CSE 1004

Network and Communication

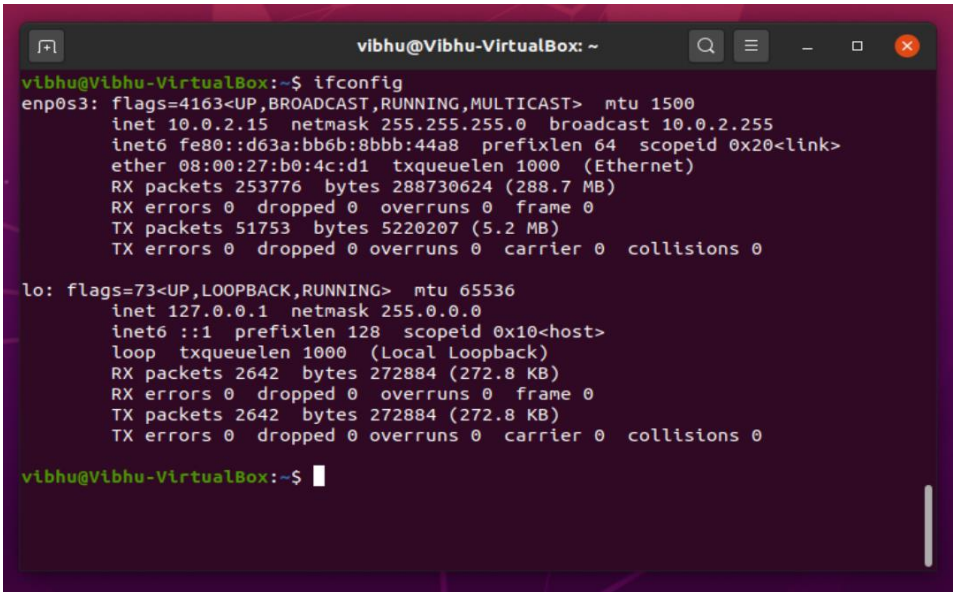
LAB ASSESSMENT - 1

NAME: Vibhu Kumar Singh
REG. NO: 19BCE0215
TEACHER: Santhi H.

Activity-1

1. ifconfig:

The "**ifconfig**" command with no arguments will display all the active interfaces details. The **ifconfig** command also used to check the assigned IP address of an server.



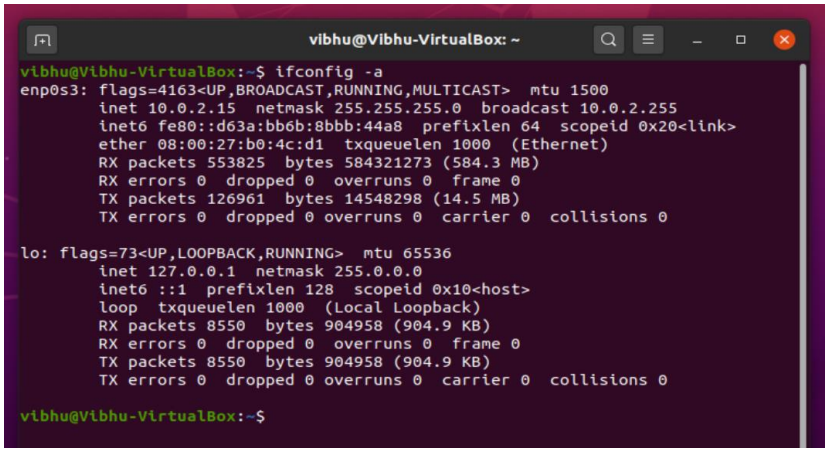
OPTIONS

i. 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 **eth0, lo, sit0 and tun0.**

Output:



Interpretation:

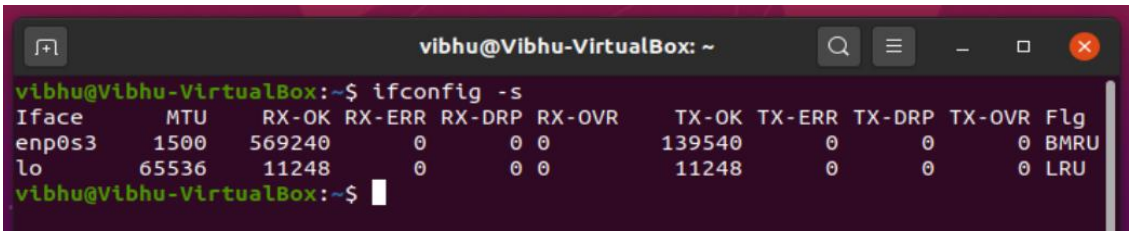
When we use this option, it returns detailed information of interfaces whether they are active or inactive. This can be used primarily when we want to see the detailed information, not just the summary.

ii. ifconfig -s:

Description:

Display a short list, instead of details using **ifconfig -s** option.

Output:



Interpretation:

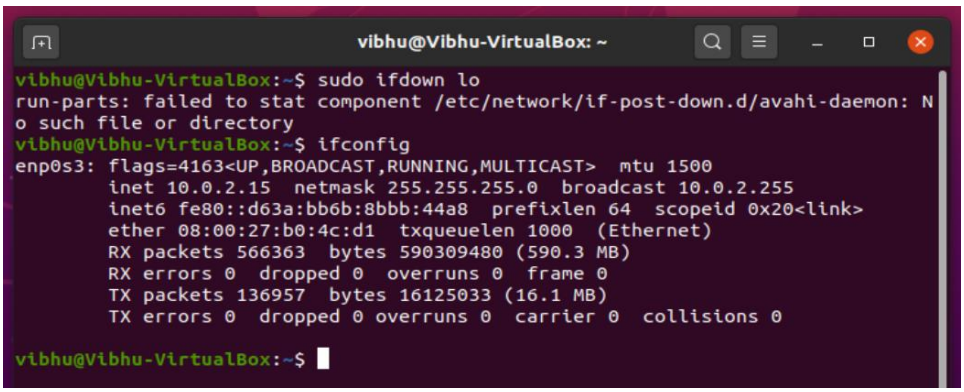
This options returns a short but important list of all the interfaces along with certain details such as MTU value, RX-OK, etc.

iii. ifconfig [interface] down/ ifdown [interface]:

Description:

The "**down**" or "**ifdown**" flag with interface name (**eth0**) deactivates the specified network interface. For example, "**ifconfig eth0 down**" or "**ifdown eth0**" command deactivates the **eth0** interface, if it is in active state.

Output:



Interpretation:

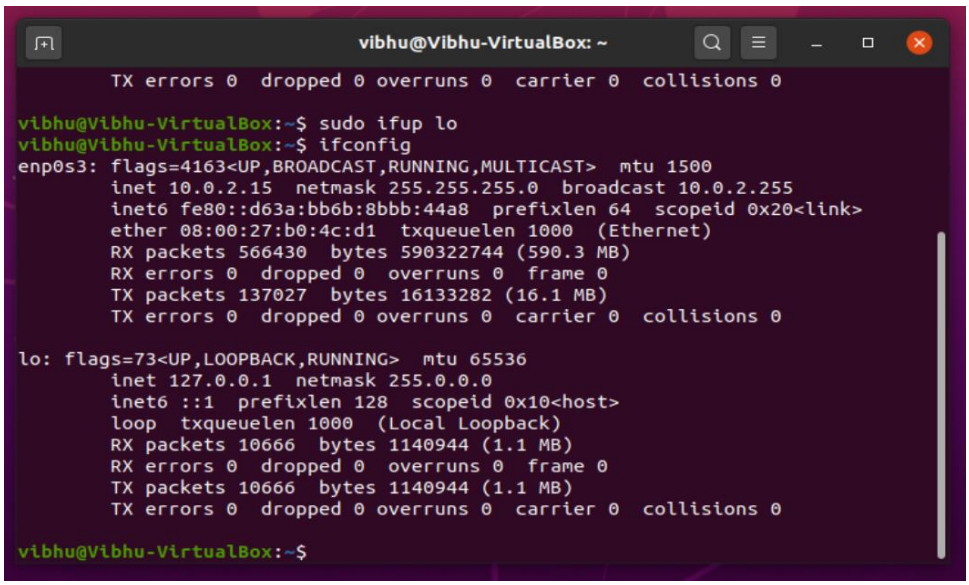
This option is used to disable any interface according to your own wish. It makes the interface disabled until the it is enable again or the system is restarted.

iv. ifconfig [interface] up/ ifup [interface]:

Description:

The "**up**" or "**ifup**" flag with interface name (**eth0**) activates an network interface, if it is not in active state and allowing to send and receive information. For example, "**ifconfig eth0 up**" or "**ifup eth0**" will activate the **eth0** interface.

Output:



Interpretation:

It is used to activate/enable any interface if it was previously disabled.

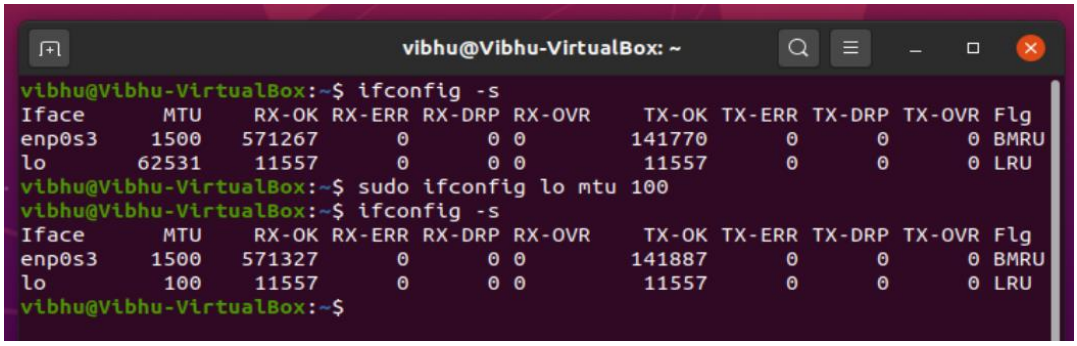
v. ifconfig [interface] mtu [value]:

Description:

The "**mtu**" argument set the maximum transmission unit to an interface. The **MTU** allows you to set the limit size of packets that are transmitted on an interface. The **MTU** able to handle maximum number of octets to an interface in one single transaction. For

example, "**ifconfig eth0 mtu 1000**" will set the maximum transmission unit to given set (i.e. **1000**). Not all network interfaces supports **MTU** settings.

