0
192.0.0.212     *               255.255.255.252 U     0      0        0 R4-eth2
mininext> R4 ip route
192.0.0.192/30 via 192.0.0.209 dev R4-eth0  proto zebra  metric 3
192.0.0.196/30 via 192.0.0.209 dev R4-eth0  proto zebra  metric 2
192.0.0.200/30 via 192.0.0.205 dev R4-eth1  proto zebra  metric 2
192.0.0.204/30 dev R4-eth1  proto kernel  scope link  src 192.0.0.206
192.0.0.208/30 dev R4-eth0  proto kernel  scope link  src 192.0.0.210
192.0.0.212/30 dev R4-eth2  proto kernel  scope link  src 192.0.0.214
```

Path between H1 and H2:

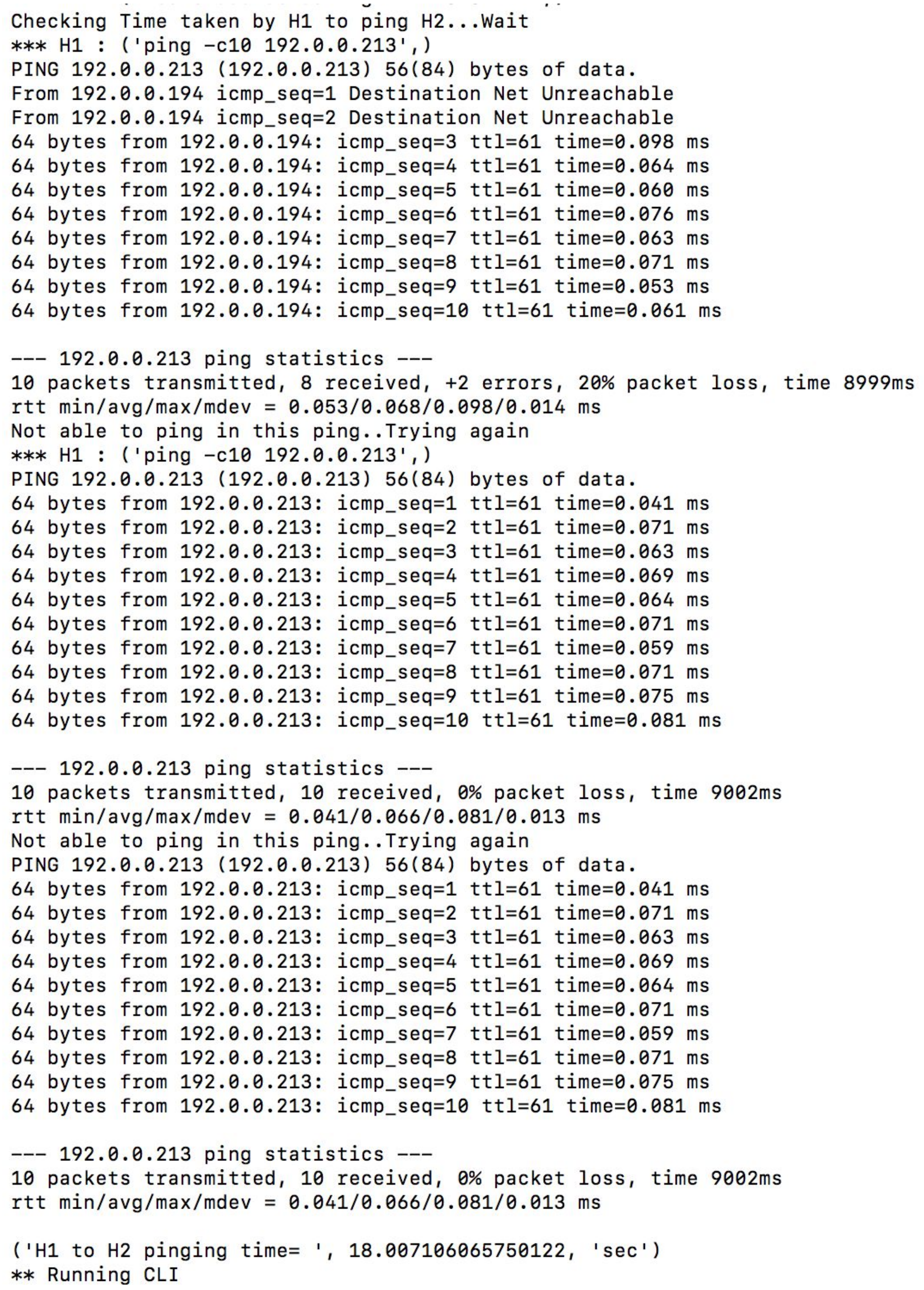


