NETWORK AND COMMUNICATION LAB ASSISNMENT -1

ANAMAYA VYAS
19BCE0568

++

Activity -1

Ifconfig

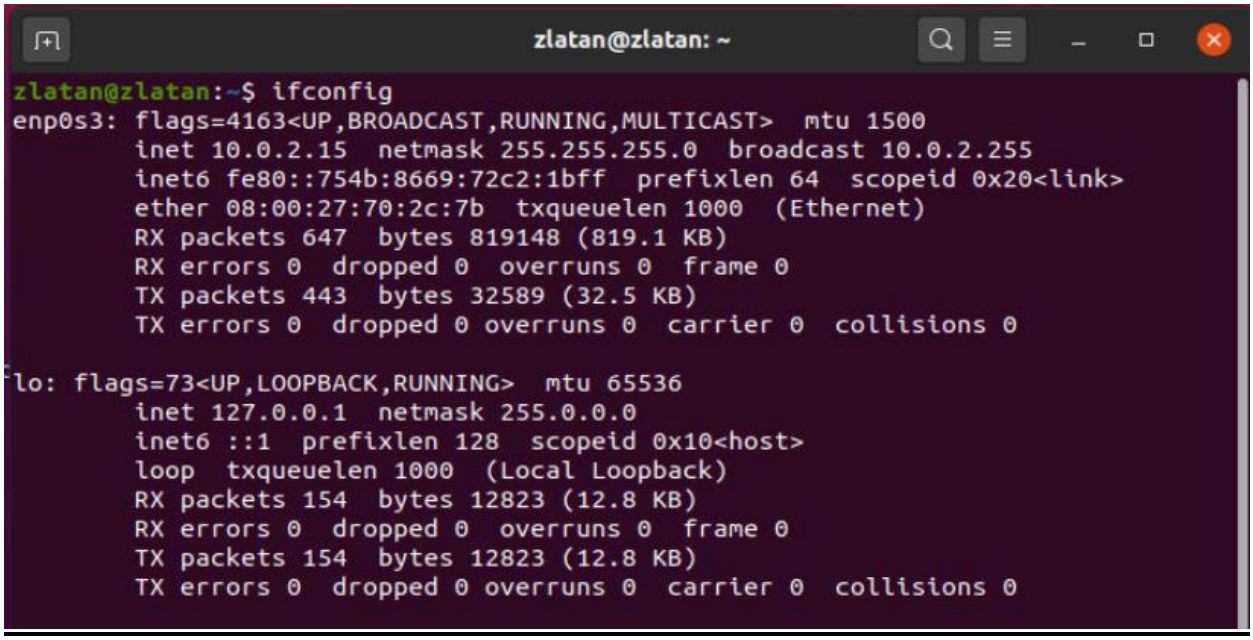
- **SYNOPSIS**

  ifconfig [interface]
  ifconfig interface [aftype] options | address ..

- **DESCRIPTION**

  Ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed. If no arguments are given, ifconfig displays the status of the currently active interfaces. If a single interface argument is given, it displays the status of the given interface only.

- **OUTPUT**



- **INTERPRETATION**

  The “ifconfig” commands with no argument will display all the active interfaces details. The ifconfig command also used to check the assigned IP address of a server.

1) ifconfig -a

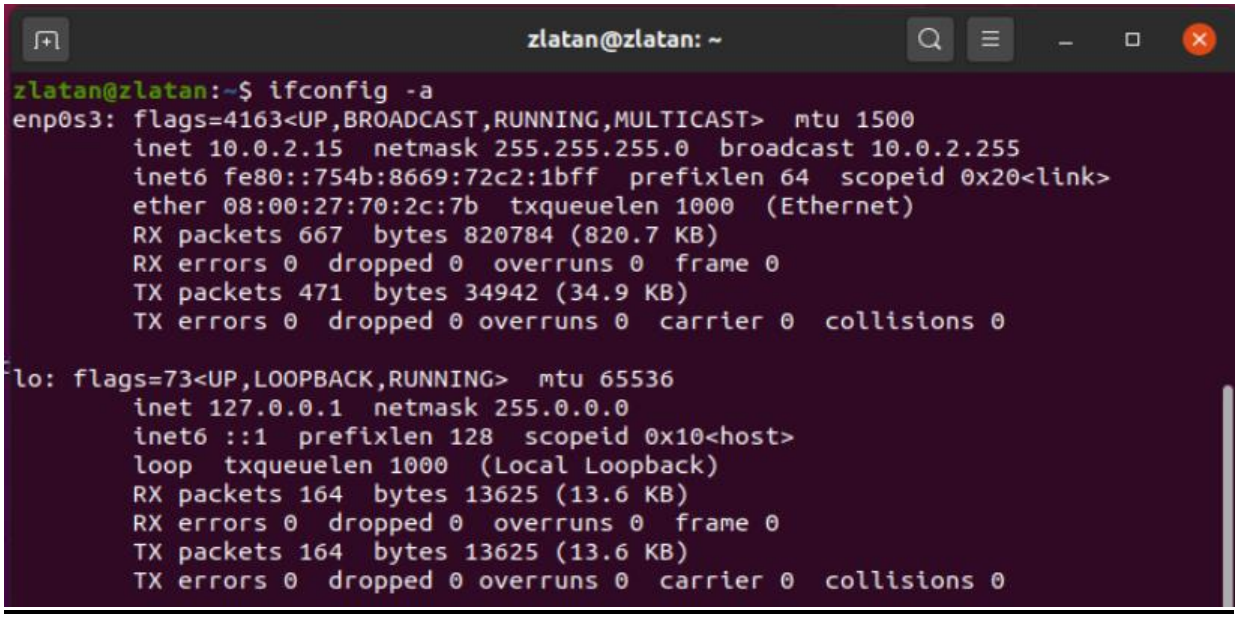
- **SYNOPSIS**

  ifconfig -a

- **DESCRIPTION**

  The following ifconfig command with -a argument will display information of all active or inactive network interfaces on server. It displays the results for enp0s3, lo, and tun0.

- **OUTPUT**



- **INTERPRETATION**

  We can see details like interface address, Netmasking address etc of every interface on server. It doesn't matter that whether it is active or inactive and if we want info for a specific interface we can use #ifconfig [interface_name] command.

2) ifconfig [interface] up

- **SYNOPSIS**

  ifconfig [interface_name] up
  OR
  ifup [interface_name]

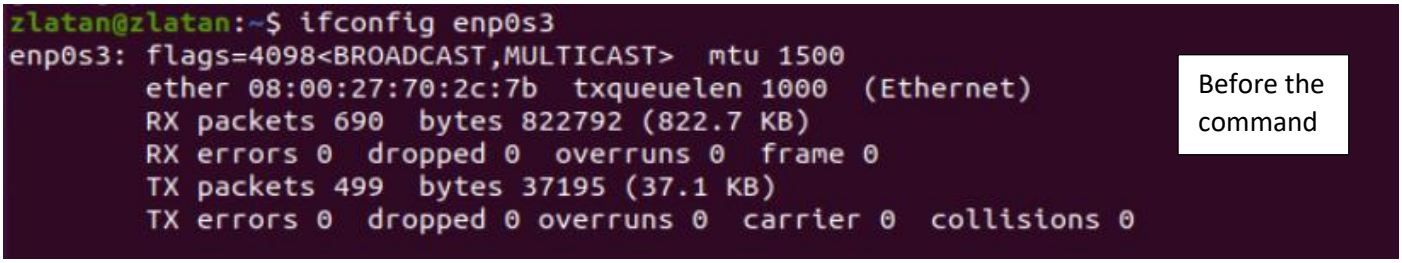
- **DESCRIPTION**

  The "up" or "ifup" flag with interface name activates an network interface, if it is not in active state and allowing to send and receive information. For example, "ifconfig enp0s3 up" or "ifup enp0s3" will activate the enp0s3 interface.

- **INTERPRETATION**

  Earlier the interface “enp0s3” was down and we couldn’t have access to internet but after that we could surf on the internet after “ifconfig enp0s3 up”. In my PC I used “sudo” with the command to avoid the following error: SIOSIFFLAGS : operation not permitted ubuntu.

- **OUTPUT**



3) ifconfig [interface] down

- **SYNOPSIS**

  ifconfig [interface_name] down
  OR
  ifdown [interface_name]

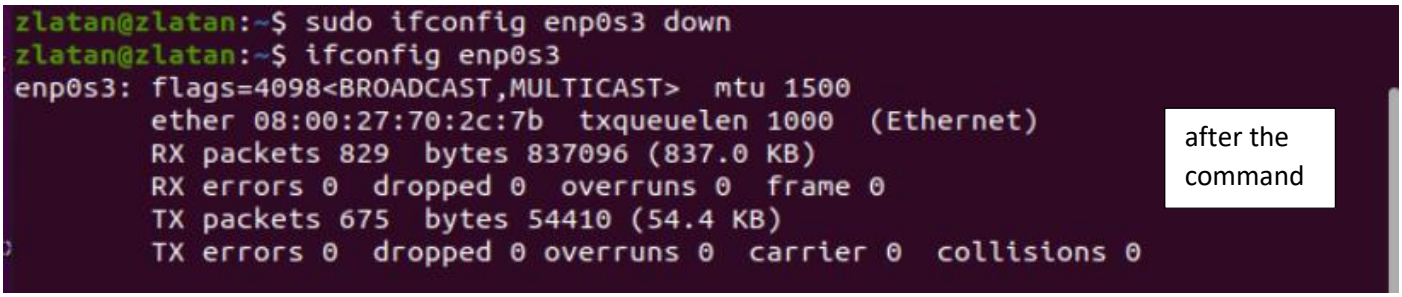
- **DESCRIPTION**

  The “down” or “ifdown” flag with interface name deactivates the specified network interface. For example, “ifconfig enp0s3 down” or “ifdown enp0s3” command deactivates the enp0s3 interface, if it is in active state.

- **INTERPRETATION**

  After the command my wifi connection which was in active state was turned down after which we couldn’t use the internet. Also we used “sudo” to avoid the error SIOSIFFLAGS : operation not permitted ubuntu.

- **OUTPUT**



4) ifconfig [interface] IP_address

- **SYNOPSIS**

  ifconfig [interface_name] IP_Address

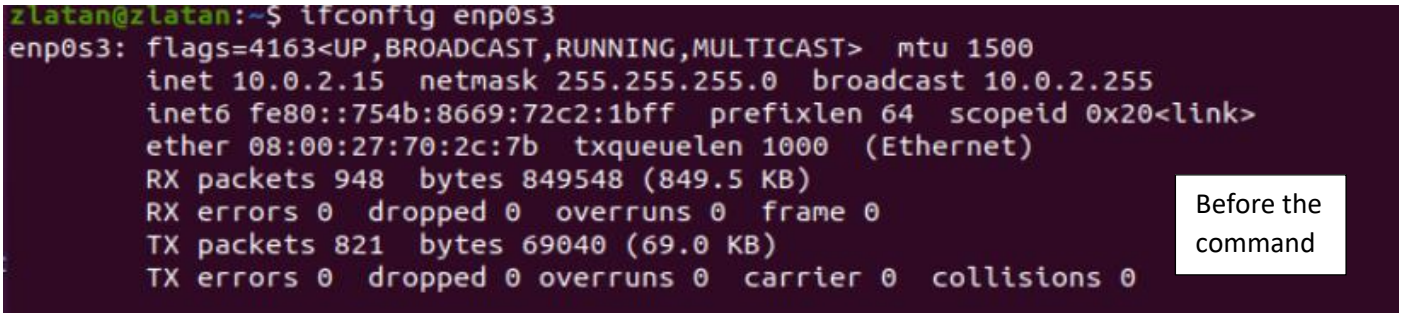
- **DESCRIPTION**

  To assign an IP address to an specific interface, use the following command with an interface name and ip address that you want to set. For example, “ifconfig enp0s3 172.16.25.125” will set the IP address to interface enp0s3.

- **INTERPRETATION**

  We can use this command to change the IP Address of the network we are using. In my case the IP address was changed to 172.16.25.125 from 10.0.2.15. We used “sudo” to avoid the error SIOSIFFLAGS : operation not permitted ubuntu.

- **OUTPUT**



5) ifconfig [interface]:0 IP_address

- **SYNOPSIS**

  ifconfig [interface_name]:0 IP_Address

- **DESCRIPTION**

  The ifconfig utility allows you to configure additional network interfaces using alias feature. To add alias network interface of enp0s3, use the following command. Please note that alias network address in same sub-net mask. For example, if your enp0s3 network ip address is 172.16.25.125, then alias ip address must be 172.16.25.127.

- **INTERPRETATION**

  We can use this command to add alias to the IP Address of the network we are using. If you no longer required an alias network interface or you incorrectly configured it, you can remove it by using the following command: **ifconfig [interface_name]:0 down**. In my case the interface name is enp0s3. We used “sudo” to avoid the error SIOSIFFLAGS : operation not permitted ubuntu.

- **OUTPUT**

Ping

- **SYNOPSIS**

  ping [option] hostname or IP address

- **DESCRIPTION**

The Linux ping command is a simple utility used to check whether a network is available and if a host is reachable. With this command, you can test if a server is up and running. It also helps with troubleshooting various connectivity issues.
The ping command allows you to:

1. Test your internet connection.
2. Check if a remote machine is online.
3. Checks if there are network issues, such as dropped packages or high latency.

- **OUTPUT**

```
zlatan@zlatan:~$ ping realmadrid.com
PING realmadrid.com (23.57.247.219) 56(84) bytes of data.
64 bytes from a23-57-247-219.deploy.static.akamaitechnologies.com (23.57.247.219): icmp_seq=1 ttl=52 time=88.9 ms
64 bytes from a23-57-247-219.deploy.static.akamaitechnologies.com (23.57.247.219): icmp_seq=2 ttl=52 time=64.5 ms
64 bytes from a23-57-247-219.deploy.static.akamaitechnologies.com (23.57.247.219): icmp_seq=3 ttl=52 time=71.4 ms
64 bytes from a23-57-247-219.deploy.static.akamaitechnologies.com (23.57.247.219): icmp_seq=4 ttl=52 time=69.9 ms
64 bytes from a23-57-247-219.deploy.static.akamaitechnologies.com (23.57.247.219): icmp_seq=5 ttl=52 time=104 ms
64 bytes from a23-57-247-219.deploy.static.akamaitechnologies.com (23.57.247.219): icmp_seq=6 ttl=52 time=106 ms
64 bytes from a23-57-247-219.deploy.static.akamaitechnologies.com (23.57.247.219): icmp_seq=7 ttl=52 time=79.9 ms
64 bytes from a23-57-247-219.deploy.static.akamaitechnologies.com (23.57.247.219): icmp_seq=8 ttl=52 time=56.9 ms
64 bytes from a23-57-247-219.deploy.static.akamaitechnologies.com (23.57.247.219): icmp_seq=9 ttl=52 time=55.5 ms
^C
--- realmadrid.com ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8114ms
rtt min/avg/max/mdev = 55.484/77.388/105.905/17.643 ms
```

- **INTERPRETATION**

- **from:** The destination and its IP address. Note that the IP address may be different for a website depending on your geographical location.
- **icmp_seq=1:** The sequence number of each ICMP packet. Increases by one for every subsequent echo request.
- **ttl=52:** The Time to Live value from 1 to 255. It represents the number of network hops a packet can take before a router discards it.
- **time=88.9 ms:** The time it took a packet to reach the destination and come back to the source. Expressed in milliseconds.
- **min:** minimum time to get a response
- **avg:** average time to get responses
- **max:** maximum time to get a response

1) ping localhost

- **SYNOPSIS**

ping localhost
OR
Ping 0

- **DESCRIPTION**

This is the quickest way to ping localhost. Once you type this command, the terminal resolves the IP address and provides a response.

- **INTERPRETATION**

You can use the name to ping localhost. The name refers to your computer, and when we use this command, we say: “ping this computer.” With this we can find the time our computer is taking to reach our localhost and returning the results.

- **OUTPUT**

```
zlatan@zlatan:~$ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.021 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.029 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.065 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.092 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.029 ms
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.023 ms
64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.029 ms
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.028 ms
^C
--- localhost ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7224ms
rtt min/avg/max/mdev = 0.021/0.039/0.092/0.023 ms
```

2) ping -c number hostname

- **SYNOPSIS**

  ping -c [number] [hostname]

- **DESCRIPTION**

  To make the ping command automatically stop after it sends a certain number of packets, use -c and a number. This sets the desired amount of ping requests.

- **INTERPRETATION**

  In my case I used this to ping “realmadrid.com” host and ordered the number of packets 4 times with that I got 79.3 ms as the minimum time to reach the packet, 112 ms as the maximum time to reach the packet and 97.26 ms as the average time to reach the packet.

- **OUTPUT**

```
zlatan@zlatan:~$ ping -c 4 realmadrid.com
PING realmadrid.com (23.57.220.117) 56(84) bytes of data.
64 bytes from a23-57-220-117.deploy.static.akamaitechnologies.com (23.57.220.117): icmp_seq=1 ttl=52 time=79.3 ms
64 bytes from a23-57-220-117.deploy.static.akamaitechnologies.com (23.57.220.117): icmp_seq=2 ttl=52 time=112 ms
64 bytes from a23-57-220-117.deploy.static.akamaitechnologies.com (23.57.220.117): icmp_seq=3 ttl=52 time=93.4 ms
64 bytes from a23-57-220-117.deploy.static.akamaitechnologies.com (23.57.220.117): icmp_seq=4 ttl=52 time=105 ms

--- realmadrid.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3025ms
rtt min/avg/max/mdev = 79.261/97.360/111.526/12.309 ms
```

3) sudo ping -f hostname-IP

- **SYNOPSIS**

  sudo ping -f hostname-IP

- **DESCRIPTION**

  We can use ping flood to test our network performance under heavy load. Ping flood -f option requires root to execute. Otherwise, we can use sudo to ping command to flood a host. This command sends a large number of packets as soon as possible.
  The output prints a dot for every sent package, and a backspace for every response. The statistics line shows a summary of the ping command.

- **INTERPRETATION**

In my case I used this to ping “realmadrid.com” host and transmitted 1400 packets and received 1389 packets, this whole process was going on for 23658ms and minimum time to send the packet was 126.25 ms , maximum time was 233.30 ms and with the average time of 167.811 ms while the process was taking place

- **OUTPUT**

```
zlatan@zlatan:~$ sudo ping -f realmadrid.com
PING realmadrid.com (23.198.127.226) 56(84) bytes of data.
...........^C
--- realmadrid.com ping statistics ---
1400 packets transmitted, 1389 received, 0.785714% packet loss, time 23658ms
rtt min/avg/max/mdev = 129.251/167.811/233.308/20.506 ms, pipe 15, ipg/ewma 16.910/177.307 ms
```

4) ping -w number hostname

- **SYNOPSIS**

ping –w number hostname

- **DESCRIPTION**

To stop receiving a ping output after a specific amount of time, add -w and an interval in seconds to your command.

