Department of Computer/Information Technology Engineering
SE: SEM IV Subject: CN

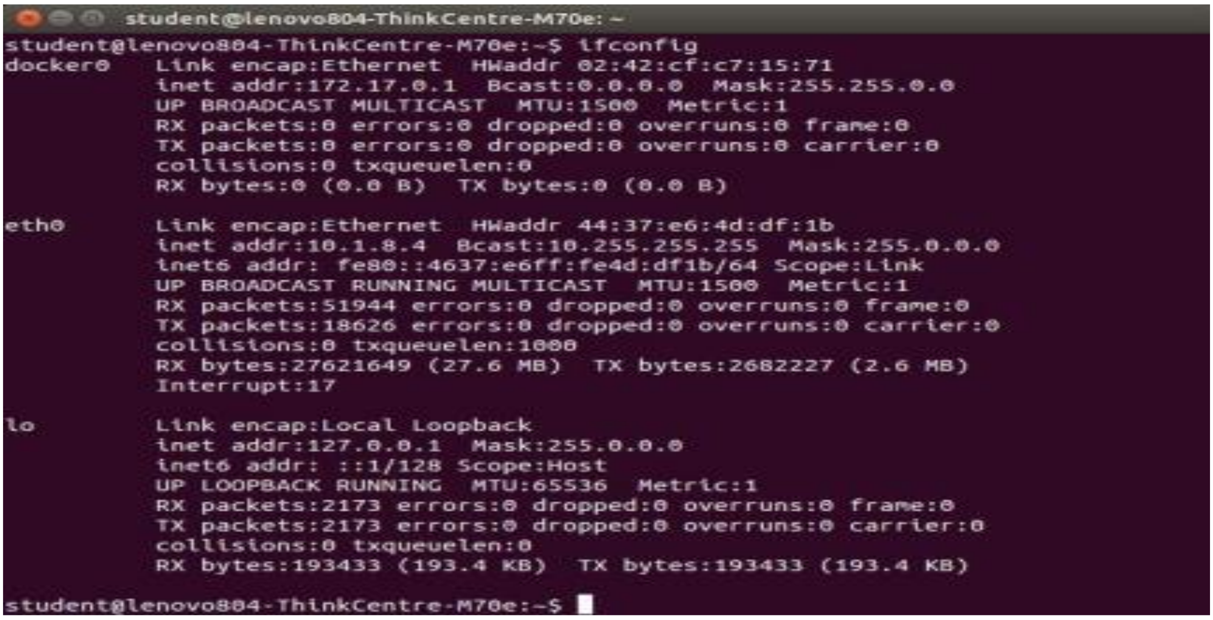
EXPERIMENT NO 01

AIM: Use basic networking commands in Linux (ping, tracert, nslookup, netstat , ARP, ifconfig, dig, route)

THEORY:

1. ifconfig

ifconfig(interface configuration) command is used to configure the kernel-resident network interfaces. It is used at the boot time to set up the interfaces as necessary. After that, it is usually used when needed during debugging or when you need system tuning. Also, this command is used to assign the IP address and netmask to an interface or to enable or disable a given interface.

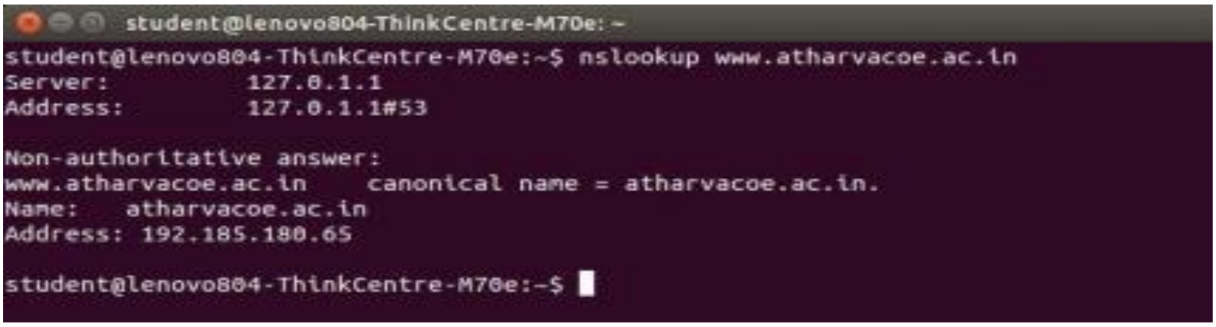


student@lenovo804-ThinkCentre-M70e: ~

```
student@lenovo804-ThinkCentre-M70e:~$ ifconfig
docker0   Link encap:Ethernet  HWaddr 02:42:cf:c7:15:71
          inet addr:172.17.0.1  Bcast:0.0.0.0  Mask:255.255.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth0      Link encap:Ethernet  HWaddr 44:37:e6:4d:df:1b
          inet addr:10.1.8.4  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::4637:e6ff:fe4d:df1b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:51944 errors:0 dropped:0 overruns:0 frame:0
          TX packets:18626 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:27621649 (27.6 MB)  TX bytes:2682227 (2.6 MB)
          Interrupt:17

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2173 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2173 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:193433 (193.4 KB)  TX bytes:193433 (193.4 KB)

student@lenovo804-ThinkCentre-M70e:~$
```

2. NSLOOKUP

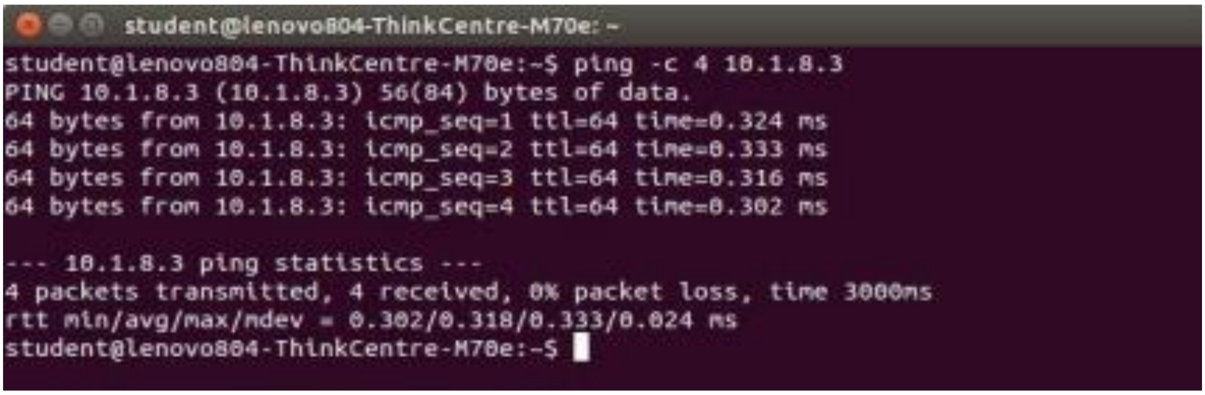
Nslookup (stands for "Name Server Lookup") is a useful command for getting information from DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNS related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



```
student@lenovo804-ThinkCentre-M70e:~$ nslookup www.atharvacoe.ac.in
Server:         127.0.1.1
Address:        127.0.1.1#53

Non-authoritative answer:
www.atharvacoe.ac.in    canonical name = atharvacoe.ac.in.
Name:   atharvacoe.ac.in
Address: 192.185.180.65

student@lenovo804-ThinkCentre-M70e:~$ 
```