```
[mininext> H1 tracepath H2
  1?: [LOCALHOST]                                         pmtu 1500
  1:  192.0.0.194                                           0.045ms
  1:  192.0.0.194                                           0.017ms
  2:  192.0.0.198                                           0.047ms
  3:  192.0.0.210                                           0.029ms
  4:  192.0.0.213                                           0.059ms reached
      Resume: pmtu 1500 hops 4 back 4
[mininext> H1 traceroute H2
 traceroute to 192.0.0.213 (192.0.0.213), 30 hops max, 60 byte packets
  1  192.0.0.194 (192.0.0.194)  0.029 ms  0.008 ms  0.006 ms
  2  192.0.0.198 (192.0.0.198)  0.024 ms  0.008 ms  0.015 ms
  3  192.0.0.210 (192.0.0.210)  0.015 ms  0.009 ms  0.009 ms
  4  192.0.0.213 (192.0.0.213)  0.020 ms  0.012 ms  0.013 ms
 mininext>
```

Convergence Time: H1 pinging time to H2: 18 sec as shown below:

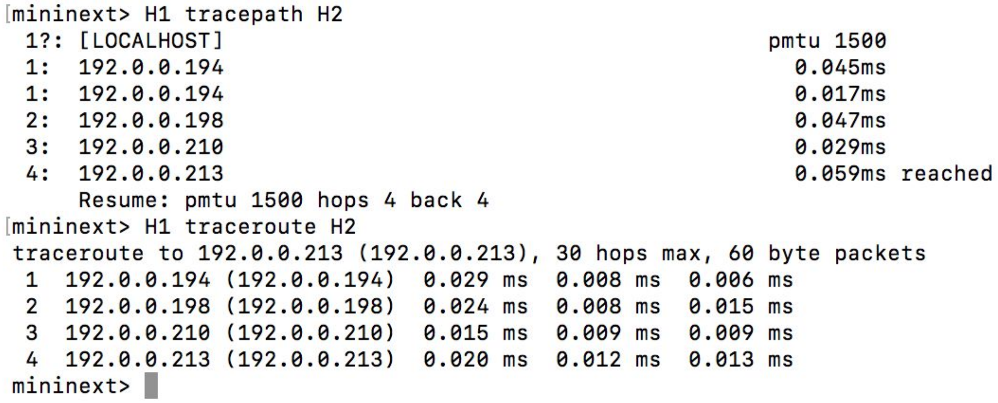
Code is in start.py of part B

Run start.py as python start.py 1 to get this output:



```
Checking Time taken by H1 to ping H2...Wait
*** H1 : ('ping -c10 192.0.0.213',)
PING 192.0.0.213 (192.0.0.213) 56(84) bytes of data.
From 192.0.0.194 icmp_seq=1 Destination Net Unreachable
From 192.0.0.194 icmp_seq=2 Destination Net Unreachable
64 bytes from 192.0.0.194: icmp_seq=3 ttl=61 time=0.098 ms
64 bytes from 192.0.0.194: icmp_seq=4 ttl=61 time=0.064 ms
64 bytes from 192.0.0.194: icmp_seq=5 ttl=61 time=0.060 ms
64 bytes from 192.0.0.194: icmp_seq=6 ttl=61 time=0.076 ms
64 bytes from 192.0.0.194: icmp_seq=7 ttl=61 time=0.063 ms
64 bytes from 192.0.0.194: icmp_seq=8 ttl=61 time=0.071 ms
64 bytes from 192.0.0.194: icmp_seq=9 ttl=61 time=0.053 ms
64 bytes from 192.0.0.194: icmp_seq=10 ttl=61 time=0.061 ms