- **INTERPRETATION**

In my case I used this to ping “google.com” host and this whole process was going on for 20 secs and minimum time to send the packet was 40.72 ms , maximum time was 444.19 ms and with the average time of 87.02 ms while the process was taking place

- **OUTPUT**

```
zlatan@zlatan:~$ ping -w 20 google.com
PING google.com (216.58.200.174) 56(84) bytes of data.
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=1 ttl=110 time=42.1 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=2 ttl=110 time=54.4 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=3 ttl=110 time=63.5 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=4 ttl=110 time=59.9 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=5 ttl=110 time=444 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=6 ttl=110 time=75.0 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=7 ttl=110 time=102 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=8 ttl=110 time=70.0 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=9 ttl=110 time=68.8 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=10 ttl=110 time=56.7 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=11 ttl=110 time=67.5 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=12 ttl=110 time=40.7 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=13 ttl=110 time=59.4 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=14 ttl=110 time=56.8 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=15 ttl=110 time=84.9 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=16 ttl=110 time=66.9 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=17 ttl=110 time=80.1 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=18 ttl=110 time=65.8 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=19 ttl=110 time=68.4 ms
64 bytes from nrt12s11-in-f174.1e100.net (216.58.200.174): icmp_seq=20 ttl=110 time=113 ms

--- google.com ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19133ms
rtt min/avg/max/mdev = 40.726/87.024/444.185/83.662 ms
```

5) ping -c number -M want [hostname]

- **SYNOPSIS**

  ping -c number -M want [hostname]

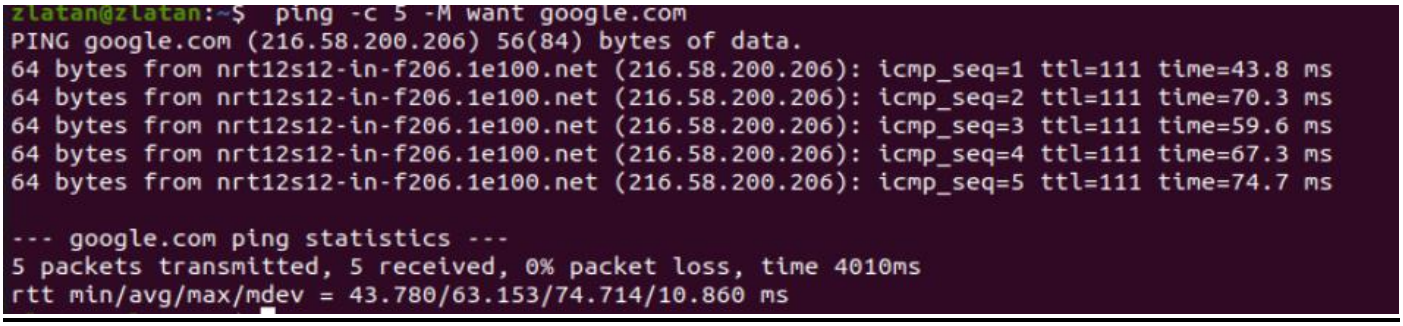
- **DESCRIPTION**

  It is a simple protocol to find out the maximum MTU(Maximum Transmission Unit) a TCP path can take.We use an option with -m do (prohibit fragmentation), want (do PMTU discovery, fragment locally when packet size is large), or dont (do not set DF flag).

- **INTERPRETATION**

  In my case I used this to ping “google.com” host and minimum time to send the packet was 43.78 ms , maximum time was 74.71 ms and with the average time of 63.15 ms while the process was taking place.

- **OUTPUT**



Traceroute

- **SYNOPSIS**

  traceroute [options] host_Address [pathlength]

- **DESCRIPTION**

  Traceroute command in Linux prints the route that a packet takes to reach the host. This command is useful when you want to know about the route and about all the hops that a packet takes.
  The first line gives us the following info:

  - The destination and its IP address.
  - The number of hops traceroute will try before giving up.
  - The size of the UDP packets we’re sending.

- **INTERPRETATION**

  The first column corresponds to the hop count. The second column represents the address of that hop and after that, you see three space-separated time in milliseconds. traceroute

command sends three packets to the hop and each of the time refers to the time taken by the packet to reach the hop.

- **OUTPUT**

```
zlatan@zlatan:~$ traceroute realmadrid.com
traceroute to realmadrid.com (184.26.204.174), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.351 ms  0.313 ms  0.292 ms
 2  _gateway (10.0.2.2)  11.843 ms  11.800 ms  11.790 ms
```

1) traceroute -n [hostname]

- **SYNOPSIS**

  Traceroute -n [hostname]

- **DESCRIPTION**

  Sometimes device names leads to a cluttered display. To make it easier to see the data, you can use the -n (no mapping) option.

- **INTERPRETATION**

  Print hop addresses numerically rather than symbolically and numerically (saves a nameserver address-to-name lookup for each gateway found on the path).

- **OUTPUT**

```
zlatan@zlatan:~$ traceroute -n realmadrid.com
traceroute to realmadrid.com (23.57.247.219), 30 hops max, 60 byte packets
 1  10.0.2.2  2.954 ms  3.016 ms  2.980 ms
 2  10.0.2.2  12.517 ms  13.974 ms  19.060 ms
```

2) traceroute -f [number] [hostname]

- **SYNOPSIS**

  Traceroute -f [number] [hostname]

- **DESCRIPTION**

  Set the initial time-to-live used in the first outgoing probe packet.

- **INTERPRETATION**

  We can set the initial value of TTL to something other than one, and skip some hops. Usually, the TTL values are set to one for the first set of tests, two for the next set of tests, and so on. If we set it to five, the first test will attempt to get to hop five and skip hops one through four.

- **OUTPUT**

```
zlatan@zlatan:~$ traceroute -f 10 realmadrid.com
traceroute to realmadrid.com (104.120.94.78), 30 hops max, 60 byte packets
10  _gateway (10.0.2.2)  126.611 ms  126.578 ms  126.500 ms
```

3) traceroute -m [number] [hostname]

- **SYNOPSIS**

  Traceroute -f [number] [hostname]

- **DESCRIPTION**

  Set the max time-to-live (max number of hops) used in outgoing probe packets. The default is 30 hops (the same default used for TCP connections).

- **INTERPRETATION**

  For example if we give the value 5 the max time-to-live will not be more than 5 in the following example we can see it is only 2.

- **OUTPUT**

```
zlatan@zlatan:~$ traceroute  -m 5 google.com
traceroute to google.com (172.217.160.238), 5 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.604 ms  0.550 ms  0.540 ms
 2  _gateway (10.0.2.2)  245.598 ms  245.522 ms  245.433 ms
```

4) traceroute -F [hostname]

- **SYNOPSIS**

  Traceroute -F [hostname]

- **DESCRIPTION**

  Set the "don't fragment" bit.

- **INTERPRETATION**

  Set the "Don’t Fragment" bit. This tells intermediate routers not to fragment the packet when they find it’s too big for a network hop’s MTU.

- **OUTPUT**

```
zlatan@zlatan:~$ traceroute  -F realmadrid.com
traceroute to realmadrid.com (23.198.127.226), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.917 ms  0.832 ms  0.748 ms
 2  _gateway (10.0.2.2)  8.892 ms  9.008 ms  8.942 ms
```

5) traceroute -F [hostname]

- **SYNOPSIS**

  Traceroute -F [hostname]

- **DESCRIPTION**

  Set the time (in seconds) to wait for a response to a probe (default 5 sec.).

- **INTERPRETATION**

  If we extend the default timeout period (five seconds), we'll get more responses. To do this, we'll use the -w (wait time) option to change it to seven seconds. Note this is a floating-point number.

- **OUTPUT**

```
zlatan@zlatan:~$ traceroute -w 5.0 realmadrid.com
traceroute to realmadrid.com (23.57.247.219), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.499 ms  0.472 ms  0.450 ms
 2  _gateway (10.0.2.2)  13.712 ms  20.255 ms  17.562 ms
```

Netstat

- **SYNOPSIS**

  netstat -a

- **DESCRIPTION**

  Netstat command displays various network related information such as network connections, routing tables, interface statistics, masquerade connections, multicast memberships etc.

- **INTERPRETATION**

  It is a command line tool for monitoring network connections both incoming and outgoing as well as viewing routing tables, interface statistics etc. It is one of the most basic network service debugging tools, telling you what ports are open and whether any programs are listening on ports.

- **OUTPUT**

```
zlatan@zlatan:~$  netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
udp        0      0 0.0.0.0:mdns            0.0.0.0:*
udp        0      0 0.0.0.0:42386           0.0.0.0:*
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp        0      0 localhost:domain        0.0.0.0:*
udp        0      0 zlatan:bootpc           _gateway:bootps         ESTABLISHED
udp6       0      0 [::]:mdns               [::]:*
udp6       0      0 [::]:34530              [::]:*
raw6       0      0 [::]:ipv6-icmp          [::]:*                  7
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     33077    @/tmp/.ICE-unix/1700
unix  2      [ ACC ]     SEQPACKET  LISTENING     1097     /run/udev/control
unix  2      [ ACC ]     STREAM     LISTENING     1070     /run/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     1072     /run/systemd/userdb/io.systemd.DynamicUser
unix  2      [ ]         DGRAM                    1081     /run/systemd/journal/syslog
unix  2      [ ]         DGRAM                    29253    /run/user/1000/systemd/notify
unix  2      [ ACC ]     STREAM     LISTENING     1083     /run/systemd/fsck.progress
unix  2      [ ACC ]     STREAM     LISTENING     29256    /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     29264    /run/user/1000/bus
unix  20     [ ]         DGRAM                    1091     /run/systemd/journal/dev-log
unix  2      [ ACC ]     STREAM     LISTENING     1093     /run/systemd/journal/stdout
unix  8      [ ]         DGRAM                    1095     /run/systemd/journal/socket
unix  2      [ ACC ]     STREAM     LISTENING     29265    /run/user/1000/gnupg/S.dirmngr
unix  2      [ ACC ]     STREAM     LISTENING     29266    /run/user/1000/gnupg/S.gpg-agent.browser
unix  2      [ ACC ]     STREAM     LISTENING     29267    /run/user/1000/gnupg/S.gpg-agent.extra
unix  2      [ ACC ]     STREAM     LISTENING     29268    /run/user/1000/gnupg/S.gpg-agent.ssh
unix  2      [ ACC ]     STREAM     LISTENING     29269    /run/user/1000/gnupg/S.gpg-agent
unix  2      [ ACC ]     STREAM     LISTENING     29270    /run/user/1000/pk-debconf-socket
unix  2      [ ACC ]     STREAM     LISTENING     29271    /run/user/1000/pulse/native
```

1) netstat -lu

- **SYNOPSIS**

  netstat -lu

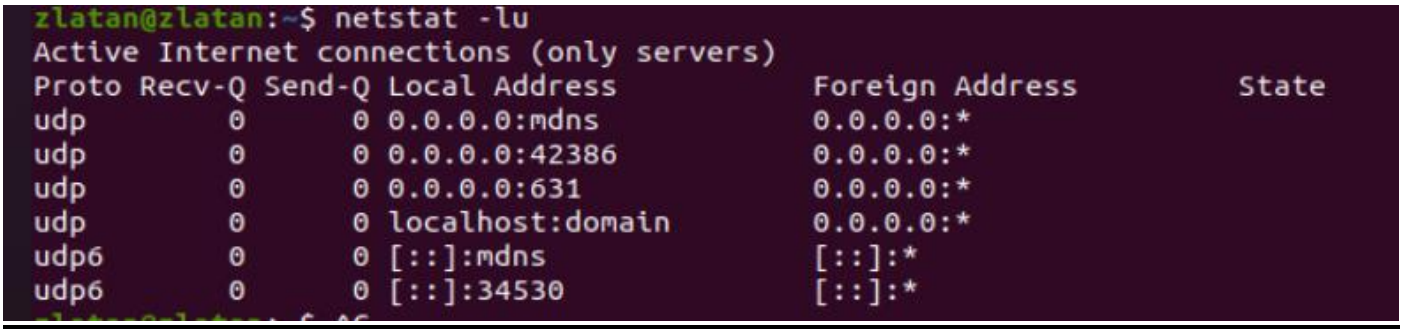
- **DESCRIPTION**

  Listing all active listening UDP ports by using option netstat -lu.

- **INTERPRETATION**

  To list only the listening udp ports.

- **OUTPUT**



2) netstat -lt

- **SYNOPSIS**

  netstat -lt

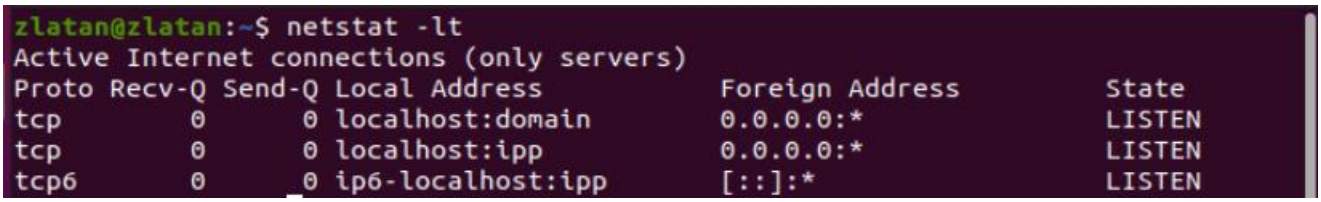
- **DESCRIPTION**

  Listing all active listening TCP ports by using option netstat -lt.

- **INTERPRETATION**

  To list only the listening tcp ports.

- **OUTPUT**



3) netstat -s

- **SYNOPSIS**

  netstat -s

- **DESCRIPTION**

  To list the statistics for all ports.

- **INTERPRETATION**

  Displays statistics by protocol. By default, statistics are shown for the TCP, UDP, ICMP, and IP protocols. The -s parameter can be used to specify a set of protocols.

- **OUTPUT**

```
zlatan@zlatan:~$ netstat -s
Ip:
    Forwarding: 2
    14607 total packets received
    5 with invalid addresses
    0 forwarded
    0 incoming packets discarded
    14600 incoming packets delivered
    14801 requests sent out
    20 outgoing packets dropped
    2262 dropped because of missing route
Icmp:
    1999 ICMP messages received
    48 input ICMP message failed
    ICMP input histogram:
        destination unreachable: 195
        timeout in transit: 27
        echo requests: 8
        echo replies: 1769
    3597 ICMP messages sent
    0 ICMP messages failed
    ICMP output histogram:
        destination unreachable: 51
        echo requests: 3538
        echo replies: 8
IcmpMsg:
        InType0: 1769
        InType3: 195
        InType8: 8
        InType11: 27
        OutType0: 8
        OutType3: 51
        OutType8: 3538
```

```
Tcp:
    456 active connection openings
    0 passive connection openings
    5 failed connection attempts
    6 connection resets received
    0 connections established
    7955 segments received
    8117 segments sent out
    15 segments retransmitted
    0 bad segments received
    163 resets sent
Udp:
    4598 packets received
    43 packets to unknown port received
    0 packet receive errors
    4900 packets sent
    0 receive buffer errors
    0 send buffer errors
    IgnoredMulti: 4
UdpLite:
TcpExt:
    208 TCP sockets finished time wait in fast timer
    66 delayed acks sent
    Quick ack mode was activated 11 times
    2844 packet headers predicted
    1417 acknowledgments not containing data payload received
    2173 predicted acknowledgments
    TCPLostRetransmit: 8
    TCPTimeouts: 15
    1 connections reset due to unexpected data
    TCPRcvCoalesce: 153
    TCPOFOQueue: 23
    TCPAutoCorking: 26
    TCPSynRetrans: 15
    TCPOrigDataSent: 2976
    TCPHystartTrainDetect: 2
    TCPHystartTrainCwnd: 41
    TCPKeepAlive: 634
    TCPDelivered: 3393
```

```
IpExt:
    InMcastPkts: 155
    OutMcastPkts: 168
    InBcastPkts: 4
    OutBcastPkts: 4
    InOctets: 7567228
    OutOctets: 1678654
    InMcastOctets: 18187
    OutMcastOctets: 18707
    InBcastOctets: 310
    OutBcastOctets: 310
    InNoECTPkts: 15591
    InECT0Pkts: 1756
```

4) netstat -r

- **SYNOPSIS**

  netstat -r

- **DESCRIPTION**

  Display Kernel IP routing table with netstat and route command.

- **INTERPRETATION**

  To get the kernel routing information.

- **OUTPUT**

```
zlatan@zlatan:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         _gateway        0.0.0.0         UG        0 0          0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
link-local      0.0.0.0         255.255.0.0     U         0 0          0 enp0s3
```

5) netstat -i

- **SYNOPSIS**

  netstat -i

- **DESCRIPTION**

  Showing network interface packet transactions including both transferring and receiving packets with MTU size.

- **INTERPRETATION**

  To get the list of network interfaces.

- **OUTPUT**

```
zlatan@zlatan:~$ netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
enp0s3    1500    14846      0      0 0          12615      0      0      0 BMRU
lo       65536     2493      0      0 0           2493      0      0      0 LRU
```

Nslookup

- **SYNOPSIS**

  nslookup [option]

- **DESCRIPTION**

  NsLookup queries the specified DNS server and retrieves the requested records that are associated with the domain name you provided. These records contain information like the domain name's IP addresses.

- **INTERPRETATION**

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.

- **OUTPUT**

```
zlatan@zlatan:~$ nslookup realmadrid.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   realmadrid.com
Address: 23.57.247.219
```

1) nslookup -query=mx [hostname]

- **SYNOPSIS**

  nslookup -query=mx [hostname]

- **DESCRIPTION**

  To display MX records (the mail servers responsible for accepting email messages on behalf of a recipient’s domain), set the DNS query type to MX.

- **INTERPRETATION**

  MX record is being used to map a domain name to a list of mail exchange servers for that domain. So that it tells that whatever mail received / sent to @realmadrid.com will be routed to mail server.

- **OUTPUT**

```
zlatan@zlatan:~$ nslookup -query=mx www.realmadrid.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
www.realmadrid.com      canonical name = realmadrid.edgekey.net.
realmadrid.edgekey.net  canonical name = e14202.g.akamaiedge.net.

Authoritative answers can be found from:
```

2) nslookup [IP_ADDRESS]

- **SYNOPSIS**

  nslookup [IP_ADDRESS]

- **DESCRIPTION**

  To perform a reverse DNS lookup, enter the IP address of a host.

- **INTERPRETATION**

You can also do the reverse DNS look-up by providing the IP Address as argument to nslookup.

- **OUTPUT**

```
zlatan@zlatan:~$ nslookup 127.0.0.53
53.0.0.127.in-addr.arpa name = localhost.

Authoritative answers can be found from:
```

3) nslookup -type=soa [hostname]

- **SYNOPSIS**

nslookup -type=soa [hostname]

- **DESCRIPTION**

Lookup for an soa record SOA record (start of authority), provides the authoritative information about the domain, the e-mail address of the domain admin, the domain serial number, etc…

- **INTERPRETATION**

For any hostname we can get info about domain, email etc after using the command. Here we used the hostname realmadrid.com found the domain email as admin.realmadrid.com and many more info.

- **OUTPUT**

```
zlatan@zlatan:~$ nslookup -type=soa realmadrid.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
realmadrid.com
        origin = pdns80.ultradns.com
        mail addr = admin.realmadrid.com
        serial = 2018011863
        refresh = 36000
        retry = 3600
        expire = 2419200
        minimum = 172800

Authoritative answers can be found from:
```

4) nslookup -type=ns [hostname]

- **SYNOPSIS**

nslookup -type=ns [hostname]

- **DESCRIPTION**

  Lookup for an ns record, NS (Name Server) record maps a domain name to a list of DNS servers authoritative for that domain. It will output the name serves which are associated with the given domain.

- **INTERPRETATION**

  One or more authoritative name server records for the domain.

- **OUTPUT**

```
zlatan@zlatan:~$ nslookup -type=ns realmadrid.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
realmadrid.com  nameserver = pdns80.ultradns.biz.
realmadrid.com  nameserver = pdns80.ultradns.org.
realmadrid.com  nameserver = pdns80.ultradns.com.
realmadrid.com  nameserver = pdns80.ultradns.net.

Authoritative answers can be found from:
```

5) nslookup -type=any [hostname]

- **SYNOPSIS**

  nslookup -type=any [hostname]

- **DESCRIPTION**

  We can also view all the available DNS records using -type=any option.

