LAB ASSIGNMENT-1

NAME	: TALANKI SUBBA PRANAV KUMAR
REG. NO	: 19BDS0060
COURSE CODE	: CSE1004
FACULTY	: SANTHI H
SLOT	: L25 + L26

Activity-1

Execute the following commands with ANY FIVE additional options:

1. ifconfig

2. ping

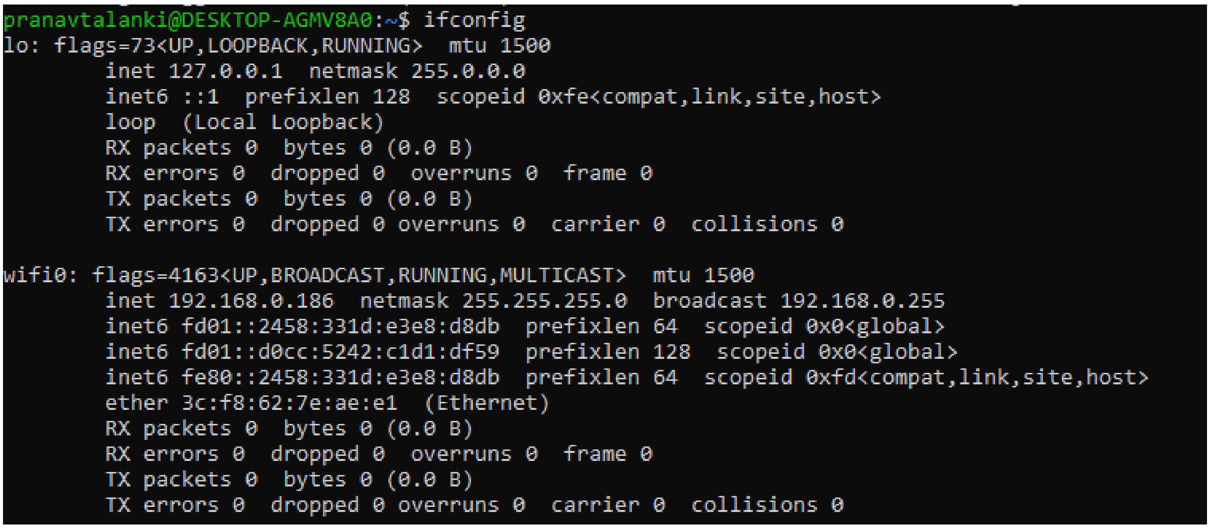
3. traceroute

4. netstat

5.nslookup

1.ifconfig

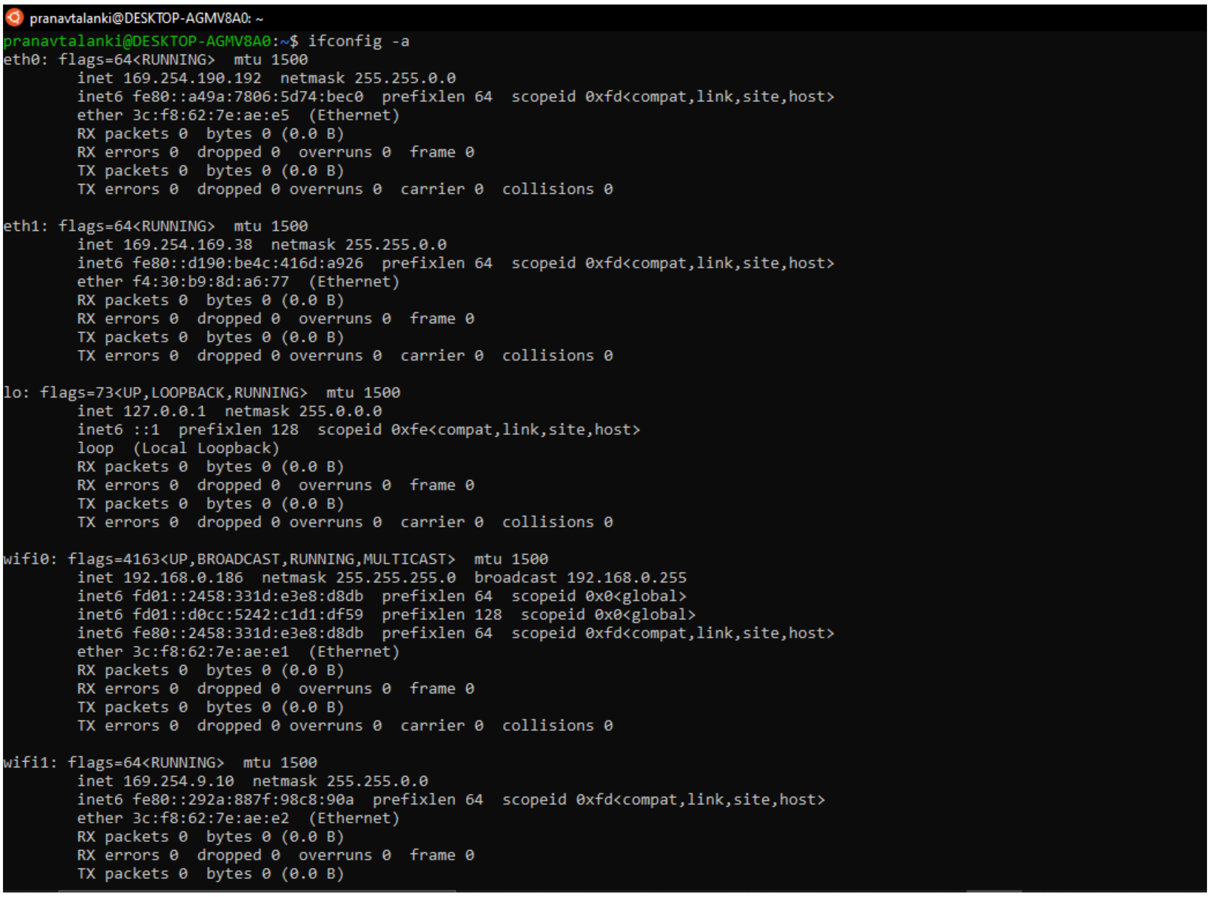
ifconfig command will provide a fairly complete description of the current state of all active network interfaces.



```
pranavtalanki@DESKTOP-AGMV8A0:~$ ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 1500
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0xfe<compat,link,site,host>
        loop  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wifi0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.186  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fd01::2458:331d:e3e8:d8db  prefixlen 64  scopeid 0x0<global>
        inet6 fd01::d0cc:5242:c1d1:df59  prefixlen 128  scopeid 0x0<global>
        inet6 fe80::2458:331d:e3e8:d8db  prefixlen 64  scopeid 0xfd<compat,link,site,host>
        ether 3c:f8:62:7e:ae:e1  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

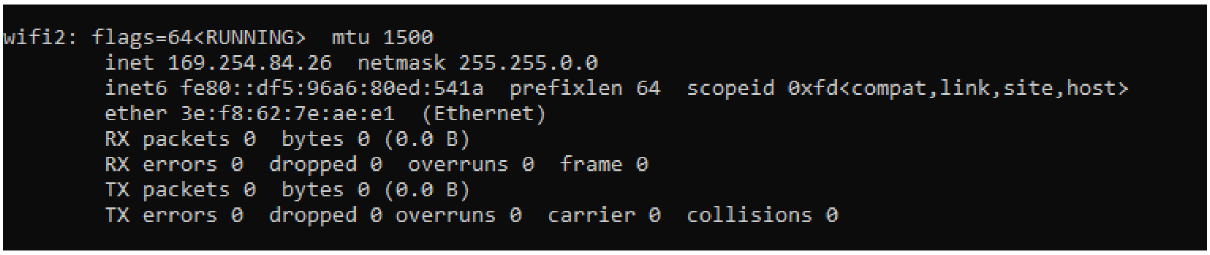
a. -a

This option will display all available interface details if it is disable also.



pranavtalanki@DESKTOP-AGMV8A0: ~

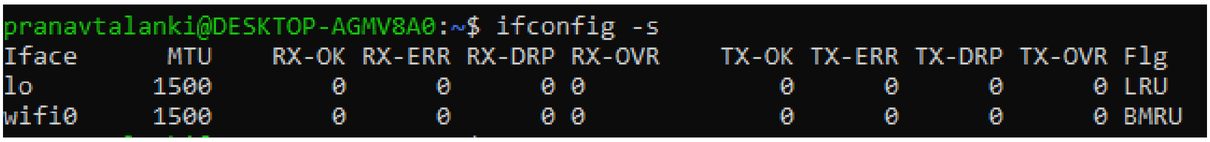
```
pranavtalanki@DESKTOP-AGMV8A0:~$ ifconfig -a
eth0: flags=64<RUNNING>  mtu 1500
        inet 169.254.190.192  netmask 255.255.0.0
        inet6 fe80::a49a:7806:5d74:bec0  prefixlen 64  scopeid 0xfd<compat,link,site,host>
        ether 3c:f8:62:7e:ae:e5  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=64<RUNNING>  mtu 1500
        inet 169.254.169.38  netmask 255.255.0.0
        inet6 fe80::d190:be4c:416d:a926  prefixlen 64  scopeid 0xfd<compat,link,site,host>
        ether f4:30:b9:8d:a6:77  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 1500
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0xfe<compat,link,site,host>
        loop  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wifi0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.186  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fd01::2458:331d:e3e8:d8db  prefixlen 64  scopeid 0x0<global>
        inet6 fd01::d0cc:5242:c1d1:df59  prefixlen 128  scopeid 0x0<global>
        inet6 fe80::2458:331d:e3e8:d8db  prefixlen 64  scopeid 0xfd<compat,link,site,host>
        ether 3c:f8:62:7e:ae:e1  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wifi1: flags=64<RUNNING>  mtu 1500
        inet 169.254.9.10  netmask 255.255.0.0
        inet6 fe80::292a:887f:98c8:90a  prefixlen 64  scopeid 0xfd<compat,link,site,host>
        ether 3c:f8:62:7e:ae:e2  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
```



```
wifi2: flags=64<RUNNING>  mtu 1500
        inet 169.254.84.26  netmask 255.255.0.0
        inet6 fe80::df5:96a6:80ed:541a  prefixlen 64  scopeid 0xfd<compat,link,site,host>
        ether 3e:f8:62:7e:ae:e1  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

b. -s

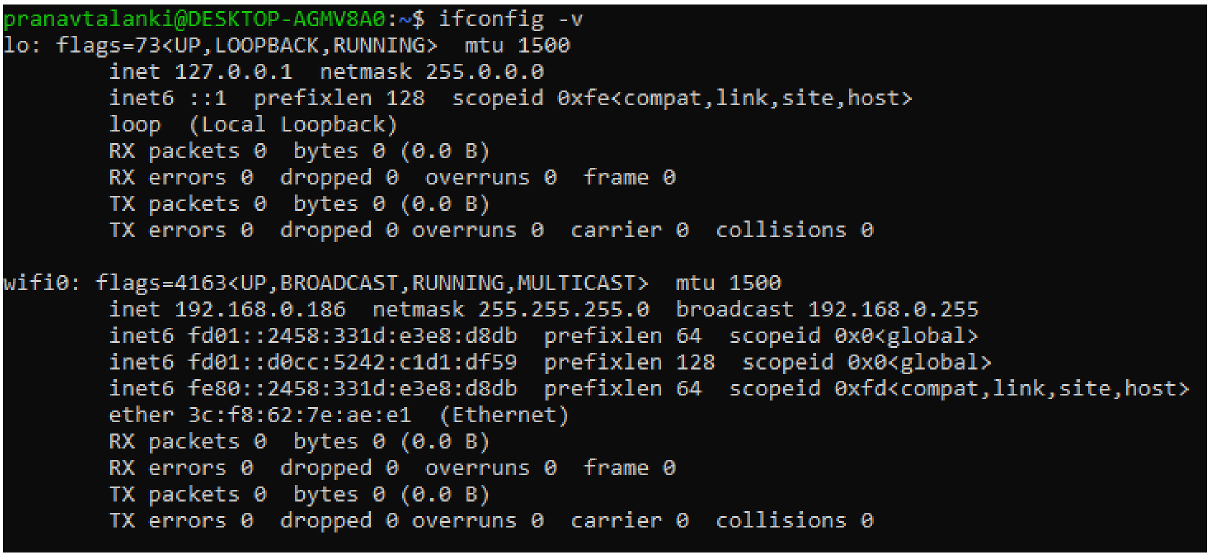
This is the "short listing" option, which shows a one-line summarized listing of data about each interface. The information returned is about interface activity, and not configuration.



```
pranavtalanki@DESKTOP-AGMV8A0:~$ ifconfig -s
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
lo        1500        0      0      0 0             0      0      0      0 LRU
wifi0     1500        0      0      0 0             0      0      0      0 BMRU
```

c. -v

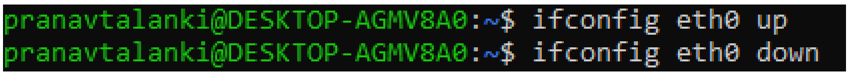
This "verbose" option returns extra information when there are certain types of error conditions to help with troubleshooting.



```
pranavtalanki@DESKTOP-AGMV8A0:~$ ifconfig -v
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 1500
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0xfe<compat,link,site,host>
        loop  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wifi0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.186  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fd01::2458:331d:e3e8:d8db  prefixlen 64  scopeid 0x0<global>
        inet6 fd01::d0cc:5242:c1d1:df59  prefixlen 128  scopeid 0x0<global>
        inet6 fe80::2458:331d:e3e8:d8db  prefixlen 64  scopeid 0xfd<compat,link,site,host>
        ether 3c:f8:62:7e:ae:e1  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

d. Up

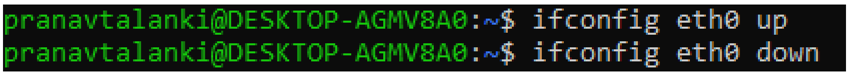
This activates an interface if it is not already active. For instance, *ifconfig eth0 up* causes eth0 to be activated.



```
pranavtalanki@DESKTOP-AGMV8A0:~$ ifconfig eth0 up
pranavtalanki@DESKTOP-AGMV8A0:~$ ifconfig eth0 down
```

e. Down

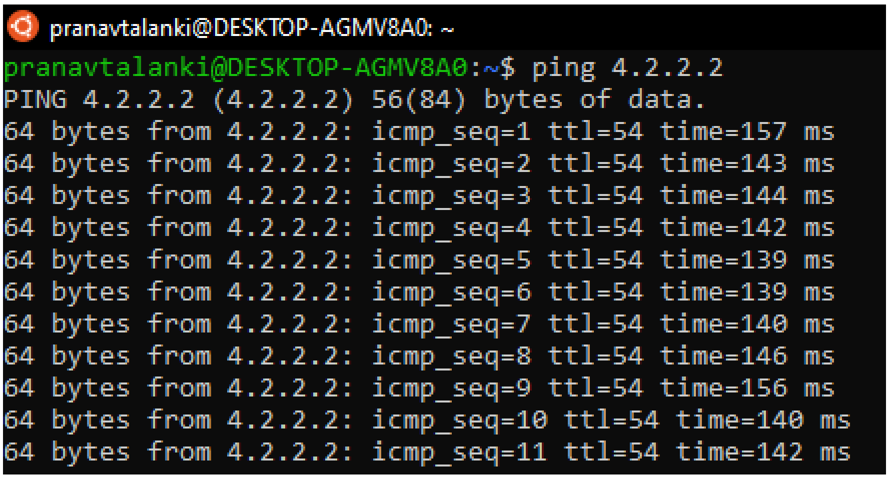
The counterpart to up, this deactivates the specified interface. Thus, *ifconfig eth0 down* causes eth0 to be deactivated if it is currently active.