--- 192.0.0.213 ping statistics ---
10 packets transmitted, 8 received, +2 errors, 20% packet loss, time 8999ms
rtt min/avg/max/mdev = 0.053/0.068/0.098/0.014 ms
Not able to ping in this ping..Trying again
*** H1 : ('ping -c10 192.0.0.213',)
PING 192.0.0.213 (192.0.0.213) 56(84) bytes of data.
64 bytes from 192.0.0.213: icmp_seq=1 ttl=61 time=0.041 ms
64 bytes from 192.0.0.213: icmp_seq=2 ttl=61 time=0.071 ms
64 bytes from 192.0.0.213: icmp_seq=3 ttl=61 time=0.063 ms
64 bytes from 192.0.0.213: icmp_seq=4 ttl=61 time=0.069 ms
64 bytes from 192.0.0.213: icmp_seq=5 ttl=61 time=0.064 ms
64 bytes from 192.0.0.213: icmp_seq=6 ttl=61 time=0.071 ms
64 bytes from 192.0.0.213: icmp_seq=7 ttl=61 time=0.059 ms
64 bytes from 192.0.0.213: icmp_seq=8 ttl=61 time=0.071 ms
64 bytes from 192.0.0.213: icmp_seq=9 ttl=61 time=0.075 ms
64 bytes from 192.0.0.213: icmp_seq=10 ttl=61 time=0.081 ms

--- 192.0.0.213 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9002ms
rtt min/avg/max/mdev = 0.041/0.066/0.081/0.013 ms
Not able to ping in this ping..Trying again
PING 192.0.0.213 (192.0.0.213) 56(84) bytes of data.
64 bytes from 192.0.0.213: icmp_seq=1 ttl=61 time=0.041 ms
64 bytes from 192.0.0.213: icmp_seq=2 ttl=61 time=0.071 ms
64 bytes from 192.0.0.213: icmp_seq=3 ttl=61 time=0.063 ms
64 bytes from 192.0.0.213: icmp_seq=4 ttl=61 time=0.069 ms
64 bytes from 192.0.0.213: icmp_seq=5 ttl=61 time=0.064 ms
64 bytes from 192.0.0.213: icmp_seq=6 ttl=61 time=0.071 ms
64 bytes from 192.0.0.213: icmp_seq=7 ttl=61 time=0.059 ms
64 bytes from 192.0.0.213: icmp_seq=8 ttl=61 time=0.071 ms
64 bytes from 192.0.0.213: icmp_seq=9 ttl=61 time=0.075 ms
64 bytes from 192.0.0.213: icmp_seq=10 ttl=61 time=0.081 ms

--- 192.0.0.213 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9002ms
rtt min/avg/max/mdev = 0.041/0.066/0.081/0.013 ms