- **INTERPRETATION**

  To display all the available DNS records, use the -type=any option.

- **OUTPUT**

```
zlatan@zlatan:~$ nslookup -type=any realmadrid.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
realmadrid.com
        origin = pdns80.ultradns.com
        mail addr = admin.realmadrid.com
        serial = 2018011863
        refresh = 36000
        retry = 3600
        expire = 2419200
        minimum = 172800
realmadrid.com  nameserver = pdns80.ultradns.net.
realmadrid.com  nameserver = pdns80.ultradns.biz.
realmadrid.com  nameserver = pdns80.ultradns.org.
realmadrid.com  nameserver = pdns80.ultradns.com.

Authoritative answers can be found from:
```

Activity -2

Perform following exercise using 'Cisco Packet Tracer'. Create the following network as shown in the figure below. Ensure that the devices in LAN 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.

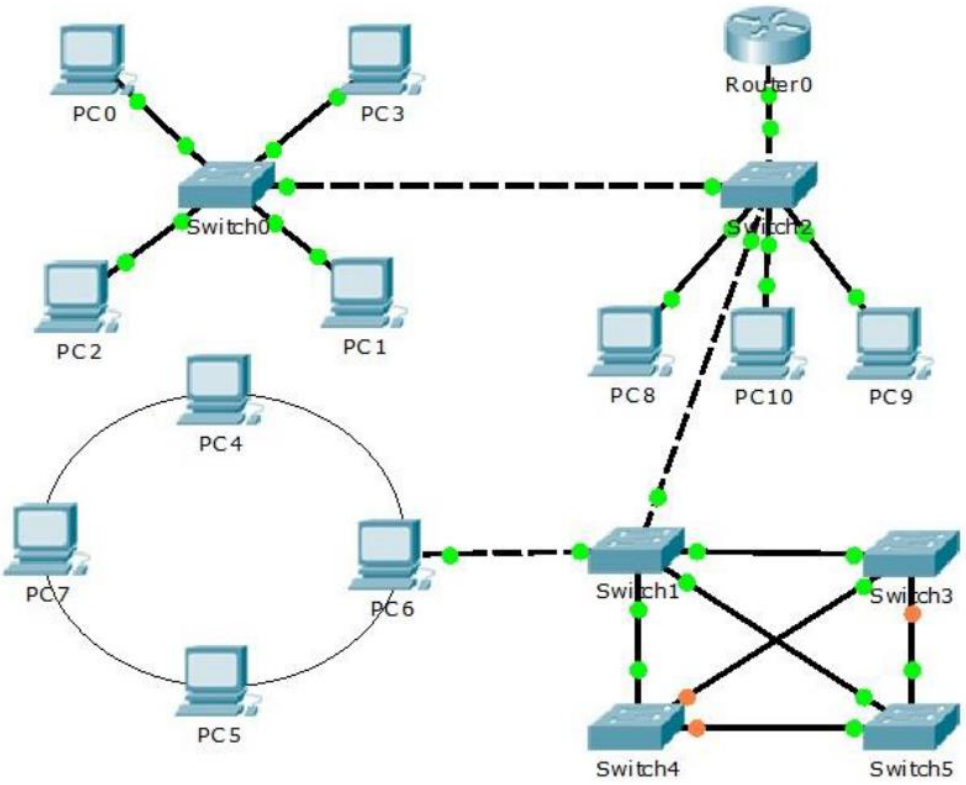


Figure 1: Network based on basic topology

1)

Ans

Method

Q1

→ Choose Switch 2960-24TT (x6)

→ Choos 7 PC

→ 1 1941 Router

→ First connect the Switch0 with 4 PCs using Straight-through wire.

→ Now do the same by connecting 3PC with Switch 1 using Straight through wire

→ Form a mesh using cross over wire b connect the mesh with Switch 1 & and connect switch 1 wit switch 2 using cross-wire.

→ Connect the router to switch 1 using 2 straight-through wire.

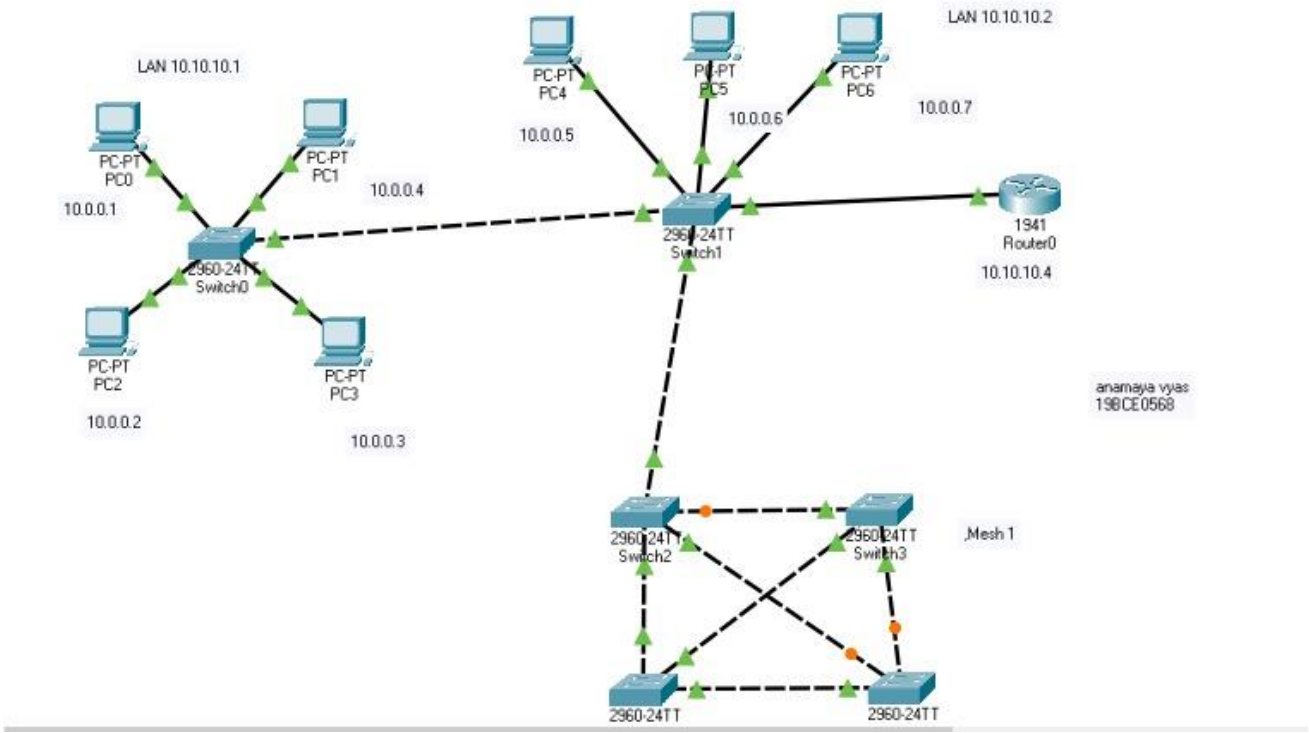
→ click on PC go to Desktop then IP config give IP address to PC

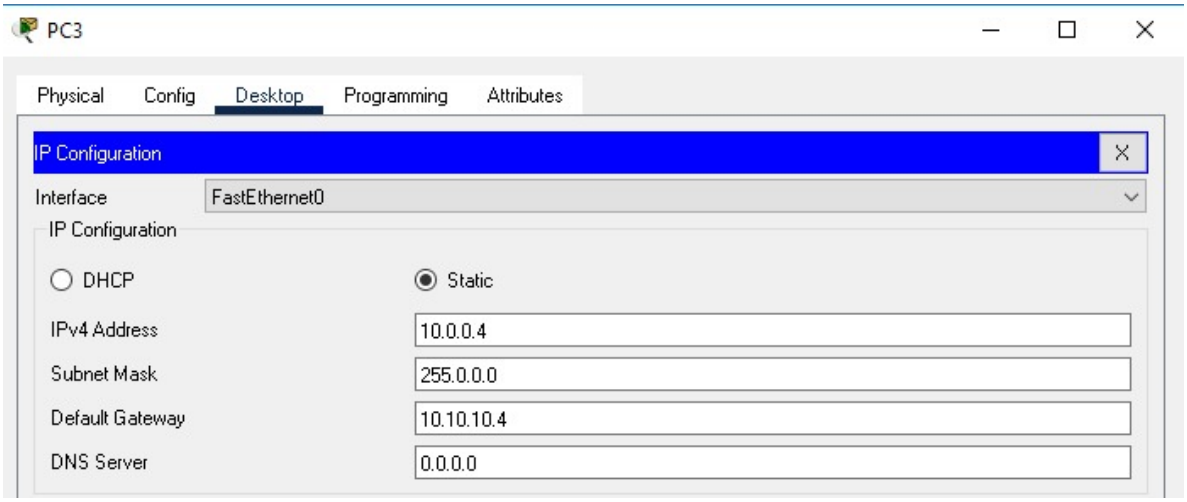
→ Also give Default gateway of the IP address of router To PC.

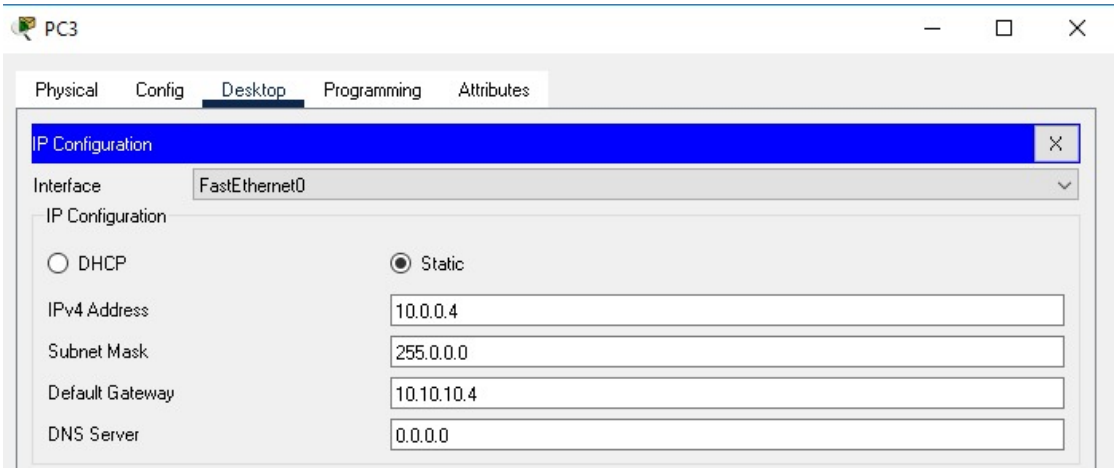
→ Click on Router → Config → give IP address to Router

→ Click on PC 10.0.0.1 and then go to desktop then cmd then run ping 10.0.0.6

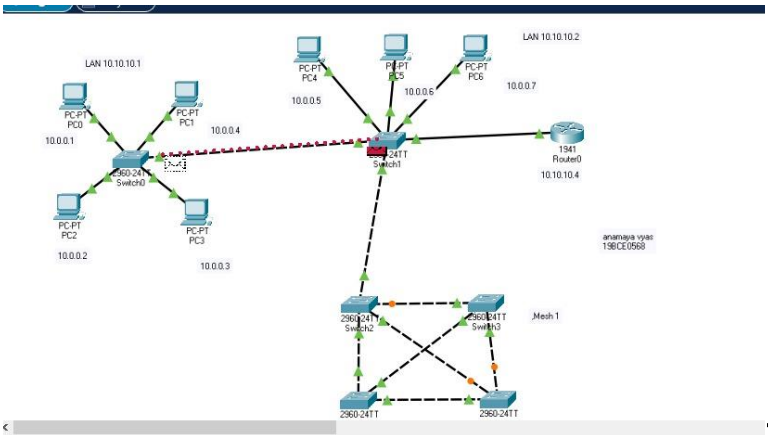
→ Your Network is made.



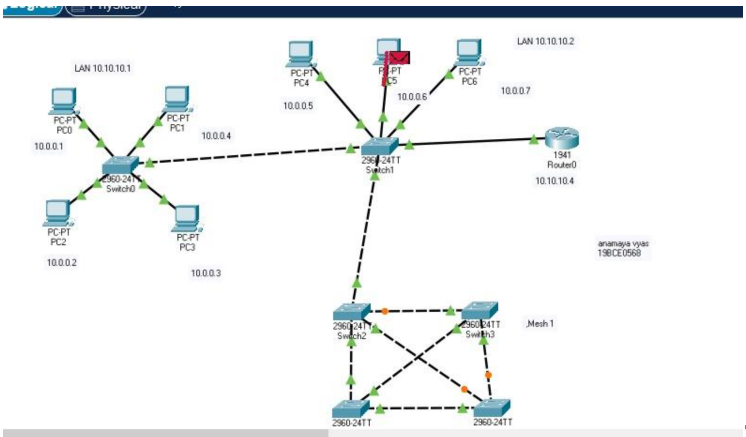




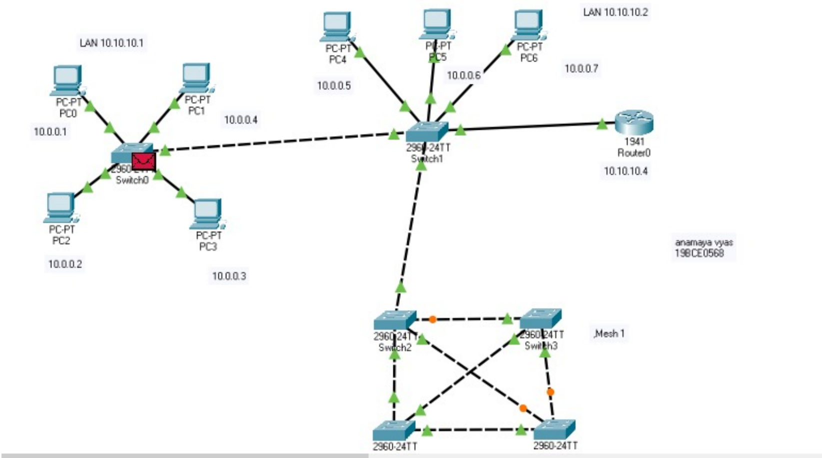
simulation photos


LAN 10.10.10.1
LAN 10.10.10.2
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
PC-PT PC4
PC-PT PC5
PC-PT PC6
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
10.0.0.5
10.0.0.6
10.0.0.7
2960-24TT Switch0
2960-24TT Switch1
1941 Router0
10.10.10.4
anamaya vyas
19BCE0568
2960-24TT Switch2
2960-24TT Switch3
Mesh 1
2960-24TT
2960-24TT

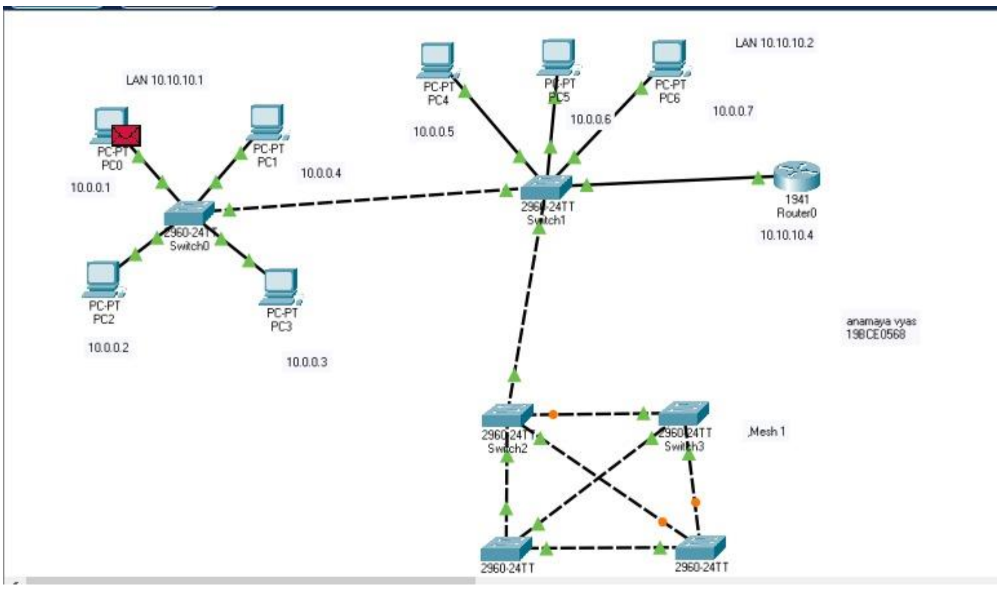
→→


LAN 10.10.10.1
LAN 10.10.10.2
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
PC-PT PC4
PC-PT PC5
PC-PT PC6
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
10.0.0.5
10.0.0.6
10.0.0.7
2960-24TT Switch0
2960-24TT Switch1
1941 Router0
10.10.10.4
anamaya vyas
19BCE0568
2960-24TT Switch2
2960-24TT Switch3
Mesh 1
2960-24TT
2960-24TT