```
pranavtalanki@DESKTOP-AGMV8A0:~$ ifconfig eth0 up
pranavtalanki@DESKTOP-AGMV8A0:~$ ifconfig eth0 down
```

2.ping

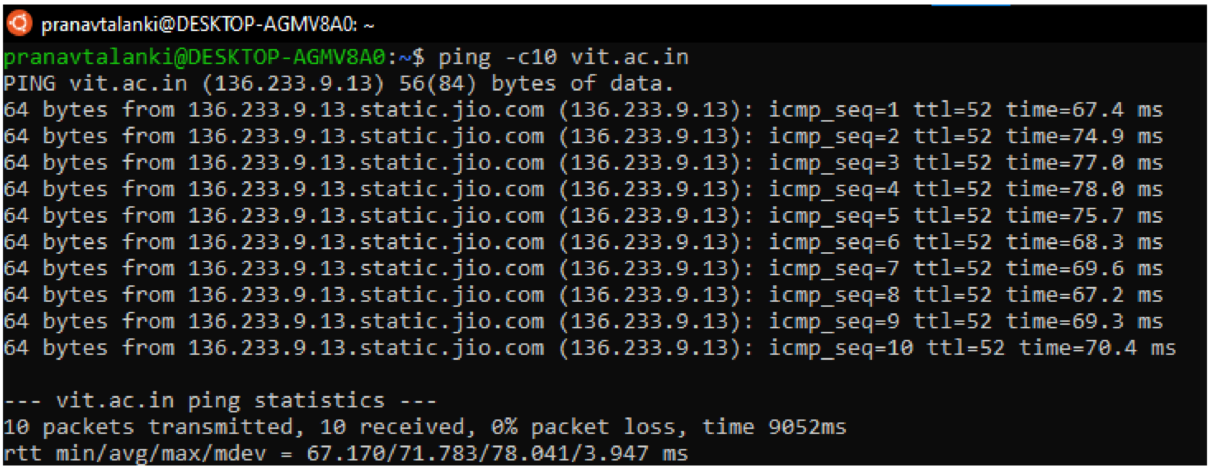
PING (Packet INternet Groper) command is the best way to test connectivity between two nodes.



```
pranavtalanki@DESKTOP-AGMV8A0:~$ ping 4.2.2.2
PING 4.2.2.2 (4.2.2.2) 56(84) bytes of data.
64 bytes from 4.2.2.2: icmp_seq=1 ttl=54 time=157 ms
64 bytes from 4.2.2.2: icmp_seq=2 ttl=54 time=143 ms
64 bytes from 4.2.2.2: icmp_seq=3 ttl=54 time=144 ms
64 bytes from 4.2.2.2: icmp_seq=4 ttl=54 time=142 ms
64 bytes from 4.2.2.2: icmp_seq=5 ttl=54 time=139 ms
64 bytes from 4.2.2.2: icmp_seq=6 ttl=54 time=139 ms
64 bytes from 4.2.2.2: icmp_seq=7 ttl=54 time=140 ms
64 bytes from 4.2.2.2: icmp_seq=8 ttl=54 time=146 ms
64 bytes from 4.2.2.2: icmp_seq=9 ttl=54 time=156 ms
64 bytes from 4.2.2.2: icmp_seq=10 ttl=54 time=140 ms
64 bytes from 4.2.2.2: icmp_seq=11 ttl=54 time=142 ms
```

a. -c

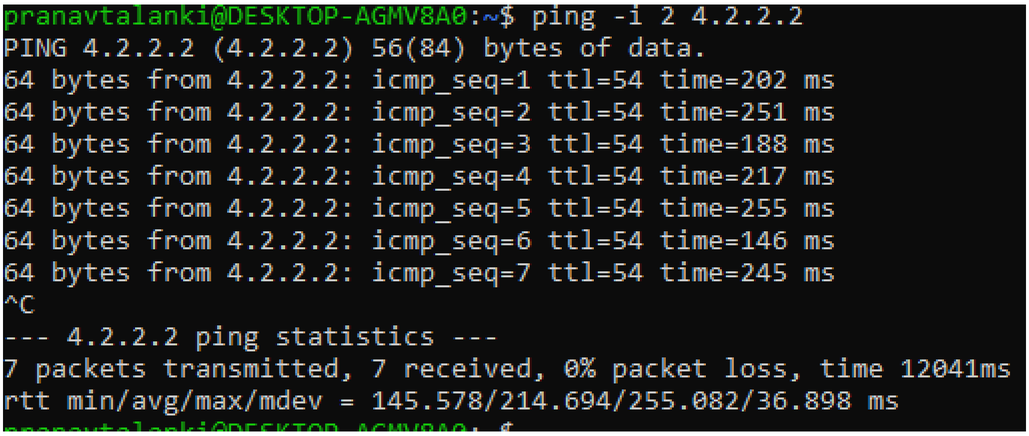
Ping with –c option exit after N number of request (success or error respond).



```
pranavtalanki@DESKTOP-AGMV8A0:~$ ping -c10 vit.ac.in
PING vit.ac.in (136.233.9.13) 56(84) bytes of data.
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=1 ttl=52 time=67.4 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=2 ttl=52 time=74.9 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=3 ttl=52 time=77.0 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=4 ttl=52 time=78.0 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=5 ttl=52 time=75.7 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=6 ttl=52 time=68.3 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=7 ttl=52 time=69.6 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=8 ttl=52 time=67.2 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=9 ttl=52 time=69.3 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=10 ttl=52 time=70.4 ms

--- vit.ac.in ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9052ms
rtt min/avg/max/mdev = 67.170/71.783/78.041/3.947 ms
```

b. -i <TTL>

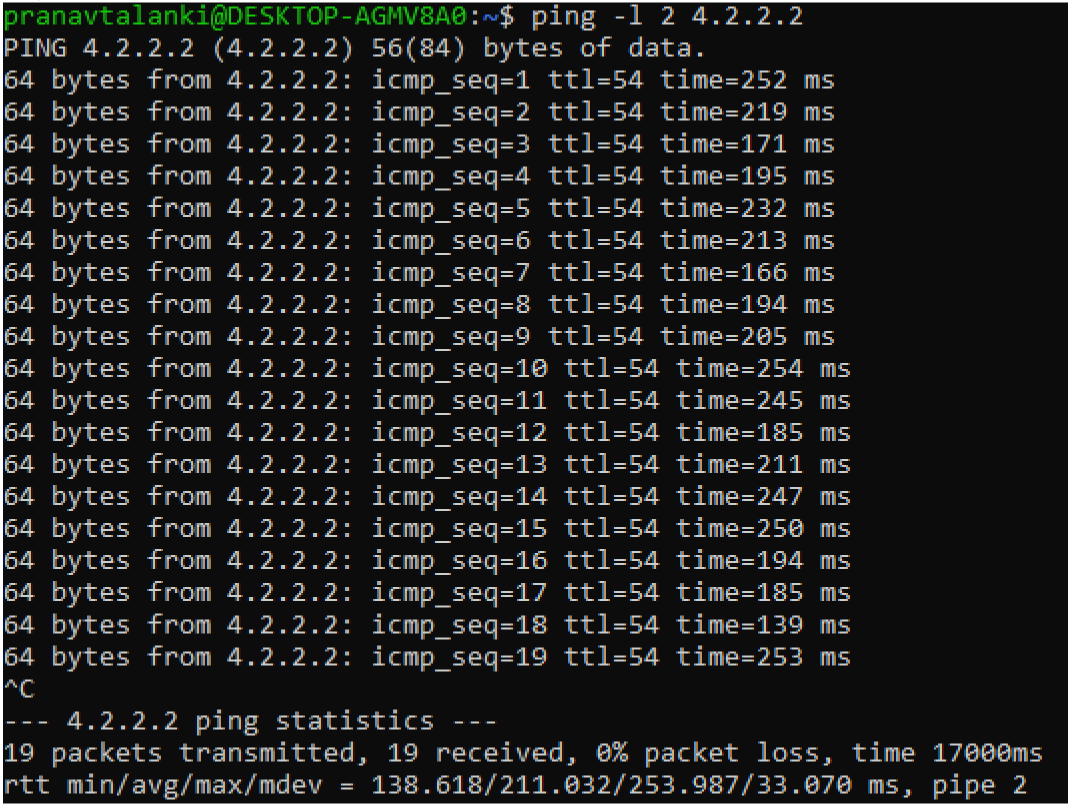
Use the -i option if you want a user-defined TTL for your ICMP echo request. The maximum is 255.



```
pranavtalanki@DESKTOP-AGMV8A0:~$ ping -i 2 4.2.2.2
PING 4.2.2.2 (4.2.2.2) 56(84) bytes of data.
64 bytes from 4.2.2.2: icmp_seq=1 ttl=54 time=202 ms
64 bytes from 4.2.2.2: icmp_seq=2 ttl=54 time=251 ms
64 bytes from 4.2.2.2: icmp_seq=3 ttl=54 time=188 ms
64 bytes from 4.2.2.2: icmp_seq=4 ttl=54 time=217 ms
64 bytes from 4.2.2.2: icmp_seq=5 ttl=54 time=255 ms
64 bytes from 4.2.2.2: icmp_seq=6 ttl=54 time=146 ms
64 bytes from 4.2.2.2: icmp_seq=7 ttl=54 time=245 ms
^C
--- 4.2.2.2 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 12041ms
rtt min/avg/max/mdev = 145.578/214.694/255.082/36.898 ms
pranavtalanki@DESKTOP-AGMV8A0:~$
```

c. -l size

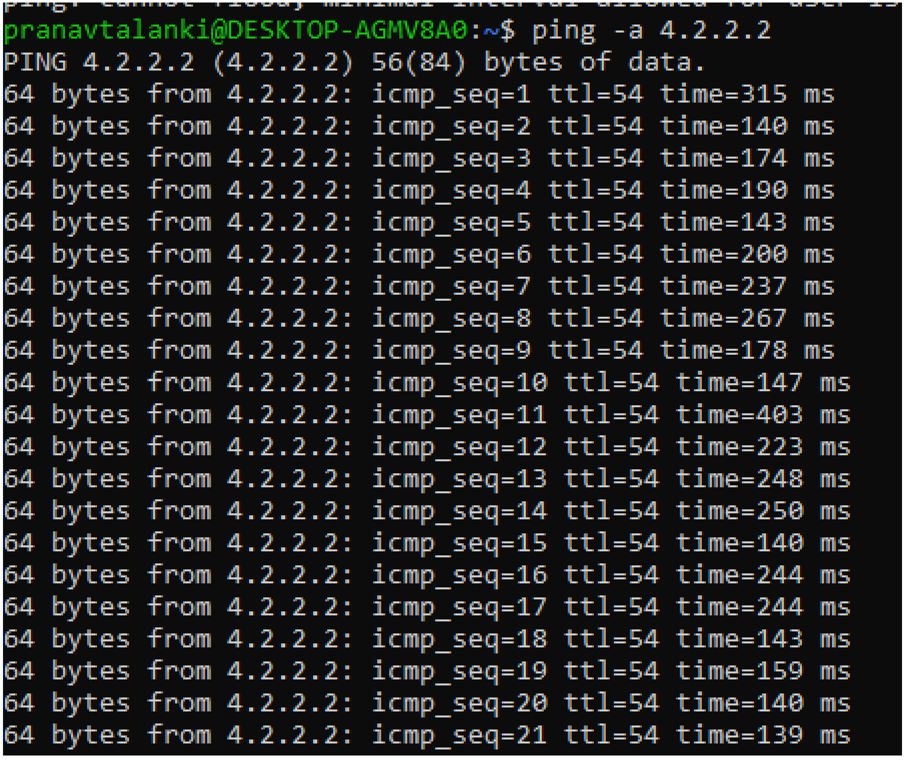
Use this option to set the size, in bytes, of the echo request packet from 32 to 65,527. The ping command will send a 32-byte echo request if you don't use the **-l** option.



```
pranavtalanki@DESKTOP-AGMV8A0:~$ ping -l 2 4.2.2.2
PING 4.2.2.2 (4.2.2.2) 56(84) bytes of data.
64 bytes from 4.2.2.2: icmp_seq=1 ttl=54 time=252 ms
64 bytes from 4.2.2.2: icmp_seq=2 ttl=54 time=219 ms
64 bytes from 4.2.2.2: icmp_seq=3 ttl=54 time=171 ms
64 bytes from 4.2.2.2: icmp_seq=4 ttl=54 time=195 ms
64 bytes from 4.2.2.2: icmp_seq=5 ttl=54 time=232 ms
64 bytes from 4.2.2.2: icmp_seq=6 ttl=54 time=213 ms
64 bytes from 4.2.2.2: icmp_seq=7 ttl=54 time=166 ms
64 bytes from 4.2.2.2: icmp_seq=8 ttl=54 time=194 ms
64 bytes from 4.2.2.2: icmp_seq=9 ttl=54 time=205 ms
64 bytes from 4.2.2.2: icmp_seq=10 ttl=54 time=254 ms
64 bytes from 4.2.2.2: icmp_seq=11 ttl=54 time=245 ms
64 bytes from 4.2.2.2: icmp_seq=12 ttl=54 time=185 ms
64 bytes from 4.2.2.2: icmp_seq=13 ttl=54 time=211 ms
64 bytes from 4.2.2.2: icmp_seq=14 ttl=54 time=247 ms
64 bytes from 4.2.2.2: icmp_seq=15 ttl=54 time=250 ms
64 bytes from 4.2.2.2: icmp_seq=16 ttl=54 time=194 ms
64 bytes from 4.2.2.2: icmp_seq=17 ttl=54 time=185 ms
64 bytes from 4.2.2.2: icmp_seq=18 ttl=54 time=139 ms
64 bytes from 4.2.2.2: icmp_seq=19 ttl=54 time=253 ms
^C
--- 4.2.2.2 ping statistics ---
19 packets transmitted, 19 received, 0% packet loss, time 17000ms
rtt min/avg/max/mdev = 138.618/211.032/253.987/33.070 ms, pipe 2
```

d. -a

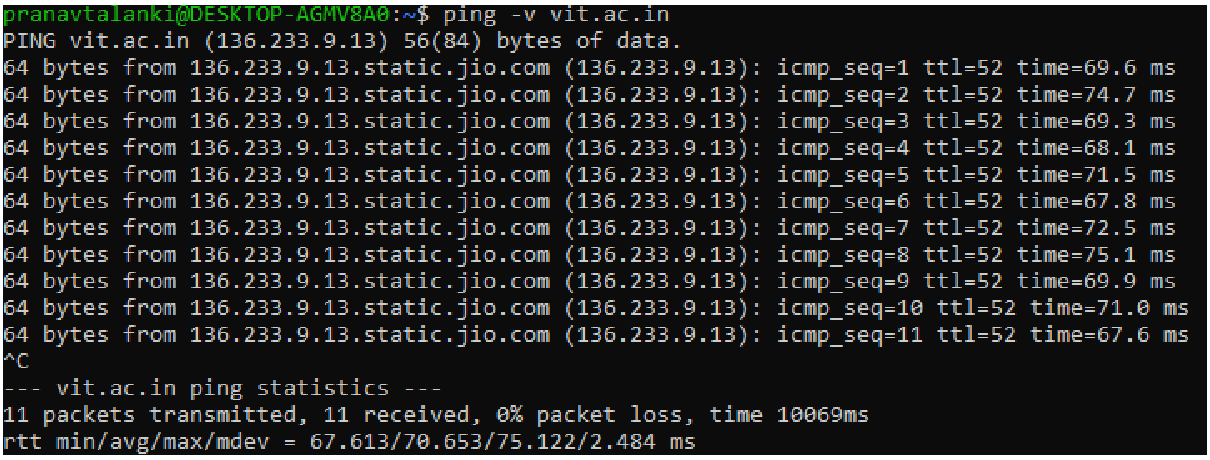
This ping command option will resolve, if possible, the hostname of an IP address *target*.