Output:



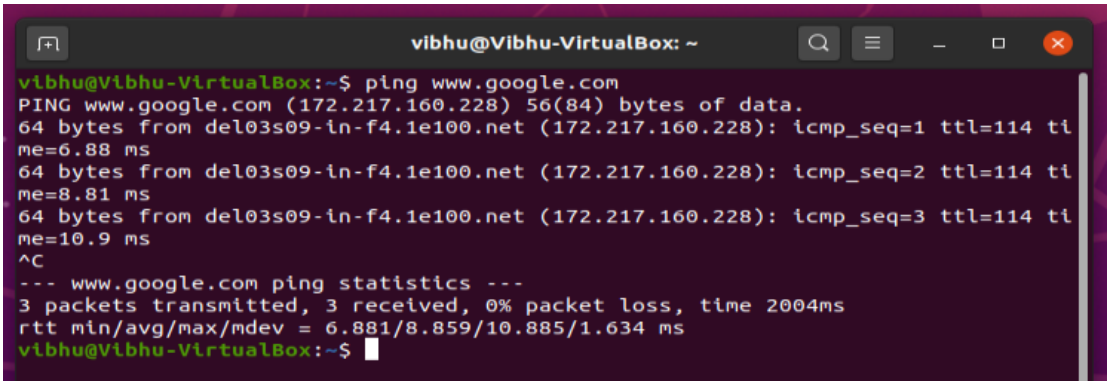
Interpretation:

This options allows the user to change the MTU value of any interface where MTU stands for Maximum Transmission Unit.

---

2. ping:

PING (Packet Internet Groper) command is used to check the network connectivity between host and server/host. This command takes as input the IP address or the URL and sends a data packet to the specified address with the message "PING" and get a response from the server/host this time is recorded which is called latency. Fast ping low latency means faster connection.



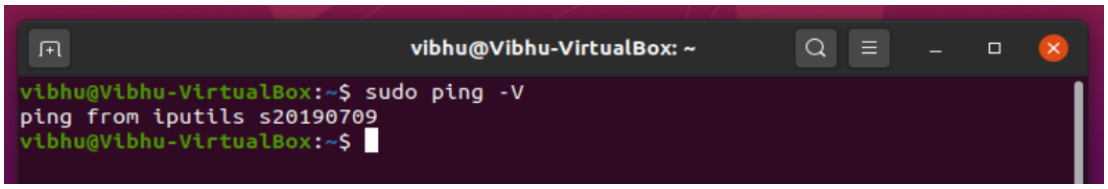
OPTIONS

i. ping -V:

Description:

Used to get the ping version installed on your system.

Output:



Interpretation:

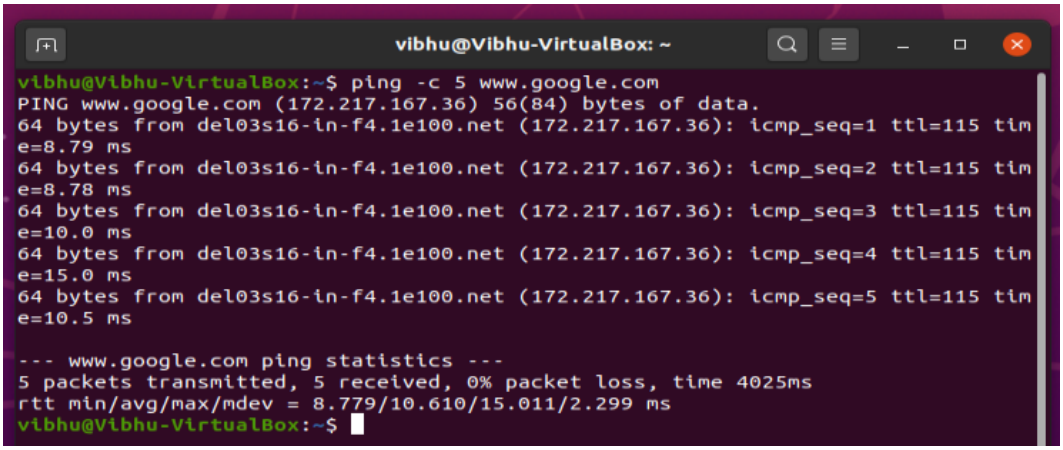
The option is used to return the current version of ping package installed on the system.

ii. ping -c:

Description:

Earlier we did not define the number of packets to send to the server/host by using **-c** option we can do so.

Output:



Interpretation:

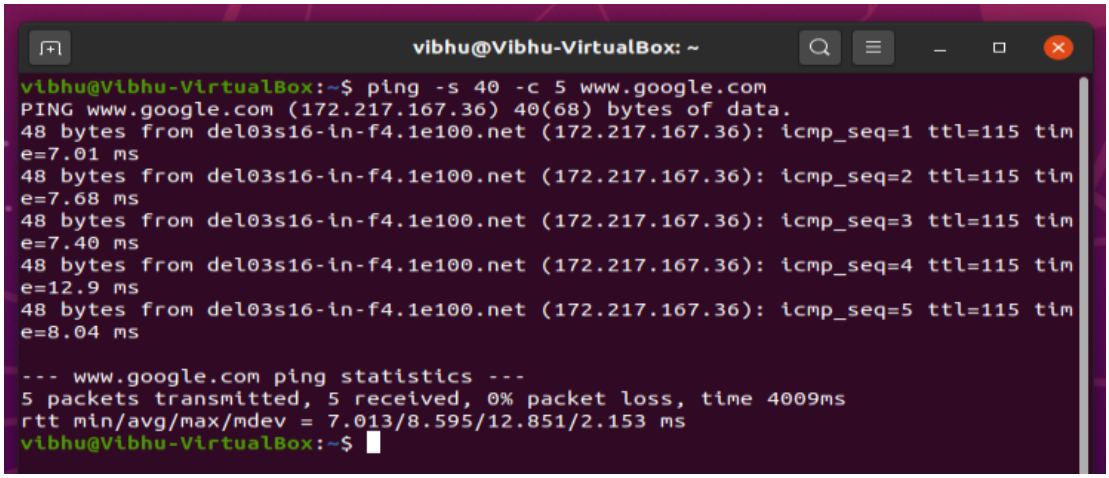
This option helps the user set the number of packets after which the ping process terminates. If nothing is specified, it pings for indefinite amount of time, until user hits ctrl+c.

iii. ping -s:

Description:

Ealier a default sized packets were sent to a host but we can send light and heavy packet by using **-s** option.

Output:



Interpretation:

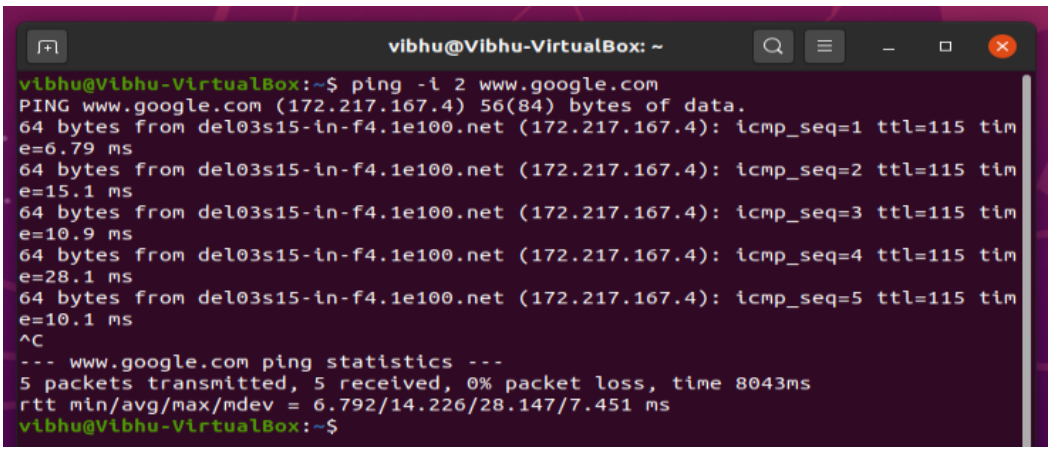
This option allows user to set the size of the packets while pinging.

iv. ping -i:

Description:

By default ping wait for 1 sec to send next packet we can change this time by using **-i** option.

Output:



Interpretation:

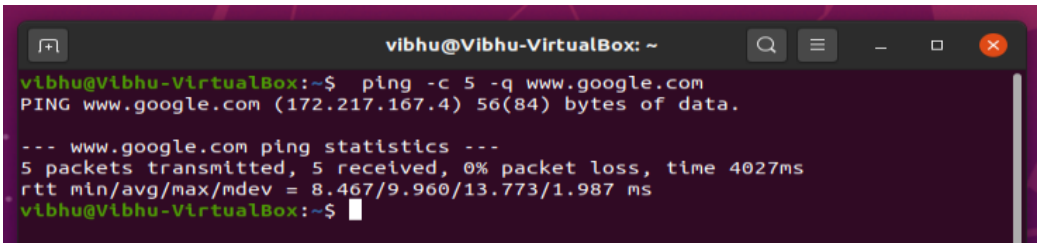
This option allows the user to set the time interval between exchange of packets. Here, we have set the time to 2s. This means that there will be a delay of 2s before each packet is transferred.

v. ping -q:

Description:

To only get the summary about the network use **-q** option.

Output:



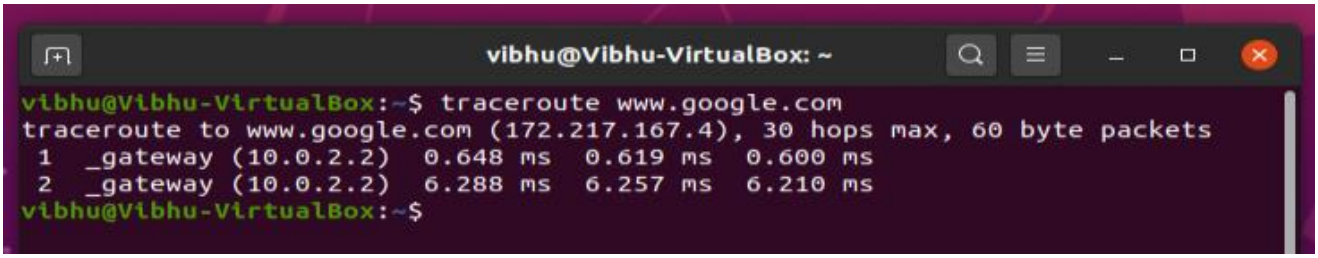
Interpretation:

This option provides the summary of the process after completing the regular pinging process.

---

3. traceroute:

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. Below image depicts how traceroute command is used to reach the Google(172.217.26.206) host from the local machine and it also prints detail about all the hops that it visits in between.



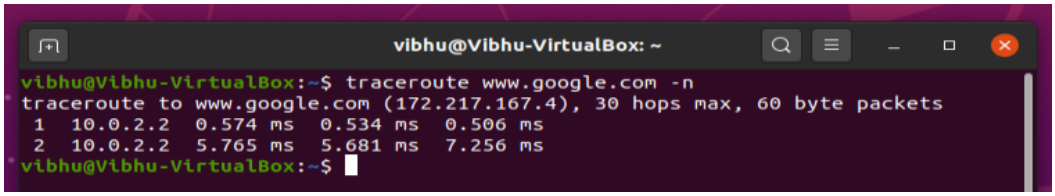
OPTIONS

i. traceroute -n:

Description:

Do not resolve IP addresses to their domain names.