3. Ping

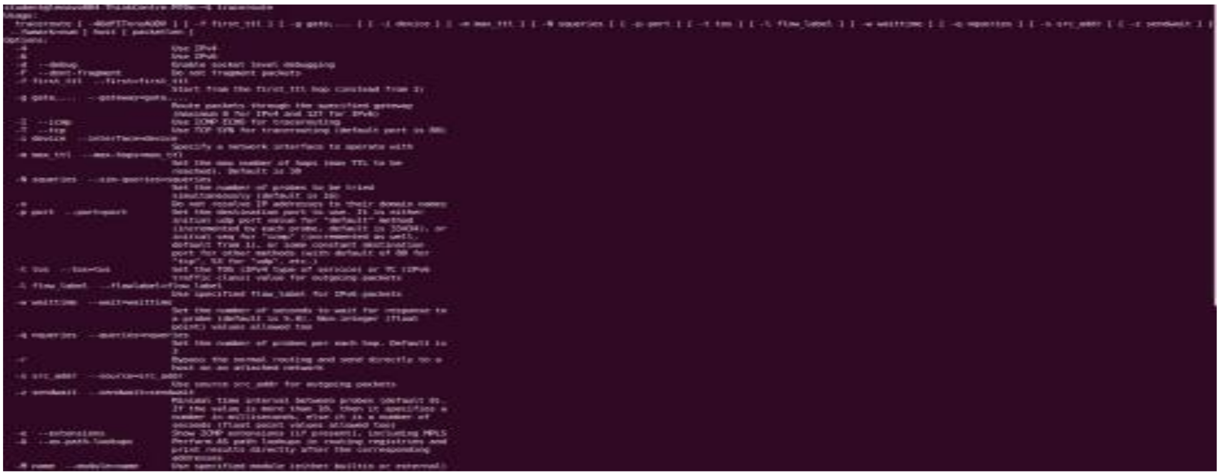
PING (Packet Internet Groper) command is used to check the network connectivity between host and server/host. This command takes as input the IP address or the URL and sends a data packet to the specified address with the message “PING” and get a response from the server/host this time is recorded which is called latency. Fast ping low latency means faster connection. Ping uses **ICMP(Internet Control Message Protocol)** to send an **ICMP echo message** to the specified host if that host is available then it sends **ICMP reply message**. Ping is generally measured in millisecond every modern operating system has this ping pre-installed.



```
student@lenovo804-ThinkCentre-M70e:~$ ping -c 4 10.1.8.3
PING 10.1.8.3 (10.1.8.3) 56(84) bytes of data.
64 bytes from 10.1.8.3: icmp_seq=1 ttl=64 time=0.324 ms
64 bytes from 10.1.8.3: icmp_seq=2 ttl=64 time=0.333 ms
64 bytes from 10.1.8.3: icmp_seq=3 ttl=64 time=0.316 ms
64 bytes from 10.1.8.3: icmp_seq=4 ttl=64 time=0.302 ms

--- 10.1.8.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.302/0.318/0.333/0.024 ms
student@lenovo804-ThinkCentre-M70e:~$ 
```

4. TRACEROUTE

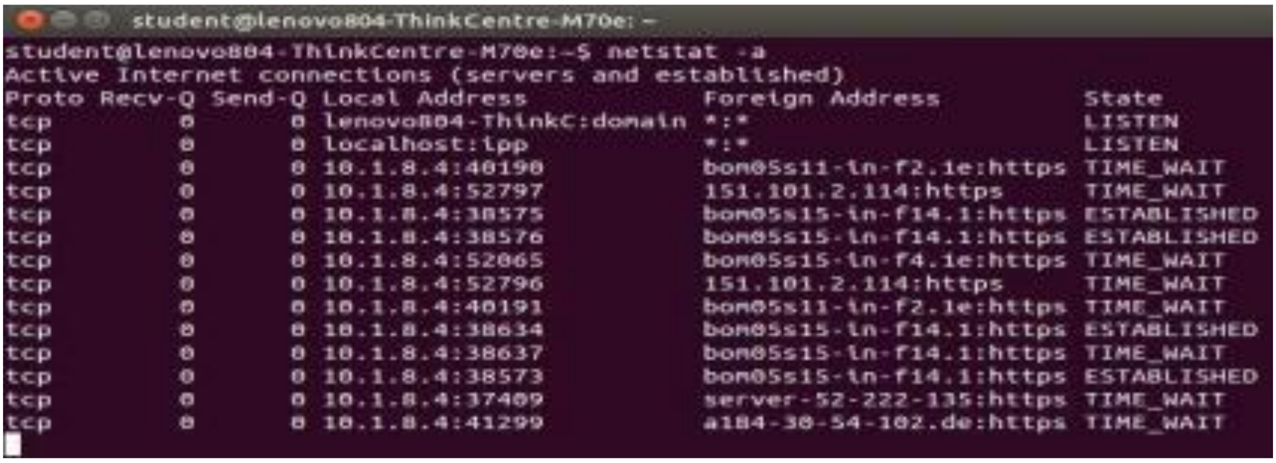
traceroute command in Linux prints the route that a packet takes to reach the host. This command is useful when you want to know about the route and about all the hops that a packet takes. Below image depicts how traceroute command is used to reach the Google(172.217.26.206) host from the local machine and it also prints detail about all the hops that it visits in between.



```
student@lenovo804-ThinkCentre-M70e:~$ traceroute
Usage:
  traceroute [ -46dFITnreAUDV ] [ -f first_ttl ] [ -g gate,... ] [ -i device ] [ -m max_ttl ] [ -N squeries ] [ -p port ] [ -t tos ] [ -l flow_label ] [ -w waittime ] [ -q nqueries ] [ -s src_addr ] [ -z sendwait ] [
  --fwmark=num ] host [ packetlen ]
Options:
  -4                          Use IPv4
  -6                          Use IPv6
  -d  --debug                 Enable socket level debugging
  -F  --dont-fragment         Do not fragment packets
  -f first_ttl  --first=first_ttl
                              Start from the first_ttl hop (instead from 1)
  -g gate,...  --gateway=gate,...
                              Route packets through the specified gateway
                              (maximum 8 for IPv4 and 127 for IPv6)
  -I  --icmp                  Use ICMP ECHO for tracerouting
  -T  --tcp                   Use TCP SYN for tracerouting (default port is 80)
  -i device  --interface=device
                              Specify a network interface to operate with
  -m max_ttl  --max-hops=max_ttl
                              Set the max number of hops (max TTL to be
                              reached). Default is 30
  -N squeries  --sim-queries=squeries
                              Set the number of probes to be tried
                              simultaneously (default is 16)
  -n                          Do not resolve IP addresses to their domain names
  -p port  --port=port        Set the destination port to use. It is either
                              initial udp port value for "default" method
                              (incremented by each probe, default is 33434), or
                              initial seq for "icmp" (incremented as well,
                              default from 1), or some constant destination
                              port for other methods (with default of 80 for
                              "tcp", 53 for "udp", etc.)
  -t tos  --tos=tos           Set the TOS (IPv4 type of service) or TC (IPv6
                              traffic class) value for outgoing packets
  -l flow_label  --flowlabel=flow_label
                              Use specified flow_label for IPv6 packets
  -w waittime  --wait=waittime
                              Set the number of seconds to wait for response to
                              a probe (default is 5.0). Non-integer (float
                              point) values allowed too
  -q nqueries  --queries=nqueries
                              Set the number of probes per each hop. Default is
                              3
  -r                          Bypass the normal routing and send directly to a
                              host on an attached network
  -s src_addr  --source=src_addr
                              Use source src_addr for outgoing packets
  -z sendwait  --sendwait=sendwait
                              Minimal time interval between probes (default 0).
                              If the value is more than 10, then it specifies a
                              number in milliseconds, else it is a number of
                              seconds (float point values allowed too)
  -e  --extensions            Show ICMP extensions (if present), including MPLS
  -A  --as-path-lookups       Perform AS path lookups in routing registries and
                              print results directly after the corresponding
                              addresses
  -M name  --module=name      Use specified module (either builtin or external)
```