```
pranavtalanki@DESKTOP-AGMV8A0:~$ ping -a 4.2.2.2
PING 4.2.2.2 (4.2.2.2) 56(84) bytes of data.
64 bytes from 4.2.2.2: icmp_seq=1 ttl=54 time=315 ms
64 bytes from 4.2.2.2: icmp_seq=2 ttl=54 time=140 ms
64 bytes from 4.2.2.2: icmp_seq=3 ttl=54 time=174 ms
64 bytes from 4.2.2.2: icmp_seq=4 ttl=54 time=190 ms
64 bytes from 4.2.2.2: icmp_seq=5 ttl=54 time=143 ms
64 bytes from 4.2.2.2: icmp_seq=6 ttl=54 time=200 ms
64 bytes from 4.2.2.2: icmp_seq=7 ttl=54 time=237 ms
64 bytes from 4.2.2.2: icmp_seq=8 ttl=54 time=267 ms
64 bytes from 4.2.2.2: icmp_seq=9 ttl=54 time=178 ms
64 bytes from 4.2.2.2: icmp_seq=10 ttl=54 time=147 ms
64 bytes from 4.2.2.2: icmp_seq=11 ttl=54 time=403 ms
64 bytes from 4.2.2.2: icmp_seq=12 ttl=54 time=223 ms
64 bytes from 4.2.2.2: icmp_seq=13 ttl=54 time=248 ms
64 bytes from 4.2.2.2: icmp_seq=14 ttl=54 time=250 ms
64 bytes from 4.2.2.2: icmp_seq=15 ttl=54 time=140 ms
64 bytes from 4.2.2.2: icmp_seq=16 ttl=54 time=244 ms
64 bytes from 4.2.2.2: icmp_seq=17 ttl=54 time=244 ms
64 bytes from 4.2.2.2: icmp_seq=18 ttl=54 time=143 ms
64 bytes from 4.2.2.2: icmp_seq=19 ttl=54 time=159 ms
64 bytes from 4.2.2.2: icmp_seq=20 ttl=54 time=140 ms
64 bytes from 4.2.2.2: icmp_seq=21 ttl=54 time=139 ms
```

e.-v

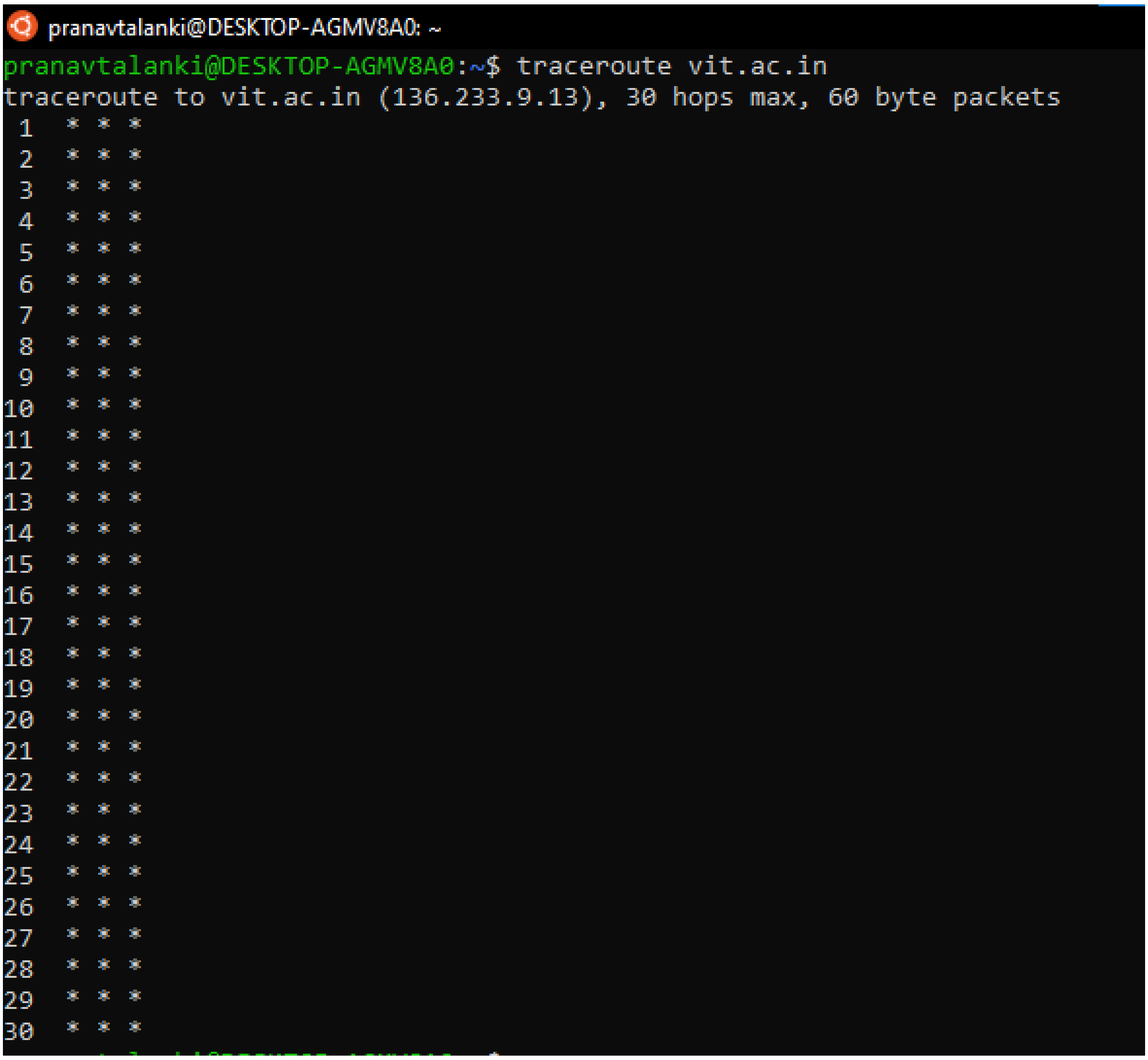
verbose output



```
pranavtalanki@DESKTOP-AGMV8A0:~$ ping -v vit.ac.in
PING vit.ac.in (136.233.9.13) 56(84) bytes of data.
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=1 ttl=52 time=69.6 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=2 ttl=52 time=74.7 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=3 ttl=52 time=69.3 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=4 ttl=52 time=68.1 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=5 ttl=52 time=71.5 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=6 ttl=52 time=67.8 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=7 ttl=52 time=72.5 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=8 ttl=52 time=75.1 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=9 ttl=52 time=69.9 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=10 ttl=52 time=71.0 ms
64 bytes from 136.233.9.13.static.jio.com (136.233.9.13): icmp_seq=11 ttl=52 time=67.6 ms
^C
--- vit.ac.in ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10069ms
rtt min/avg/max/mdev = 67.613/70.653/75.122/2.484 ms
```

3.traceroute

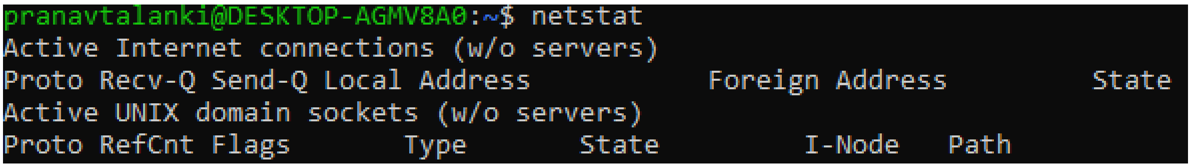
traceroute is a network troubleshooting utility which shows number of hops taken to reach destination also determine packets traveling path.



```
pranavtalanki@DESKTOP-AGMV8A0:~$ traceroute vit.ac.in
traceroute to vit.ac.in (136.233.9.13), 30 hops max, 60 byte packets
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

4.Netstat

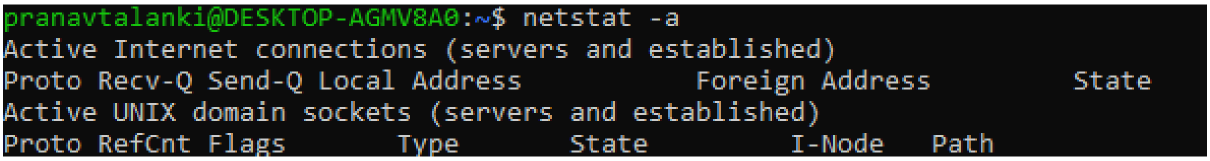
It delivers basic statistics on all network activities and informs users on which portsand addresses the corresponding connections (TCP, UDP) are running and which ports are open for tasks.



```
pranavtalanki@DESKTOP-AGMV8A0:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
```

- -a

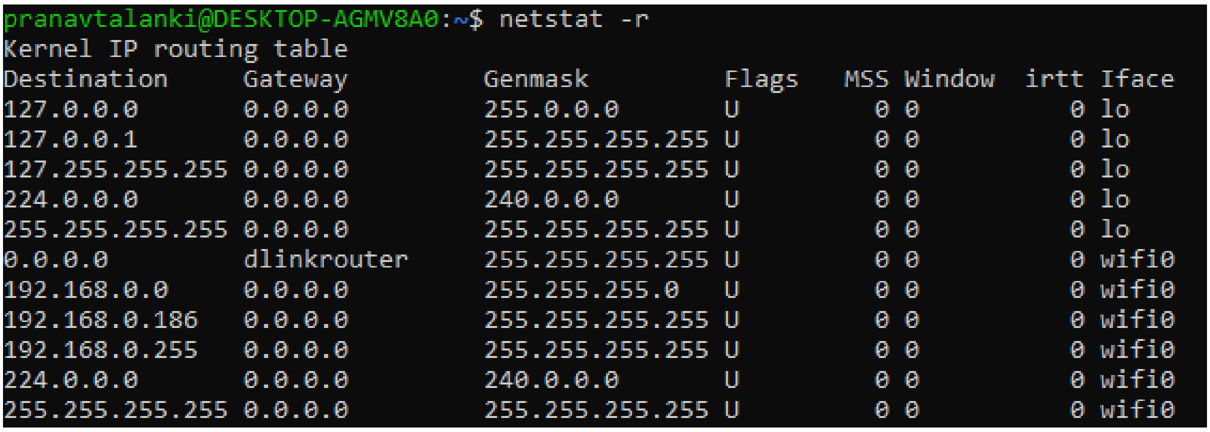
  Display all the active ports



```
pranavtalanki@DESKTOP-AGMV8A0:~$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
```

- -r

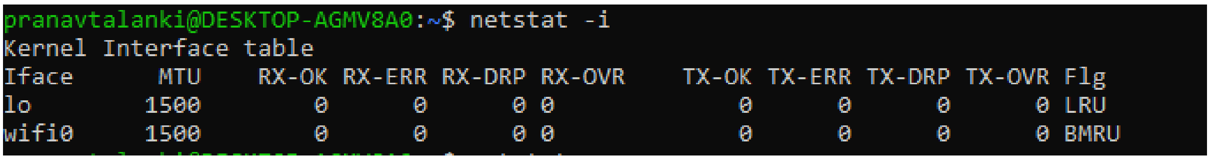
  Display routing table



```
pranavtalanki@DESKTOP-AGMV8A0:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
127.0.0.0       0.0.0.0         255.0.0.0       U         0 0          0 lo
127.0.0.1       0.0.0.0         255.255.255.255 U         0 0          0 lo
127.255.255.255 0.0.0.0         255.255.255.255 U         0 0          0 lo
224.0.0.0       0.0.0.0         240.0.0.0       U         0 0          0 lo
255.255.255.255 0.0.0.0         255.255.255.255 U         0 0          0 lo
0.0.0.0         dlinkrouter     255.255.255.255 U         0 0          0 wifi0
192.168.0.0     0.0.0.0         255.255.255.0   U         0 0          0 wifi0
192.168.0.186   0.0.0.0         255.255.255.255 U         0 0          0 wifi0
192.168.0.255   0.0.0.0         255.255.255.255 U         0 0          0 wifi0
224.0.0.0       0.0.0.0         240.0.0.0       U         0 0          0 wifi0
255.255.255.255 0.0.0.0         255.255.255.255 U         0 0          0 wifi0
```

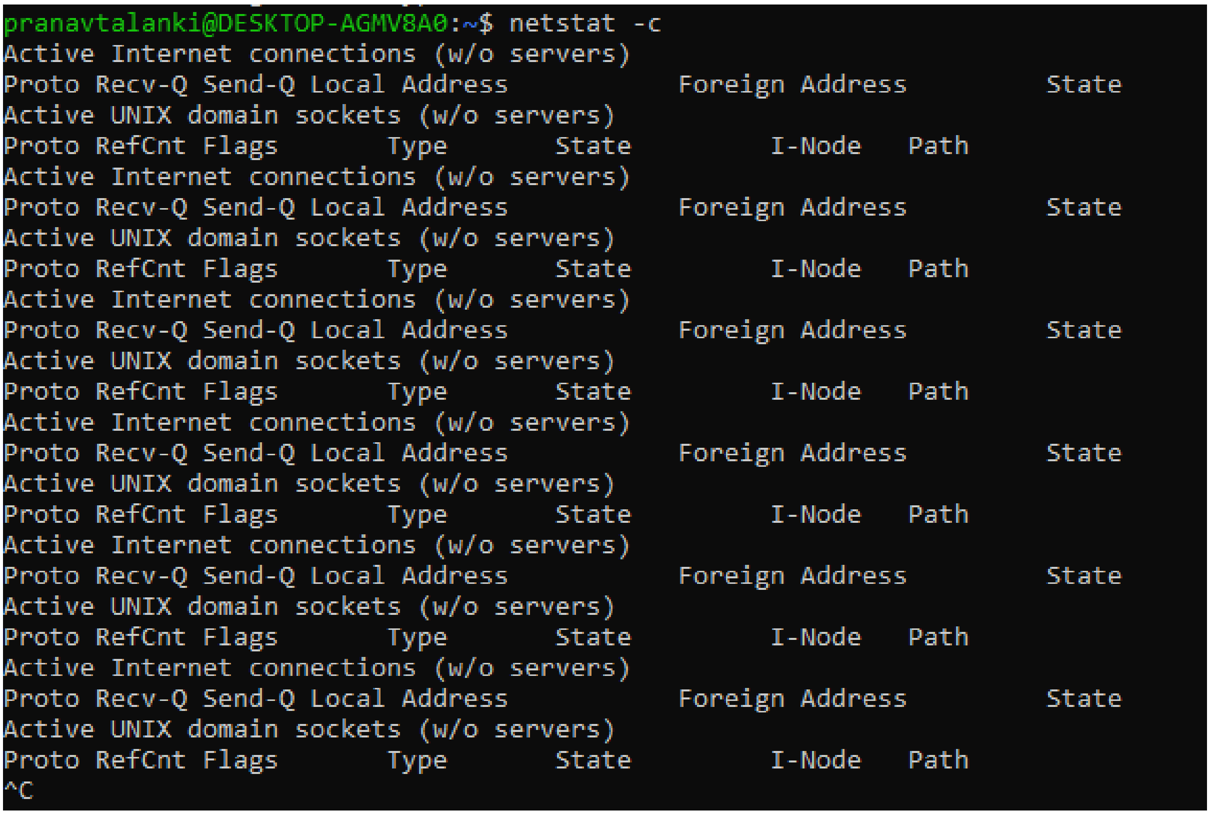
- -i

  Displays interface table



```
pranavtalanki@DESKTOP-AGMV8A0:~$ netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
lo        1500        0      0      0 0             0      0      0      0 LRU
wifi0     1500        0      0      0 0             0      0      0      0 BMRU
```