('H1 to H2 pinging time= ', 18.007106065750122, 'sec')
** Running CLI
```

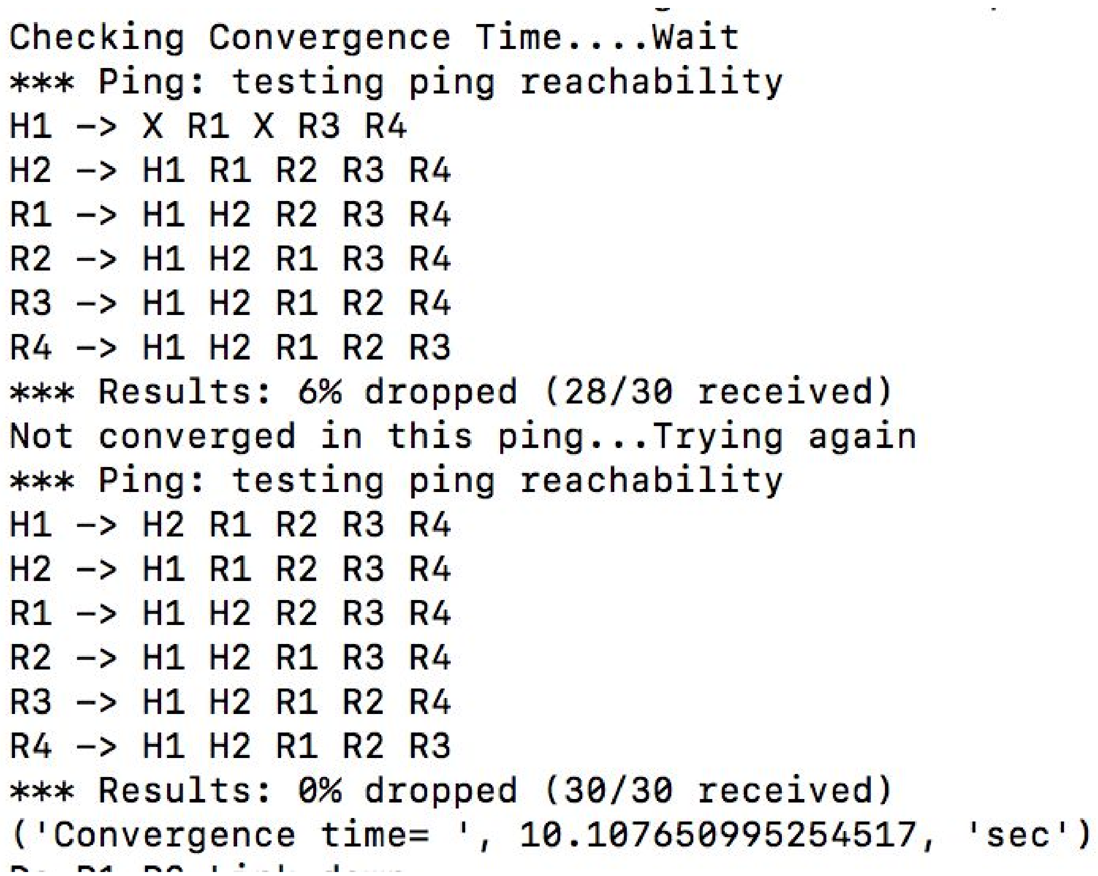
H1 to H2 path:



```
[mininext> H1 tracepath H2
 1?: [LOCALHOST]                                         pmtu 1500
 1:  192.0.0.194                                           0.045ms
 1:  192.0.0.194                                           0.017ms
 2:  192.0.0.198                                           0.047ms
 3:  192.0.0.210                                           0.029ms
 4:  192.0.0.213                                           0.059ms reached
     Resume: pmtu 1500 hops 4 back 4
[mininext> H1 traceroute H2
traceroute to 192.0.0.213 (192.0.0.213), 30 hops max, 60 byte packets
 1  192.0.0.194 (192.0.0.194)  0.029 ms  0.008 ms  0.006 ms
 2  192.0.0.198 (192.0.0.198)  0.024 ms  0.008 ms  0.015 ms
 3  192.0.0.210 (192.0.0.210)  0.015 ms  0.009 ms  0.009 ms
 4  192.0.0.213 (192.0.0.213)  0.020 ms  0.012 ms  0.013 ms
mininext>
```

All to all convergence time:

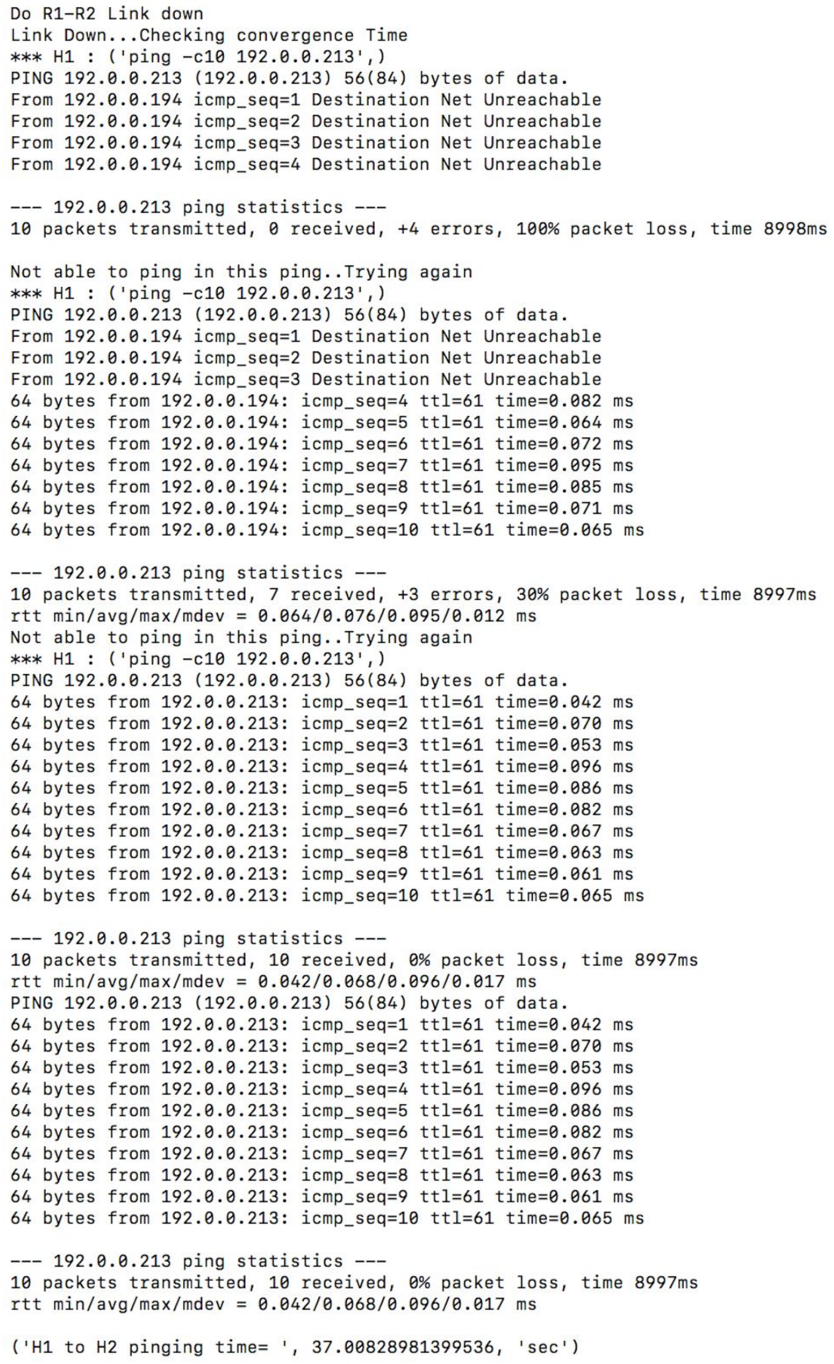
Used pingAll() function in start.py