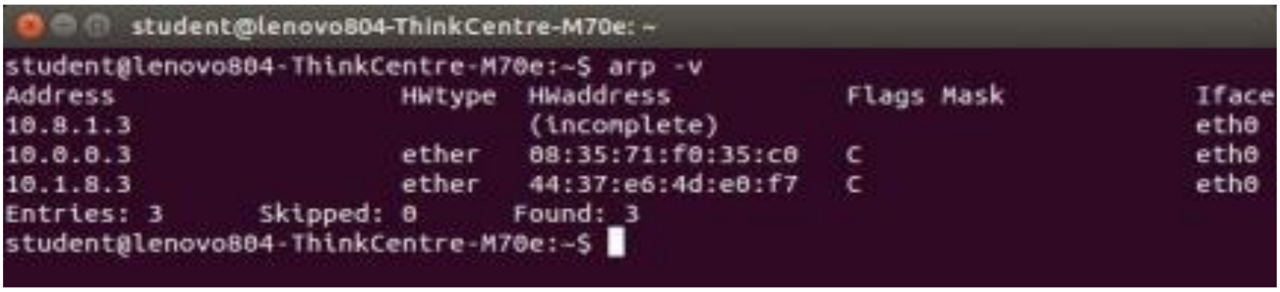
5. Netstat

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



```
student@lenovo804-ThinkCentre-M70e:~$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 lenovo804-ThinkC:domain *:*                     LISTEN
tcp        0      0 localhost:ipp           *:*                     LISTEN
tcp        0      0 10.1.8.4:40190          bom05s11-in-f2.1e:https TIME_WAIT
tcp        0      0 10.1.8.4:52797          151.101.2.114:https     TIME_WAIT
tcp        0      0 10.1.8.4:38575          bom05s15-in-f14.1:https ESTABLISHED
tcp        0      0 10.1.8.4:38576          bom05s15-in-f14.1:https ESTABLISHED
tcp        0      0 10.1.8.4:52065          bom05s15-in-f4.1e:https TIME_WAIT
tcp        0      0 10.1.8.4:52796          151.101.2.114:https     TIME_WAIT
tcp        0      0 10.1.8.4:40191          bom05s11-in-f2.1e:https TIME_WAIT
tcp        0      0 10.1.8.4:38634          bom05s15-in-f14.1:https ESTABLISHED
tcp        0      0 10.1.8.4:38637          bom05s15-in-f14.1:https TIME_WAIT
tcp        0      0 10.1.8.4:38573          bom05s15-in-f14.1:https ESTABLISHED
tcp        0      0 10.1.8.4:37409          server-52-222-135:https TIME_WAIT
tcp        0      0 10.1.8.4:41299          a184-30-54-102.de:https TIME_WAIT
```

6. ARP

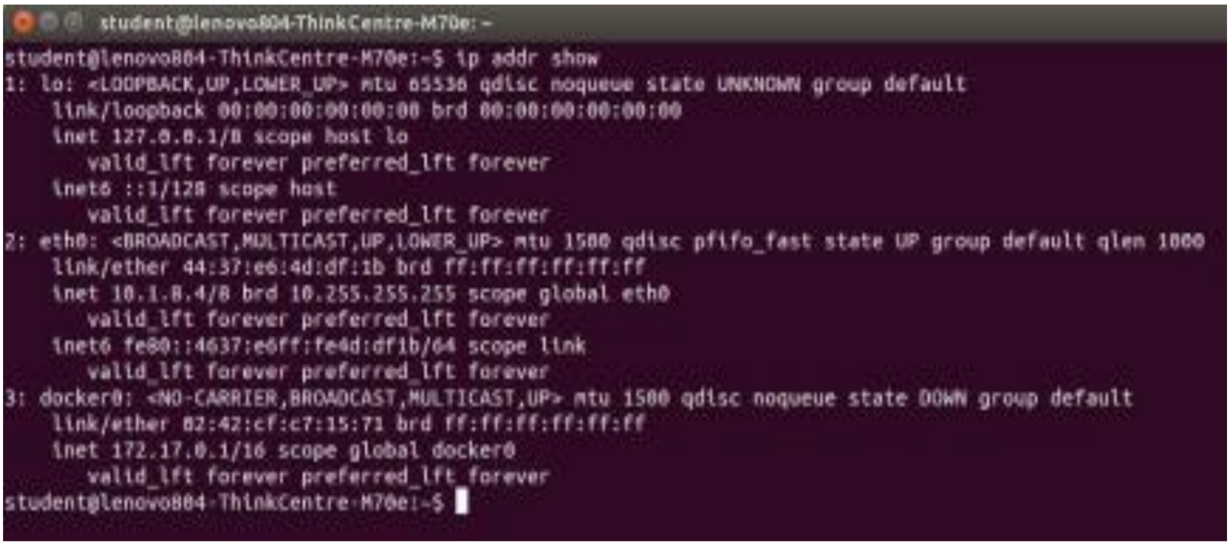


```
student@lenovo804-ThinkCentre-M70e:~$ arp -v
Address                  HWtype  HWaddress           Flags Mask            Iface
10.8.1.3                         (incomplete)                              eth0
10.0.0.3                 ether   08:35:71:f0:35:c0   C                     eth0
10.1.8.3                 ether   44:37:e6:4d:e0:f7   C                     eth0
Entries: 3      Skipped: 0      Found: 3
student@lenovo804-ThinkCentre-M70e:~$
```

ARP command manipulates the System’s ARP cache. It also allows a complete dump of the ARP cache. ARP stands for Address Resolution Protocol. The primary function of this protocol is to resolve the IP address of a system to its mac address, and hence it works between level 2(Data link layer) and level 3(Network layer).

7. IP

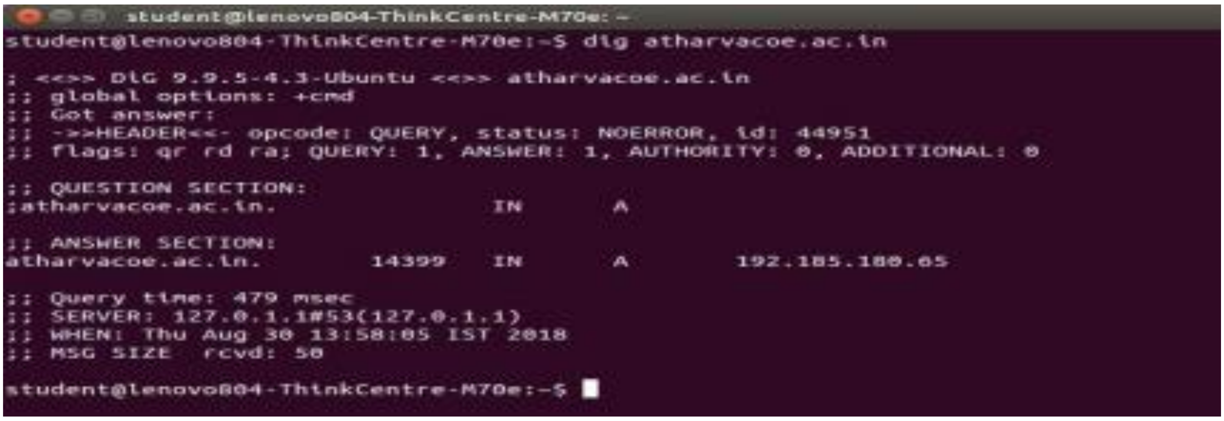
ip command in Linux is present in the net-tools which is used for performing several network administration tasks. IP stands for Internet Protocol. This command is used to show or manipulate routing, devices, and tunnels. It is similar
to *ifconfig* command but it is much more powerful with more functions and facilities attached to it. *ifconfig* is one of the deprecated commands in the net-tools of Linux that has not been maintained for many years. ip command is used to perform several tasks like assigning an address to a network interface or configuring network interface parameters.
It can perform several other tasks like configuring and modifying the default and static routing, setting up tunnel over IP, listing IP addresses and property information,
modifying the status of the interface, assigning, deleting and setting up IP addresses and routes.



```
student@lenovo804-ThinkCentre-M70e: ~
student@lenovo804-ThinkCentre-M70e:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 44:37:e6:4d:df:1b brd ff:ff:ff:ff:ff:ff
    inet 10.1.8.4/8 brd 10.255.255.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::4637:e6ff:fe4d:df1b/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:cf:c7:15:71 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 scope global docker0
       valid_lft forever preferred_lft forever
student@lenovo804-ThinkCentre-M70e:~$ 
```