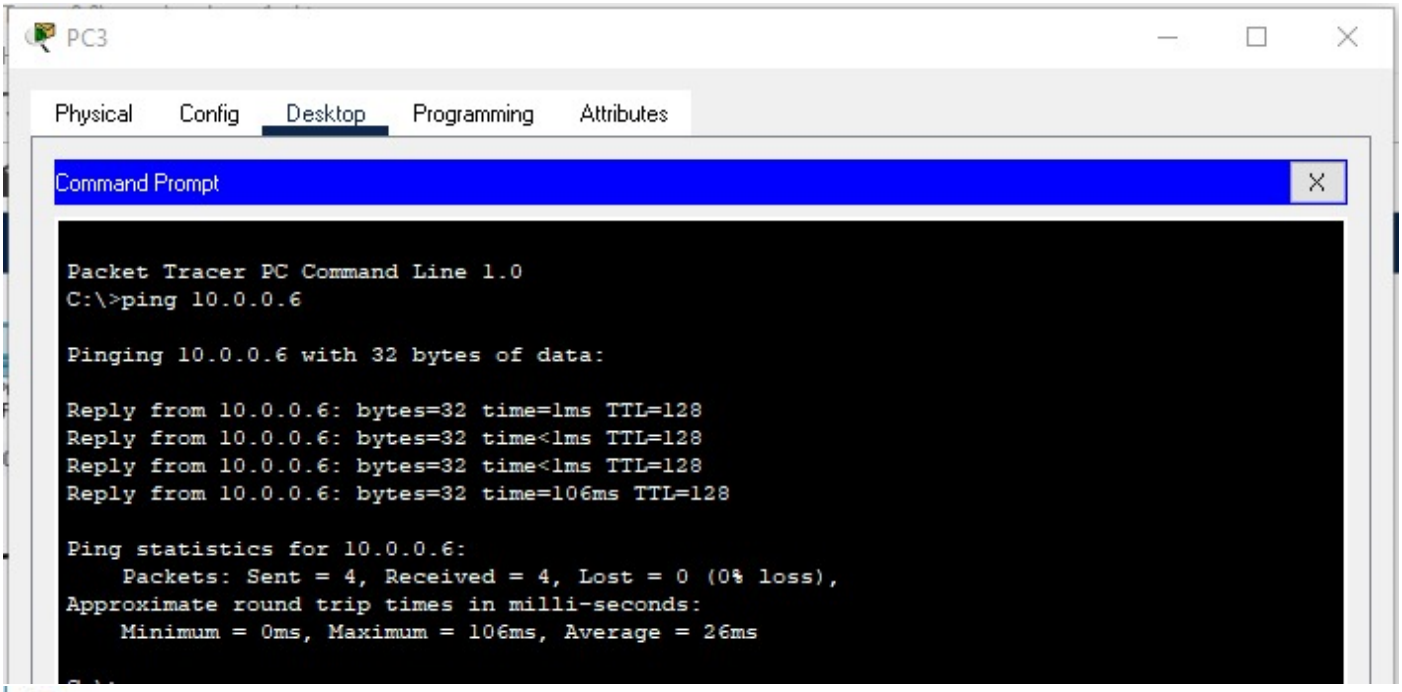
→→

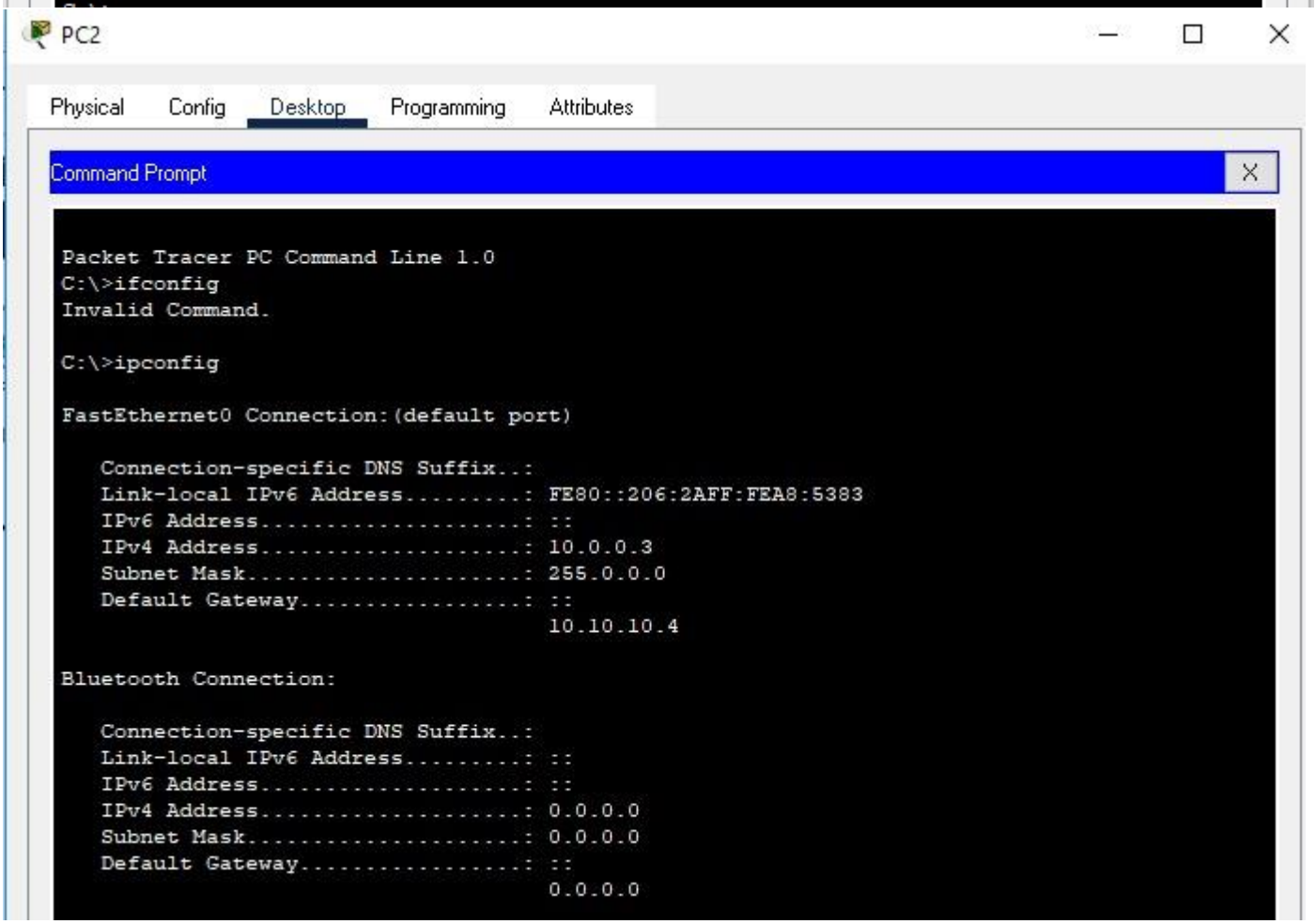

LAN 10.10.10.1
LAN 10.10.10.2
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
PC-PT PC4
PC-PT PC5
PC-PT PC6
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
10.0.0.5
10.0.0.6
10.0.0.7
2960-24TT Switch0
2960-24TT Switch1
1941 Router0
10.10.10.4
anamaya vyas
19BCE0568
2960-24TT Switch2
2960-24TT Switch3
Mesh 1
2960-24TT
2960-24TT

→→


LAN 10.10.10.1
LAN 10.10.10.2
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
PC-PT PC4
PC-PT PC5
PC-PT PC6
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
10.0.0.5
10.0.0.6
10.0.0.7
2960-24TT Switch0
2960-24TT Switch1
1941 Router0
10.10.10.4
anamaya vyas
19BCE0568
2960-24TT Switch2
2960-24TT Switch3
Mesh 1
2960-24TT
2960-24TT

Commands on cmd





2)

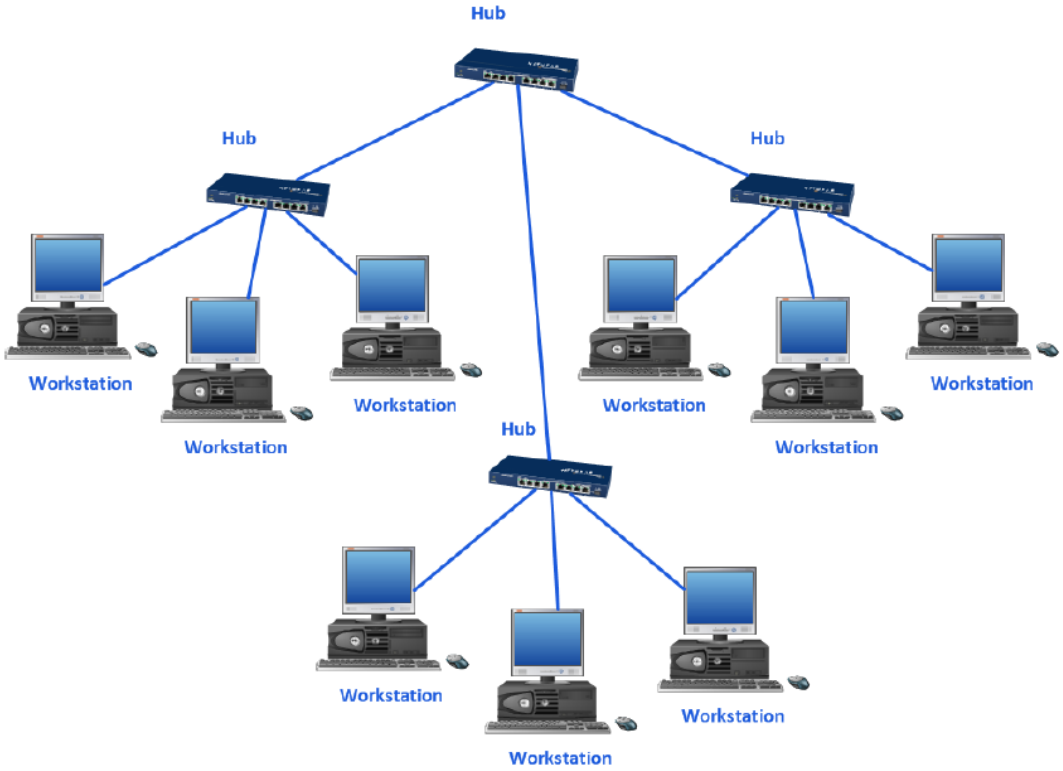
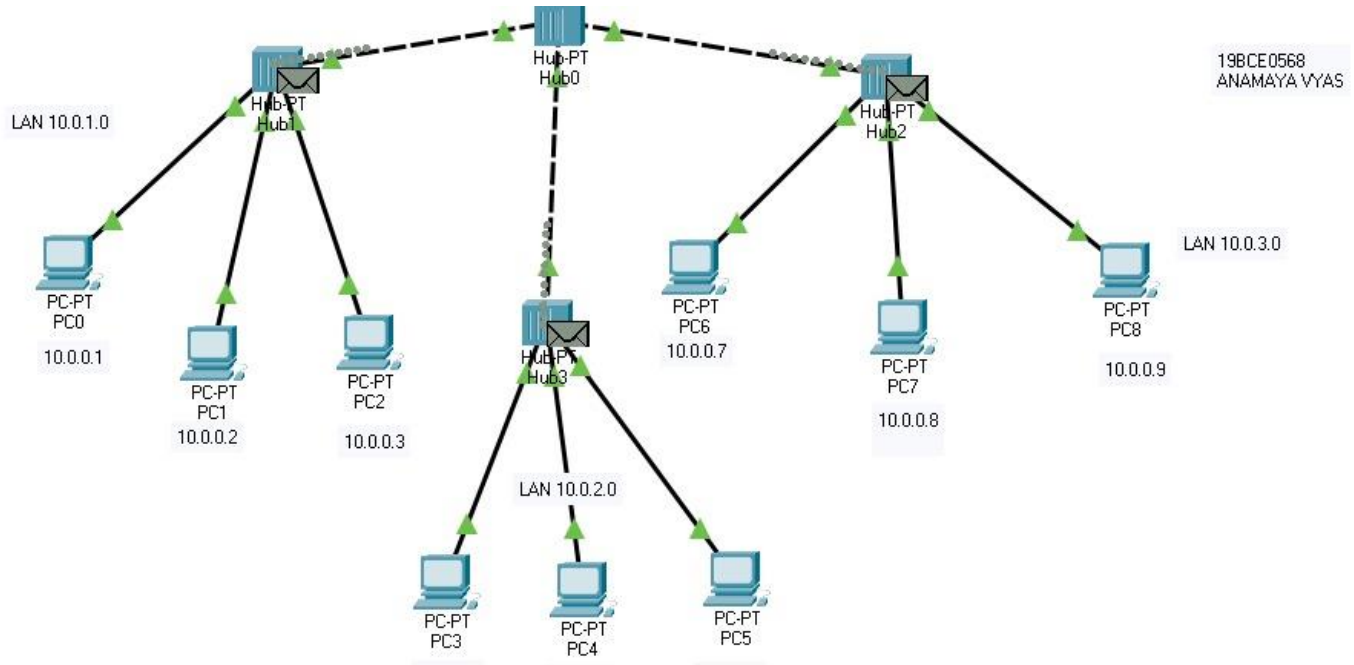


Figure 2: Network based on Hub

Ans

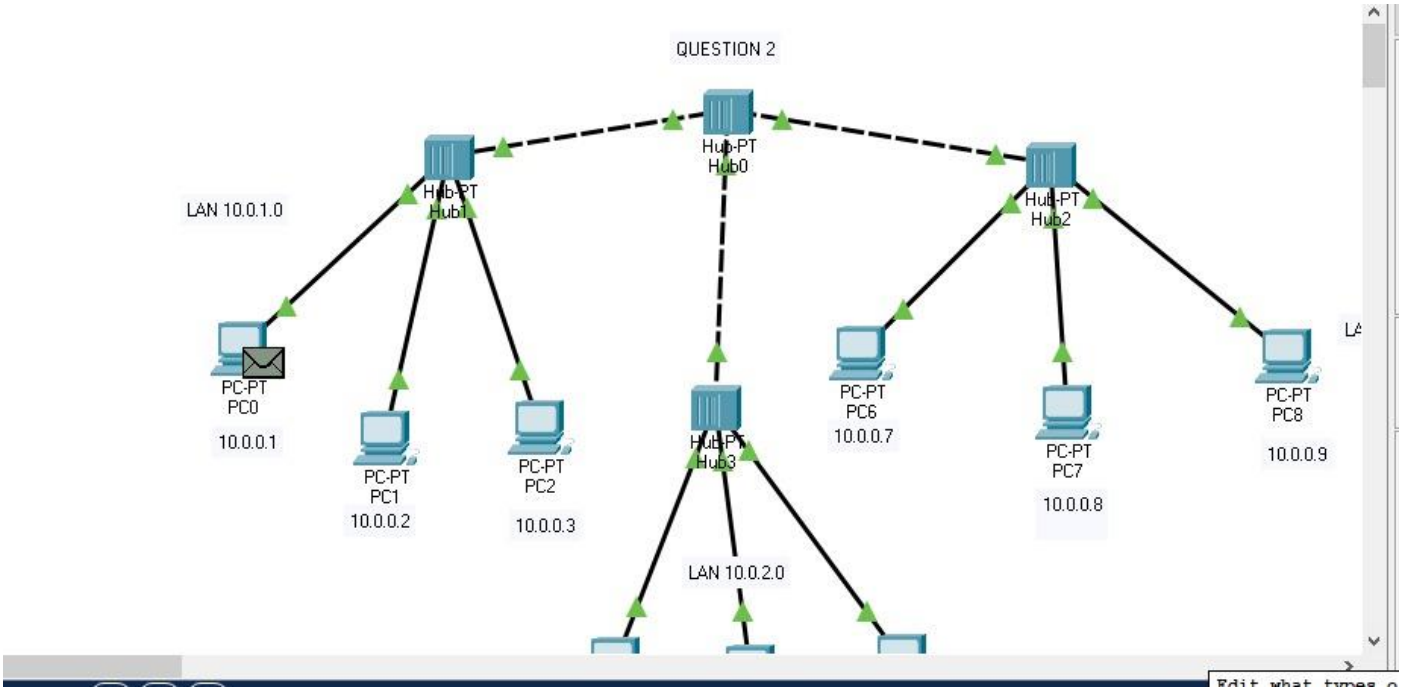


Method

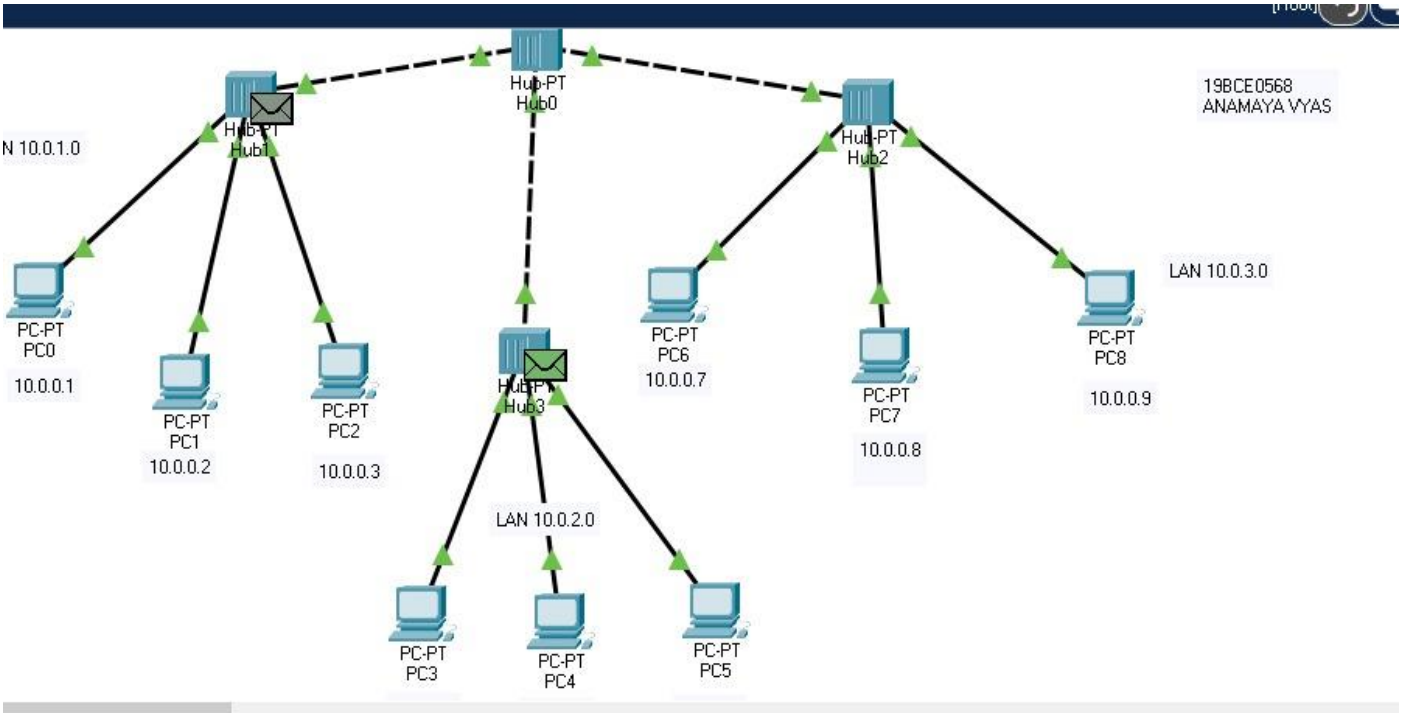
Q2

* use 9x General PCs
* use 4x Hub-PT
* using Copper Straight Through wire Connect 3 PC (PC0, PC1, PC2) to Hub 0.
* Repeat the same process for Hub 1 & Hub 2
* Now Connect Hub 0, Hub 1, Hub 2 with Hub 3 using Cross-over wire
* We made our network.
* Now assign the IP address to the PCs by going in the desktop > IP configuration > assign IP address
* Now click on PC0 and go to desktop then go to cmd then type ping 10.0.0.6.