```
Checking Convergence Time....Wait
*** Ping: testing ping reachability
H1 -> X R1 X R3 R4
H2 -> H1 R1 R2 R3 R4
R1 -> H1 H2 R2 R3 R4
R2 -> H1 H2 R1 R3 R4
R3 -> H1 H2 R1 R2 R4
R4 -> H1 H2 R1 R2 R3
*** Results: 6% dropped (28/30 received)
Not converged in this ping...Trying again
*** Ping: testing ping reachability
H1 -> H2 R1 R2 R3 R4
H2 -> H1 R1 R2 R3 R4
R1 -> H1 H2 R2 R3 R4
R2 -> H1 H2 R1 R3 R4
R3 -> H1 H2 R1 R2 R4
R4 -> H1 H2 R1 R2 R3
*** Results: 0% dropped (30/30 received)
('Convergence time= ', 10.107650995254517, 'sec')
```

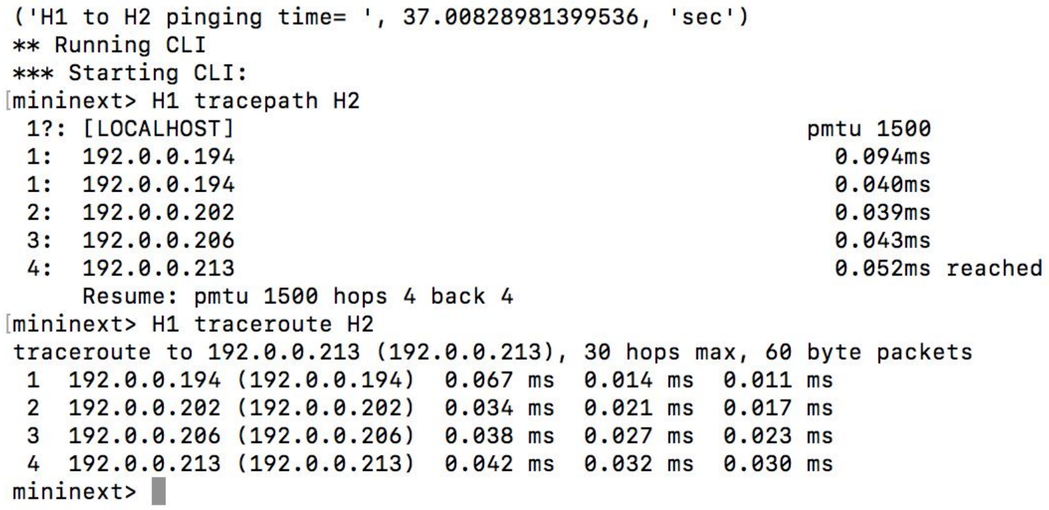
Run start.py as python start.py 2 to get above output and it will also bring the link down (R1-R2) and will start converging again:

Convergence time after link down: 37 sec.



```
Do R1-R2 Link down
Link Down...Checking convergence Time
*** H1 : ('ping -c10 192.0.0.213',)
PING 192.0.0.213 (192.0.0.213) 56(84) bytes of data.
From 192.0.0.194 icmp_seq=1 Destination Net Unreachable
From 192.0.0.194 icmp_seq=2 Destination Net Unreachable
From 192.0.0.194 icmp_seq=3 Destination Net Unreachable
From 192.0.0.194 icmp_seq=4 Destination Net Unreachable

--- 192.0.0.213 ping statistics ---
10 packets transmitted, 0 received, +4 errors, 100% packet loss, time 8998ms

Not able to ping in this ping..Trying again
*** H1 : ('ping -c10 192.0.0.213',)
PING 192.0.0.213 (192.0.0.213) 56(84) bytes of data.
From 192.0.0.194 icmp_seq=1 Destination Net Unreachable
From 192.0.0.194 icmp_seq=2 Destination Net Unreachable
From 192.0.0.194 icmp_seq=3 Destination Net Unreachable
64 bytes from 192.0.0.194: icmp_seq=4 ttl=61 time=0.082 ms
64 bytes from 192.0.0.194: icmp_seq=5 ttl=61 time=0.064 ms
64 bytes from 192.0.0.194: icmp_seq=6 ttl=61 time=0.072 ms
64 bytes from 192.0.0.194: icmp_seq=7 ttl=61 time=0.095 ms
64 bytes from 192.0.0.194: icmp_seq=8 ttl=61 time=0.085 ms
64 bytes from 192.0.0.194: icmp_seq=9 ttl=61 time=0.071 ms
64 bytes from 192.0.0.194: icmp_seq=10 ttl=61 time=0.065 ms

--- 192.0.0.213 ping statistics ---
10 packets transmitted, 7 received, +3 errors, 30% packet loss, time 8997ms
rtt min/avg/max/mdev = 0.064/0.076/0.095/0.012 ms
Not able to ping in this ping..Trying again
*** H1 : ('ping -c10 192.0.0.213',)
PING 192.0.0.213 (192.0.0.213) 56(84) bytes of data.
64 bytes from 192.0.0.213: icmp_seq=1 ttl=61 time=0.042 ms
64 bytes from 192.0.0.213: icmp_seq=2 ttl=61 time=0.070 ms
64 bytes from 192.0.0.213: icmp_seq=3 ttl=61 time=0.053 ms
64 bytes from 192.0.0.213: icmp_seq=4 ttl=61 time=0.096 ms
64 bytes from 192.0.0.213: icmp_seq=5 ttl=61 time=0.086 ms
64 bytes from 192.0.0.213: icmp_seq=6 ttl=61 time=0.082 ms
64 bytes from 192.0.0.213: icmp_seq=7 ttl=61 time=0.067 ms
64 bytes from 192.0.0.213: icmp_seq=8 ttl=61 time=0.063 ms
64 bytes from 192.0.0.213: icmp_seq=9 ttl=61 time=0.061 ms
64 bytes from 192.0.0.213: icmp_seq=10 ttl=61 time=0.065 ms