8. Dig

dig command stands for ***Domain Information Groper***. It is used for retrieving information about DNS name servers. It is basically used by network
administrators. It is used for verifying and troubleshooting DNS problems and to perform DNS lookups. Dig command replaces older tools such as nslookup and the host.



```
student@lenovo804-ThinkCentre-M70e: ~
student@lenovo804-ThinkCentre-M70e:~$ dig atharvacoe.ac.in

; <<>> DiG 9.9.5-4.3-Ubuntu <<>> atharvacoe.ac.in
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44951
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;atharvacoe.ac.in.              IN      A

;; ANSWER SECTION:
atharvacoe.ac.in.       14399   IN      A       192.185.180.65

;; Query time: 479 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Thu Aug 30 13:58:05 IST 2018
;; MSG SIZE  rcvd: 50

student@lenovo804-ThinkCentre-M70e:~$ 
```

CONCLUSION: Hence, in this experiment, we have successfully studied some important networking command and also implemented them in Linux

Experiment: 02

Aim: Use Wireshark to understand the operation of TCP/IP layers :
Ethernet Layer : Frame header, ● Frame size etc.
● Data Link Layer : MAC address, ARP (IP and MAC address binding)
● Network Layer : IP Packet (header, fragmentation), ICMP (Query and Echo)
● Transport Layer: TCP Ports, TCP handshake segments etc.
● Application Layer: DHCP, FTP, HTTP header formats

Description: Wireshark is an open source tool for profiling network traffic and analyzing packets. Such a tool is often referred to as a network analyzer, network protocol analyzer or sniffer.
Wireshark, formerly known as Ethereal, can be used to examine the details of traffic at a variety of levels ranging from connection-level information to the bits that make up a single packet.Packet capture can provide a network administrator with information about individual packets such as transmit time, source, destination, protocol type and header data. This information can be useful for evaluating security events and troubleshooting network security device issues.
Wireshark will typically display information in three panels. The top panel lists frames individually with key data on a single line. Any single frame selected in the top pane is further explained in the tool's middle panel. In this section of the display, Wireshark shows packet details, illustrating how various aspects of the frame can be understood as belonging to the data link layer, network layer, transport layer or application layer. Finally, Wireshark's bottom panel displays the raw frame, with a hexadecimal rendition on the left and the corresponding ASCII values on the right.
Wireshark has a rich feature set which includes the following:

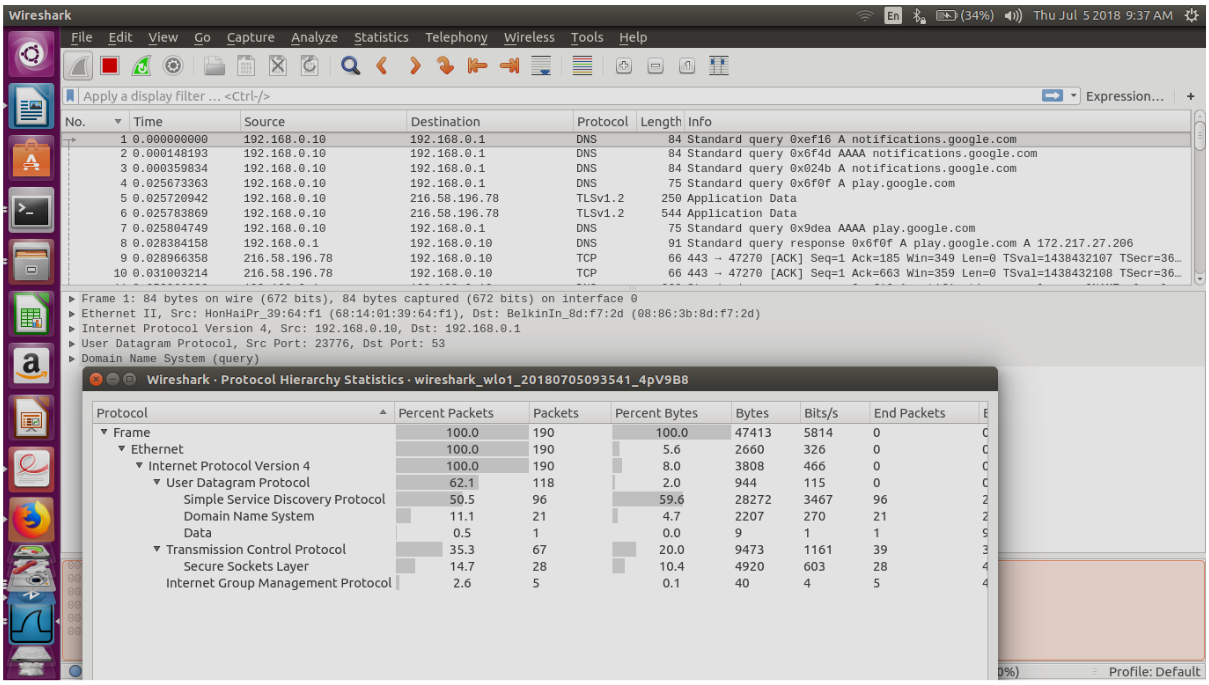
- Deep inspection of hundreds of protocols, with more being added all the time
- Live capture and offline analysis
- Standard three-pane packet browser
- Multi-platform: Runs on Windows, Linux, OS X, Solaris, FreeBSD, NetBSD, and many others
- Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility
- The most powerful display filters in the industry
- Rich VoIP analysis
- Read/write many different capture file formats: tcpdump (libpcap), Pcap NG, Catapult DCT2000, Cisco Secure IDS iplog, Microsoft Network Monitor, Network General Sniffer® (compressed and uncompressed), Sniffer® Pro, and NetXray®, Network Instruments Observer, NetScreen snoop, Novell LANalyzer, RADCOM WAN/LAN Analyzer, Shomiti/Finisar Surveyor, Tektronix K12xx, Visual Networks Visual UpTime, WildPackets EtherPeek/TokenPeek/AiroPeek, and many others
- Capture files compressed with gzip can be decompressed on the fly
- Live data can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth,

USB, Token Ring, Frame Relay, FDDI, and others (depending on your platform)

- Decryption support for many protocols, including IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2
- Coloring rules can be applied to the packet list for quick, intuitive analysis
- Output can be exported to XML, PostScript®, CSV, or plain text