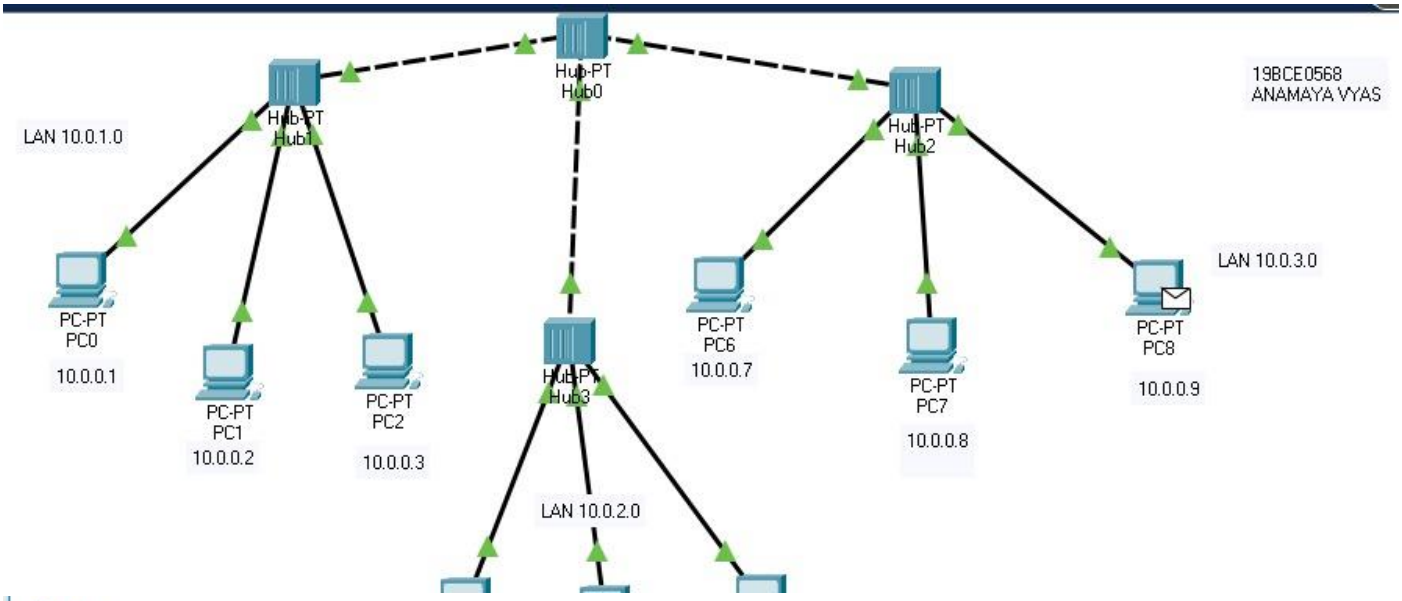
Simulation

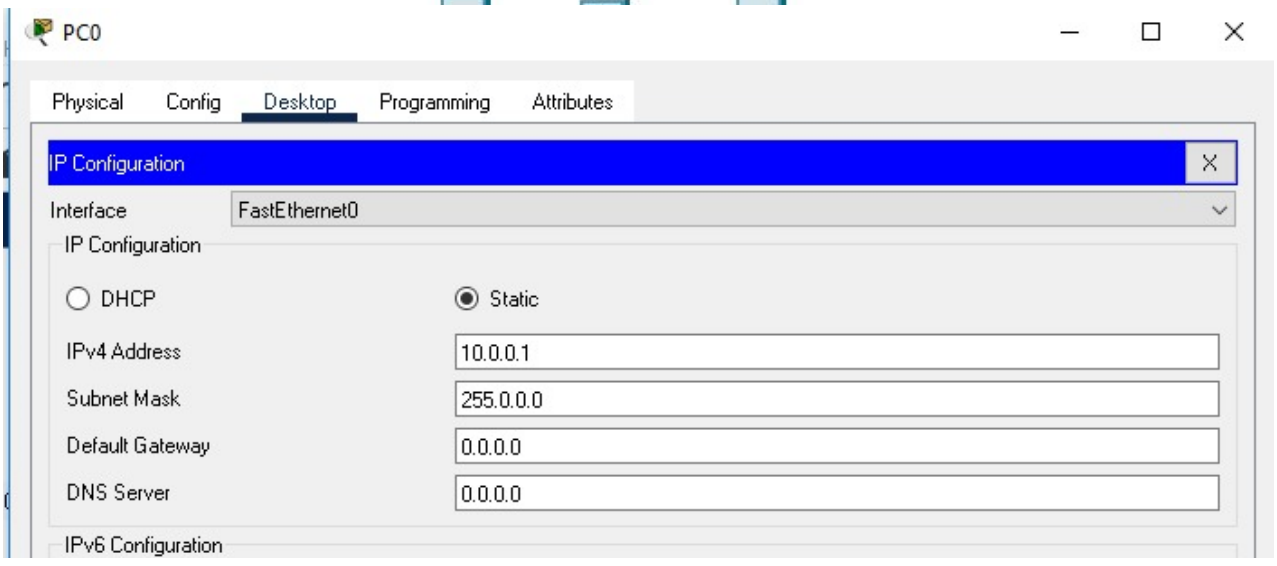


→→

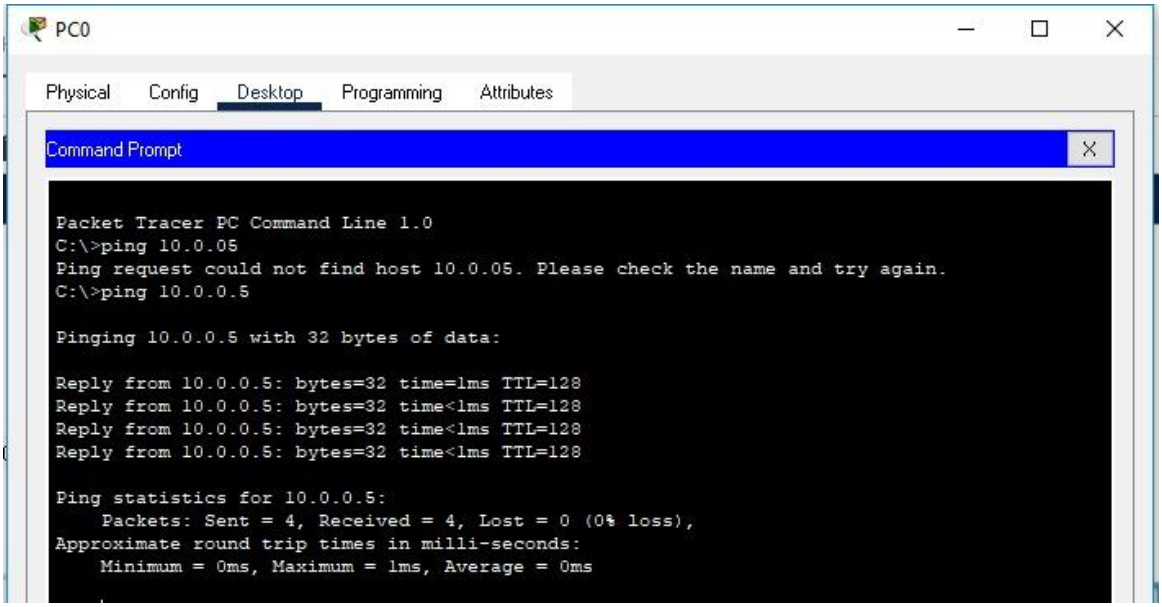

Hub-PT
Hub0
Hub-PT
Hub1
Hub-PT
Hub2
Hub-PT
Hub3
19BCE0568
ANAMAYA VYAS
N 10.0.1.0
LAN 10.0.3.0
LAN 10.0.2.0
PC-PT
PC0
10.0.0.1
PC-PT
PC1
10.0.0.2
PC-PT
PC2
10.0.0.3
PC-PT
PC3
PC-PT
PC4
PC-PT
PC5
PC-PT
PC6
10.0.0.7
PC-PT
PC7
10.0.0.8
PC-PT
PC8
10.0.0.9

→→


Hub-PT
Hub0
Hub-PT
Hub1
Hub-PT
Hub2
Hub-PT
Hub3
19BCE0568
ANAMAYA VYAS
LAN 10.0.1.0
LAN 10.0.3.0
LAN 10.0.2.0
PC-PT
PC0
10.0.0.1
PC-PT
PC1
10.0.0.2
PC-PT
PC2
10.0.0.3
PC-PT
PC6
10.0.0.7
PC-PT
PC7
10.0.0.8
PC-PT
PC8
10.0.0.9


PC0
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Interface
FastEthernet0
IP Configuration
DHCP
Static
IPv4 Address
10.0.0.1
Subnet Mask
255.0.0.0
Default Gateway
0.0.0.0
DNS Server
0.0.0.0
IPv6 Configuration

Commands



3)

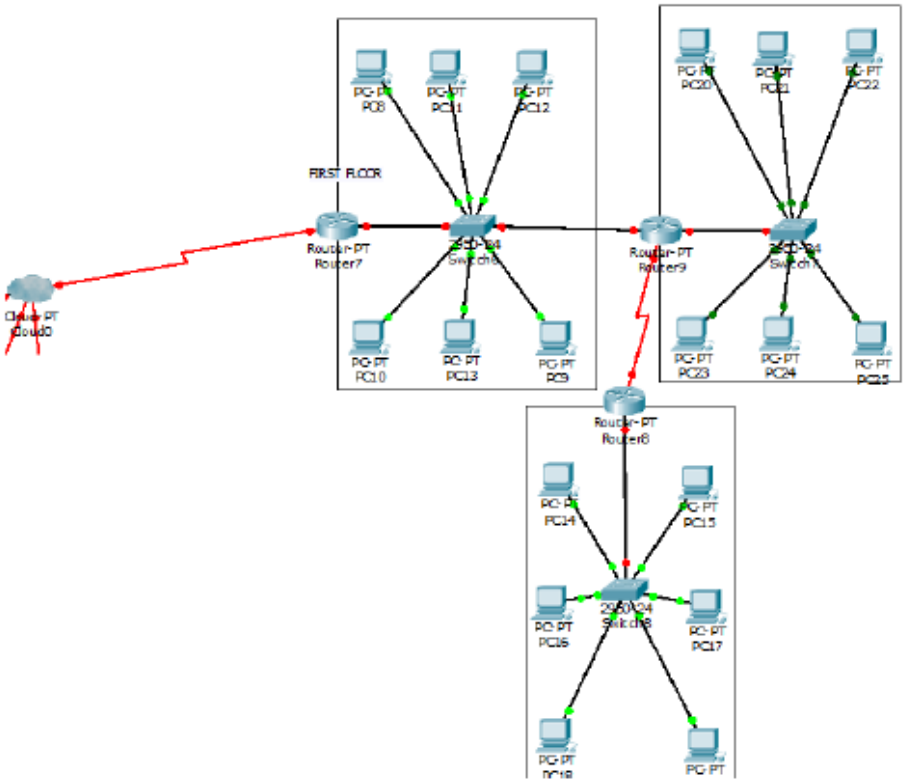
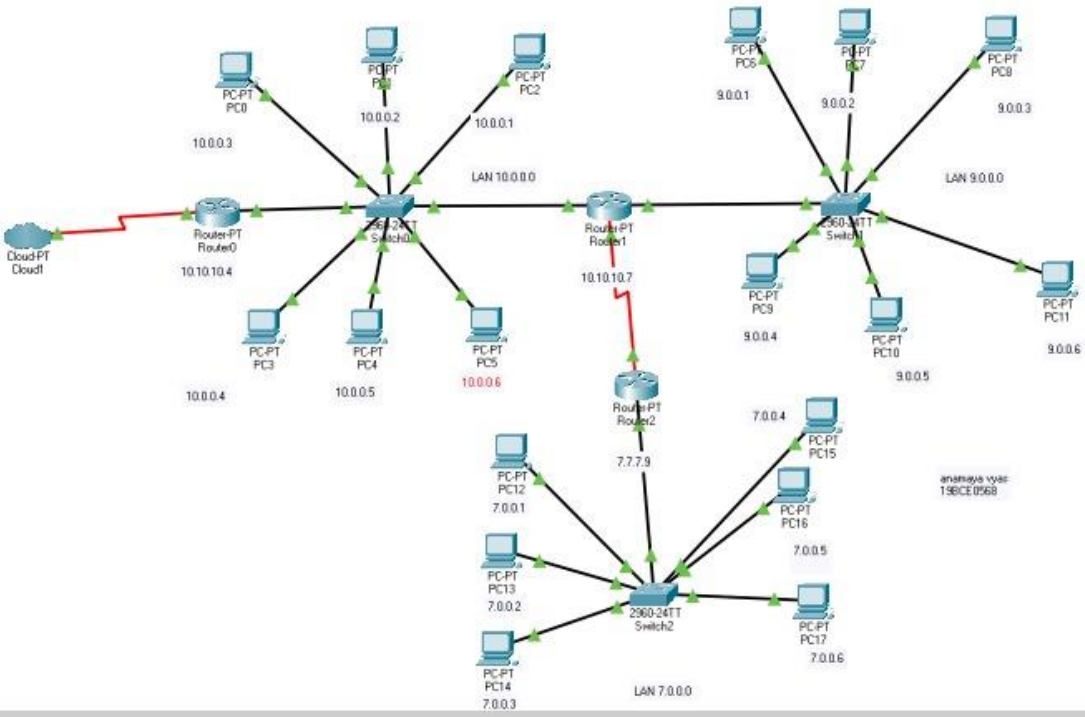


Figure 3: Network based on Switch

Ans

diagram



Explanation

Q3

Requirements

→ 18 × General PC
→ 3 PT-Routers
→ 1 Cloud PT
→ 3 2960-24 TT switches

→ We will make 3 lan connection and will connect them to a router which will act as the bridge b/w the lan networks to transfer info from one lan to the other

→ Take a switch and connect 6 PCs to it with the straight-through wire. Now connect the switch to the fast ethernet port of Router 0.

→ Connect 2 such lans to this router.

→ From one of the LAN connect the switch to another router. which will be connected to Cobbs PT.

→ ~~Connect~~ ~~one~~ Make another LAN and connect it to another routers.

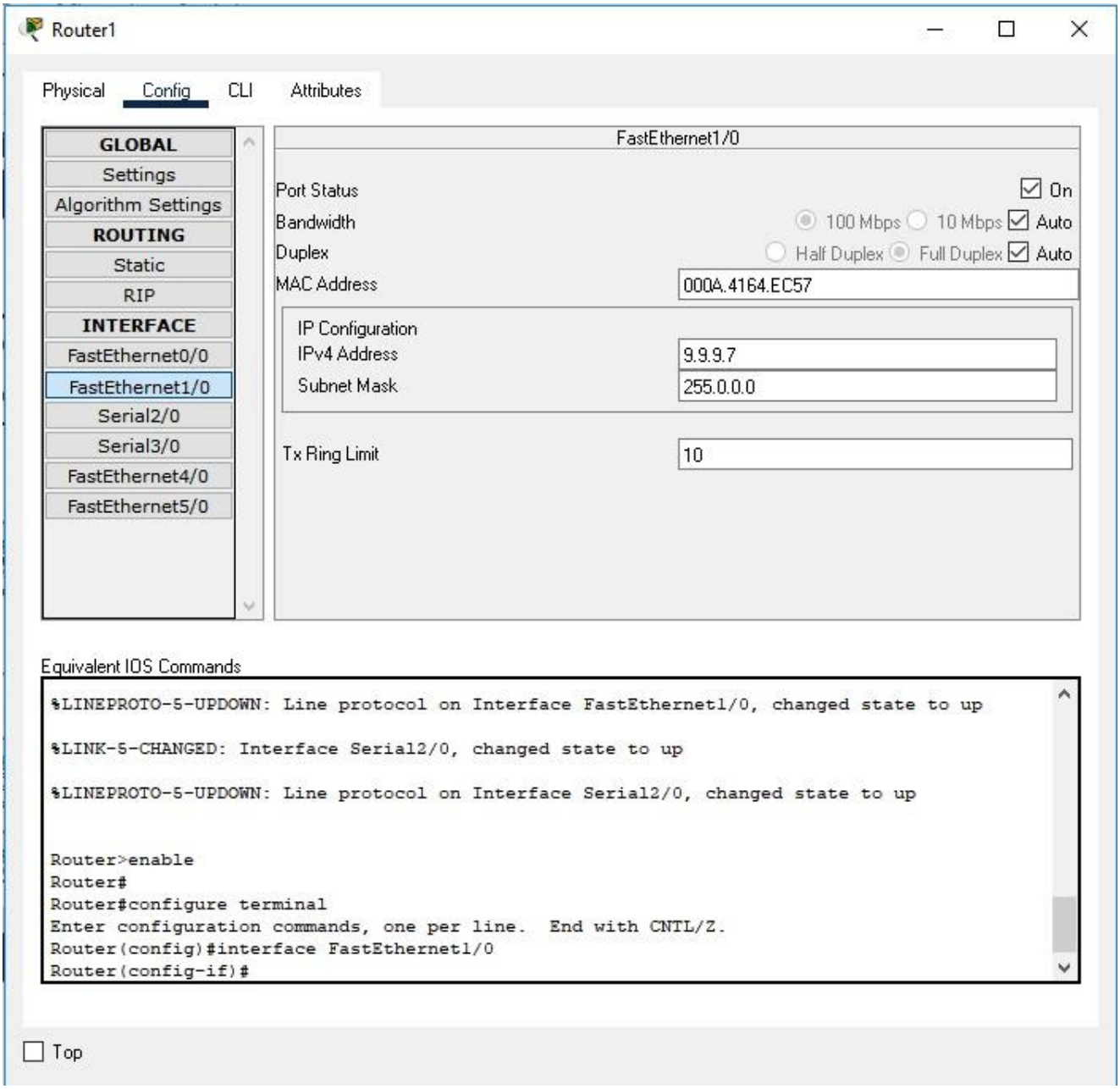
→ Connect these routers 1 & 0 using Serial DTE.

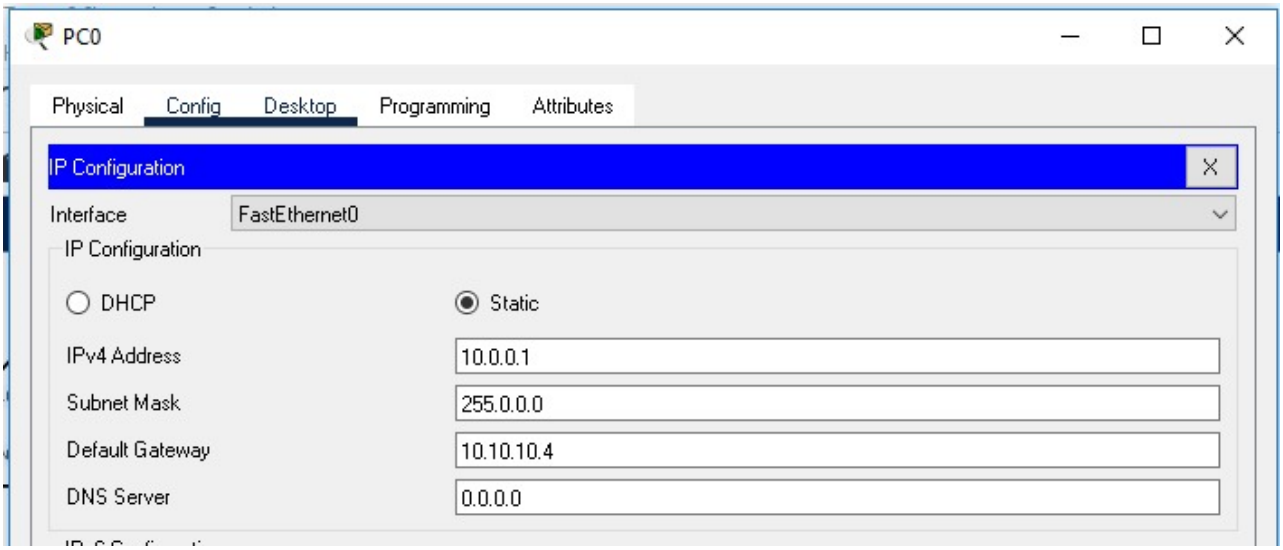
→ Now click on PC and go to ipConfig assign gateways & ip address