Output:



Interpretation:

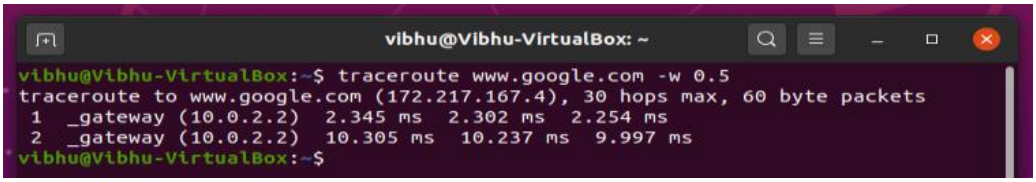
Each line gives the details of interaction with each router encountered. The traceroute not only gives the IP addresses of the intermediate routers, but also three round trip times for that particular router as for each router the traceroute commands fires three packets.

ii. traceroute -w:

Description:

Configures Response Wait Time.This traceroute utility waits after issuing a probe can also be configured.

Output:



Interpretation:

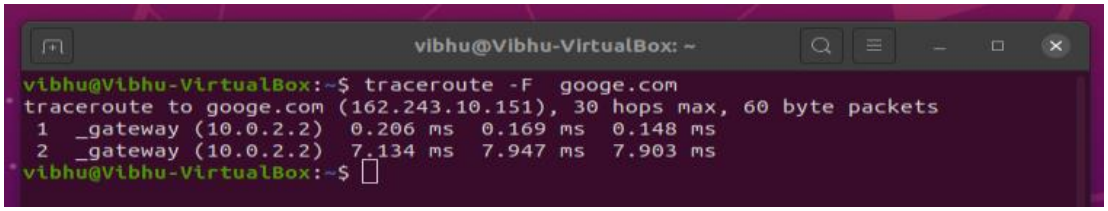
The **-w** option expects a value which the utility will take as the response time to wait.

iii. traceroute -F:

Description:

Do not fragment packet.

Output:



Interpretation:

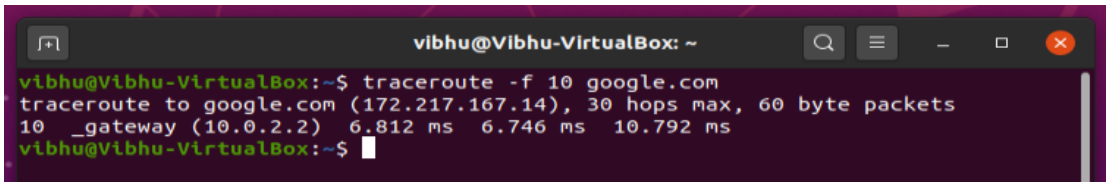
The option doesn't allow the packets to be fragmented. This means that all the packets are transferred in their entirety.

iv. traceroute -f:

Description:

Start from the first ttl hop (instead from 1).

Output:



Interpretation:

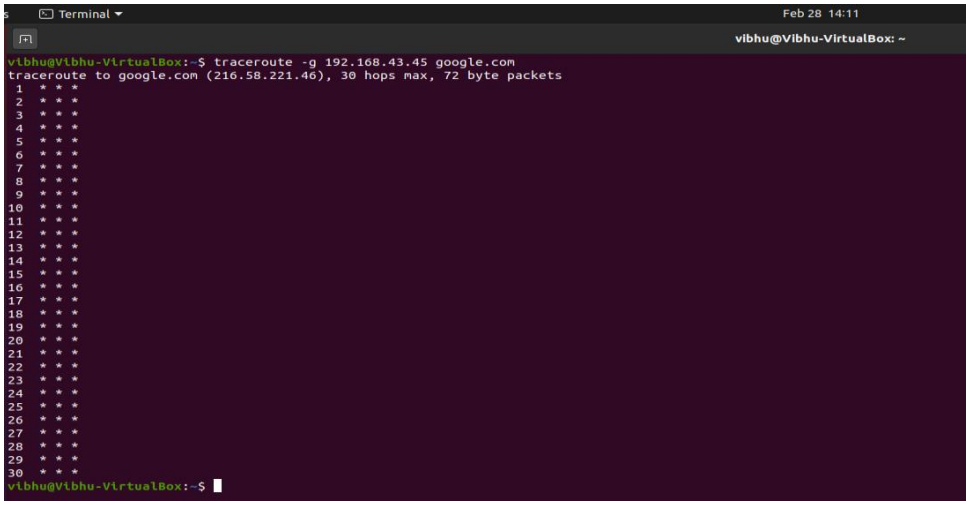
The default value of TTL is 1 which means it starts off with the first router in the path but using the **-f** option a new value of the TTL field can be set. That gives us a nice, condensed report on the state of the connection to the destination.

v. traceroute -g:

Description:

Routes the packet through gate.

Output:



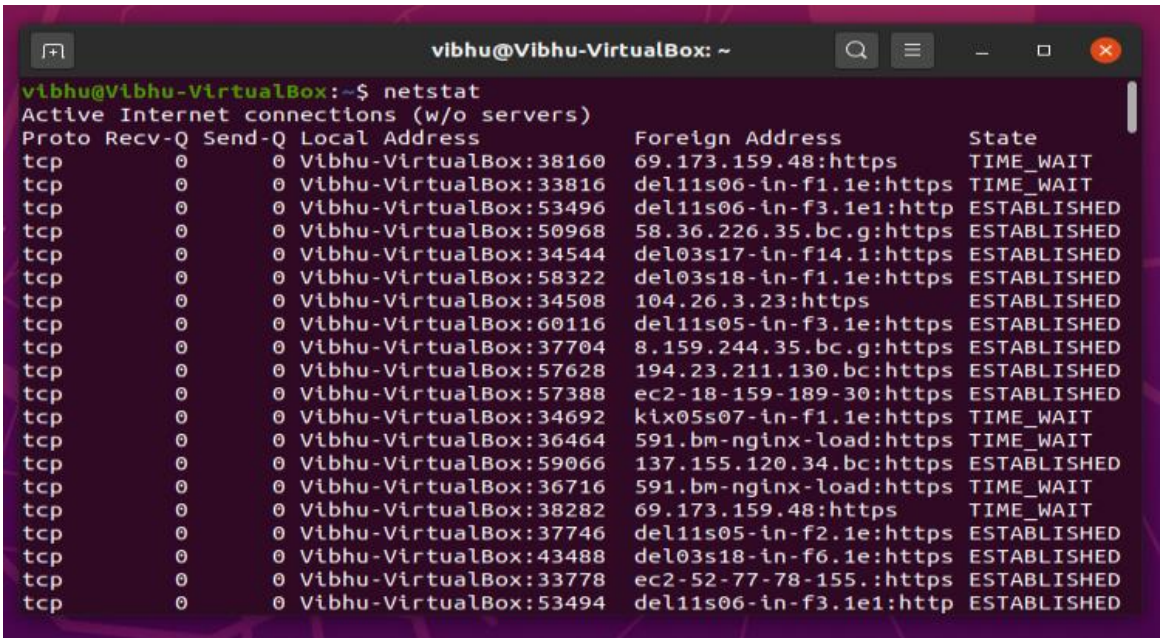
Interpretation:

Here, you can see in the screenshot it makes maximum of 30 hops through gates.

---

4. netstat:

This tool is very important and much useful for Linux network administrators as well as system administrators to monitor and troubleshoot their network-related problems and determine network traffic performance. This article shows usages of netstat command with their examples which may be useful in daily operation.



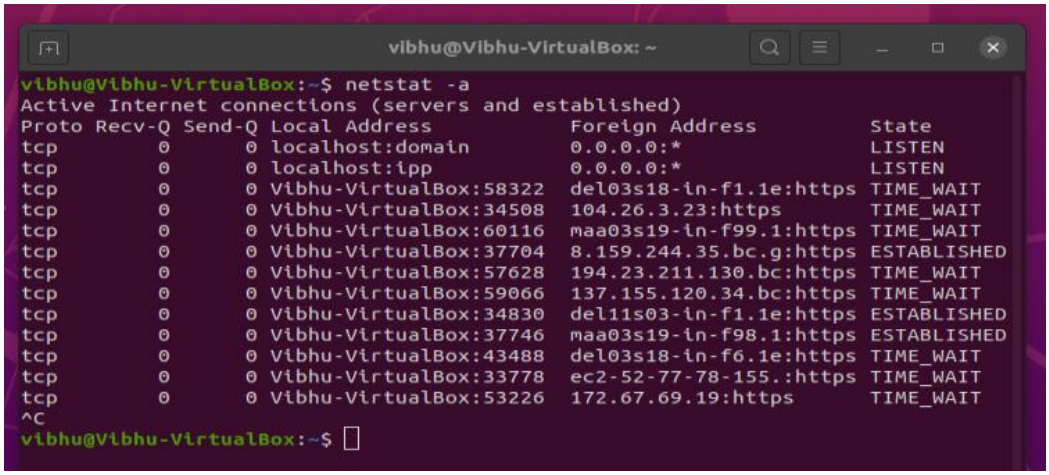
OPTIONS

i. netstat -a:

Description:

Listing all ports (both TCP and UDP) using **netstat** -a option.

Output:



Interpretation:

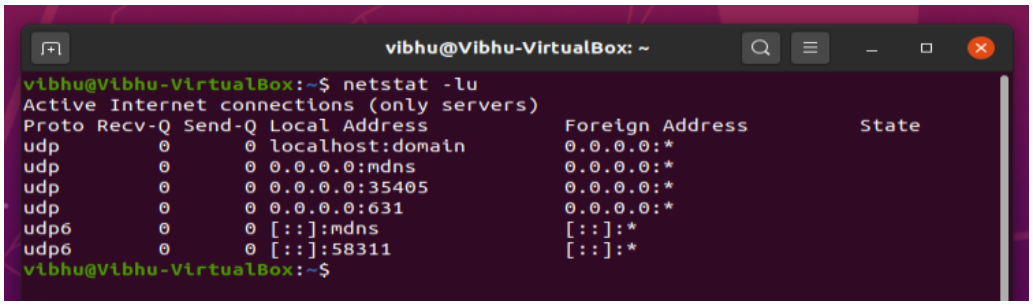
This option allows the user to enlist all the ports including TCP and UDP all in one command.

ii. netstat -lu:

Description:

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

Output:



Interpretation:

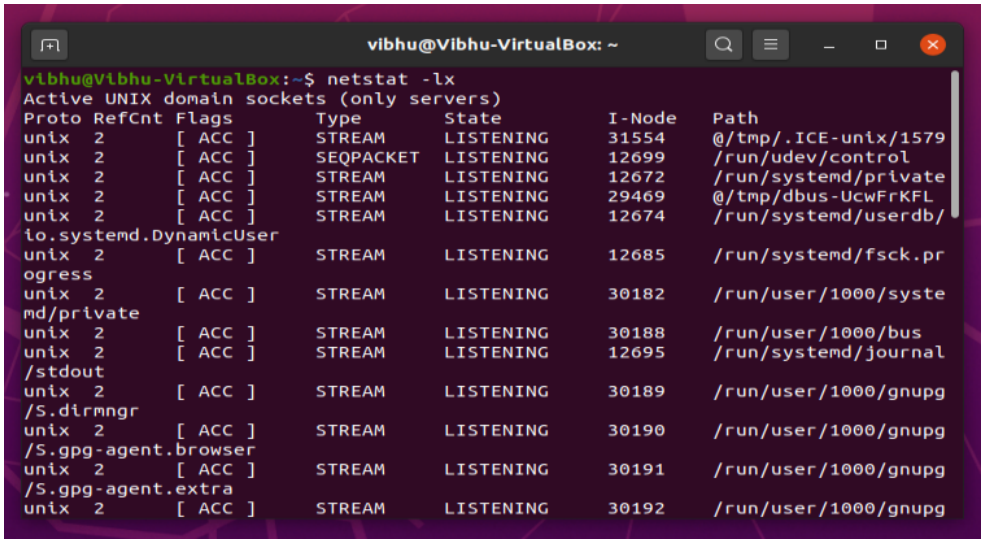
This option allows the user to enlist all the active listening UDP ports.

iii. netstat -lx:

Description:

Listing all active UNIX listening ports using **netstat -lx**.