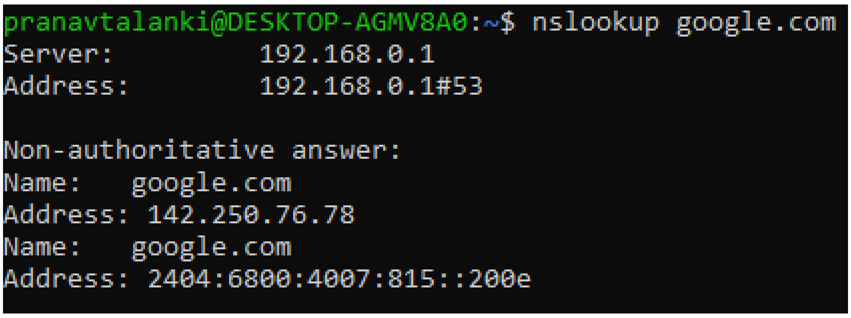
- -c
  Continuous listing



```
pranavtalanki@DESKTOP-AGMV8A0:~$ netstat -c
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
^C
```

5.nslookup

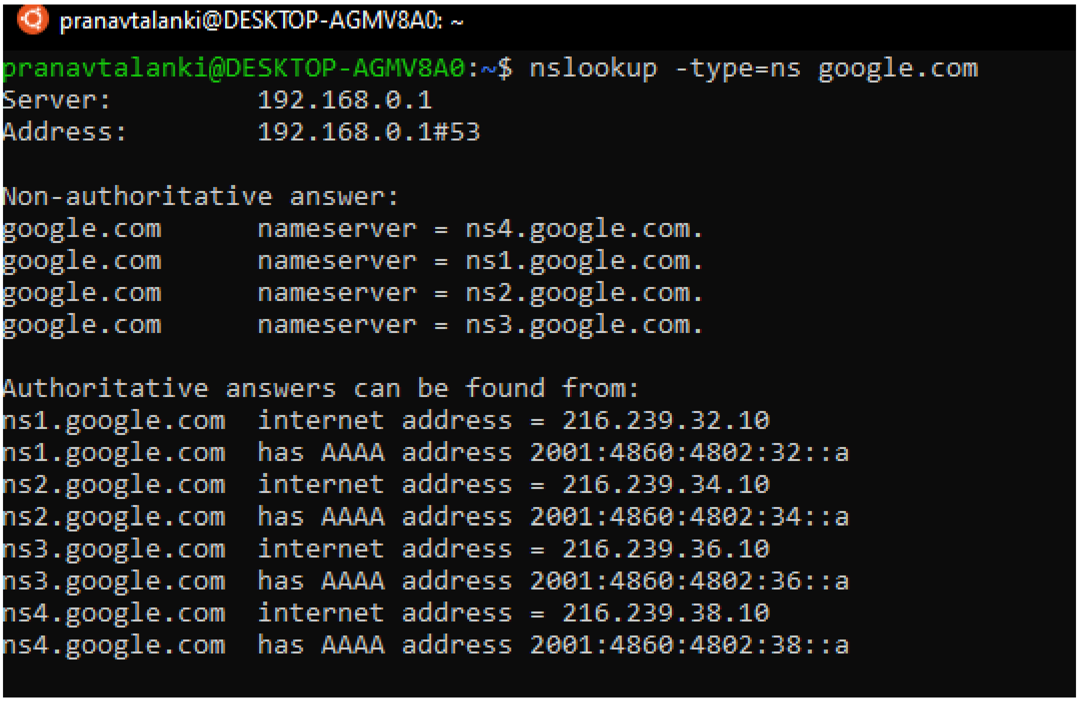
nslookup followed by the domain name will display the "A Record" (IP Address) of the domain. Use this command to find the address record for a domain. It queries to domain name servers and get the details.



```
pranavtalanki@DESKTOP-AGMV8A0:~$ nslookup google.com
Server:         192.168.0.1
Address:        192.168.0.1#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.76.78
Name:   google.com
Address: 2404:6800:4007:815::200e
```

- -type=ns

  We can check the NS records

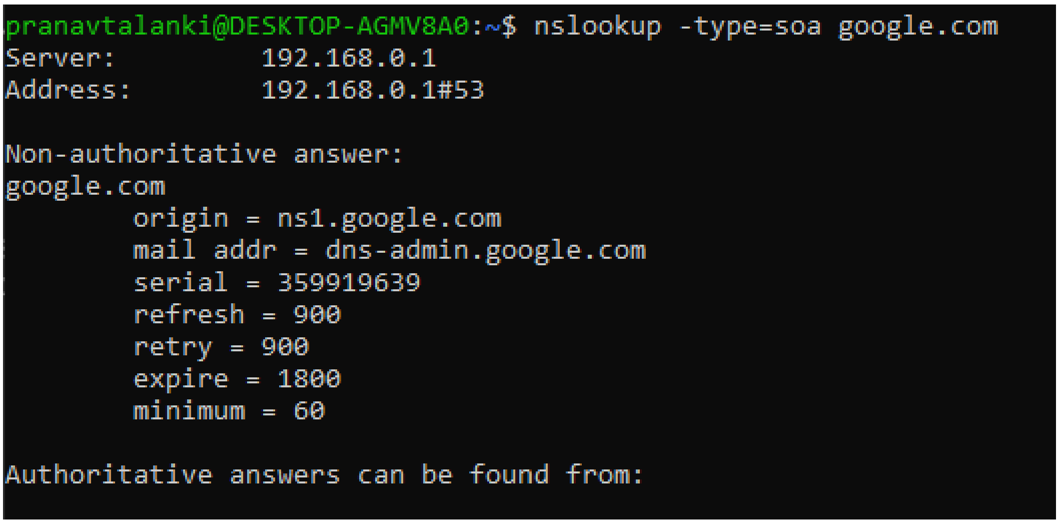


pranavtalanki@DESKTOP-AGMV8A0: ~

```
pranavtalanki@DESKTOP-AGMV8A0:~$ nslookup -type=ns google.com
Server:         192.168.0.1
Address:        192.168.0.1#53

Non-authoritative answer:
google.com      nameserver = ns4.google.com.
google.com      nameserver = ns1.google.com.
google.com      nameserver = ns2.google.com.
google.com      nameserver = ns3.google.com.

Authoritative answers can be found from:
ns1.google.com  internet address = 216.239.32.10
ns1.google.com  has AAAA address 2001:4860:4802:32::a
ns2.google.com  internet address = 216.239.34.10
ns2.google.com  has AAAA address 2001:4860:4802:34::a
ns3.google.com  internet address = 216.239.36.10
ns3.google.com  has AAAA address 2001:4860:4802:36::a
ns4.google.com  internet address = 216.239.38.10
ns4.google.com  has AAAA address 2001:4860:4802:38::a
```

- -type = soa

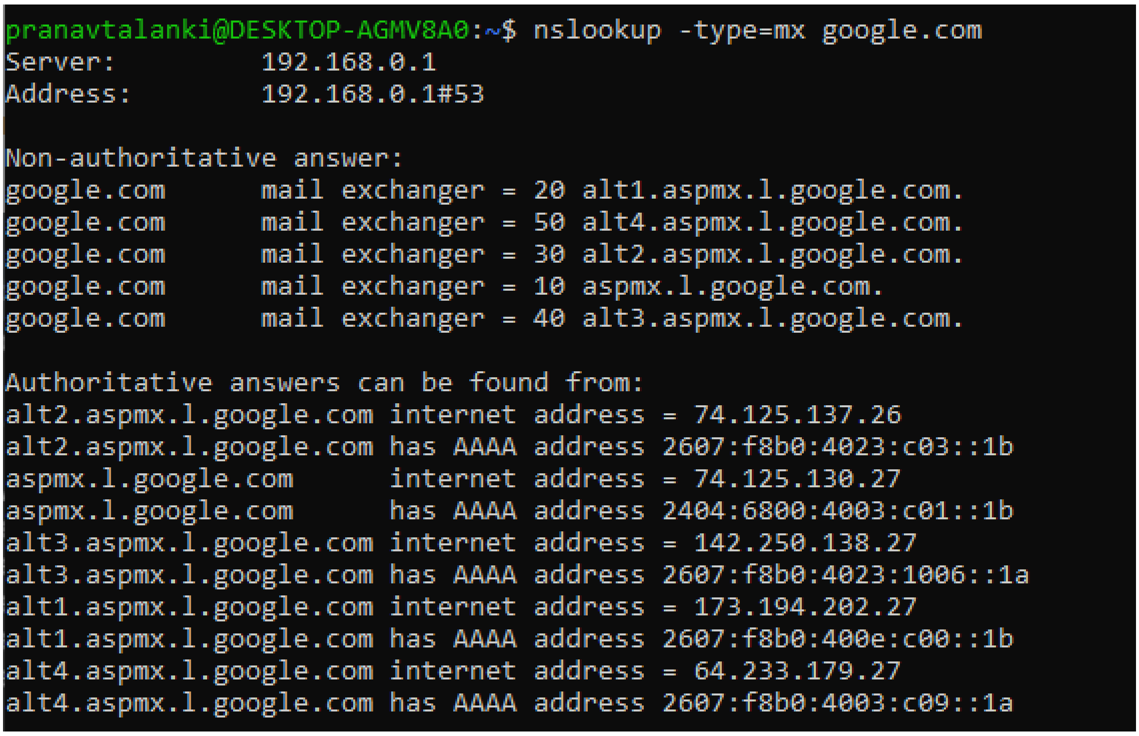
  We can check the start of authority and get information about the zone



```
pranavtalanki@DESKTOP-AGMV8A0:~$ nslookup -type=soa google.com
Server:         192.168.0.1
Address:        192.168.0.1#53

Non-authoritative answer:
google.com
        origin = ns1.google.com
        mail addr = dns-admin.google.com
        serial = 359919639
        refresh = 900
        retry = 900
        expire = 1800
        minimum = 60

Authoritative answers can be found from:
```

- -type = mx

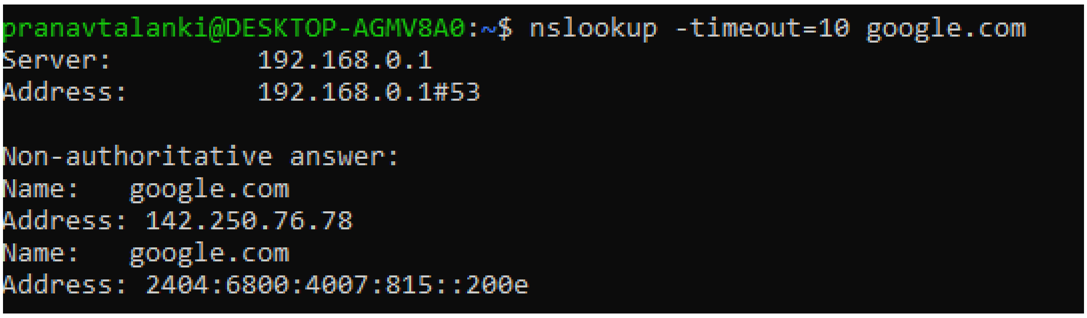
  we can check the MX records of the mail servers. You can see if all the mail servers are working well.



```
pranavtalanki@DESKTOP-AGMV8A0:~$ nslookup -type=mx google.com
Server:         192.168.0.1
Address:        192.168.0.1#53

Non-authoritative answer:
google.com      mail exchanger = 20 alt1.aspmx.l.google.com.
google.com      mail exchanger = 50 alt4.aspmx.l.google.com.
google.com      mail exchanger = 30 alt2.aspmx.l.google.com.
google.com      mail exchanger = 10 aspmx.l.google.com.
google.com      mail exchanger = 40 alt3.aspmx.l.google.com.

Authoritative answers can be found from:
alt2.aspmx.l.google.com internet address = 74.125.137.26
alt2.aspmx.l.google.com has AAAA address 2607:f8b0:4023:c03::1b
aspmx.l.google.com      internet address = 74.125.130.27
aspmx.l.google.com      has AAAA address 2404:6800:4003:c01::1b
alt3.aspmx.l.google.com internet address = 142.250.138.27
alt3.aspmx.l.google.com has AAAA address 2607:f8b0:4023:1006::1a
alt1.aspmx.l.google.com internet address = 173.194.202.27
alt1.aspmx.l.google.com has AAAA address 2607:f8b0:400e:c00::1b
alt4.aspmx.l.google.com internet address = 64.233.179.27
alt4.aspmx.l.google.com has AAAA address 2607:f8b0:4003:c09::1a
```

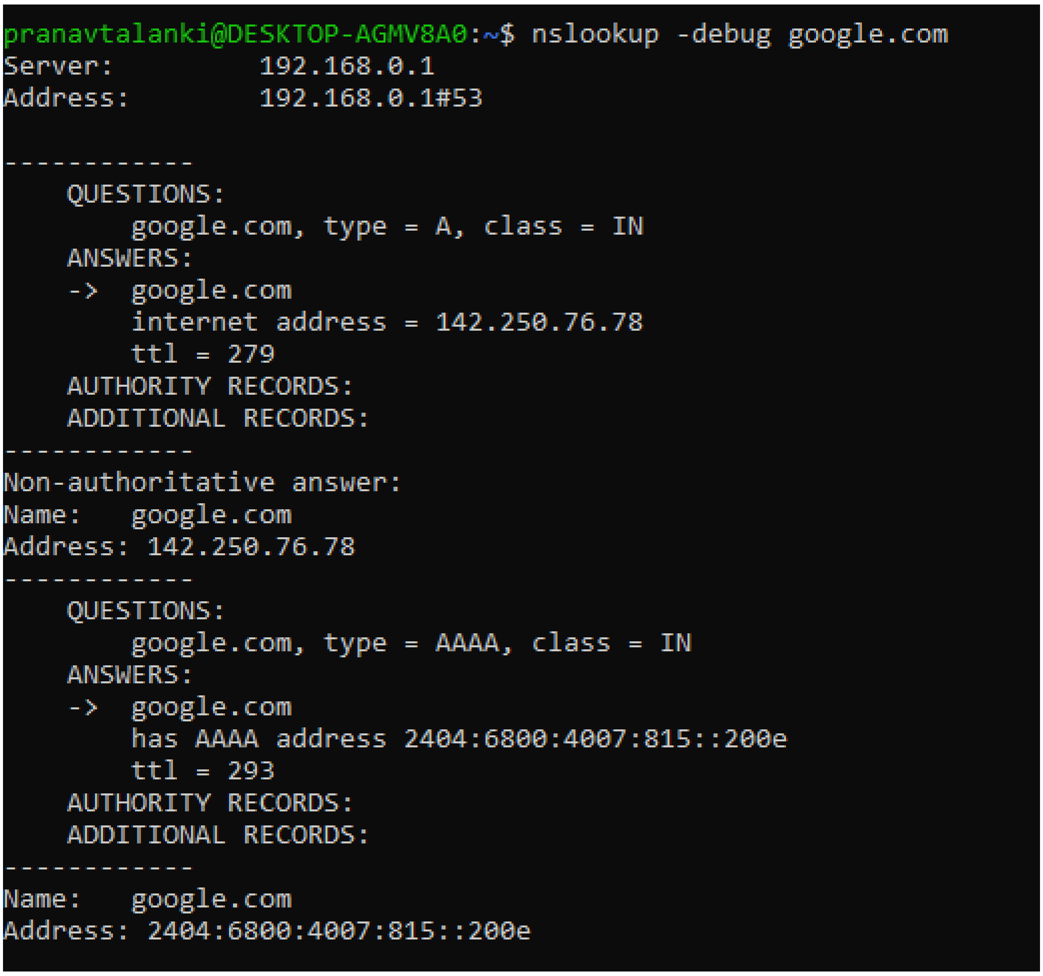
- -timeout=number

  we can manually choose the timeout time in seconds. We can increase it to give more time for the server to respond. we can also shorter it to see which servers can respond quicker



```
pranavtalanki@DESKTOP-AGMV8A0:~$ nslookup -timeout=10 google.com
Server:         192.168.0.1
Address:        192.168.0.1#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.76.78
Name:   google.com
Address: 2404:6800:4007:815::200e
```