Installation

Sudo apt-get install wireshark

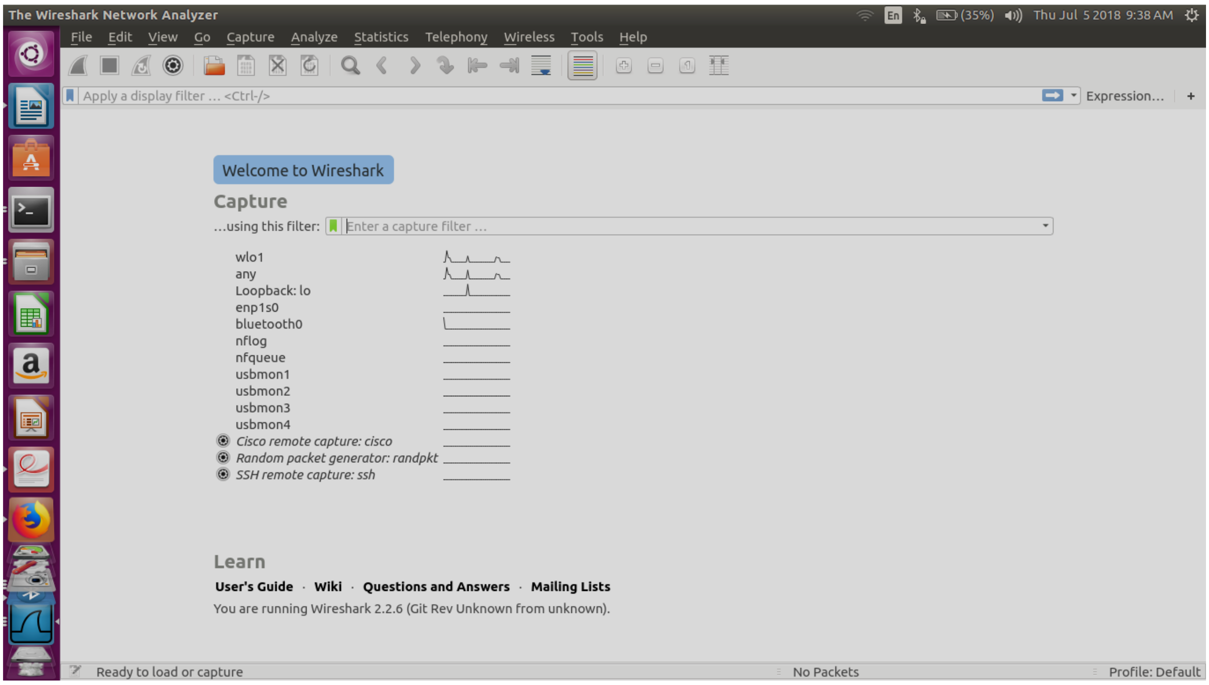

Wireshark
Thu Jul 5 2018 9:37 AM
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…

```
No.   Time         Source          Destination     Protocol Length Info
    1 0.000000000  192.168.0.10    192.168.0.1     DNS          84 Standard query 0xef16 A notifications.google.com
    2 0.000148193  192.168.0.10    192.168.0.1     DNS          84 Standard query 0x6f4d AAAA notifications.google.com
    3 0.000359834  192.168.0.10    192.168.0.1     DNS          84 Standard query 0x024b A notifications.google.com
    4 0.025673363  192.168.0.10    192.168.0.1     DNS          75 Standard query 0x6f0f A play.google.com
    5 0.025720942  192.168.0.10    216.58.196.78   TLSv1.2     250 Application Data
    6 0.025783869  192.168.0.10    216.58.196.78   TLSv1.2     544 Application Data
    7 0.025804749  192.168.0.10    192.168.0.1     DNS          75 Standard query 0x9dea AAAA play.google.com
    8 0.028384158  192.168.0.1     192.168.0.10    DNS          91 Standard query response 0x6f0f A play.google.com A 172.217.27.206
    9 0.028966358  216.58.196.78   192.168.0.10    TCP          66 443 → 47270 [ACK] Seq=1 Ack=185 Win=349 Len=0 TSval=1438432107 TSecr=36…
   10 0.031003214  216.58.196.78   192.168.0.10    TCP          66 443 → 47270 [ACK] Seq=1 Ack=663 Win=359 Len=0 TSval=1438432108 TSecr=36…
```

```
▶ Frame 1: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface 0
▶ Ethernet II, Src: HonHaiPr_39:64:f1 (68:14:01:39:64:f1), Dst: BelkinIn_8d:f7:2d (08:86:3b:8d:f7:2d)
▶ Internet Protocol Version 4, Src: 192.168.0.10, Dst: 192.168.0.1
▶ User Datagram Protocol, Src Port: 23776, Dst Port: 53
▶ Domain Name System (query)
```

Wireshark · Protocol Hierarchy Statistics · wireshark_wlo1_20180705093541_4pV9B8

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets
▼ Frame	100.0	190	100.0	47413	5814	0
▼ Ethernet	100.0	190	5.6	2660	326	0
▼ Internet Protocol Version 4	100.0	190	8.0	3808	466	0
▼ User Datagram Protocol	62.1	118	2.0	944	115	0
Simple Service Discovery Protocol	50.5	96	59.6	28272	3467	96
Domain Name System	11.1	21	4.7	2207	270	21
Data	0.5	1	0.0	9	1	1
▼ Transmission Control Protocol	35.3	67	20.0	9473	1161	39
Secure Sockets Layer	14.7	28	10.4	4920	603	28
Internet Group Management Protocol	2.6	5	0.1	40	4	5


The Wireshark Network Analyzer
Thu Jul 5 2018 9:38 AM
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…
Welcome to Wireshark
Capture

…using this filter: Enter a capture filter …

wlo1
any
Loopback: lo
enp1s0
bluetooth0
nflog
nfqueue
usbmon1
usbmon2
usbmon3
usbmon4
Cisco remote capture: cisco
Random packet generator: randpkt
SSH remote capture: ssh

Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists

You are running Wireshark 2.2.6 (Git Rev Unknown from unknown).

Ready to load or capture

No Packets

Packet List

The packet list pane, located at the top of the window, shows all packets found in the active capture
file. Each packet has its own row and corresponding number assigned to it, along with each of these data points.

- Time: The timestamp of when the packet was captured is displayed in this column, with the default format being the number of seconds (or partial seconds) since this specific capture file was first created. To modify this format to something that may be a bit more useful, such as the actual time of day, select the Time Display Format option from Wireshark's View menu - located at the top of the main interface.
- Source: This column contains the address (IP or other) where the packet originated.
- Destination: This column contains the address that the packet is being sent to.
- Protocol: The packet's protocol name (i.e., TCP) can be found in this column.
- Length: The packet length, in bytes, is displayed in this column.
- Info: Additional details about the packet are presented here. The contents of this column can vary greatly depending on packet contents.

When a packet is selected in the top pane, you may notice one or more symbols appear in the first
column. Open and/or closed brackets, as well as a straight horizontal line, can indicate whether or
not a packet or group of packets are all part of the same back-and-forth conversation on the network. A broken horizontal line signifies that a packet is not part of said conversation.

Some filters :

ip.src == 10.0.0.5
ip.src != 10.0.0.5
frame.len > 10
frame.len < 128
sip.To contains "a1762"
ip.src==192.168.5.63&&192.168.10.5
arp
dns
http
tcp
udp
tcp.port == 80 || tcp.port == 443 || tcp.port == 8080

Conclusion:

Studied Wireshark as a data analyser in TCP/IP layers.

Experiment: 03

AIM: Build a simple network topology and configure it for static routing protocol using packet tracer.

THEORY: Cisco Packet Tracer is a cross-platform visual simulation tool designed by Cisco Systems that allows users to create network topologies and imitate modern computer networks. The software allows users to simulate the configuration of Cisco routers and switches using a simulated command line interface. Packet Tracer makes use of a drag and drop user interface, allowing users to add and remove simulated network devices as they see fit. The software is mainly focused towards Certified Cisco Network Associate Academy students as an educational tool for helping them learn fundamental CCNA concepts.