Output:



Interpretation:

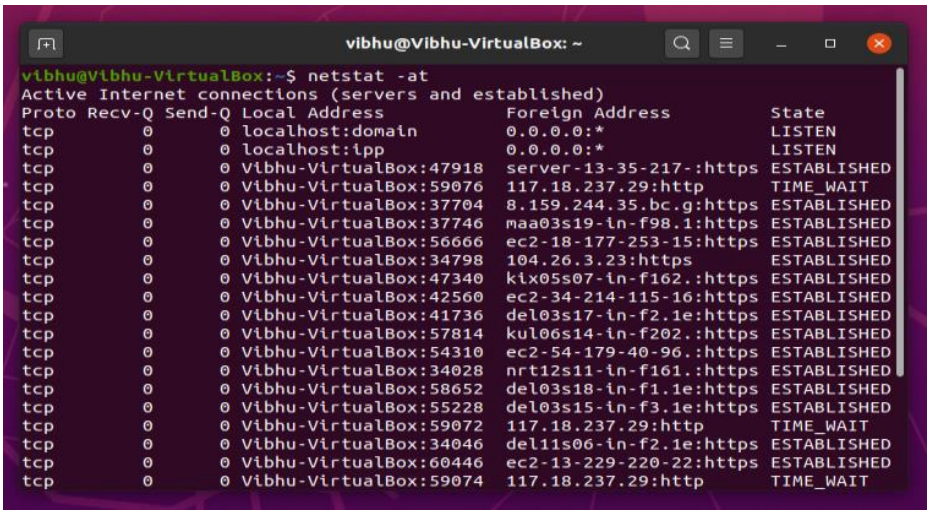
This option allows the user to get the details of all the active listening linux ports.

iv. netstat -at:

Description:

Listing only **TCP** (**Transmission Control Protocol**) port connections using **netstat -at**.

Output:



Interpretation:

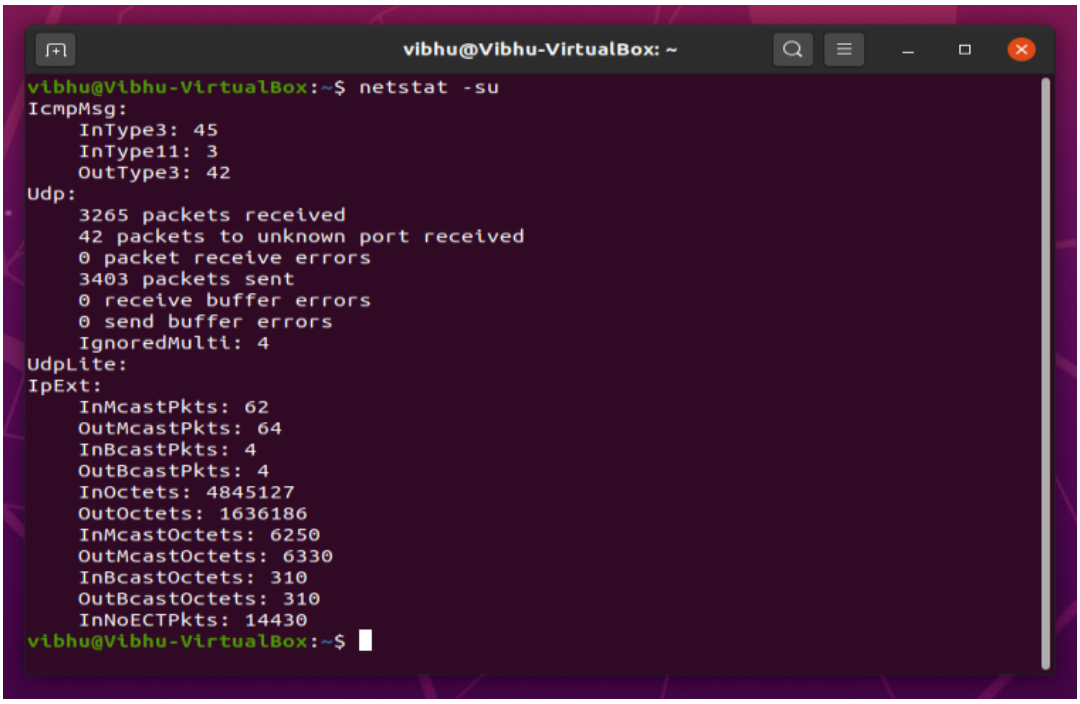
This options returns the details of all the TCP ports in one command.

v. netstat -su:

Description:

To Show Statistics by UDP Protocol, use **netstat -su** option.

Output:



Interpretation:

This option can be used to show the statistics by UDP Protocol.

---

5. nslookup:

Nslookup (stands for "Name Server Lookup") is a useful command for getting information from DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNS related problems.

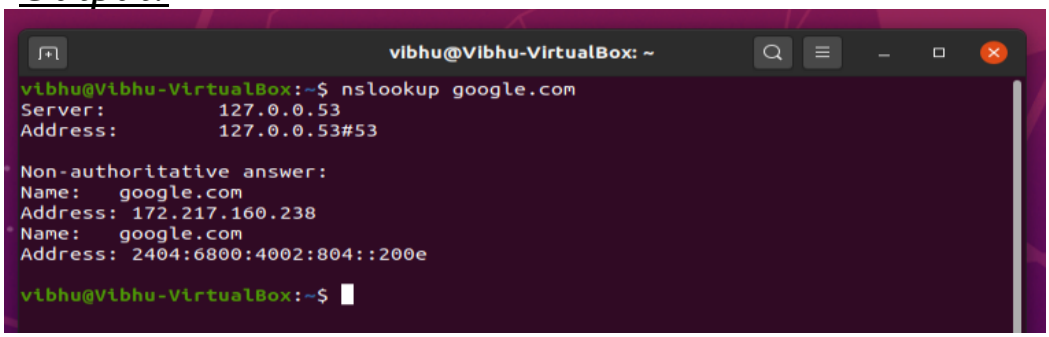
OPTIONS

1. nslookup [website]:

Description:

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:



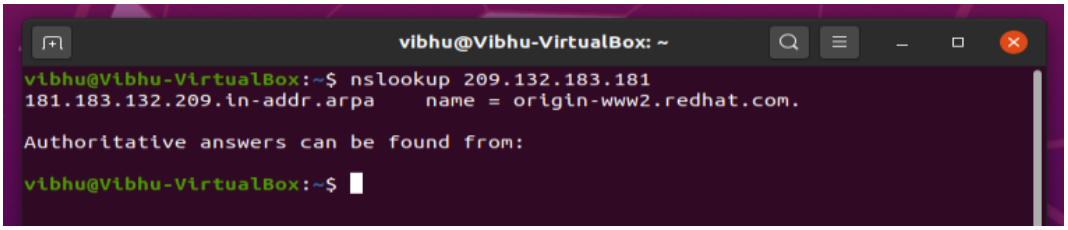
Interpretation:

2. nslookup [I.P. Address]:

Description:

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

Output:



Interpretation:

We can see from the above screenshot that we can do the reverse DNS lookup and search for the domain name associated with an IP

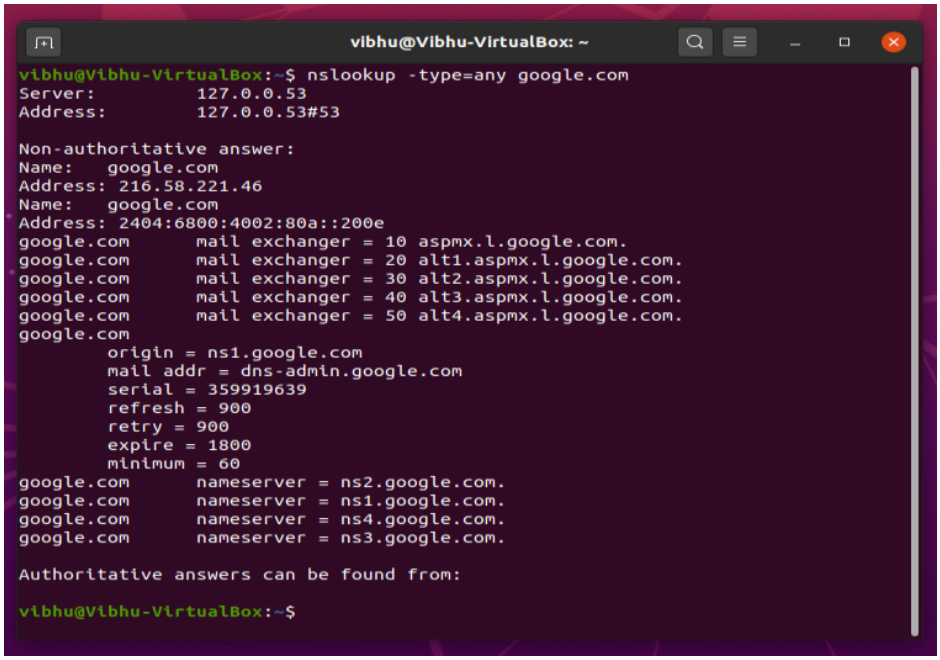
address. The IP address can be founded by simply typing- “nslookup redhat.com”.

3. nslookup -type=any [website]:

Description:

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

Output:



Interpretation:

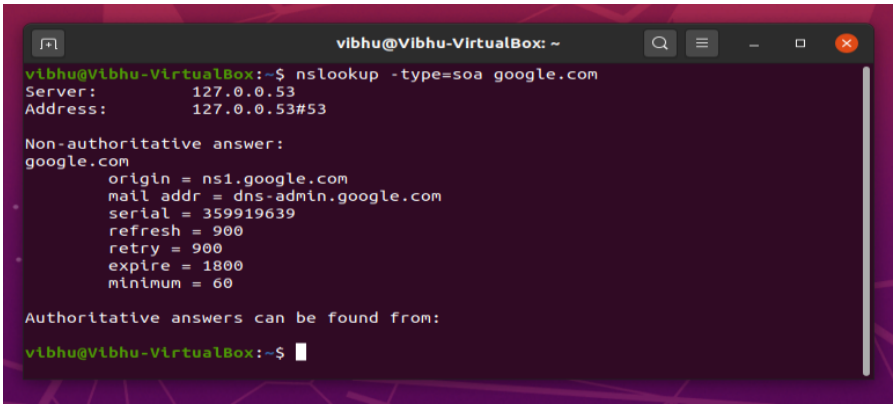
This option provides will all the available DNS records in a list.
In this example, the option returns the list of all the available DNS records for ‘www.google.com’.