- -debug
  Debug mode provides important and detailed information both for the question and for the received answer



```
pranavtalanki@DESKTOP-AGMV8A0:~$ nslookup -debug google.com
Server:         192.168.0.1
Address:        192.168.0.1#53

------------
    QUESTIONS:
        google.com, type = A, class = IN
    ANSWERS:
    ->  google.com
        internet address = 142.250.76.78
        ttl = 279
    AUTHORITY RECORDS:
    ADDITIONAL RECORDS:
------------
Non-authoritative answer:
Name:   google.com
Address: 142.250.76.78
------------
    QUESTIONS:
        google.com, type = AAAA, class = IN
    ANSWERS:
    ->  google.com
        has AAAA address 2404:6800:4007:815::200e
        ttl = 293
    AUTHORITY RECORDS:
    ADDITIONAL RECORDS:
------------
Name:   google.com
Address: 2404:6800:4007:815::200e
```

Activity-2

Perform following exercise using ‘Cisco Packet Tracer’. Create the followingnetwork asshown in the figure below. Ensure that the devicesinLAN are configured with an IP address and can ping each other. Also ensure that switches can telnet to each other. Simulate the network to find the data communication between any two devices in the network is successful or not.

a.

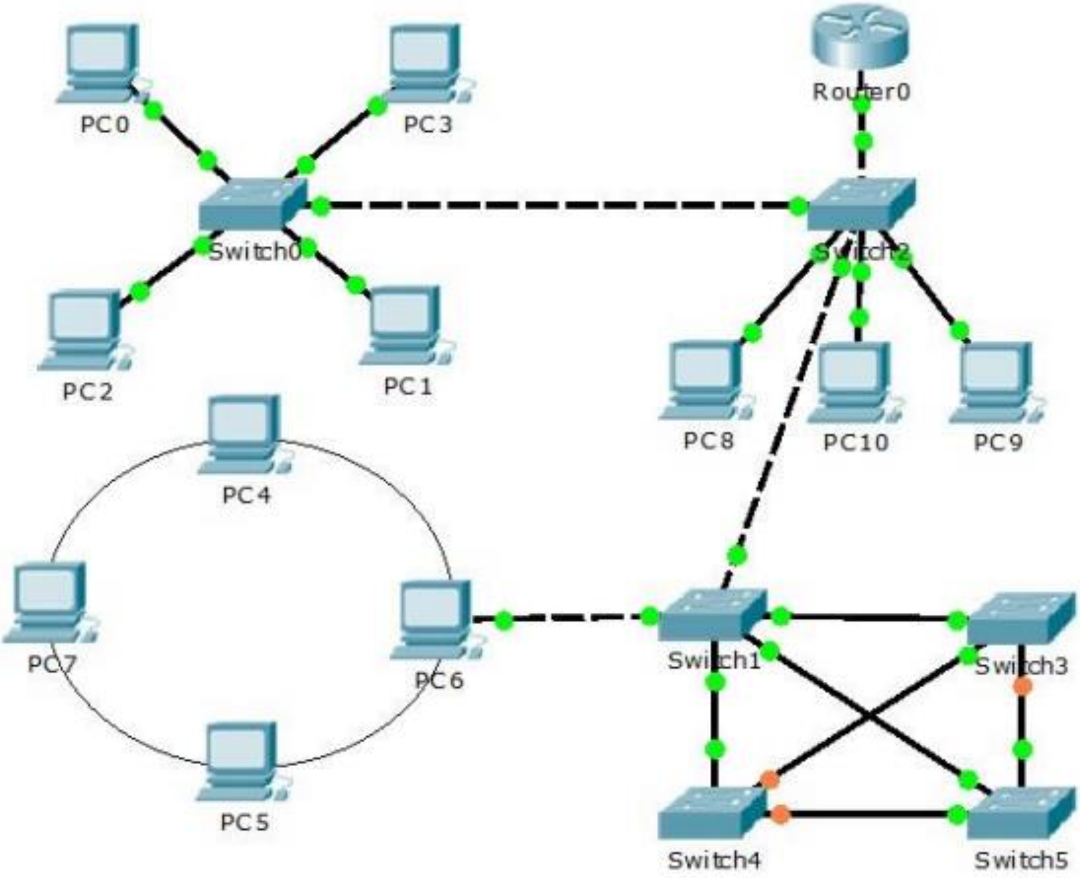
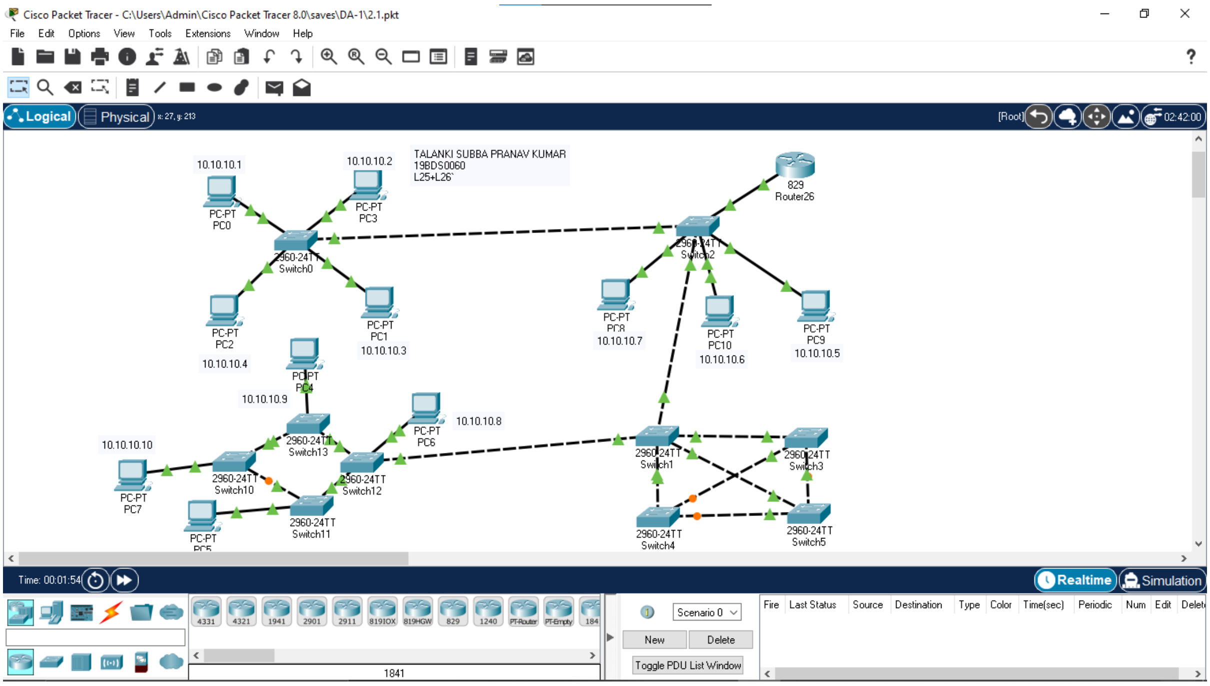


Figure 1: Network based on basic topology

Ans:-

Network connection:-



This is the network connection

For switches

->Network devices

->switches

-> 2960-24 model

For Pc's

->end devices

->Pc

For routers

->Network Devices

->routers

->829 model

For Connections

->connections

For connections between switches and PC's use copper straight-through

For connections between switche and switch use copper cross-over

For connections between switche and router use copper straight-through

<u>I.P.ADDRESSES</u>

PC0 – 10.10.10.1

PC1 – 10.10.10.3

PC2 – 10.10.10.4

PC3 – 10.10.10.2

PC4 – 10.10.10.9

PC5 – 10.10.10.11

PC6 – 10.10.10.8

PC7 – 10.10.10.10

PC8 – 10.10.10.7

PC9 – 10.10.10.5

PC10 – 10.10.10.6

<u>Simulation results:-</u>

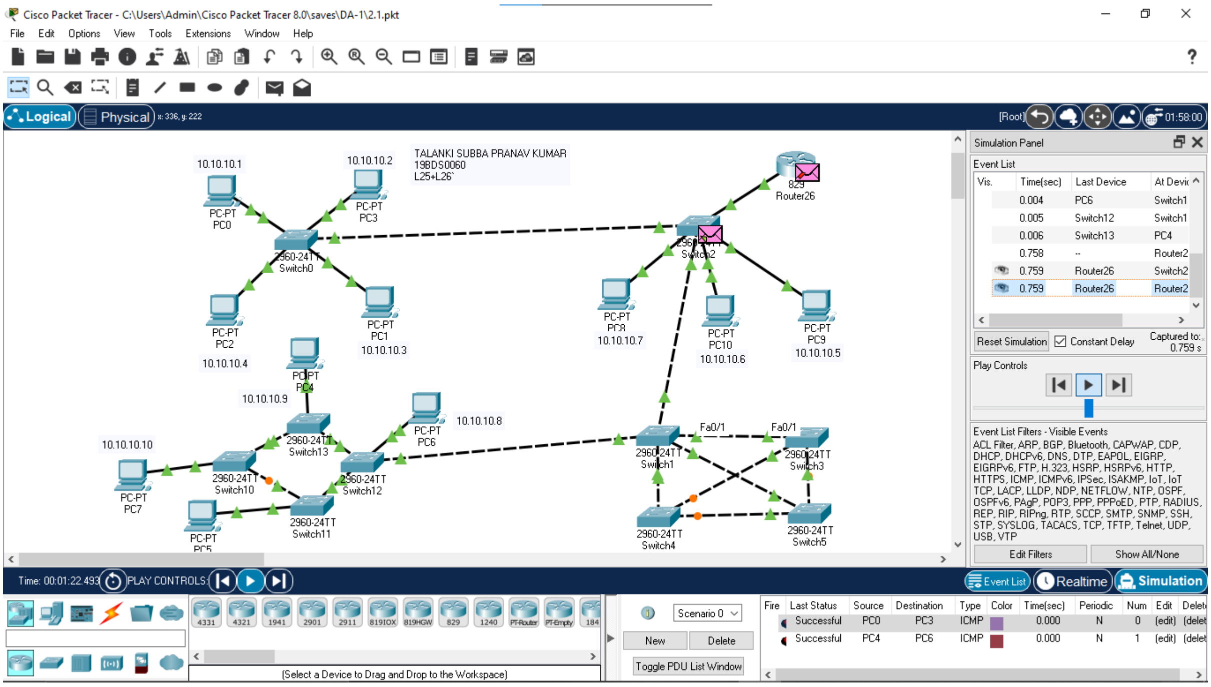
For this I did 3 simulations

1.Source->PC0

Destination->PC3

2.Source->PC4

Destination->PC6



Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Devic
	0.004	PC6	Switch1
	0.005	Switch12	Switch1
	0.006	Switch13	PC4
	0.758	--	Router2
👁	0.759	Router26	Switch2
👁	0.759	Router26	Router2

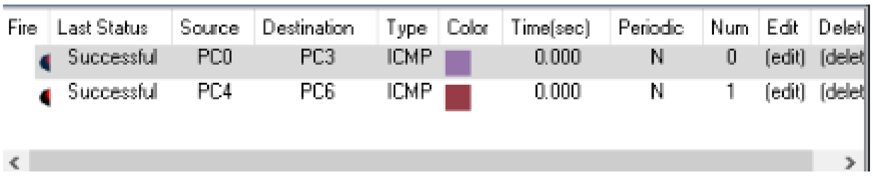
Reset Simulation ☑ Constant Delay Captured to: 0.759 s

Play Controls

Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP

Edit Filters Show All/None

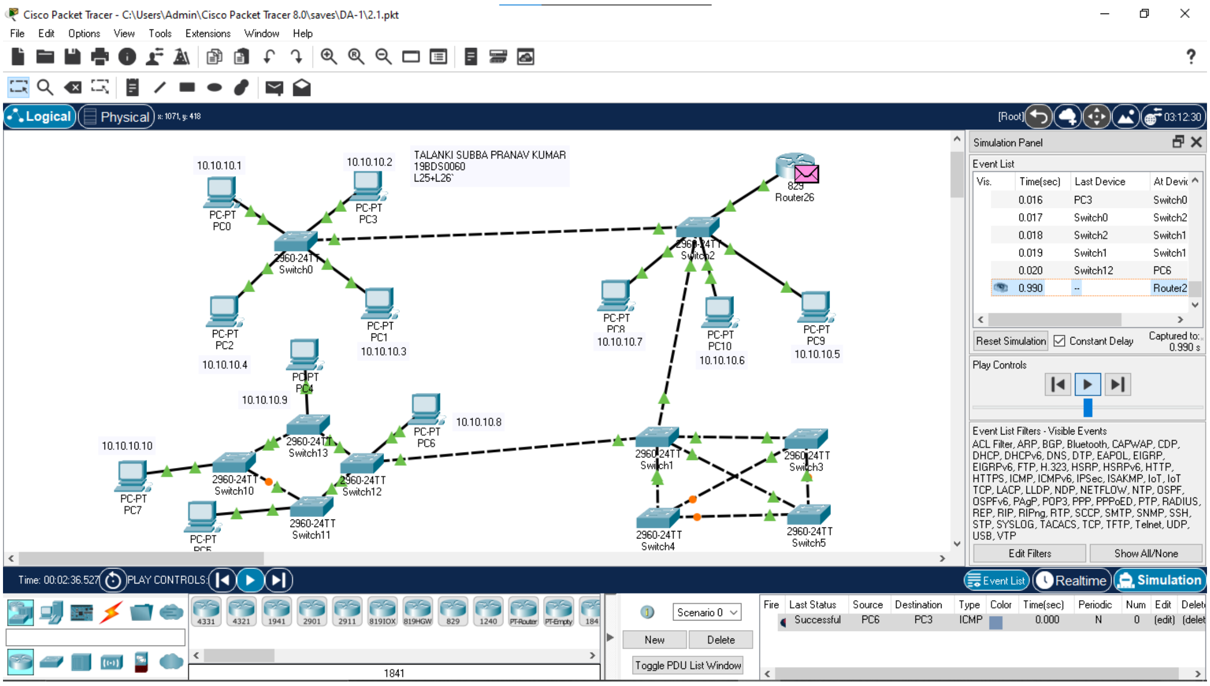
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delet
●	Successful	PC0	PC3	ICMP	🟪	0.000	N	0	(edit)	(delet
●	Successful	PC4	PC6	ICMP	🟥	0.000	N	1	(edit)	(delet



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delet
●	Successful	PC0	PC3	ICMP	🟪	0.000	N	0	(edit)	(delet
●	Successful	PC4	PC6	ICMP	🟥	0.000	N	1	(edit)	(delet

3.Source->PC6

Destination->PC3



Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Devic
	0.016	PC3	Switch0
	0.017	Switch0	Switch2
	0.018	Switch2	Switch1
	0.019	Switch1	Switch1
	0.020	Switch12	PC6
👁	0.990	--	Router2

Reset Simulation ☑ Constant Delay Captured to: 0.990 s

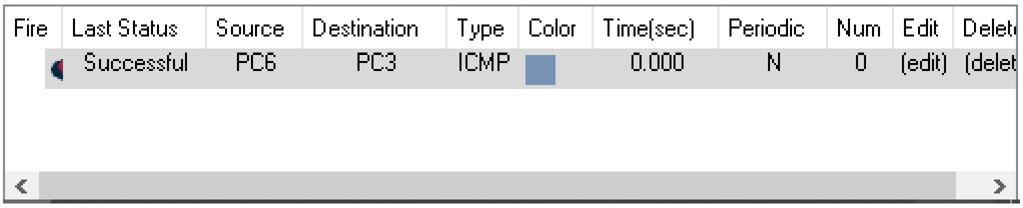
Play Controls

Event List Filters - Visible Events

ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP

Edit Filters Show All/None

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delet
●	Successful	PC6	PC3	ICMP	■	0.000	N	0	(edit)	(delet



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delet
●	Successful	PC6	PC3	ICMP	■	0.000	N	0	(edit)	(delet

b.