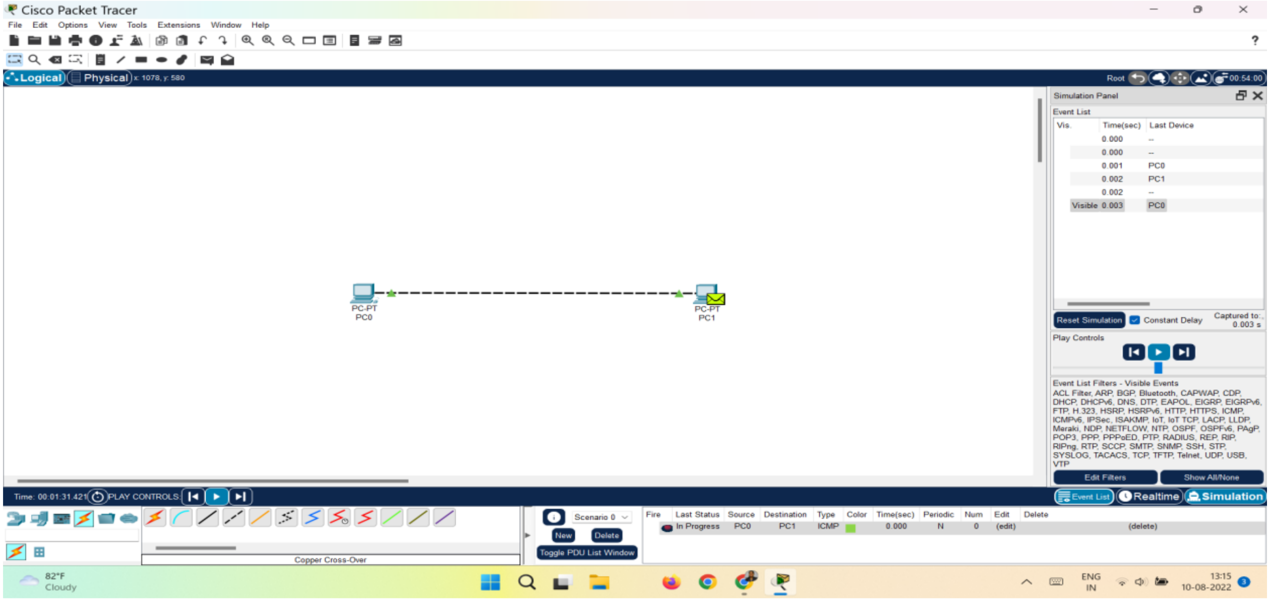
To install cisco packet tracer-

1.Go to www.netacad.com
2.Sign up and then login
3.Go to resources
4.Go to packet tracer (https://www.netacad.com/portal/resources/packet-tracer)
5.Download packet tracer with respect to OS and processor
6.Agree to all terms and conditions and proceed.
7.Packet tracer environment is ready to use.

1.To create peer to peer network

1.Select two pcs
2.Select cable to connect
3.Set up ip address for the pcs
4.Now check for connection by performing ping operation.

Connect the devices as shown below:



Vis.	Time(sec)	Last Device
	0.000	--
	0.000	--
	0.001	PC0
	0.002	PC1
	0.002	--
Visible	0.003	PC0

☑ Constant Delay   Captured to: 0.003 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, 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

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	In Progress	PC0	PC1	ICMP	🟩	0.000	N	0	(edit)	(delete)

2.To set up network using Hub

1.Select hub

2.Select 6 pcs(hub has 6 max ports)

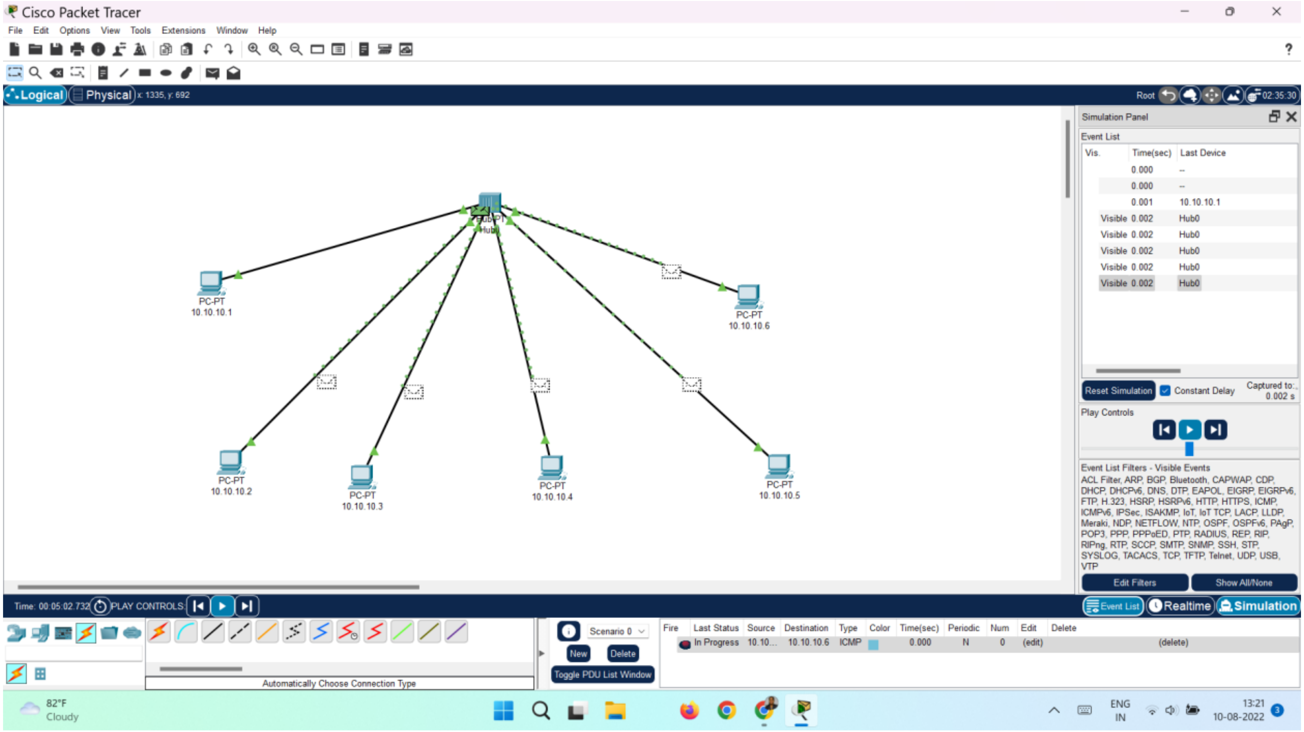
3.Connect with cable.

4.Set up ip address for the pcs

5.Now check for connection by performing ping operation

6.Use simulation to check packet transmission.

Connect the devices as shown below:



Simulation Panel

Event List

Vis.	Time(sec)	Last Device
	0.000	--
	0.000	--
	0.001	10.10.10.1
Visible	0.002	Hub0
Visible	0.002	Hub0
Visible	0.002	Hub0
Visible	0.002	Hub0
Visible	0.002	Hub0

Reset Simulation ☑ Constant Delay Captured to: 0.002 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	Delete
●	In Progress	10.10...	10.10.10.6	ICMP	🟦	0.000	N	0	(edit)	(delete)

3.To set up network in LAN using switch

1.Select switch

2.Select 5 pcs(Switch has 24 max ports and 2 gigabit ports)