→ Now do the same for routers

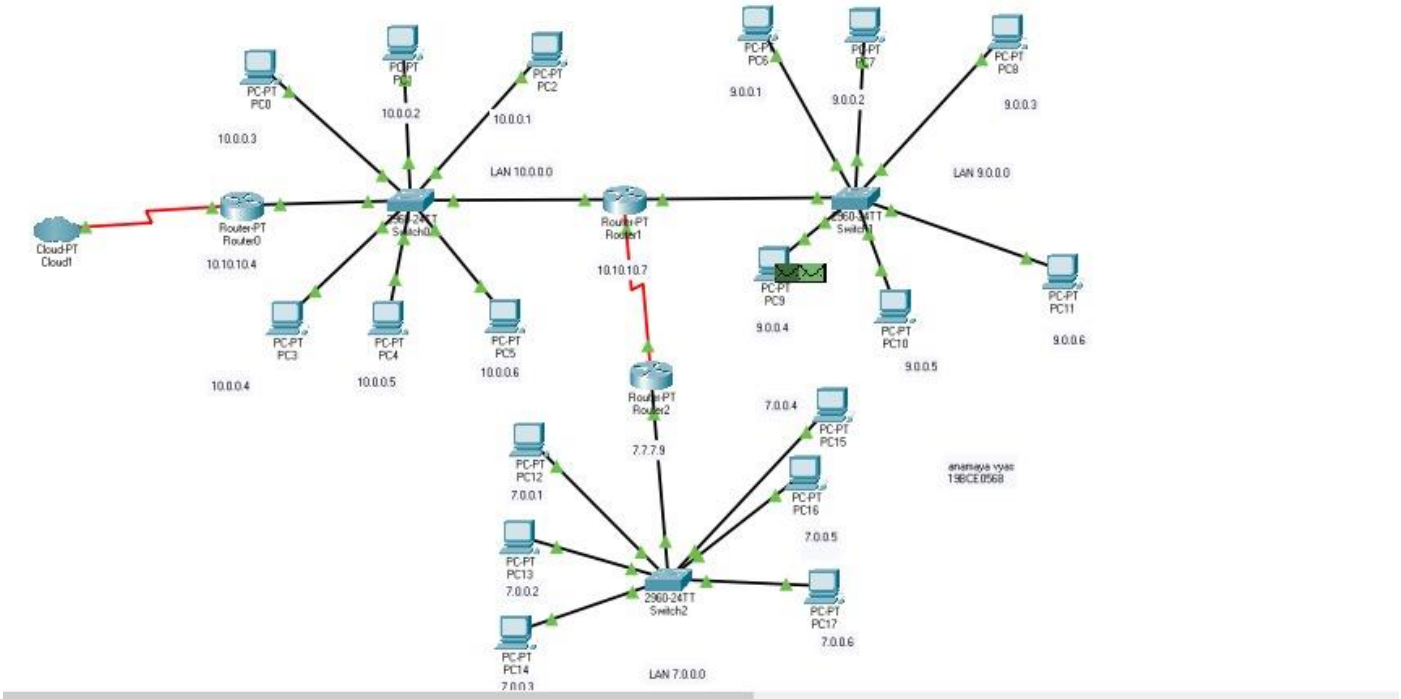
→ Click on any PC and ping it to other IP address and we have our network

Assigning IP address

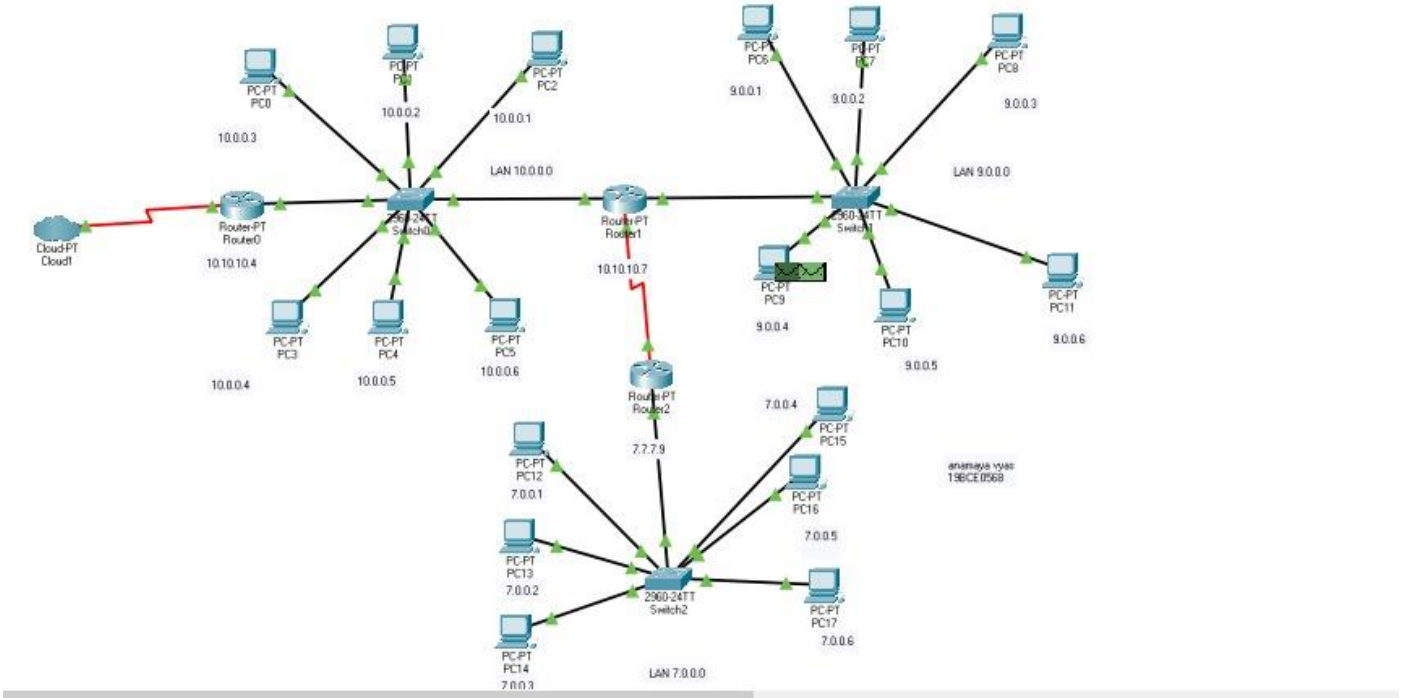




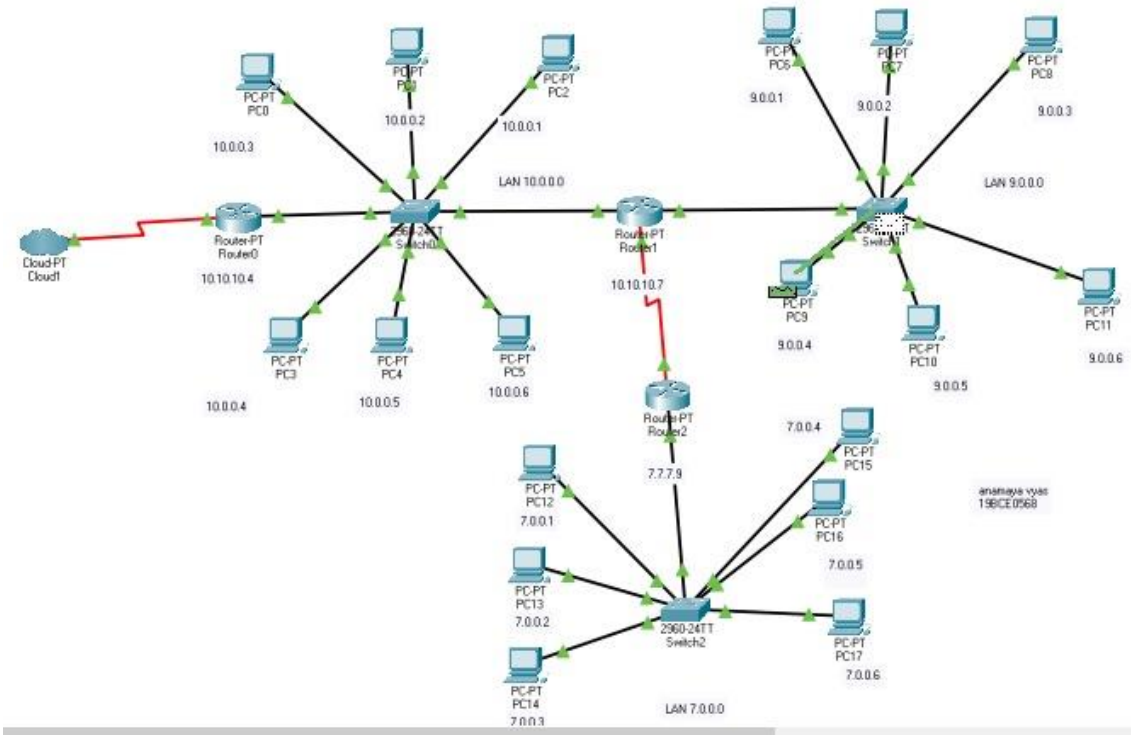
Simulation


PC-PT PC0
PC-PT PC1
PC-PT PC2
10.0.0.3
10.0.0.2
10.0.0.1
LAN 10.0.0.0
PC-PT PC6
PC-PT PC7
PC-PT PC8
9.0.0.1
9.0.0.2
9.0.0.3
LAN 9.0.0.0
Cloud-PT Cloud1
Router-PT Router0
10.10.10.4
2960-24TT Switch0
Router-PT Router1
10.10.10.7
2960-24TT Switch1
PC-PT PC9
PC-PT PC10
PC-PT PC11
9.0.0.4
9.0.0.5
9.0.0.6
PC-PT PC3
PC-PT PC4
PC-PT PC5
10.0.0.4
10.0.0.5
10.0.0.6
Router-PT Router2
7.7.7.9
7.0.0.4
PC-PT PC15
PC-PT PC12
PC-PT PC16
7.0.0.1
7.0.0.5
PC-PT PC13
7.0.0.2
2960-24TT Switch2
PC-PT PC17
7.0.0.6
PC-PT PC14
7.0.0.3
LAN 7.0.0.0
anamaya vyas
19BCE0568

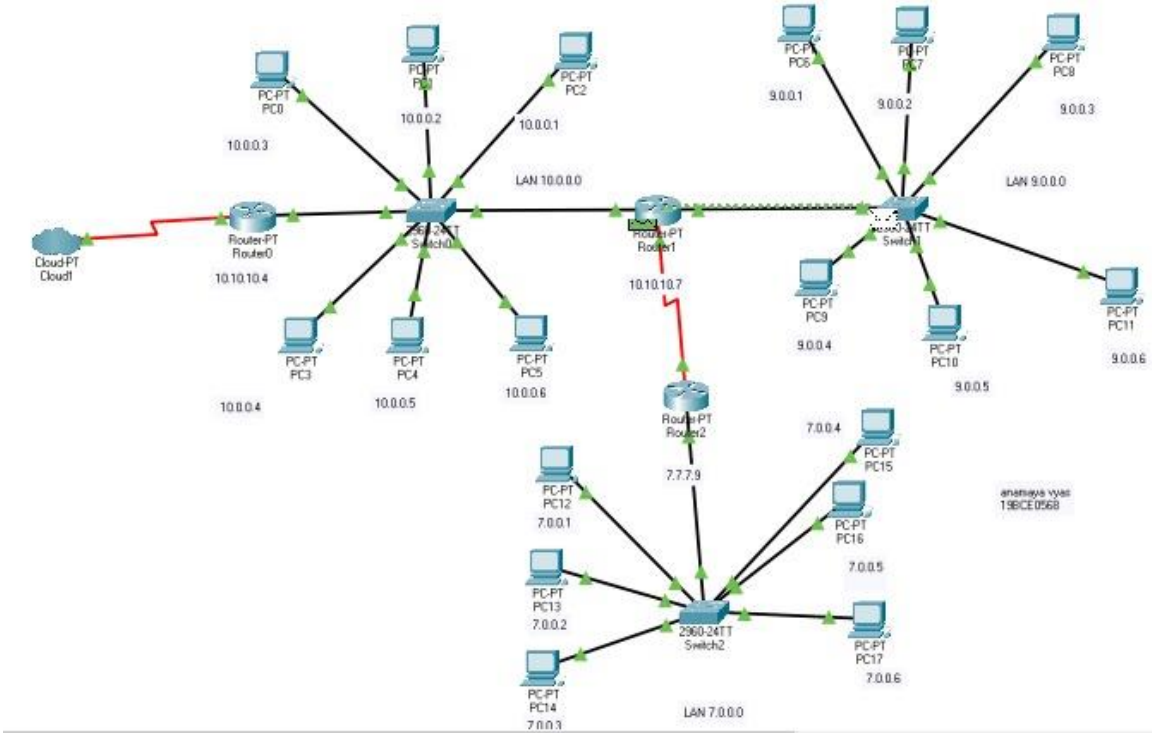
→→


PC-PT PC0
PC-PT PC1
PC-PT PC2
10.0.0.3
10.0.0.2
10.0.0.1
LAN 10.0.0.0
PC-PT PC6
PC-PT PC7
PC-PT PC8
9.0.0.1
9.0.0.2
9.0.0.3
LAN 9.0.0.0
Cloud-PT Cloud1
Router-PT Router0
10.10.10.4
2960-24TT Switch0
Router-PT Router1
10.10.10.7
2960-24TT Switch1
PC-PT PC9
PC-PT PC10
PC-PT PC11
9.0.0.4
9.0.0.5
9.0.0.6
PC-PT PC3
PC-PT PC4
PC-PT PC5
10.0.0.4
10.0.0.5
10.0.0.6
Router-PT Router2
7.7.7.9
7.0.0.4
PC-PT PC15
PC-PT PC12
PC-PT PC16
7.0.0.1
7.0.0.5
PC-PT PC13
7.0.0.2
2960-24TT Switch2
PC-PT PC17
7.0.0.6
PC-PT PC14
7.0.0.3
LAN 7.0.0.0
anamaya vyas
19BCE0568

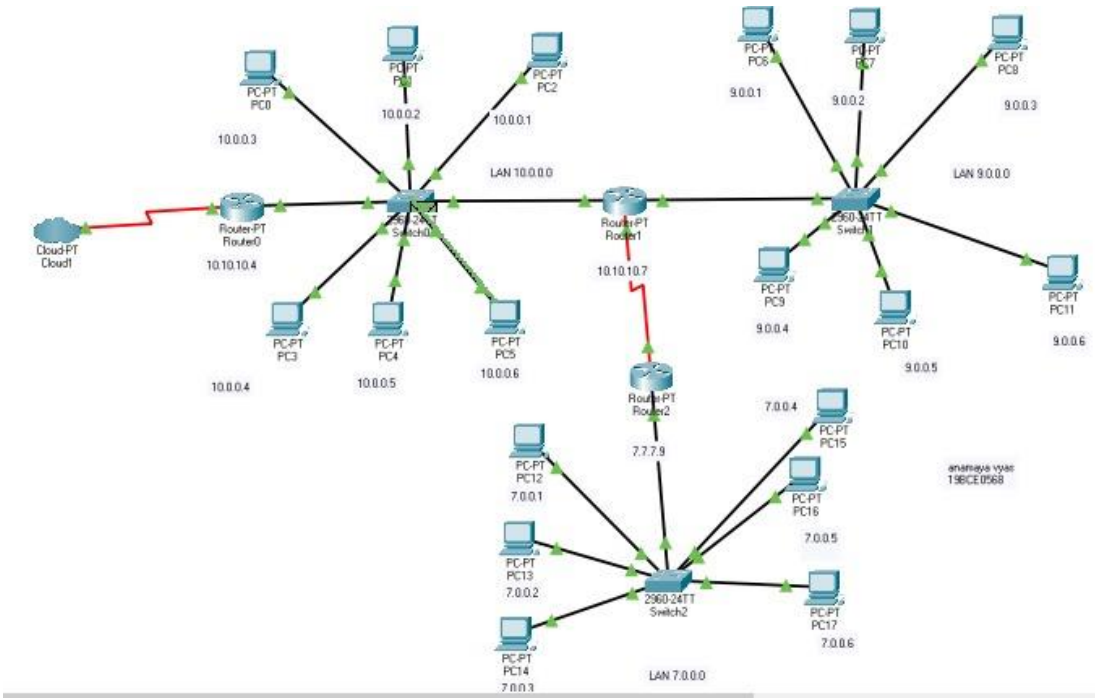
→→


PC-PT PC0
PC-PT PC1
PC-PT PC2
10.0.0.3
10.0.0.2
10.0.0.1
LAN 10.0.0.0
Cloud-PT Cloud1
Router-PT Router0
10.10.10.4
2960-24TT Switch0
Router-PT Router1
10.10.10.7
PC-PT PC3
PC-PT PC4
PC-PT PC5
10.0.0.4
10.0.0.5
10.0.0.6
PC-PT PC6
PC-PT PC7
PC-PT PC8
9.0.0.1
9.0.0.2
9.0.0.3
LAN 9.0.0.0
PC-PT PC9
PC-PT PC10
PC-PT PC11
9.0.0.4
9.0.0.5
9.0.0.6
Router-PT Router2
7.7.7.9
PC-PT PC12
PC-PT PC13
PC-PT PC14
7.0.0.1
7.0.0.2
7.0.0.3
PC-PT PC15
PC-PT PC16
PC-PT PC17
7.0.0.4
7.0.0.5
7.0.0.6
2960-24TT Switch2
LAN 7.0.0.0
anamaya vyas
19BCE0568

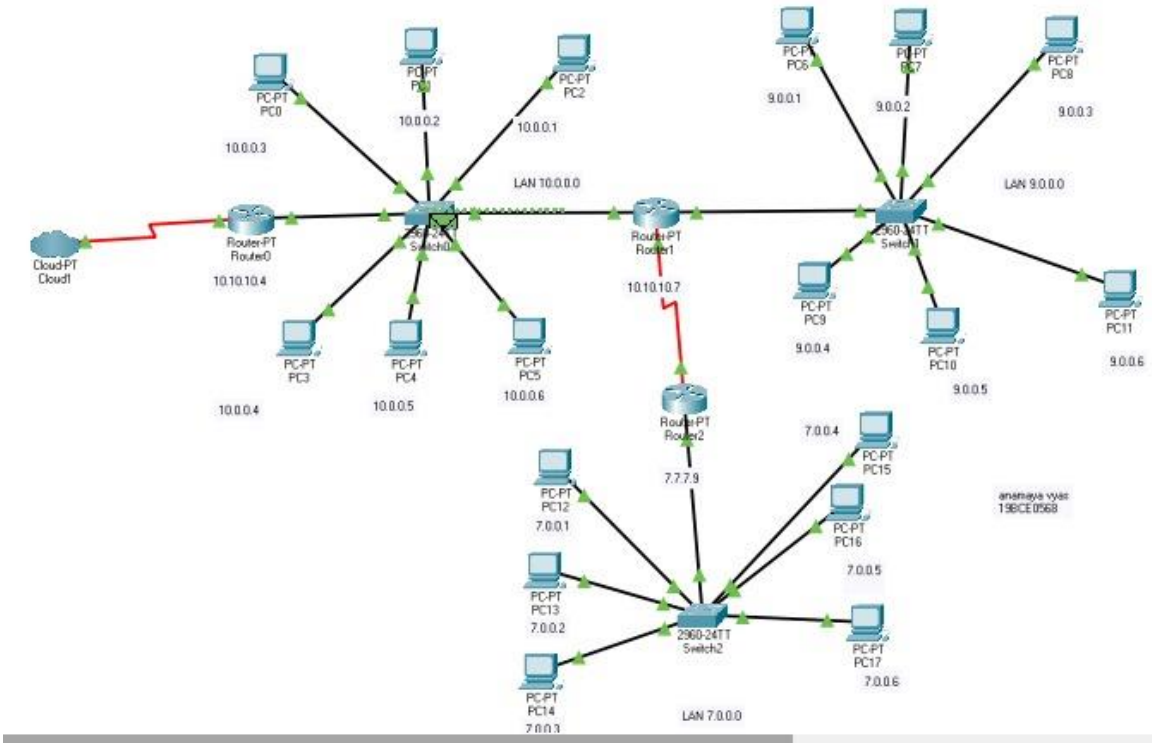
→→


PC-PT PC0
PC-PT PC1
PC-PT PC2
10.0.0.3
10.0.0.2
10.0.0.1
LAN 10.0.0.0
Cloud-PT Cloud1
Router-PT Router0
10.10.10.4
2960-24TT Switch0
Router-PT Router1
10.10.10.7
PC-PT PC3
PC-PT PC4
PC-PT PC5
10.0.0.4
10.0.0.5
10.0.0.6
PC-PT PC6
PC-PT PC7
PC-PT PC8
9.0.0.1
9.0.0.2
9.0.0.3
LAN 9.0.0.0
PC-PT PC9
PC-PT PC10
PC-PT PC11
9.0.0.4
9.0.0.5
9.0.0.6
Router-PT Router2
7.7.7.9
PC-PT PC12
PC-PT PC13
PC-PT PC14
7.0.0.1
7.0.0.2
7.0.0.3
PC-PT PC15
PC-PT PC16
PC-PT PC17
7.0.0.4
7.0.0.5
7.0.0.6
2960-24TT Switch2
LAN 7.0.0.0
anamaya vyas
19BCE0568