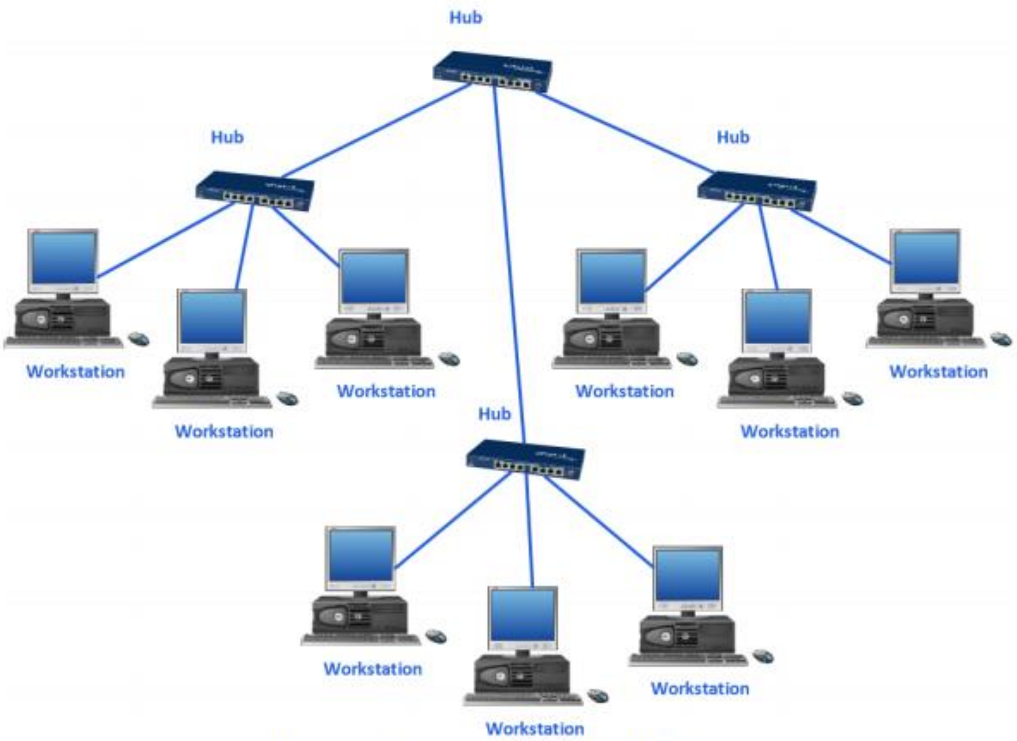
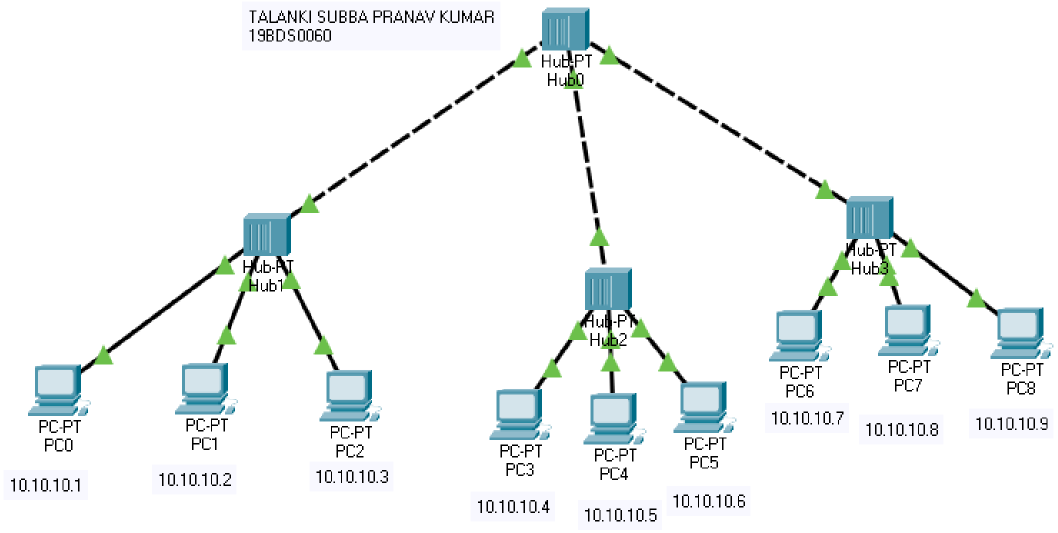


Figure 2: Network based on Hub

ANS:-

Network connection:-



For Pc's

->end devices

->Pc

For Hubs

->network devices

->hubs

->PT model

For Connections

->connections

For connections between Hub's and PC's use copper straight-through

For connections between Hub and Hub use copper cross-over

I.P.ADDRESSES:-

PC0 – 10.10.10.1

PC1 – 10.10.10.2

PC2 – 10.10.10.3

PC3 – 10.10.10.4

PC4 – 10.10.10.5

PC5 – 10.10.10.6

PC6 – 10.10.10.7

PC7 – 10.10.10.8

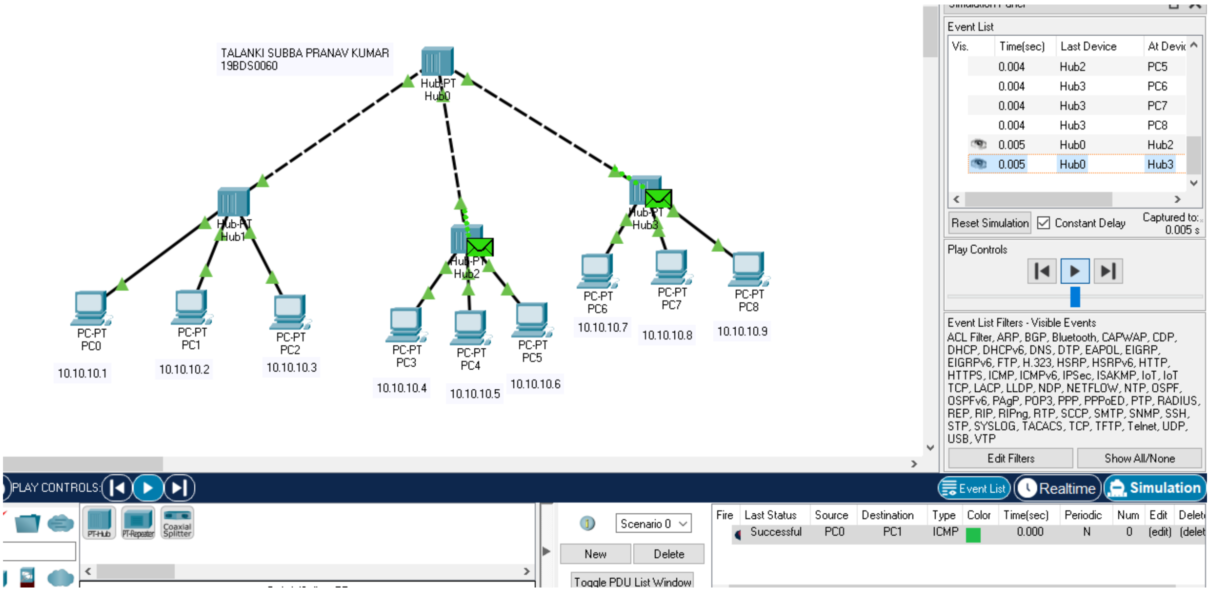
PC8 – 10.10.10.9

Simulation Results:-

1.

Source->PC0

Destination->PC1



Event List

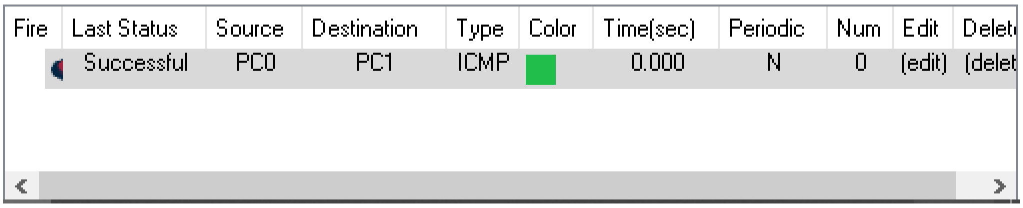
Vis.	Time(sec)	Last Device	At Devic
	0.004	Hub2	PC5
	0.004	Hub3	PC6
	0.004	Hub3	PC7
	0.004	Hub3	PC8
	0.005	Hub0	Hub2
	0.005	Hub0	Hub3

☑ Constant Delay

Captured to: 0.005 s

Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delet
●	Successful	PC0	PC1	ICMP	🟩	0.000	N	0	(edit)	(delet

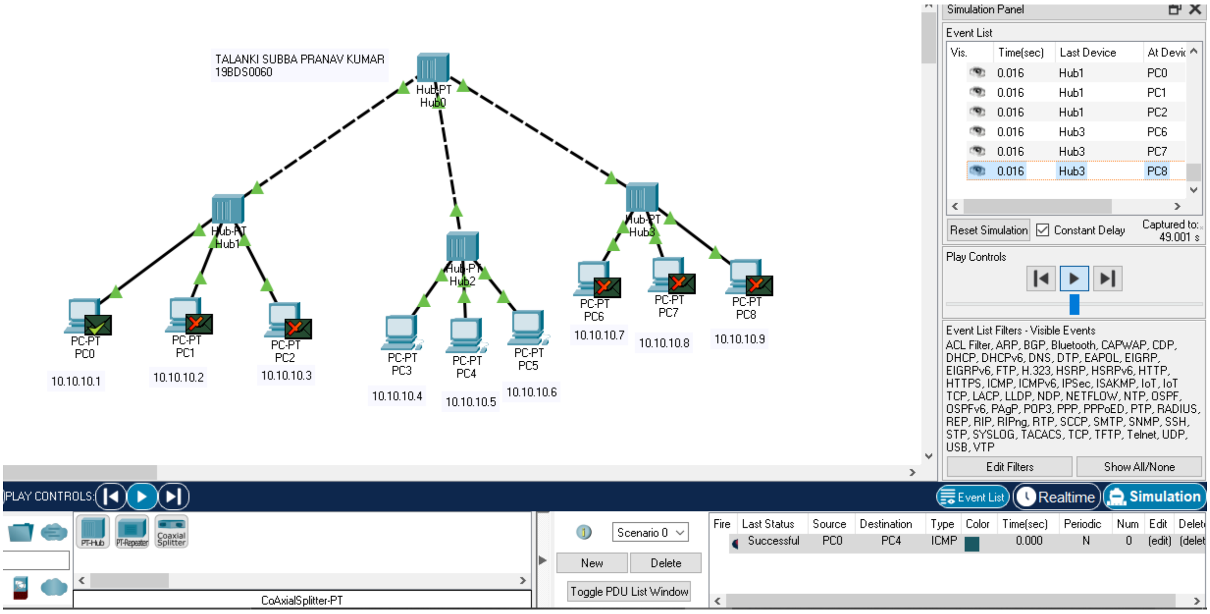


Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delet
●	Successful	PC0	PC1	ICMP	🟩	0.000	N	0	(edit)	(delet

2.

Source->PC0

Destination->PC4



Event List

Vis.	Time(sec)	Last Device	At Devic
	0.016	Hub1	PC0
	0.016	Hub1	PC1
	0.016	Hub1	PC2
	0.016	Hub3	PC6
	0.016	Hub3	PC7
	0.016	Hub3	PC8

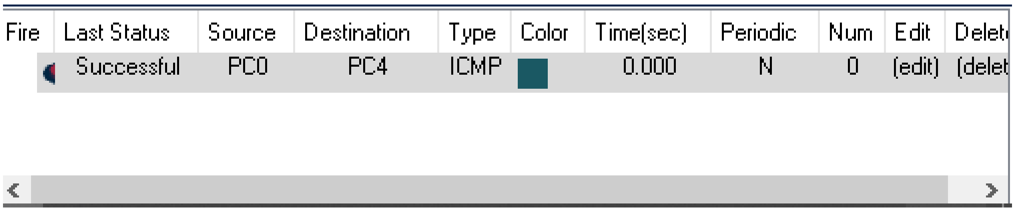
Reset Simulation ☑ Constant Delay Captured to: 49.001 s

Play Controls

Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP

Edit Filters | Show All/None

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delet
●	Successful	PC0	PC4	ICMP	■	0.000	N	0	(edit)	(delet



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delet
●	Successful	PC0	PC4	ICMP	■	0.000	N	0	(edit)	(delet

C.