--- 192.0.0.213 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8997ms
rtt min/avg/max/mdev = 0.042/0.068/0.096/0.017 ms
PING 192.0.0.213 (192.0.0.213) 56(84) bytes of data.
64 bytes from 192.0.0.213: icmp_seq=1 ttl=61 time=0.042 ms
64 bytes from 192.0.0.213: icmp_seq=2 ttl=61 time=0.070 ms
64 bytes from 192.0.0.213: icmp_seq=3 ttl=61 time=0.053 ms
64 bytes from 192.0.0.213: icmp_seq=4 ttl=61 time=0.096 ms
64 bytes from 192.0.0.213: icmp_seq=5 ttl=61 time=0.086 ms
64 bytes from 192.0.0.213: icmp_seq=6 ttl=61 time=0.082 ms
64 bytes from 192.0.0.213: icmp_seq=7 ttl=61 time=0.067 ms
64 bytes from 192.0.0.213: icmp_seq=8 ttl=61 time=0.063 ms
64 bytes from 192.0.0.213: icmp_seq=9 ttl=61 time=0.061 ms
64 bytes from 192.0.0.213: icmp_seq=10 ttl=61 time=0.065 ms

--- 192.0.0.213 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8997ms
rtt min/avg/max/mdev = 0.042/0.068/0.096/0.017 ms

('H1 to H2 pinging time= ', 37.00828981399536, 'sec')
```

Path between H1-H2 after link was down. Path changed and packets were not going from R3.



```
('H1 to H2 pinging time= ', 37.00828981399536, 'sec')
** Running CLI
*** Starting CLI:
[mininext> H1 tracepath H2
 1?: [LOCALHOST]                                          pmtu 1500
 1:  192.0.0.194                                            0.094ms
 1:  192.0.0.194                                            0.040ms
 2:  192.0.0.202                                            0.039ms
 3:  192.0.0.206                                            0.043ms
 4:  192.0.0.213                                            0.052ms reached
     Resume: pmtu 1500 hops 4 back 4
[mininext> H1 traceroute H2
traceroute to 192.0.0.213 (192.0.0.213), 30 hops max, 60 byte packets
 1  192.0.0.194 (192.0.0.194)  0.067 ms  0.014 ms  0.011 ms
 2  192.0.0.202 (192.0.0.202)  0.034 ms  0.021 ms  0.017 ms
 3  192.0.0.206 (192.0.0.206)  0.038 ms  0.027 ms  0.023 ms
 4  192.0.0.213 (192.0.0.213)  0.042 ms  0.032 ms  0.030 ms
mininext>
```

Also to get the link down, I used the command in start.py
net.configLinkStatus('R1', 'R2', 'down')

Question .3) C1,2 and 3.) Code is in folder RIPLite.py.
I had 1 file for each host and router to just supply the neighbor info to each node initially and the main logic is in RIPLite.py.
After starting the mininext, run **source runn** script to start the protocol.

Application routing table at each node is present in **RipLogsInitial folder**. Each node has its own file as the algo was distributed.

0.6798 seconds was the time taken by protocol to converge. Basically I printed the logs after each update and then analysed the logs to find the last time a dist vector of some node updated. As the algo was distributed there was no way to centrally declare convergence. So choose this path.

Now after changing the weight from 6 to 1 between R1-R3, we got new routing tables which are present in **RipLogs-6-to-1** folder.
Found the convergence time in same way.
0.3702 seconds was the convergence time.

In the last, if the weight becomes negative I treat this as link down and set his weight as infinity. So we got new routing tables after this which is present in **RipLogs-NegWeight** folder.
In this I change weight in outside Routing Info.txt by sshing from other terminal.