4. nslookup -type=soa [website]:

Description:

SOA record (start of authority), provides the authoritative information about the domain, the e-mail address of the domain admin, the domain serial number.

Output:



Interpretation:

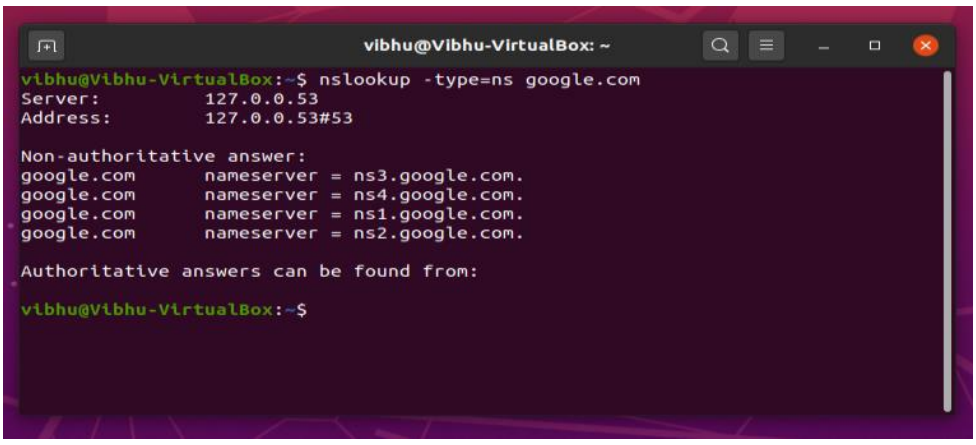
From the above screenshot we can see that SOA record (start of authority), provides the authoritative information about the domain, the e-mail address of the domain admin, the domain serial number, mail address, serial, refresh, retry, expire and minimum.

5. nslookup -type=ns [website]:

Description:

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.

Output:

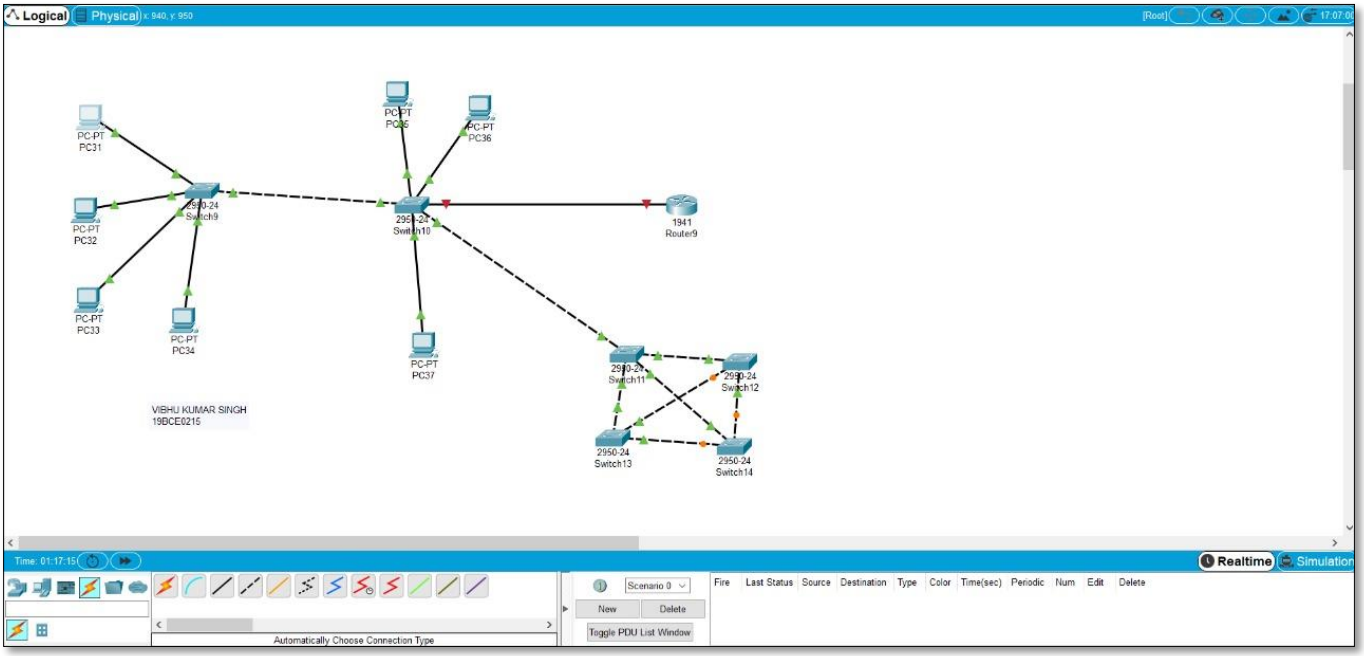


Interpretation:

This option provides will all the Name Server (NS) records in a list. In this example, the option returns the list of all the Name Server records for ‘www.google.com’.

Activity-2

1. Network based on basic topology:



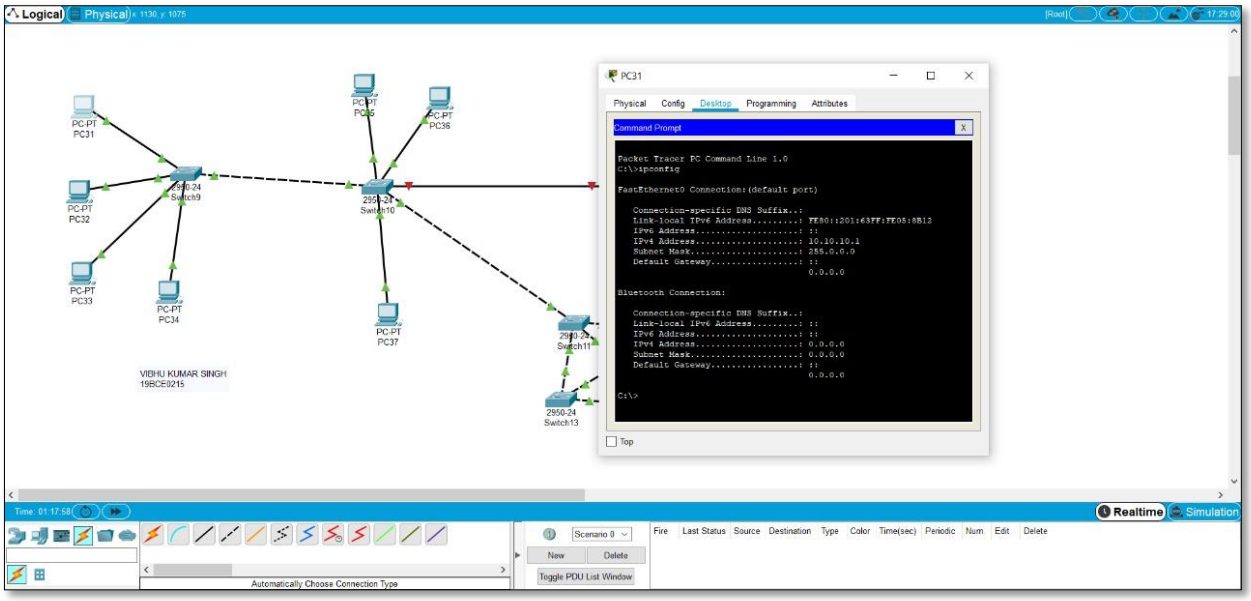
For switches
- Network devices
- Switches
- 2960-24 model

For PCs
- End Devices
- PC

For Routers
- Network Devices
- Routers
- 1941 model

For Connections
- Connections

For connections between switches and PC's, use copper straight-through.
For connections between switches and switch, use copper cross-over.
For connections between switches and router, use copper straight-through.

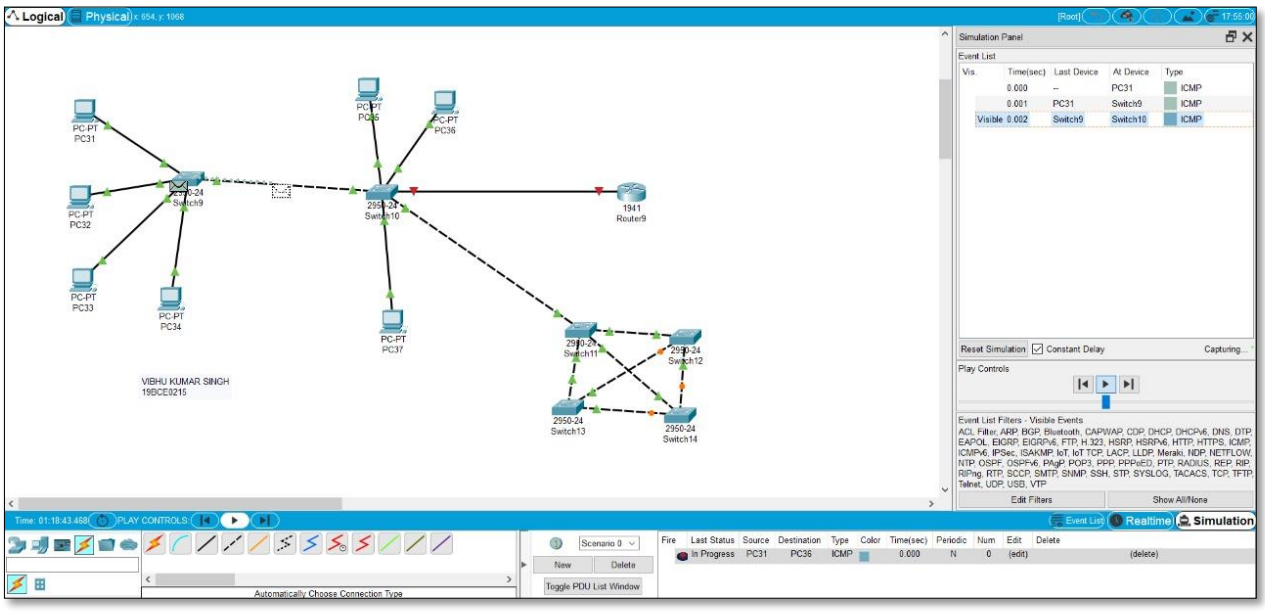


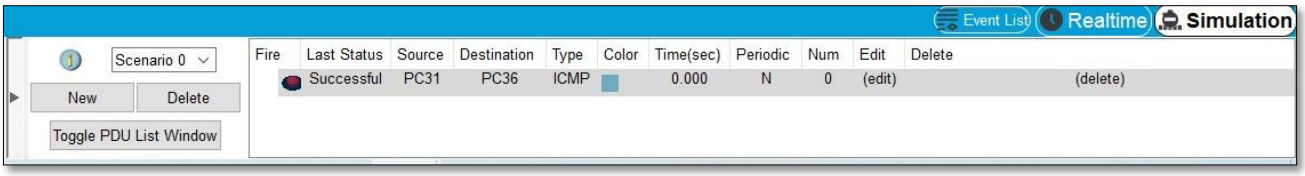
I.P. Configuration

PC31 – 10.10.10.1
PC32 – 10.10.10.2
PC33 – 10.10.10.3
PC34 – 10.10.10.4
PC35 – 10.10.10.5
PC36 – 10.10.10.6
PC37 – 10.10.10.7