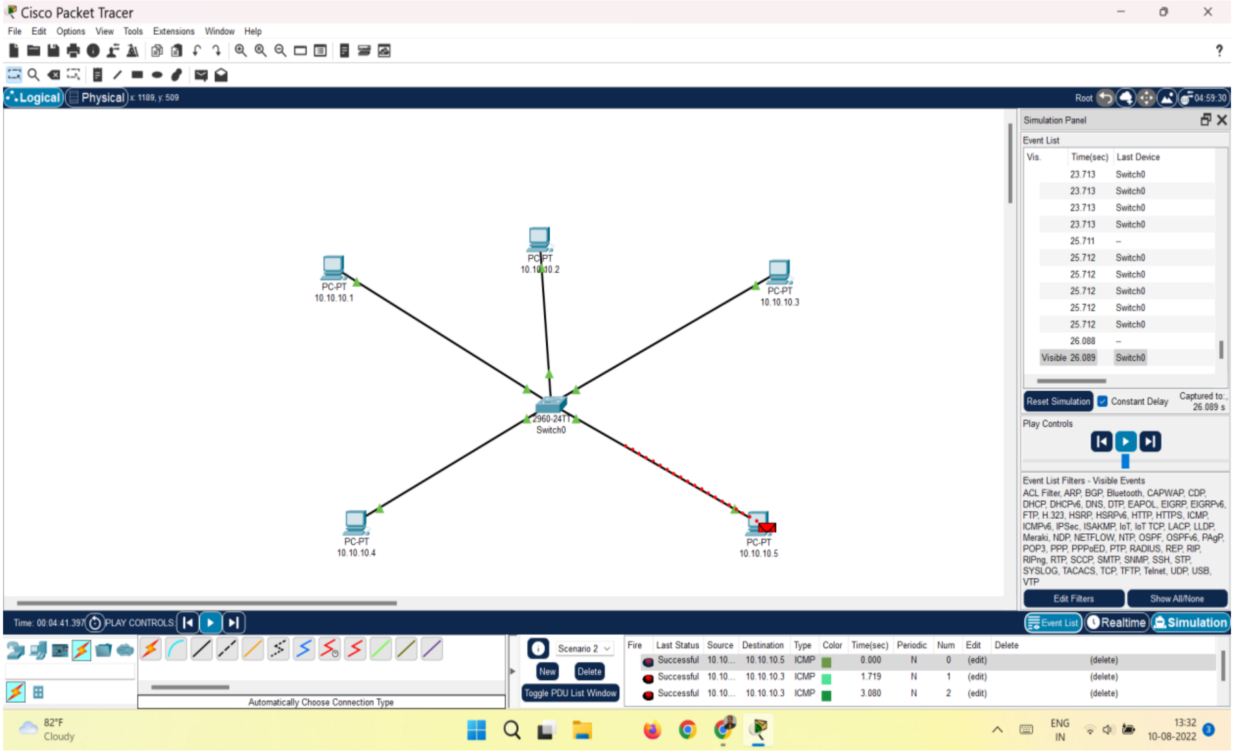
3.Connect with cable.

4.Set up IP address for the pcs

5.Now check for connection by performing ping operation

6.Use simulation to check packet transmission.

Connect the devices as shown below:



Vis.	Time(sec)	Last Device
	23.713	Switch0
	23.713	Switch0
	23.713	Switch0
	23.713	Switch0
	25.711	--
	25.712	Switch0
	25.712	Switch0
	25.712	Switch0
	25.712	Switch0
	25.712	Switch0
	26.088	--
Visible	26.089	Switch0

☑ Constant Delay

Captured to:, 26.089 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, 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

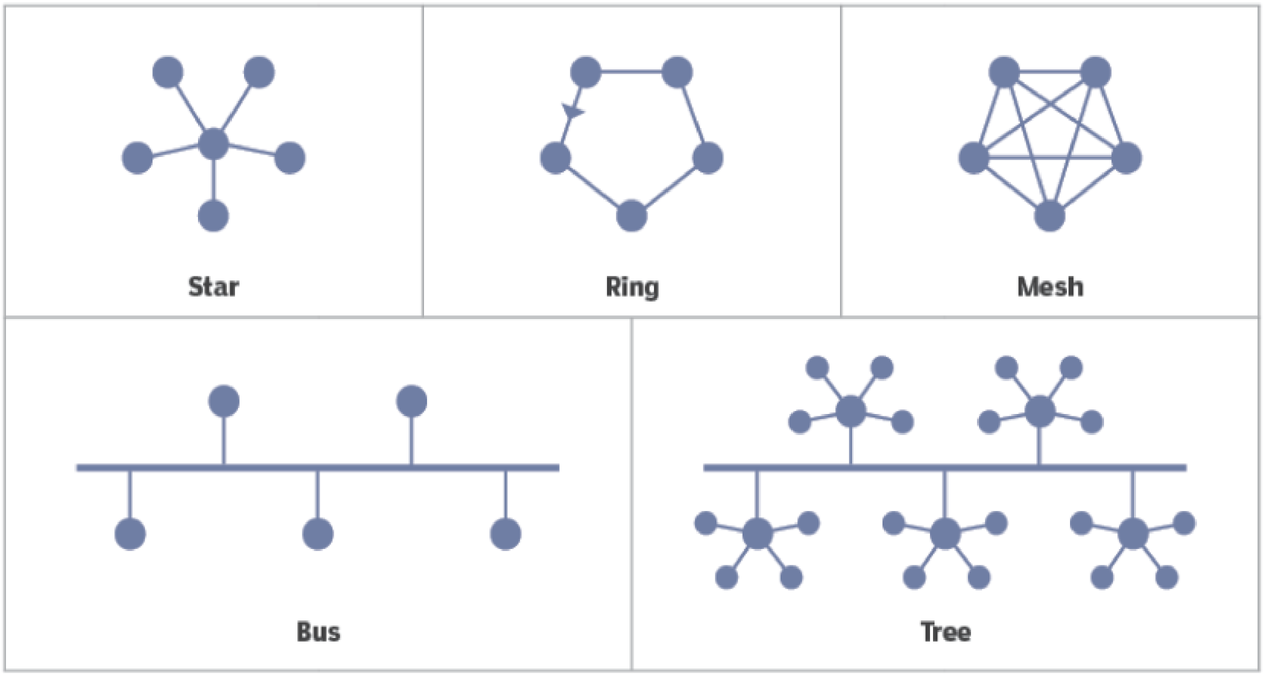
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	10.10...	10.10.10.5	ICMP	🟩	0.000	N	0	(edit)	(delete)
●	Successful	10.10...	10.10.10.3	ICMP	🟩	1.719	N	1	(edit)	(delete)
●	Successful	10.10...	10.10.10.3	ICMP	🟩	3.080	N	2	(edit)	(delete)

CONCLUSION: Hence we have successfully created peer to peer network and used hub and switch devices to create network using CISCO PACKET TRACER

Experiment: 04

AIM: Create a network to show different network Topologies (Bus, ring, star, mesh, tree, hybrid topologies).

THEORY:
A network topology is the physical and logical arrangement of nodes and connections in a network. Nodes usually include devices such as switches, routers and software with switch and router features. Network topologies are often represented as a graph.
Network topologies describe the arrangement of networks and the relative location of traffic flows. Administrators can use network topology diagrams to determine the best placements for each node and the optimal path for traffic flow. With a well-defined and planned-out network topology, an organization can more easily locate faults and fix issues, improving its data transfer efficiency.
Following are the types of topologies:
1.Bus Topology
2.Ring Topology
3.Star Topology
4.Mesh Topology
5.Hybrid Topology
6.Tree topology



CONCLUSION: In this experiment we have created different networks using different topologies and checked packet transmission using simulation in Cisco packet tracer

Experiment: 05

Aim: a. Using netstat and route commands of Linux, do the following:

● View current routing table

● Add and delete routes

● Change default gateway

b. Perform packet filtering by enabling IP forwarding using IPtables in Linux.

Description: Routing is the transfer of an IP packet from one point to another across the network. When you send someone an email, you're actually transmitting a series of IP packets or datagrams from your system to the other person's computer. The packets sent from your computer pass through several gateways or routers to get to the destination computer system. The same is true for all Internet protocols such as HTTP, IRC, FTP, etc.

In all Linux and UNIX systems, the information about how the IP packets should be routed is stored in a kernel structure. These structures are called routing tables. If you want your system to communicate with other computers, you may want to configure these routing tables. First, it is important to know how to view these routing tables on your Linux system.