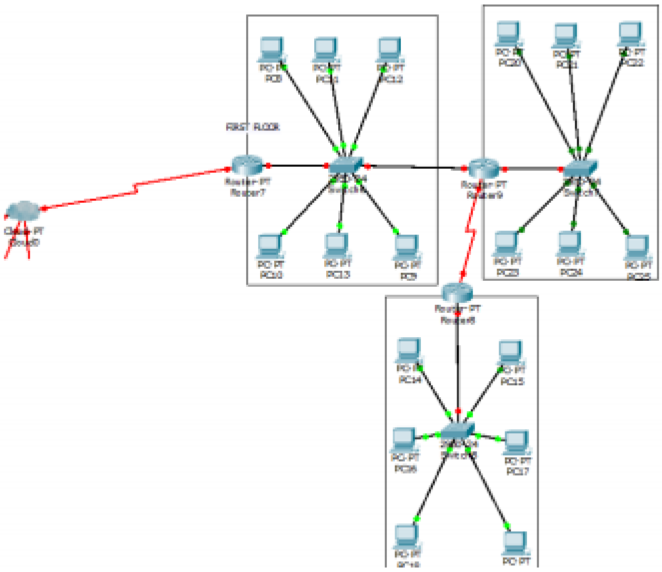
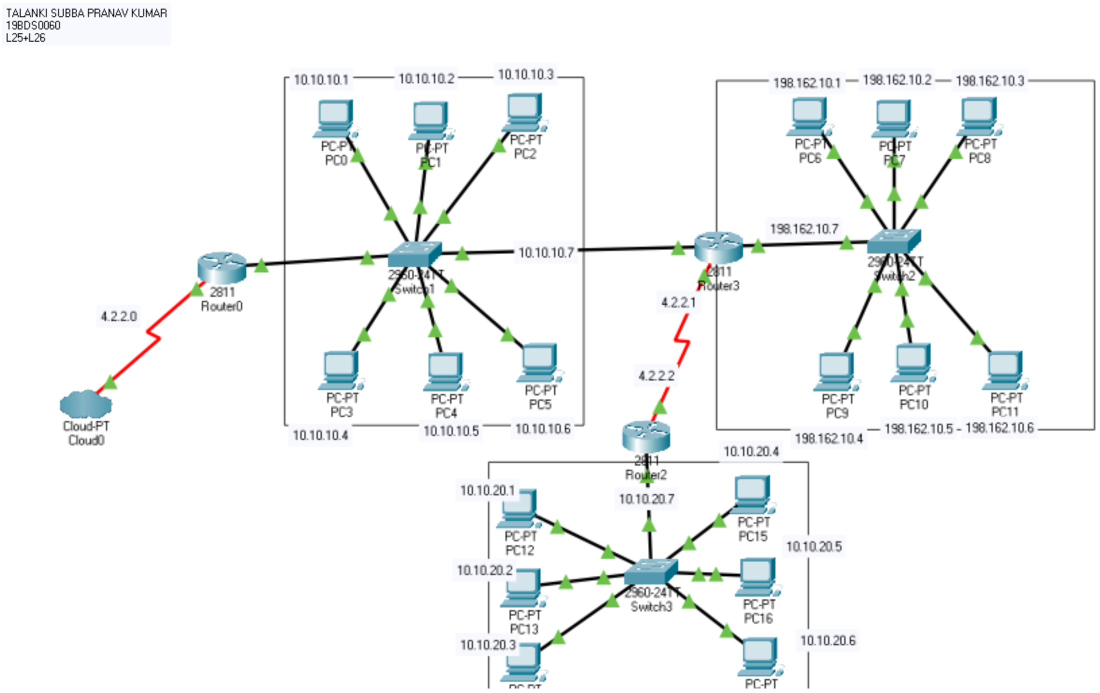


Figure 3: Network based on Switch

Ans:-

Network Connection:-



For switches

->Network devices

->switches

-> 2960-24 model

For Pc's

->end devices

->Pc

For routers

->Network Devices

->routers

->2811 model

For Connections

->connections

For cloud

->Network Devices

->WAN emulation

->PT cloud

For connections between switches and PC’s use copper straight-through

For connections between switch and switch use copper cross-over

For connections between switch and router use copper straight-through

For connections between router and router use Serial DTE

For connection between cloud and router use Serial DTE

I.P.ADDRESS:-

PC0-10.10.10.1

PC1-10.10.10.2

PC2-10.10.10.3

PC3-10.10.10.4

PC4-10.10.10.5

PC5-10.10.10.6

Default gateway for PC0-PC5 :- 10.10.10.7

PC6-198.162.10.1

PC7-198.162.10.2

PC8-198.162.10.3

PC9-198.162.10.4

PC10-198.162.10.5

PC11-198.162.10.6

Default gateway for PC6-PC11 :- 198.162.10.7

PC12-10.10.20.1

PC13-10.10.20.2

PC14-10.10.20.3

PC15-10.10.20.4

PC16-10.10.20.5

PC17-10.10.20.6

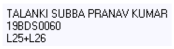
Default gateway for PC12-PC17:- 10.10.20.7

Simulation Results :-

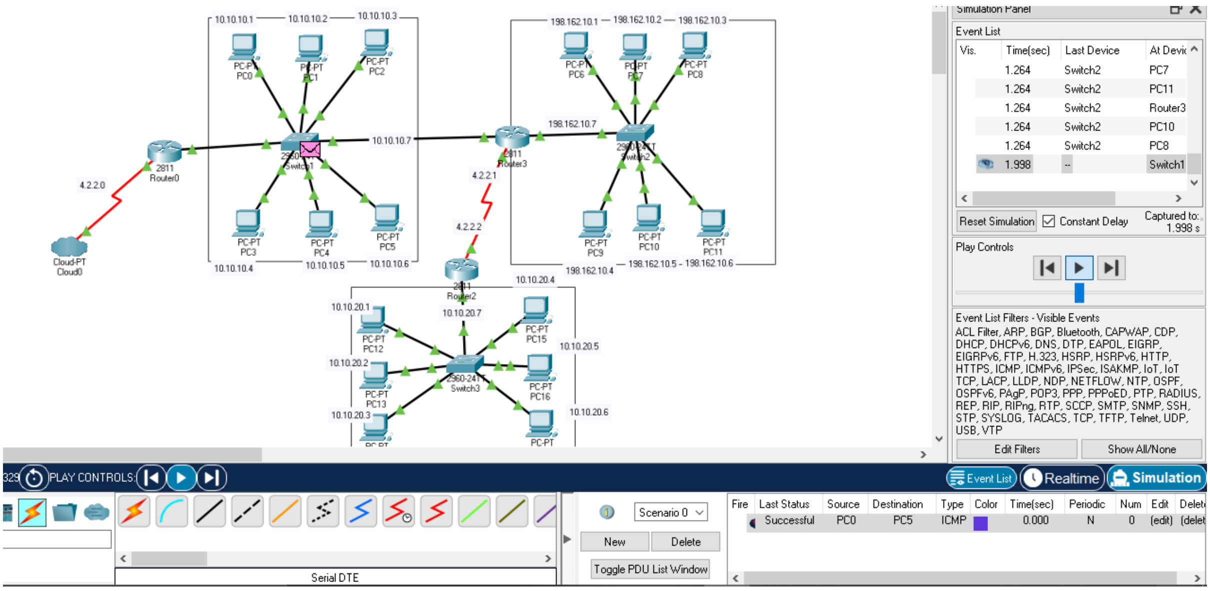
1.

Source :- PC0

destination :-PC5



TALANKI SUBBA PRANAV KUMAR
19BDS0060
L25+L26



Simulation Panel

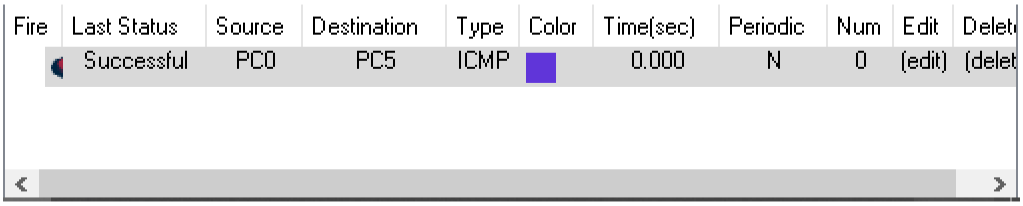
Event List

Vis.	Time(sec)	Last Device	At Devic
	1.264	Switch2	PC7
	1.264	Switch2	PC11
	1.264	Switch2	Router3
	1.264	Switch2	PC10
	1.264	Switch2	PC8
👁	1.998	--	Switch1

Reset Simulation ☑ Constant Delay Captured to: 1.998 s

Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	PC5	ICMP	🟪	0.000	N	0	(edit)	(delet

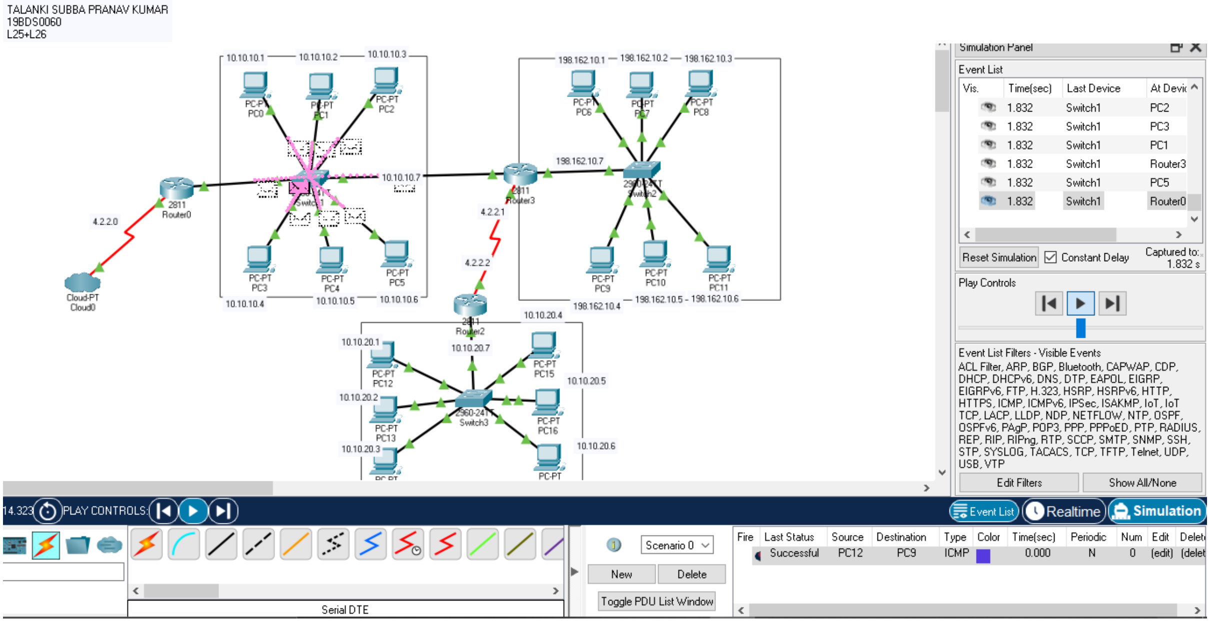


Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delet
●	Successful	PC0	PC5	ICMP	🟪	0.000	N	0	(edit)	(delet

2.

Source:-PC12

Destination:-PC9



TALANKI SUBBA PRANAV KUMAR
19BDS0060
L25+L26

Simulation Panel

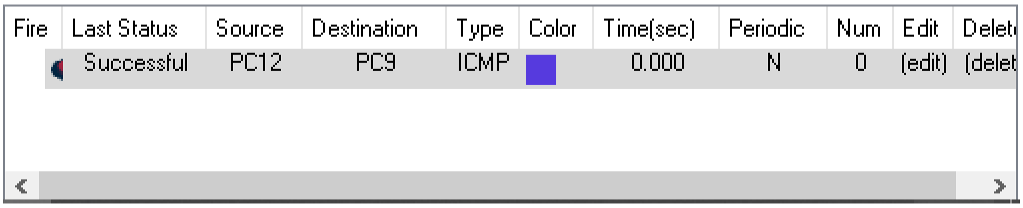
Event List

Vis.	Time(sec)	Last Device	At Devic
👁	1.832	Switch1	PC2
👁	1.832	Switch1	PC3
👁	1.832	Switch1	PC1
👁	1.832	Switch1	Router3
👁	1.832	Switch1	PC5
👁	1.832	Switch1	Router0

Reset Simulation ☑ Constant Delay Captured to: 1.832 s

Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC12	PC9	ICMP	🟪	0.000	N	0	(edit)	(delet



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delet
●	Successful	PC12	PC9	ICMP	🟦	0.000	N	0	(edit)	(delet

d.

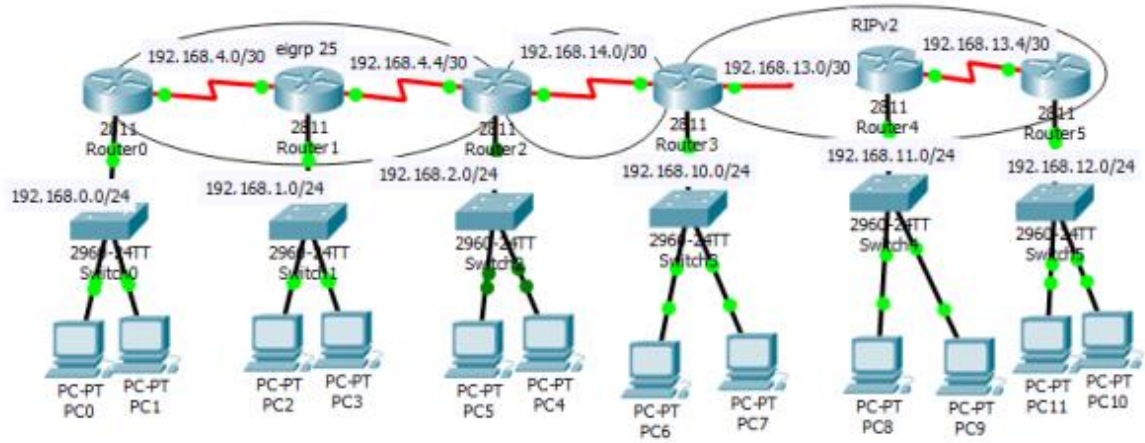
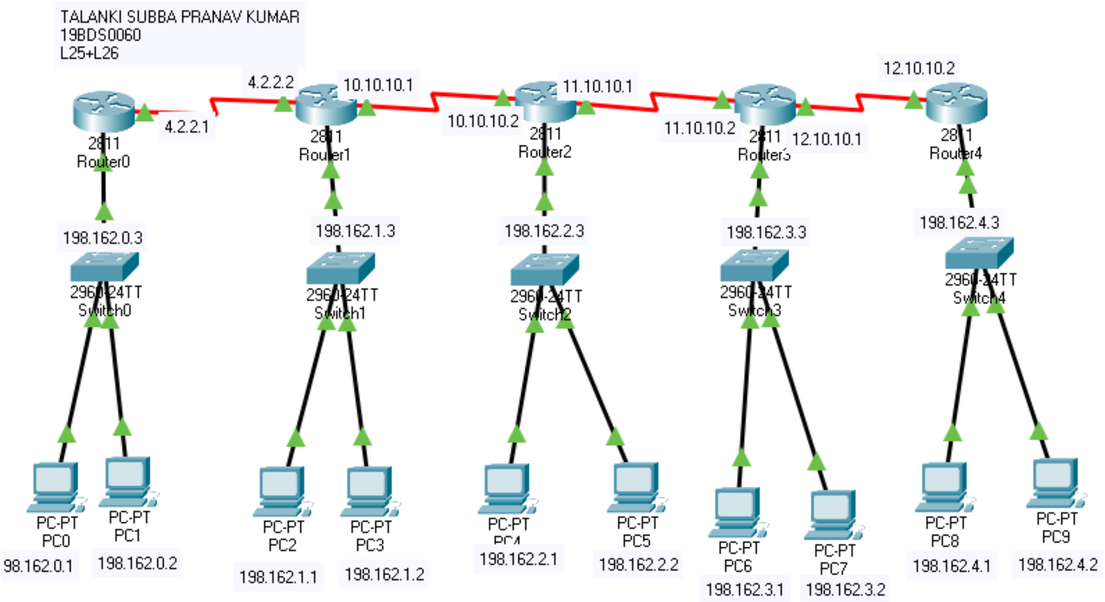