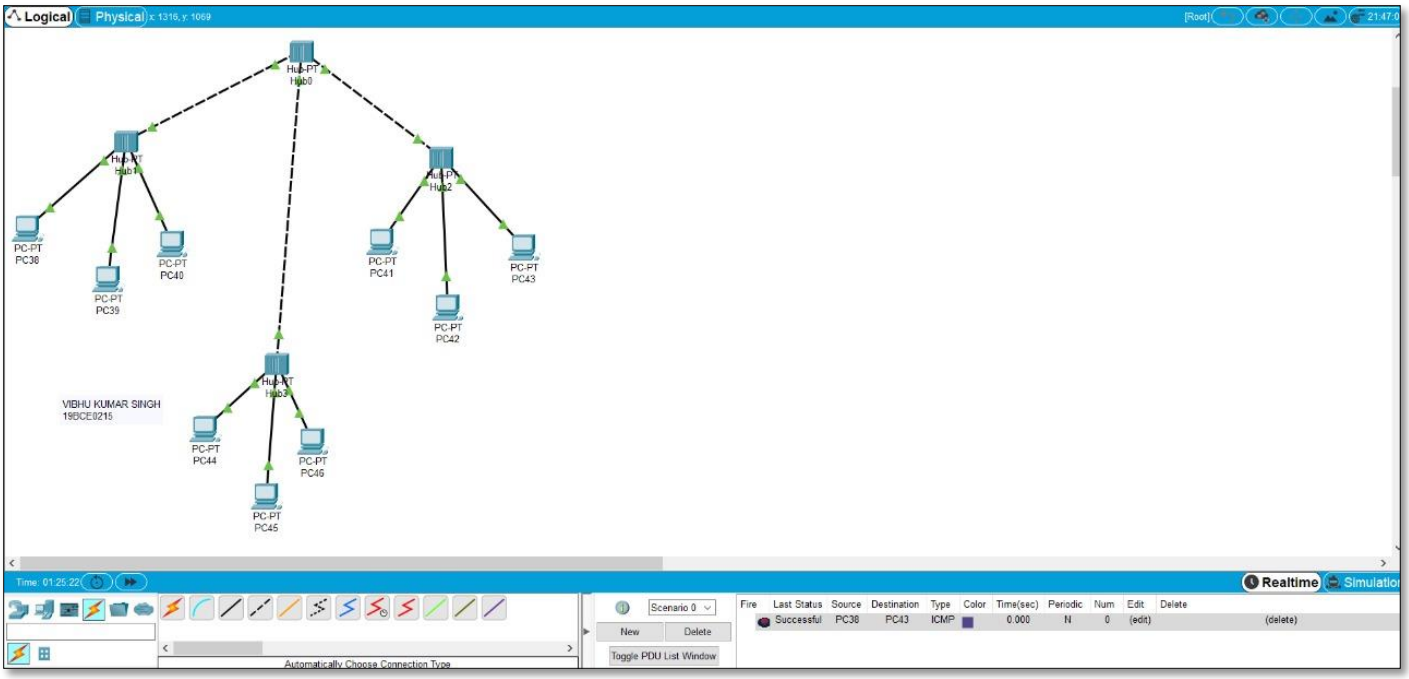
Simulation Results:

- Source: P31
- Destination: P36





2. Network based on Hub:



For PCs

- End devices
- PC

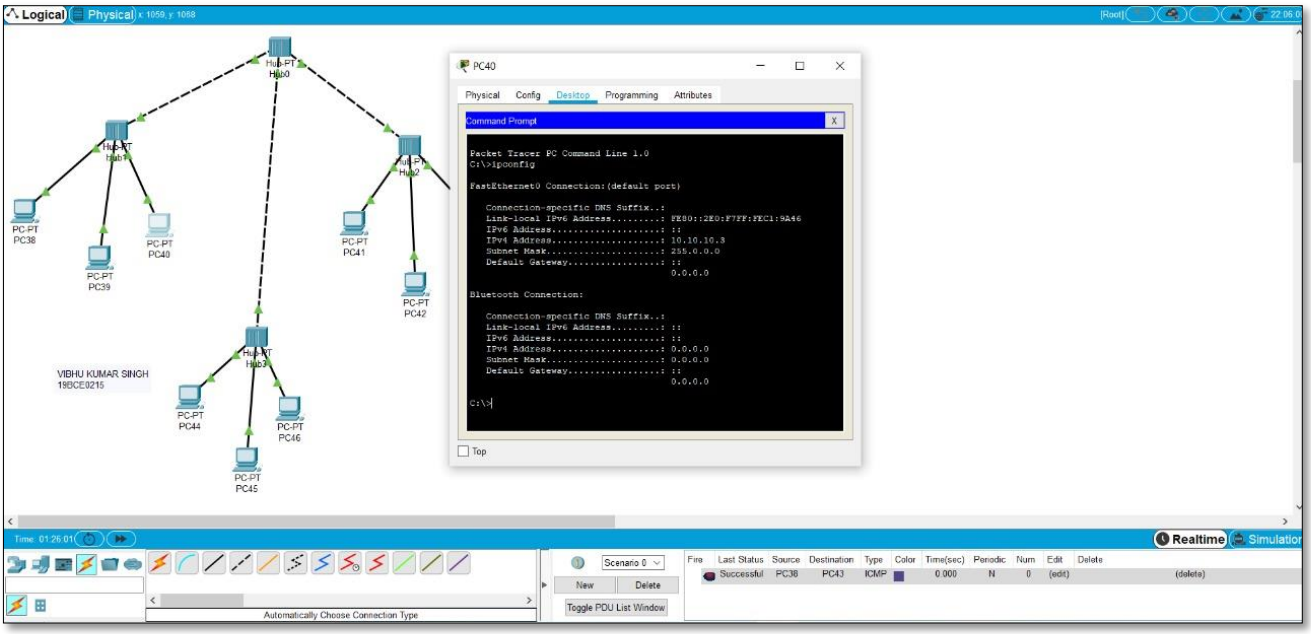
For Hubs

- 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.

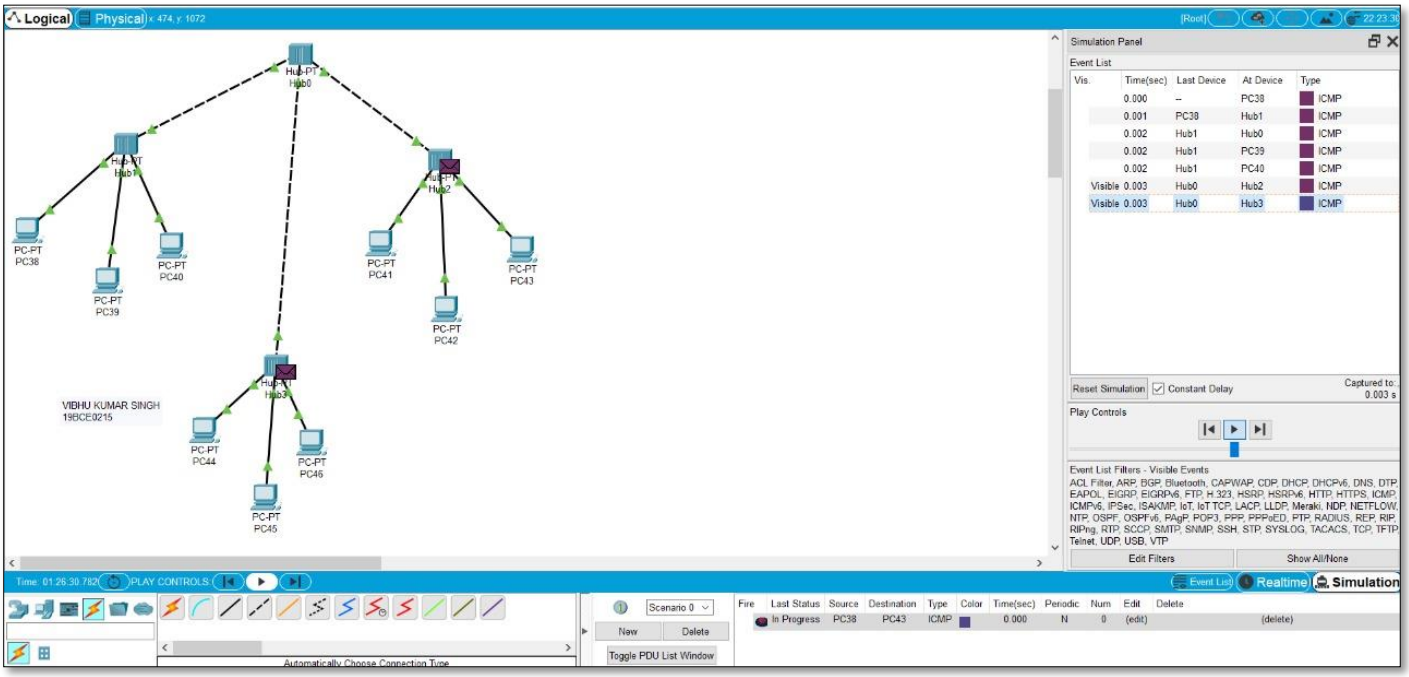


IP Configuration:

PC38 – 10.10.10.1
PC39 – 10.10.10.2
PC40 – 10.10.10.3
PC41 – 10.10.10.4
PC42 – 10.10.10.5
PC43 – 10.10.10.6
PC44 – 10.10.10.7
PC45 – 10.10.10.8
PC46 – 10.10.10.9

Simulation Results:

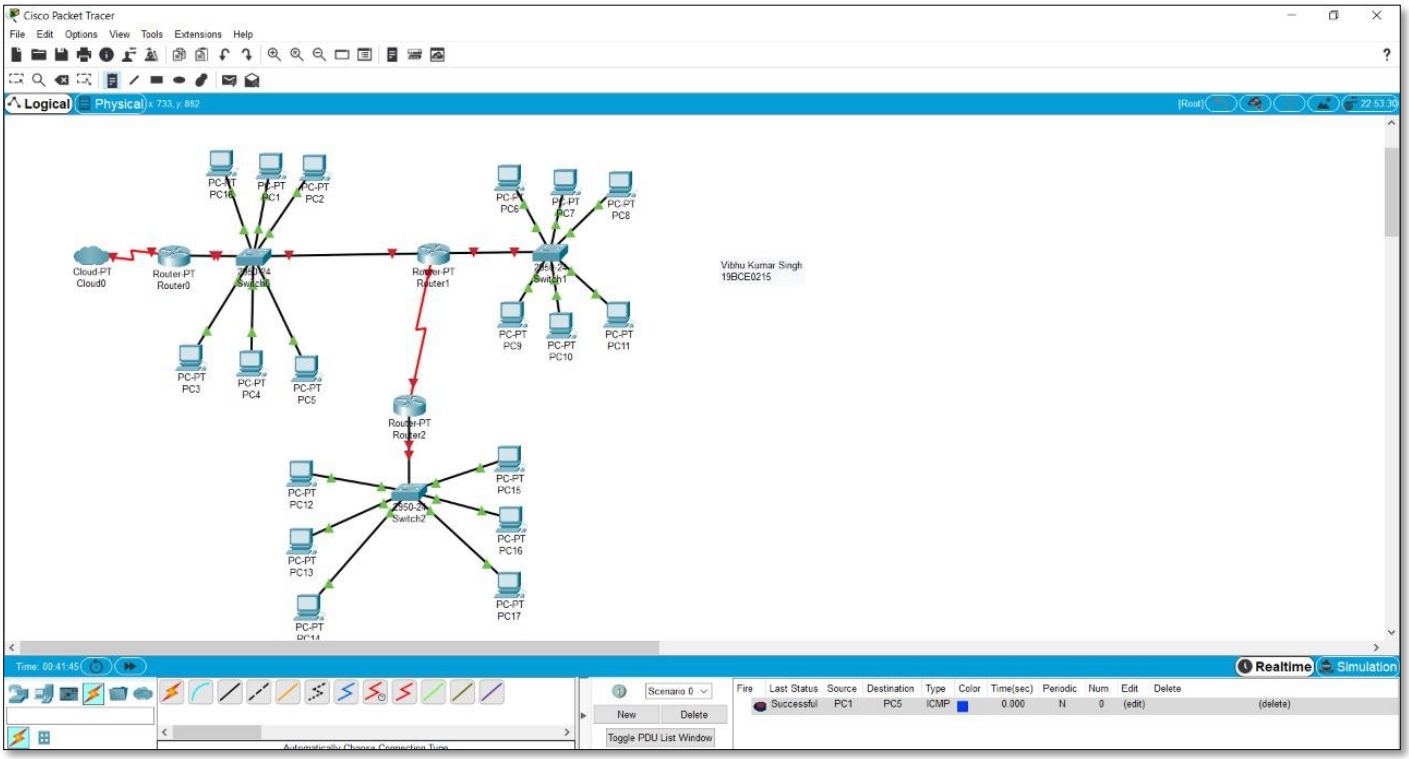
- Source: PC38
- Destination: PC43



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC38	PC43	ICMP		0.000	N	0	(edit)	(delete)

---

3. Network Based on switch



For switches

- Network devices
- Switches
- 2960-24 model

For PCs

- End Devices
- PC

For Routers

- Network Devices
- Routers
- 2811 model

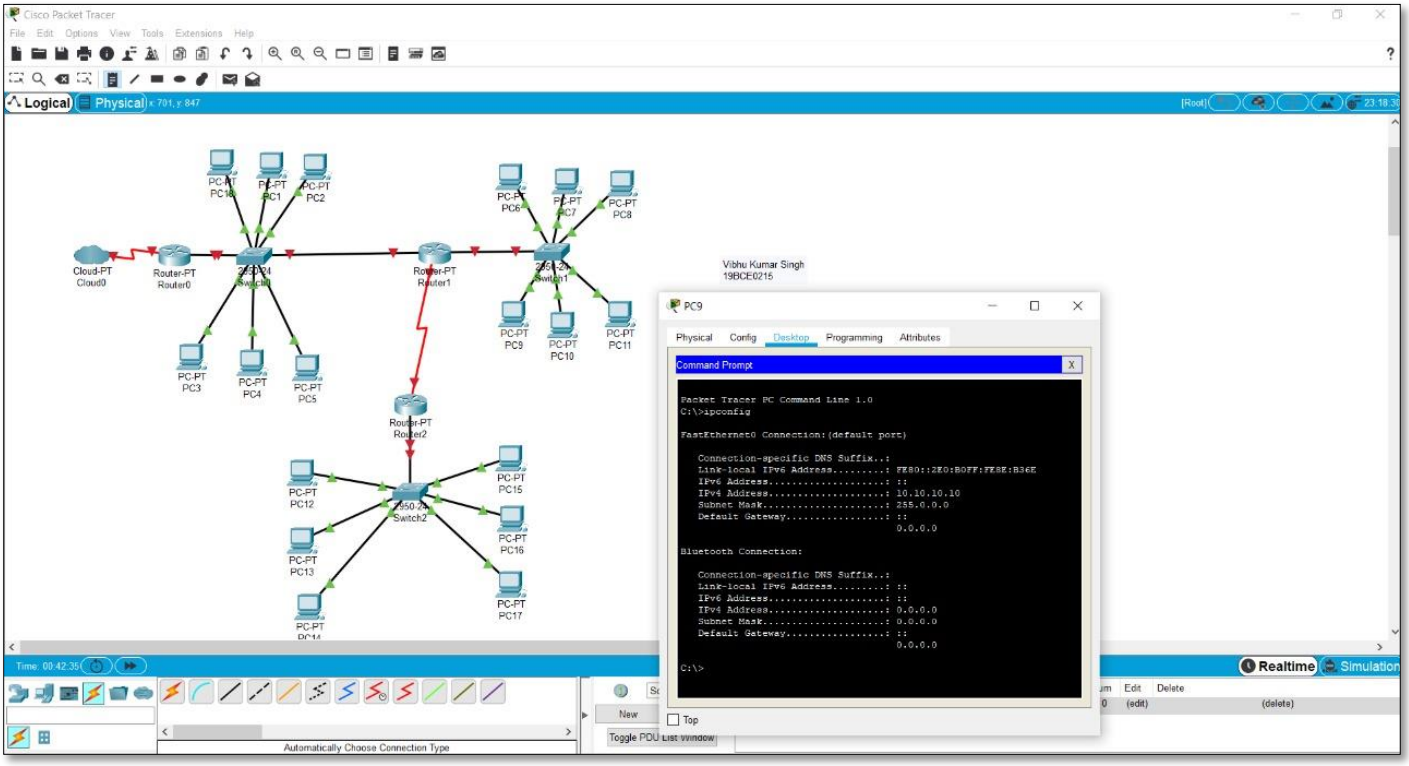
For Connections

- Connections

For cloud

- Network Devices
- WAN Emulation
- PT Cloud

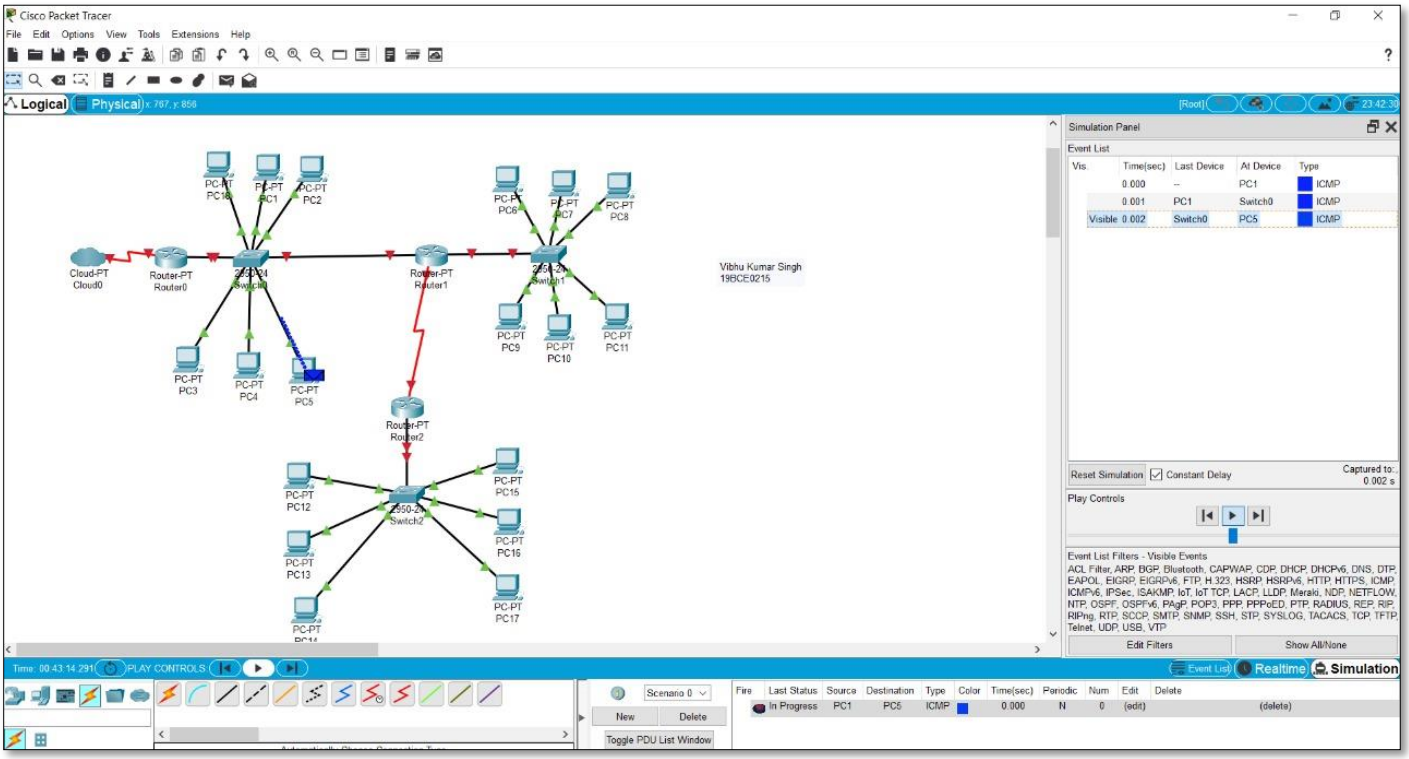
For connections between switches and PC's use copper straight-through
For connections between switch and switch use copper cross-over
For connections between switch and router use copper straight-through
For connections between router and router use Serial DTE
For connection between cloud and router use Serial DTE

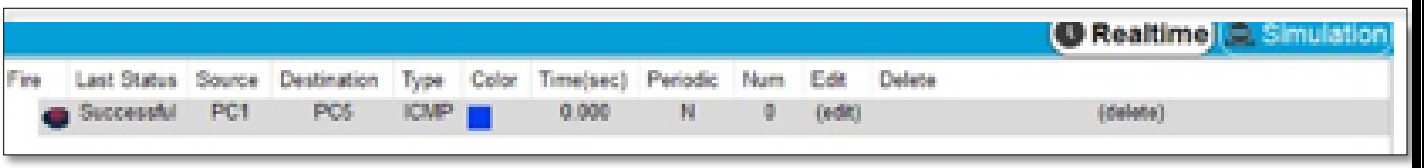


<u>IP configrations</u>
PC0-10.10.10.1
PC1-10.10.10.2
PC2-10.10.10.3
PC3-10.10.10.4
PC4-10.10.10.5
PC5-10.10.10.6
PC6 -10.10.10.7
PC7 - 10.10.10.8
PC8 - 10.10.10.9
PC9 - 10.10.10.10
PC10 - 10.10.10.11
PC11 - 10.10.10.12
PC12 - 10.10.10.13
PC14 - 10.10.10.15
PC15 - 10.10.10.16
PC16 - 10.10.10.17
PC17 - 10.10.10.18
PC18 - 10.10.10.19