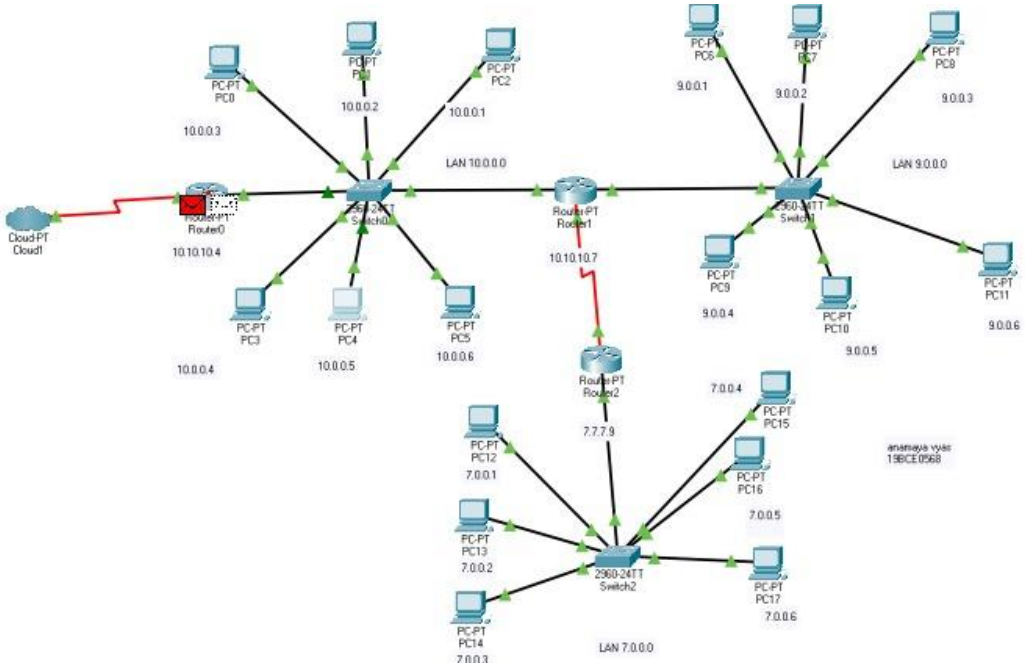
→→


PC-PT PC0
PC-PT PC1
PC-PT PC2
10.0.0.3
10.0.0.2
10.0.0.1
LAN 10.0.0.0
Cloud-PT Cloud1
Router-PT Router0
10.10.10.4
2960-24TT Switch0
Router-PT Router1
10.10.10.7
PC-PT PC3
PC-PT PC4
PC-PT PC5
10.0.0.4
10.0.0.5
10.0.0.6
PC-PT PC6
PC-PT PC7
PC-PT PC8
9.0.0.1
9.0.0.2
9.0.0.3
LAN 9.0.0.0
2960-24TT Switch1
PC-PT PC9
PC-PT PC10
PC-PT PC11
9.0.0.4
9.0.0.5
9.0.0.6
Router-PT Router2
7.7.7.9
PC-PT PC12
PC-PT PC13
PC-PT PC14
7.0.0.1
7.0.0.2
7.0.0.3
2960-24TT Switch2
LAN 7.0.0.0
PC-PT PC15
PC-PT PC16
PC-PT PC17
7.0.0.4
7.0.0.5
7.0.0.6
anamaya vyas
19BCE0568

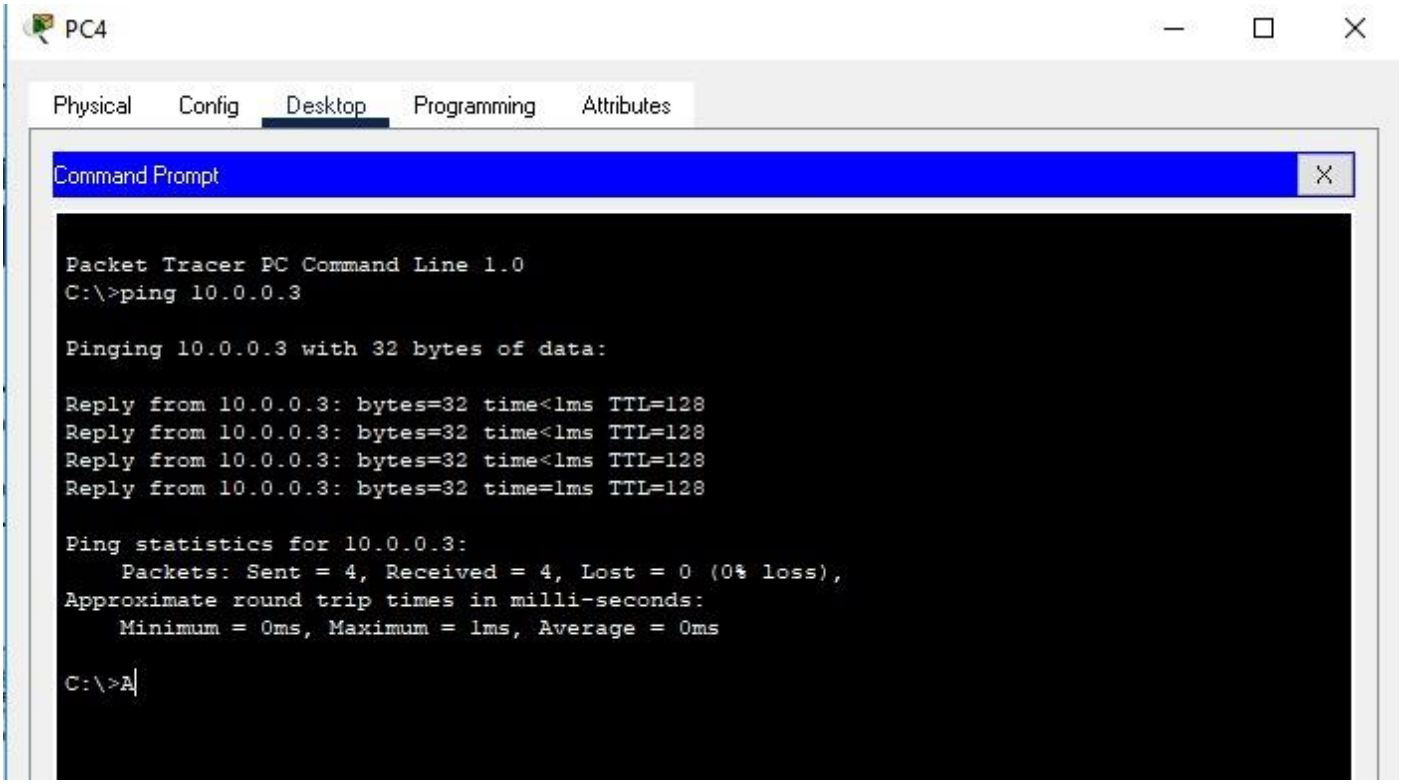
→→


PC-PT PC0
PC-PT PC1
PC-PT PC2
10.0.0.3
10.0.0.2
10.0.0.1
LAN 10.0.0.0
Cloud-PT Cloud1
Router-PT Router0
10.10.10.4
2960-24TT Switch0
Router-PT Router1
10.10.10.7
PC-PT PC3
PC-PT PC4
PC-PT PC5
10.0.0.4
10.0.0.5
10.0.0.6
PC-PT PC6
PC-PT PC7
PC-PT PC8
9.0.0.1
9.0.0.2
9.0.0.3
LAN 9.0.0.0
2960-24TT Switch1
PC-PT PC9
PC-PT PC10
PC-PT PC11
9.0.0.4
9.0.0.5
9.0.0.6
Router-PT Router2
7.7.7.9
PC-PT PC12
PC-PT PC13
PC-PT PC14
7.0.0.1
7.0.0.2
7.0.0.3
2960-24TT Switch2
LAN 7.0.0.0
PC-PT PC15
PC-PT PC16
PC-PT PC17
7.0.0.4
7.0.0.5
7.0.0.6
anamaya vyas
19BCE0568

→→



CMD commands



PC12

Physical Config Desktop Programming Attributes

Command Prompt

```
Packet Tracer PC Command Line 1.0
C:\>ipconfig

FastEthernet0 Connection:(default port)

   Connection-specific DNS Suffix..:
   Link-local IPv6 Address.....: FE80::202:16FF:FE0C:33C6
   IPv6 Address.....: ::
   IPv4 Address.....: 7.0.0.1
   Subnet Mask.....: 255.0.0.0
   Default Gateway.....: ::
                                 7.7.7.9

Bluetooth Connection:

   Connection-specific DNS Suffix..:
   Link-local IPv6 Address.....: ::
   IPv6 Address.....: ::
   IPv4 Address.....: 0.0.0.0
   Subnet Mask.....: 0.0.0.0
   Default Gateway.....: ::
                                 0.0.0.0

C:\>
```

4)

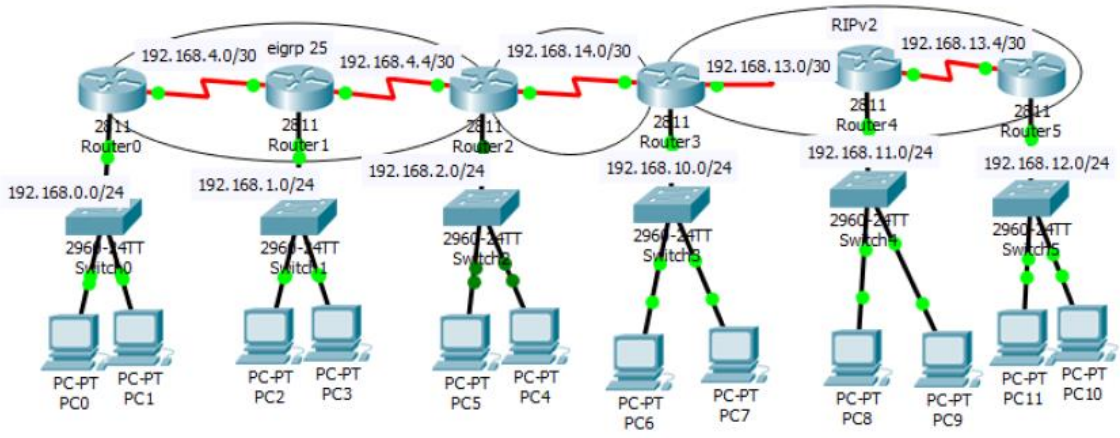
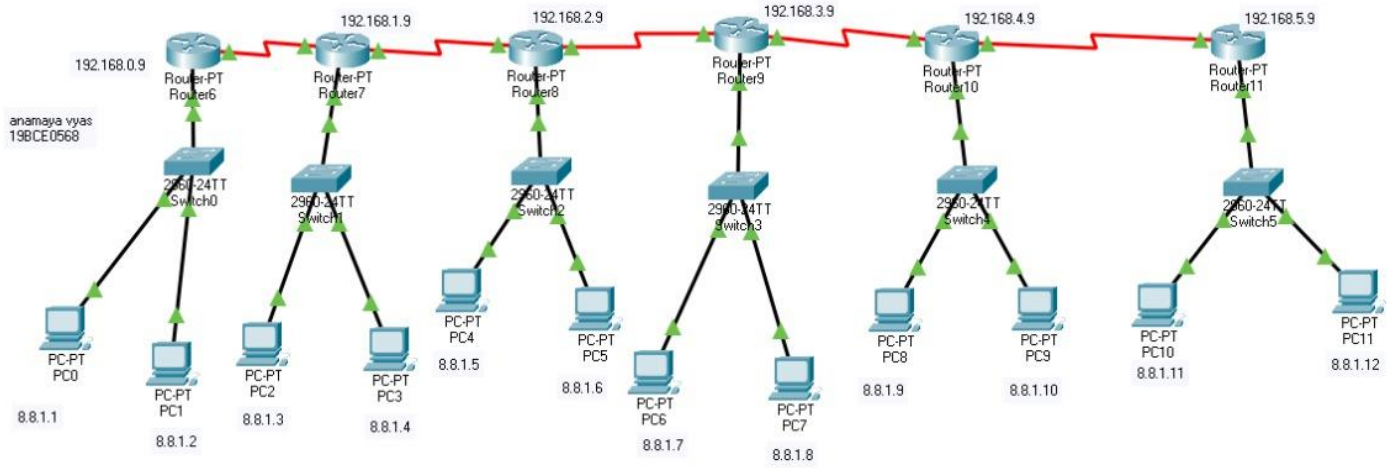


Figure 4: Network based on Switch and Router

Ans

Network circuit



Explanation

Q4

Requirements

6x Router-PT
6x 2960-24TT Switches
12x General PC

→ We will make 6 LAN connections and will connect each LAN with a router which will serially form a network.

→ Here we will use 2 PCs to make one LAN. Connect PC1 & PC2 to the switch with Straight-Through wire

→ Make 6 such LANS

→ Now connect each LAN with a router & then connect each router with serial DTE

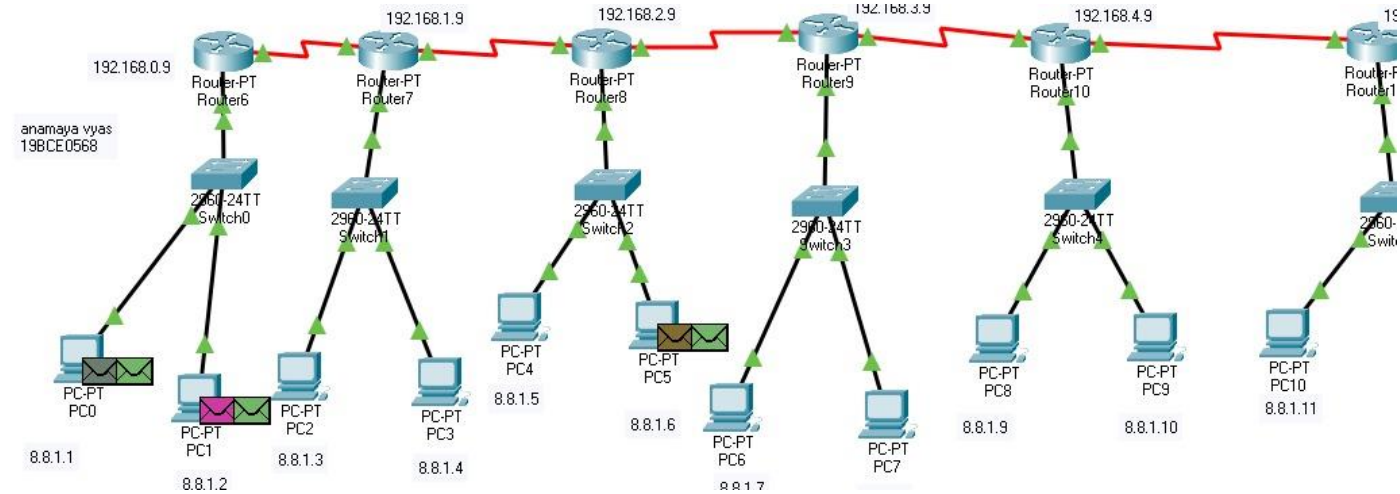
– Here we formed a serial network.

– Now click on PC and go to ipconfig and give gateways & ip address.

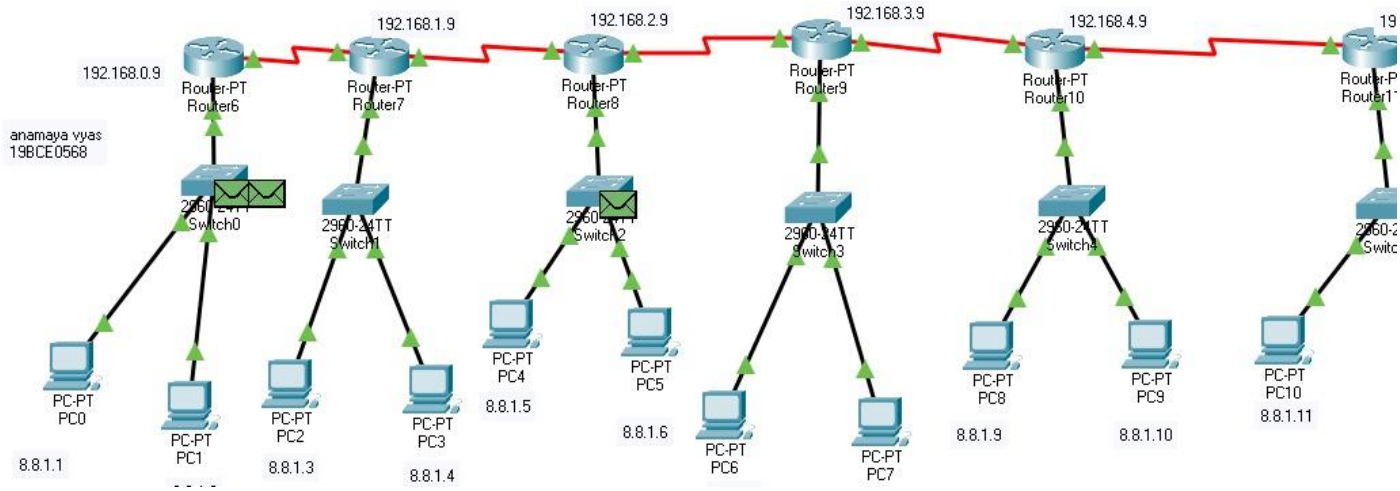
→ Do the same for Routers

→ Now click on PC and go to cmd and use ping command to start the network.