Figure 4: Network based on Switch and Router

Ans:-

Network Connection:-



For switches

->Network devices

->switches

-> 2960-24 model

For Pc's

->end devices

->Pc

For routers

->Network Devices

->routers

->2811 model

For Connections

->connections

For connections between switches and PC’s use copper straight-through

For connections between switch and router use copper straight-through

For connections between router and router use Serial DTE

I.P.ADDRESSES:-

PC0 – 198.162.0.1

PC1 – 198.162.0.2

Default gateway for PC0 and PC1 is 198.162.0.3

PC2 – 198.162.1.1

PC3 – 198.162.1.2

Default gateway for PC2 and PC3 is 198.162.1.3

PC4 – 198.162.2.1

PC5 – 198.162.2.2

Default gateway for PC4 and PC5 is 198.162.2.3

PC6 – 198.162.3.1

PC7 – 198.162.3.2

Default gateway for PC6 and PC7 is 198.162.3.3

Between Router 0 and Router 1 I.P addresses are 4.2.2.1 and 4.2.2.2

Between Router 1 and Router 2 I.P addresses are 10.10.10.1 and 10.10.10.2

Between Router 2 and Router 3 I.P addresses are 11.10.10.1 and 11.10.10.2

Between Router 3 and Router 4 I.P addresses are 12.10.10.1 and 12.10.10.2

For communication between 2 routers static routes are to be enabled

To enable static routes

Click on router→confi→static

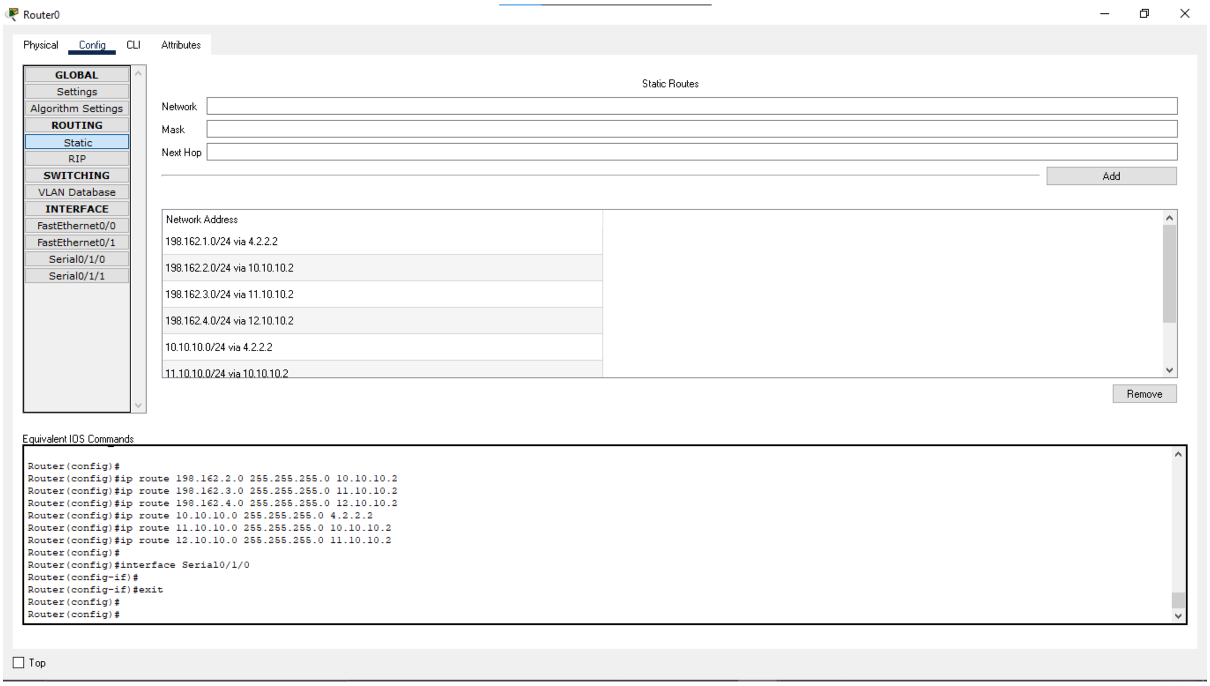
1.In network tab give the ip address of the route u want to connect

2.In mask tab give the mask of the ip address

3.In next hop tab give the address of the router

Follow the above steps to create all the routes available from a single router

Do the same with all the routers



Static Routes

Network

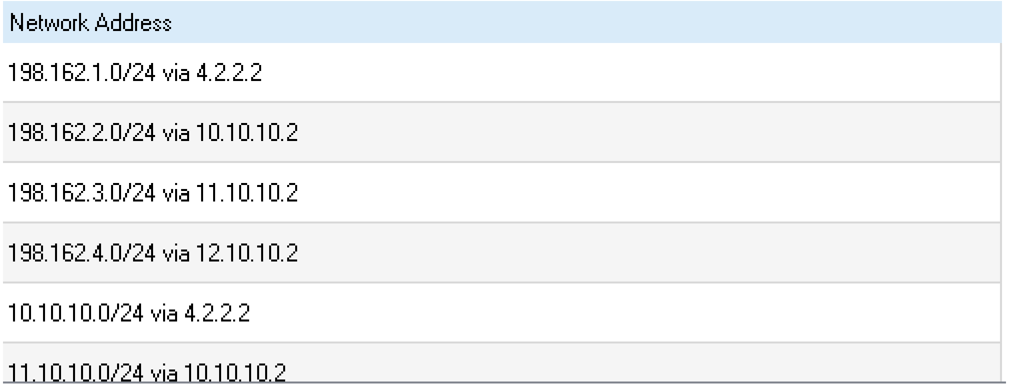
Mask

Next Hop

Network Address
198.162.1.0/24 via 4.2.2.2
198.162.2.0/24 via 10.10.10.2
198.162.3.0/24 via 11.10.10.2
198.162.4.0/24 via 12.10.10.2
10.10.10.0/24 via 4.2.2.2
11.10.10.0/24 via 10.10.10.2

Equivalent IOS Commands

```
Router(config)#
Router(config)#ip route 198.162.2.0 255.255.255.0 10.10.10.2
Router(config)#ip route 198.162.3.0 255.255.255.0 11.10.10.2
Router(config)#ip route 198.162.4.0 255.255.255.0 12.10.10.2
Router(config)#ip route 10.10.10.0 255.255.255.0 4.2.2.2
Router(config)#ip route 11.10.10.0 255.255.255.0 10.10.10.2
Router(config)#ip route 12.10.10.0 255.255.255.0 11.10.10.2
Router(config)#
Router(config)#interface Serial0/1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#
Router(config)#
```



Network Address
198.162.1.0/24 via 4.2.2.2
198.162.2.0/24 via 10.10.10.2
198.162.3.0/24 via 11.10.10.2
198.162.4.0/24 via 12.10.10.2
10.10.10.0/24 via 4.2.2.2
11.10.10.0/24 via 10.10.10.2

Simulation Results :-

1.

Source ->PC0

Destination ->PC1

2.

Source ->PC2

Destination ->PC3

3.

Source ->PC4

Destination ->PC5

4.

Source ->PC6

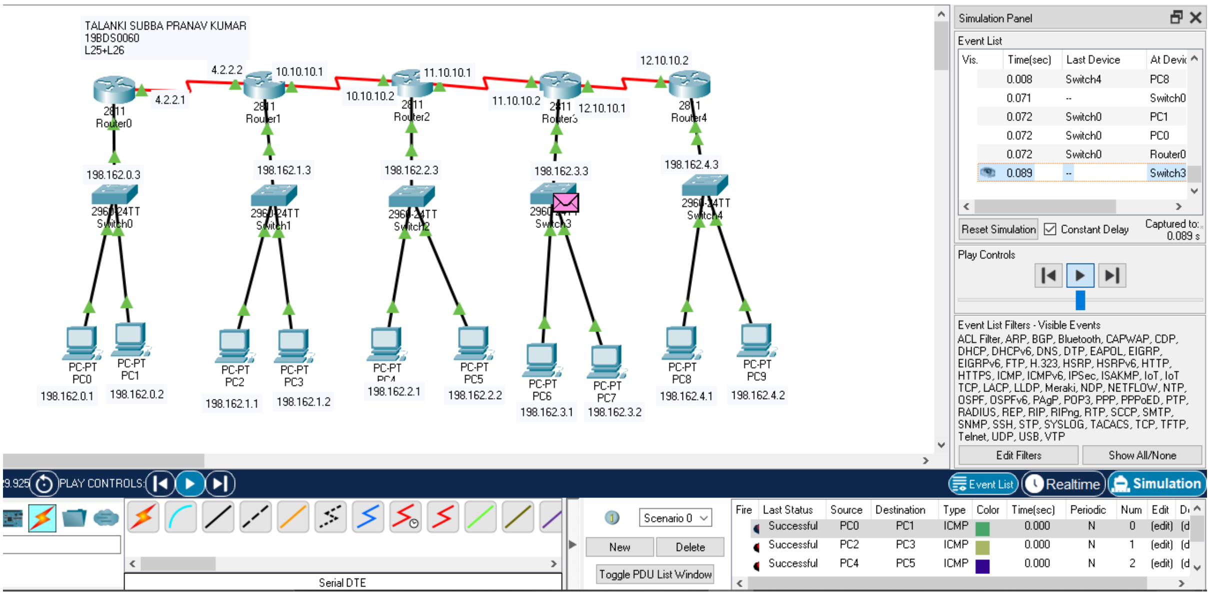
Destination ->PC7

5.

Source ->PC8

Destination ->PC9

All of the above simulations are done in a single simulation



Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Devic
	0.008	Switch4	PC8
	0.071	--	Switch0
	0.072	Switch0	PC1
	0.072	Switch0	PC0
	0.072	Switch0	Router0
	0.089	--	Switch3

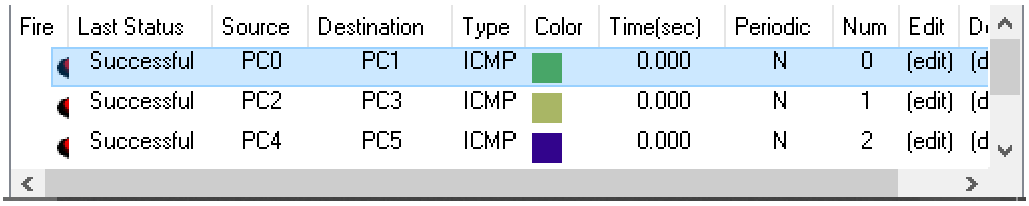
Reset Simulation ☑ Constant Delay Captured to: 0.089 s

Play Controls

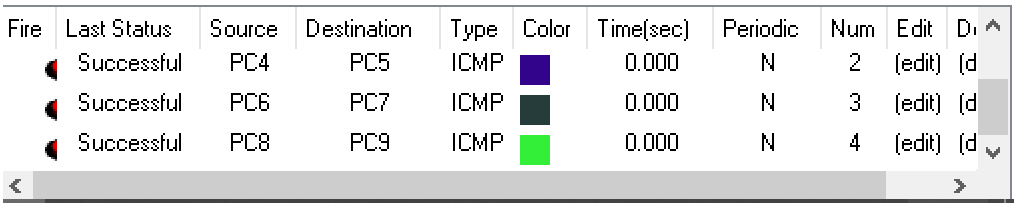
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, Meraki, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP

Edit Filters Show All/None

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	De
	Successful	PC0	PC1	ICMP	🟩	0.000	N	0	(edit)	(d
	Successful	PC2	PC3	ICMP	🟩	0.000	N	1	(edit)	(d
	Successful	PC4	PC5	ICMP	🟪	0.000	N	2	(edit)	(d



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	D
●	Successful	PC0	PC1	ICMP	🟩	0.000	N	0	(edit)	(d
●	Successful	PC2	PC3	ICMP	🟩	0.000	N	1	(edit)	(d
●	Successful	PC4	PC5	ICMP	🟪	0.000	N	2	(edit)	(d

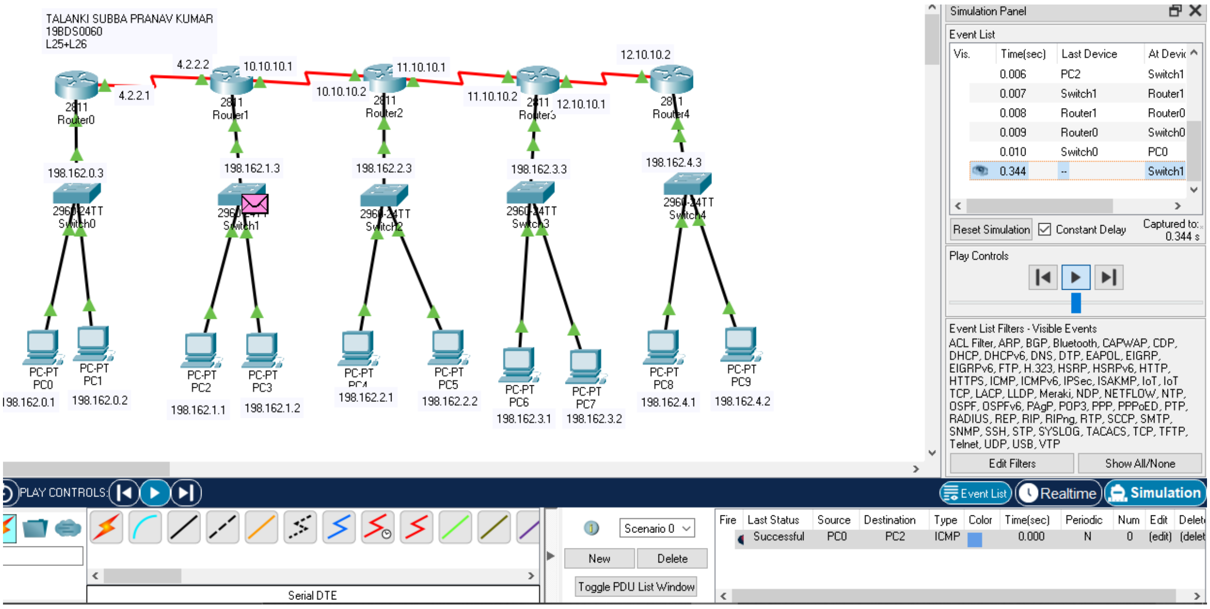


Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	D
●	Successful	PC4	PC5	ICMP	🟪	0.000	N	2	(edit)	(d
●	Successful	PC6	PC7	ICMP	⬛	0.000	N	3	(edit)	(d
●	Successful	PC8	PC9	ICMP	🟩	0.000	N	4	(edit)	(d

6.

Source ->PC0

Destination ->PC2



Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Devic
	0.006	PC2	Switch1
	0.007	Switch1	Router1
	0.008	Router1	Router0
	0.009	Router0	Switch0
	0.010	Switch0	PC0
👁	0.344	--	Switch1

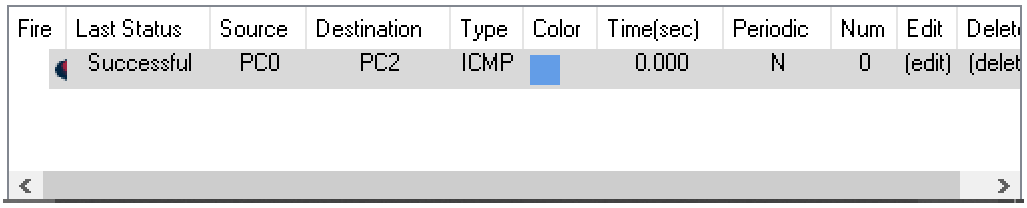
Reset Simulation ☑ Constant Delay Captured to: 0.344 s

Play Controls

Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, Meraki, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP

Edit Filters | Show All/None

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delet
●	Successful	PC0	PC2	ICMP	🟦	0.000	N	0	(edit)	(delet



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delet
●	Successful	PC0	PC2	ICMP	🟦	0.000	N	0	(edit)	(delet