Simulation Results:

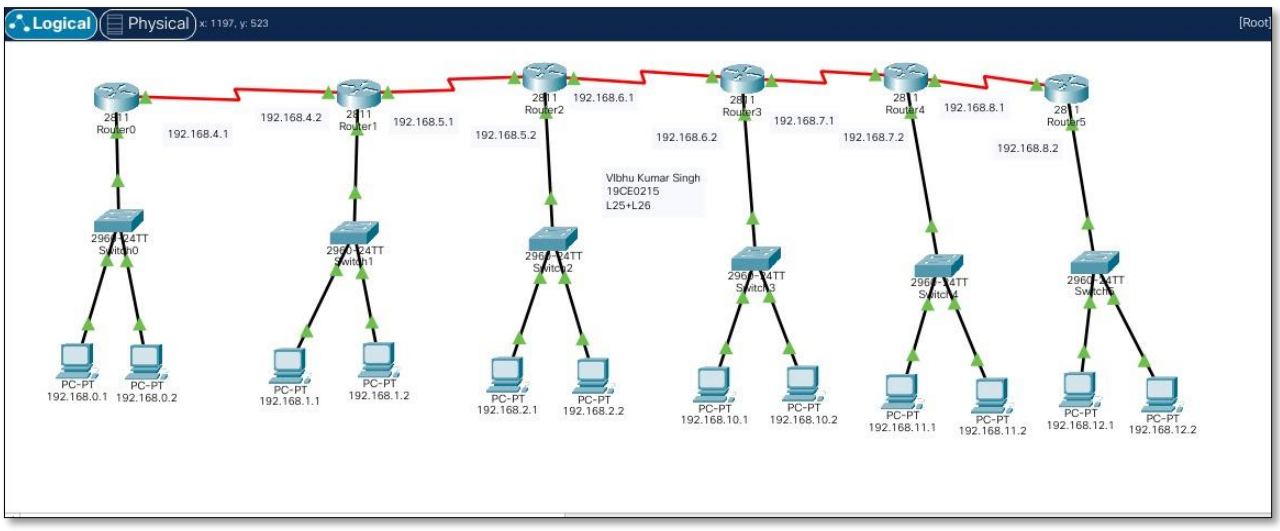
- Source: PC1
- Destination: PC5





---

4. Based on Switch and Router:



For switches

- Network devices
- Switches
- 2960-24 model

For PC's

- End Devices
- PC

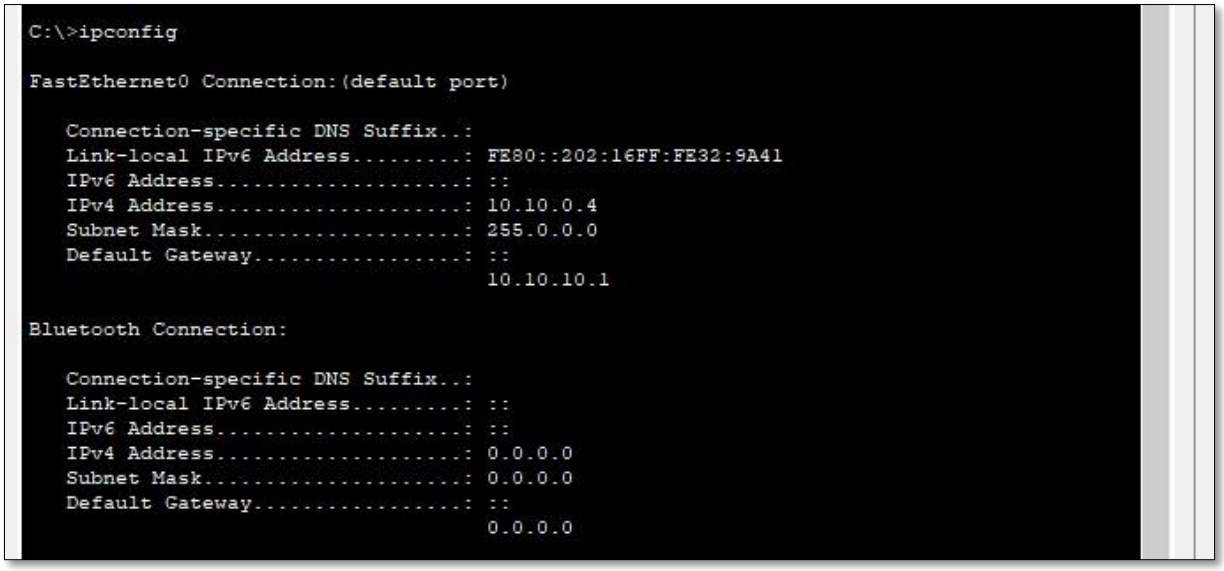
For Routers

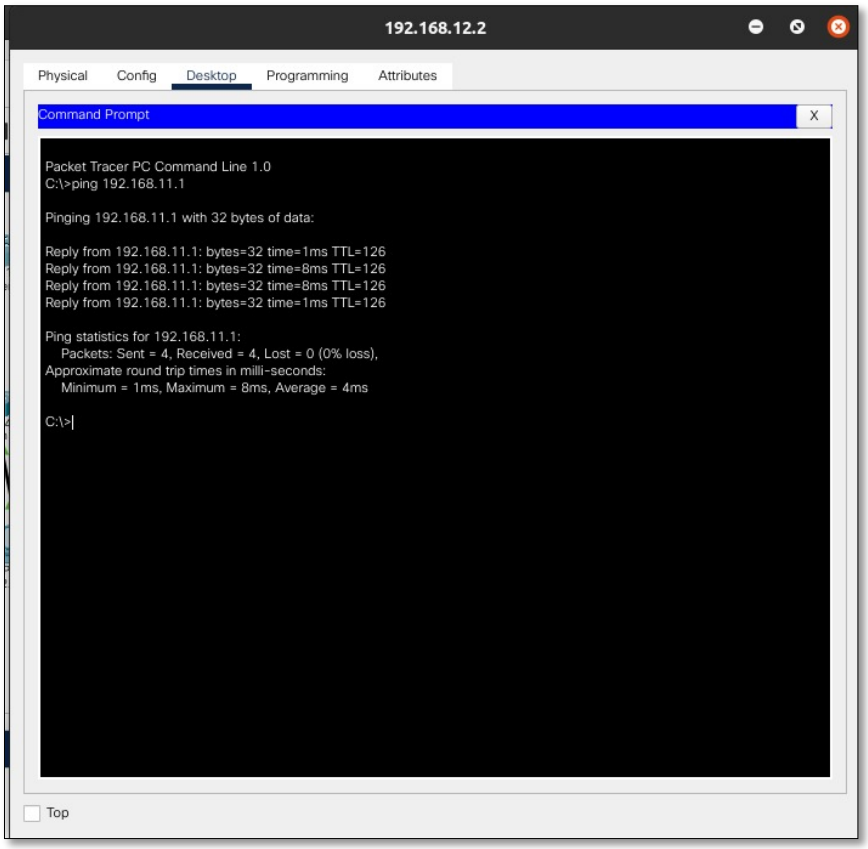
- Network Devices
- Routers
- 2811 model

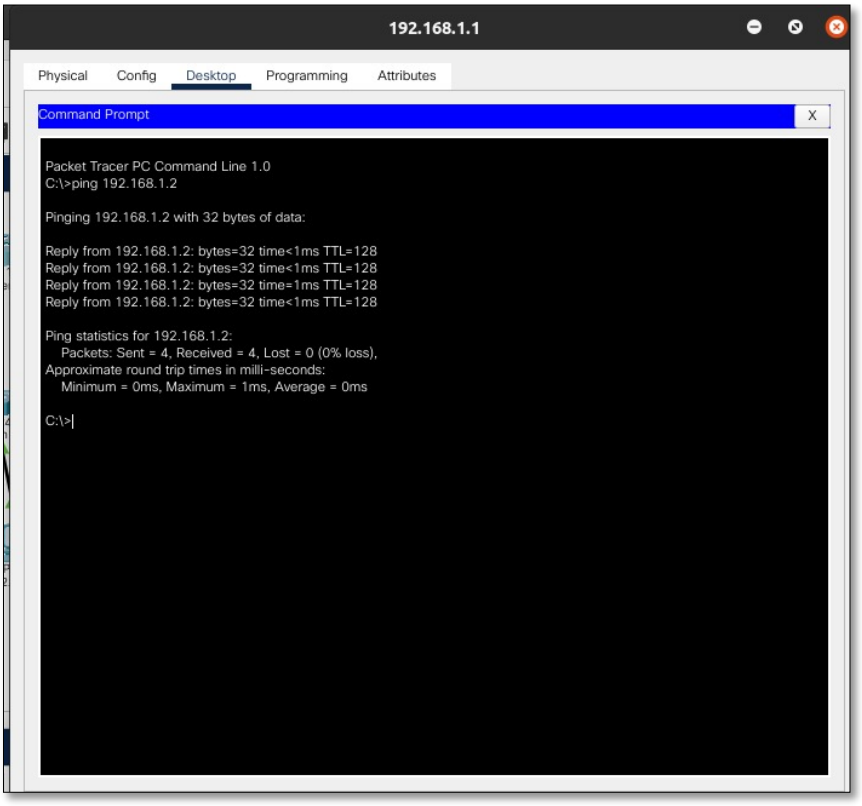
For Connections

- 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. Configurations:
PC0 – 198.168.0.1
PC1 – 198.168.0.2
Default gateway for PC0 and PC1 is 198.168.0.3
PC2 – 198.168.1.1

PC3 – 198.168.1.2
Default gateway for PC2 and PC3 is 198.168.1.3
PC4 – 198.168.2.1
PC5 – 198.168.2.2
Default gateway for PC4 and PC5 is 198.168.2.3
PC6 – 198.168.3.1
PC7 – 198.168.3.2
Default gateway for PC6 and PC7 is 198.168.3.3

Between Router 0 and Router 1 I.P addresses are 198.168.4.1 and 198.168.4.2
Between Router 1 and Router 2 I.P addresses are 198.168.5.1 and 198.168.5.2
Between Router 2 and Router 3 I.P addresses are 198.168.6.1 and 198.168.6.2
Between Router 3 and Router 4 I.P addresses are 198.168.7.1 and 198.168.7.2
Between Router 4 and Router 5 I.P addresses are 198.168.8.1 and 198.168.8.2

For communication between 2 routers static routes are to be enabled

To enable static routes:
Click on router→config→static.

1. In network tab give the I.P. address of the route you want to connect.
2. In mask tab give the mask of the I.P. 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.