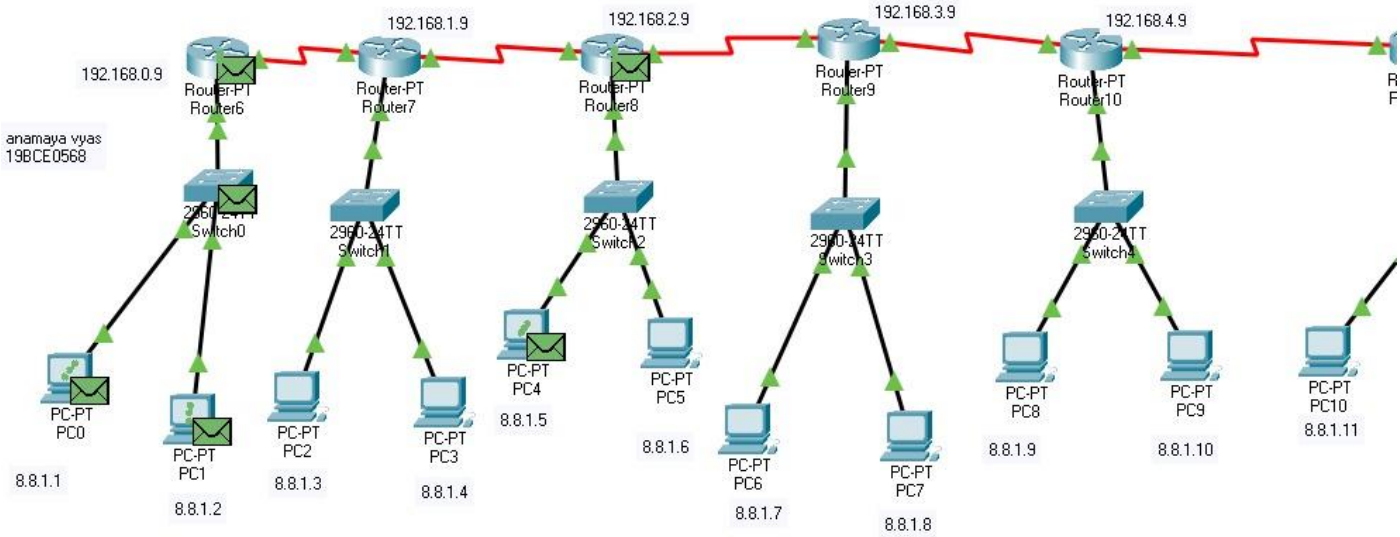
Stimulation



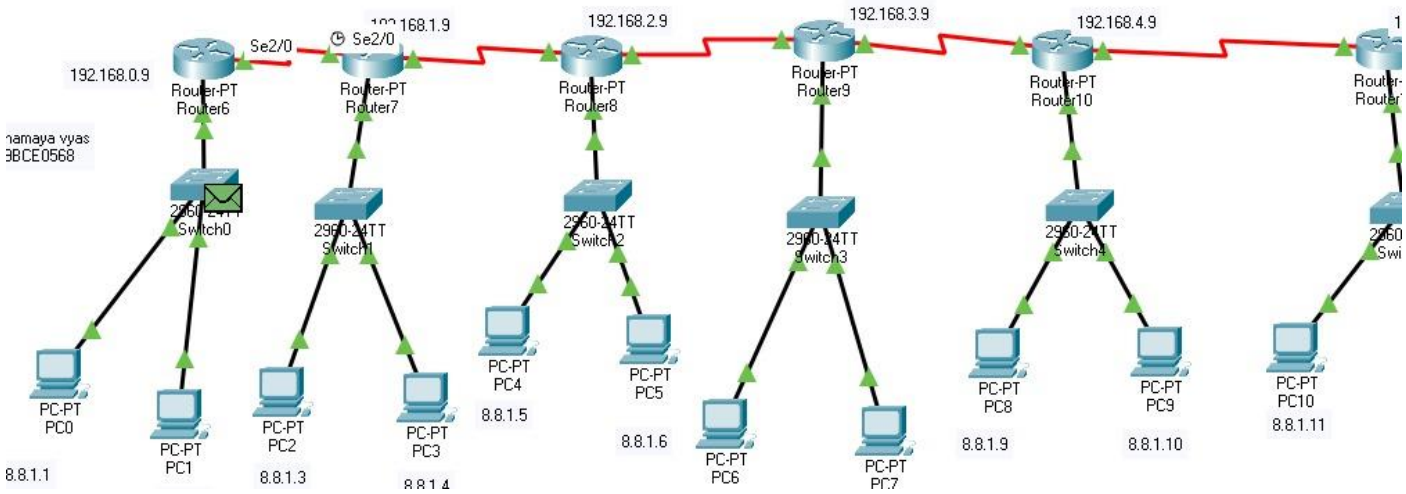
→→



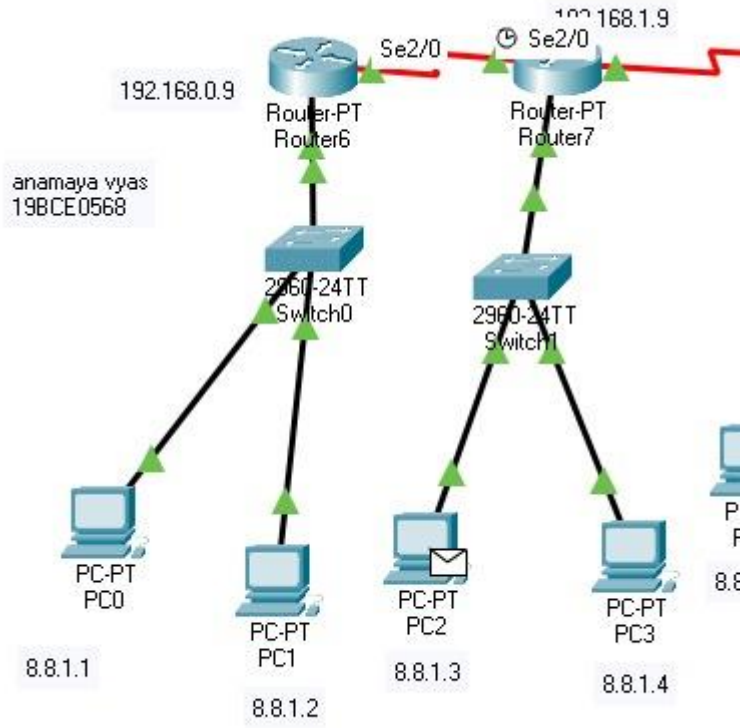
→→



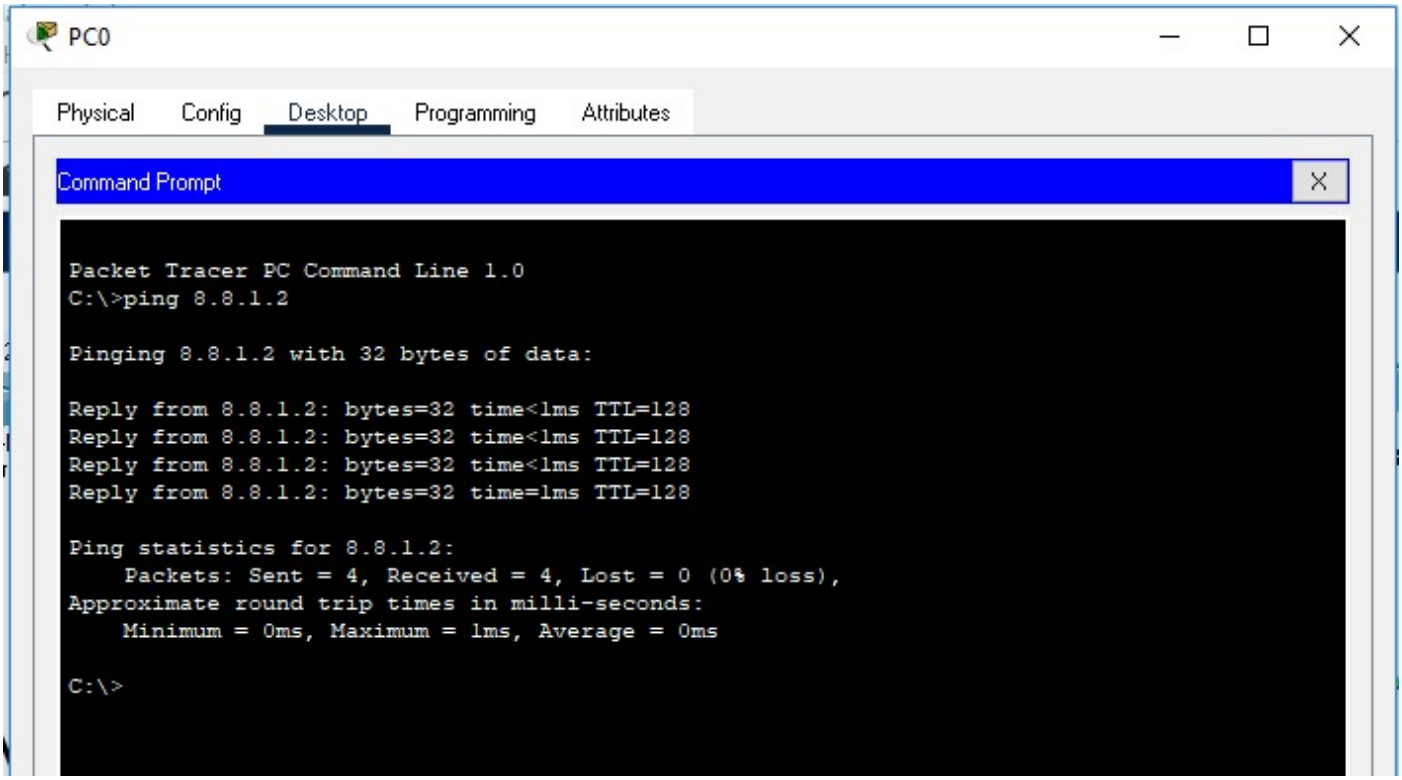
→→

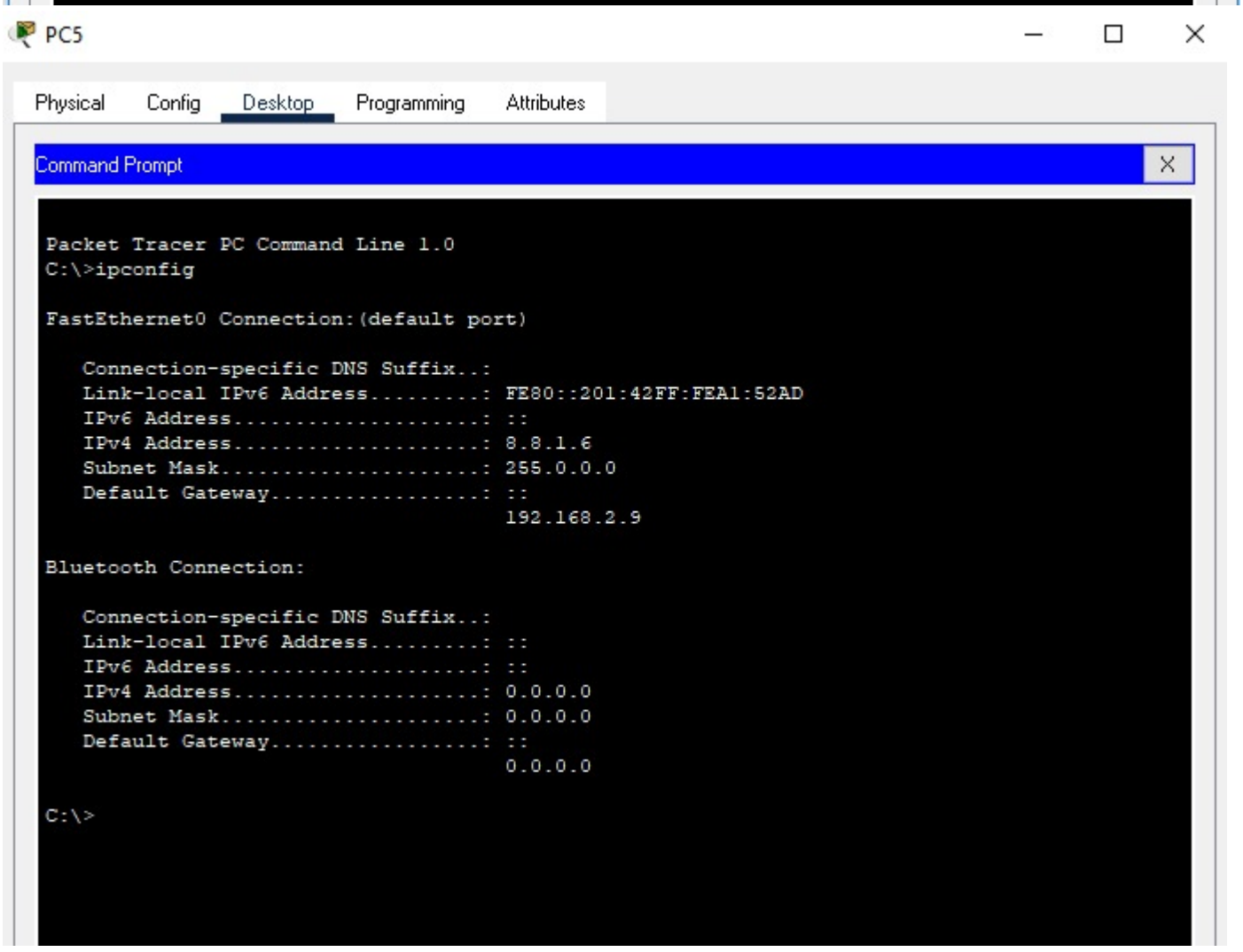


→→

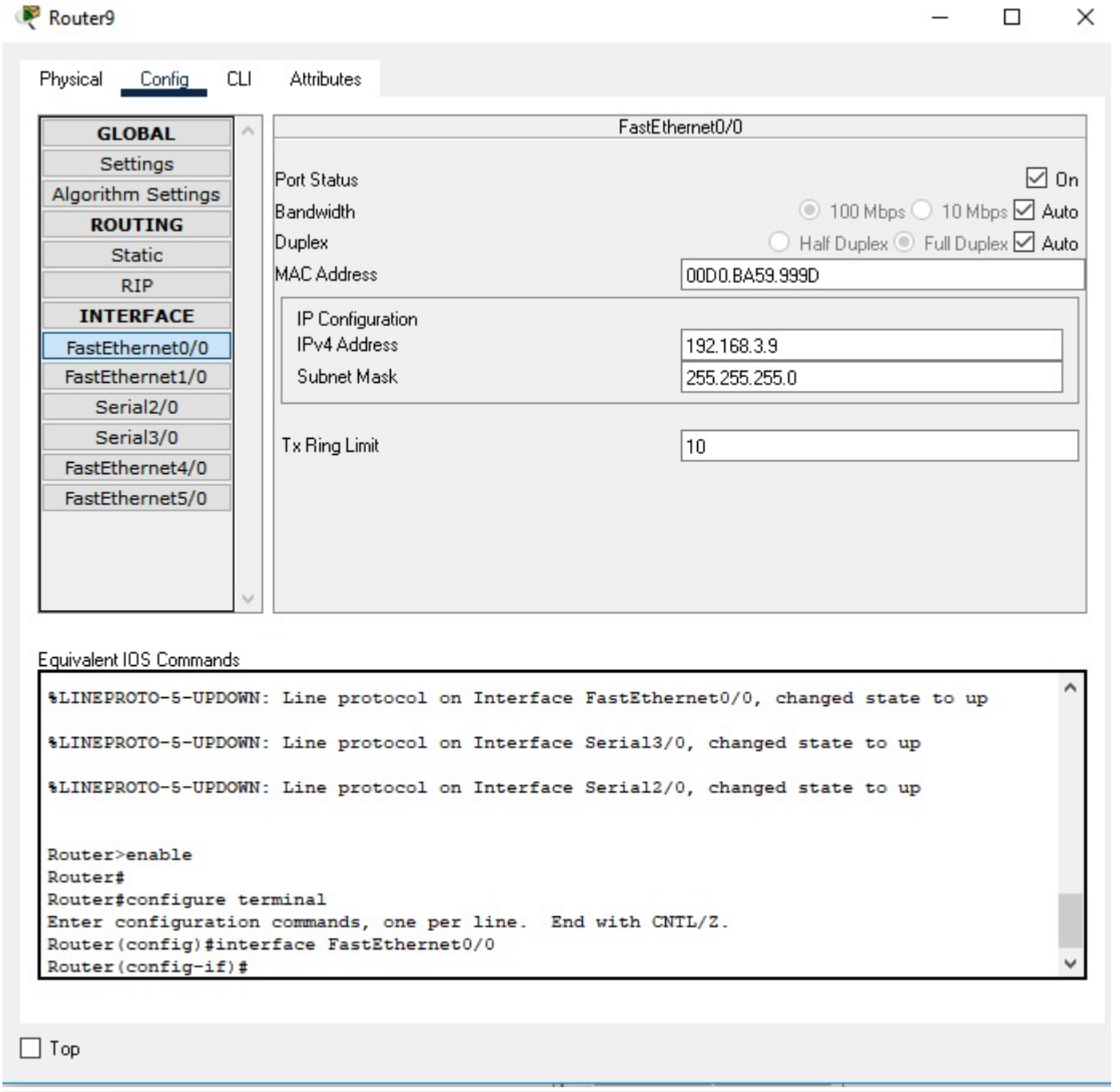


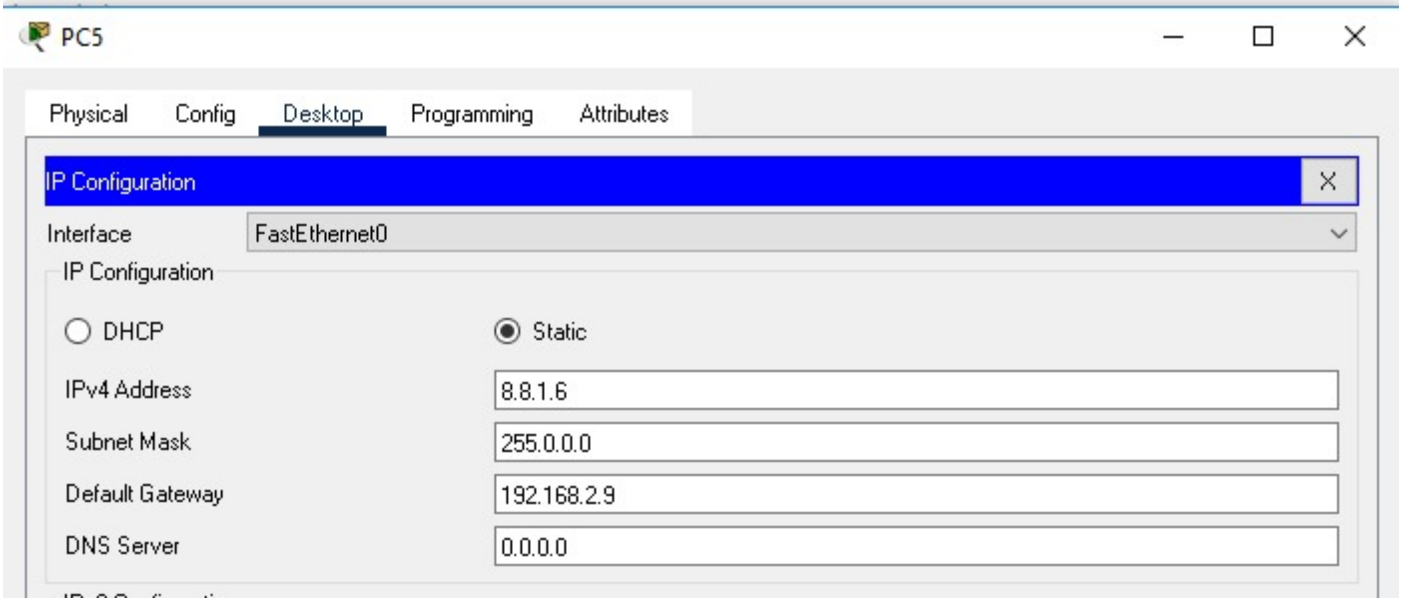
Command in cmd





Providing ip address


Router9
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
FastEthernet4/0
FastEthernet5/0
FastEthernet0/0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
00D0.BA59.999D
IP Configuration
IPv4 Address
192.168.3.9
Subnet Mask
255.255.255.0
Tx Ring Limit
10
Equivalent IOS Commands
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#
Top


PC5
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Interface
FastEthernet0
IP Configuration
DHCP
Static
IPv4 Address
8.8.1.6
Subnet Mask
255.0.0.0
Default Gateway
192.168.2.9
DNS Server
0.0.0.0