To view the routing tables in Ubuntu using the following three common commands:

▪ The netstat command

▪ The route commands

▪ The ip route command

Method 1: Through the netstat command

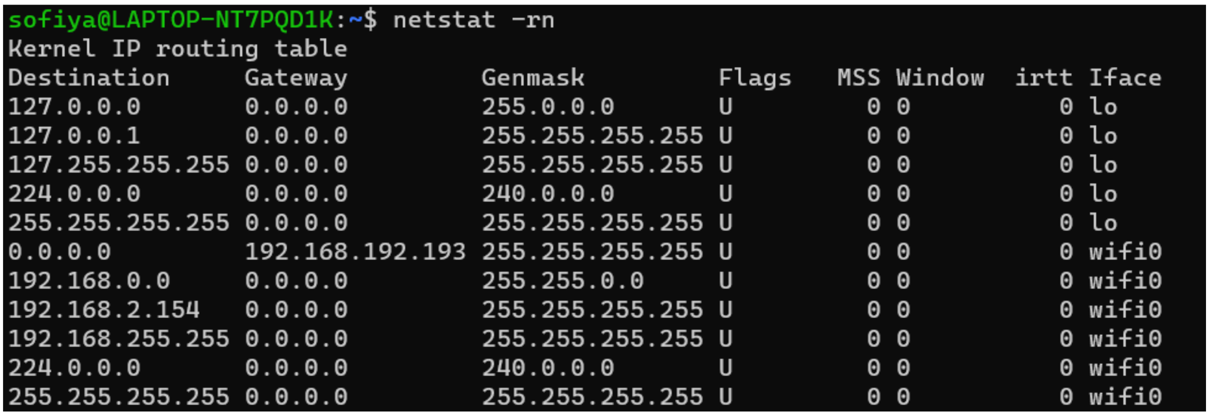
The netstat command has always been a widely used method of printing routing table information in Linux. However, it is officially replaced by the ip route command. We are including it anyway as it is still an approach to retrieve the required information.

Here is how you can use this command:

$ netstat -rn

-r This flag is used to display the Kernel routing tables

-n This flag is used to display the numerical addresses



```
sofiya@LAPTOP-NT7PQD1K:~$ netstat -rn
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         192.168.192.193 255.255.255.255 U         0 0          0 wifi0
192.168.0.0     0.0.0.0         255.255.0.0     U         0 0          0 wifi0
192.168.2.154   0.0.0.0         255.255.255.255 U         0 0          0 wifi0
192.168.255.255 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
```

This is what the output indicates: Destination	This column indicates the destination network.
Gateway	This column indicates the defined gateway for the network. If you see an * in this column, it means that no forwarding gateway is needed for the specified network.
Genmask	This column indicates the netmask of the network.
Flags	The U output in this columns means that the route is up. The G output indicates that a specified gateway should be used for this route. D stands for dynamically installed, M stands for modified, and R means reinstated. The H flag indicates that the destination is a fully qualified host address, rather than a network.
MSS	This column indicates the default Maximum Segment Size(MSS) for TCP connections for this route.
Window	This column indicates the default window size for TCP connections over this route.
Irtt	This column indicates the Initial Round Trip Time for this route.
Iface	The Iface column shows the network interface. If you had more than one interface, you would see *lo* (for loopback), *eth0* (first Ethernet device), and *eth1* (for the second Ethernet device), and so on for the number of interfaces, you have installed.
matric	The metric field has a number of different meanings: The Metric field indicates the cost of a route. If multiple routes exist to a given destination network ID, the metric is used to decide which route is to be taken. The route with the lowest metric is the preferred route.

Method 2: Through the route command

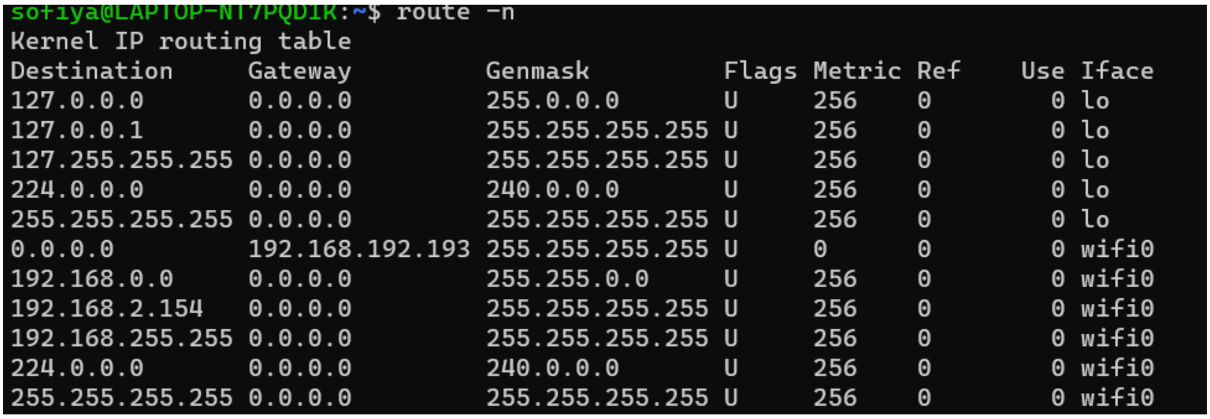
The route command also falls under the category of once widely used but now obsolete command to view routing tables. The manual page of this command also mentions that the command is now replaced by the ip route command.

Through this command, you can view exactly the same information that you could, through the netstat command. Here is how you can use it:

$ route -n

-n This flag is used to display the numerical addresses only

Method

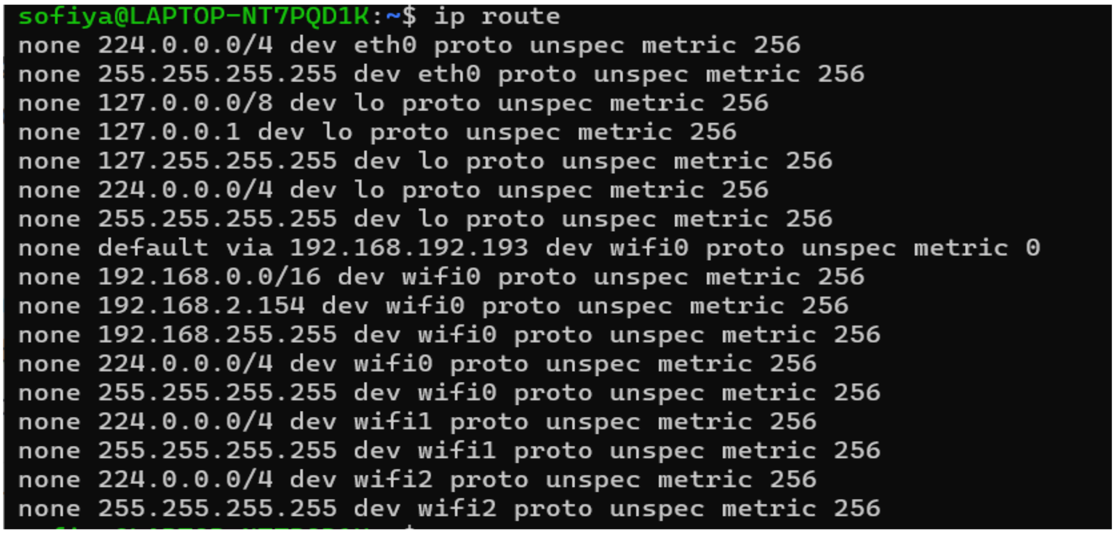


```
sofiya@LAPTOP-NT7PQD1K:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
127.0.0.0       0.0.0.0         255.0.0.0       U     256    0        0 lo
127.0.0.1       0.0.0.0         255.255.255.255 U     256    0        0 lo
127.255.255.255 0.0.0.0         255.255.255.255 U     256    0        0 lo
224.0.0.0       0.0.0.0         240.0.0.0       U     256    0        0 lo
255.255.255.255 0.0.0.0         255.255.255.255 U     256    0        0 lo
0.0.0.0         192.168.192.193 255.255.255.255 U     0      0        0 wifi0
192.168.0.0     0.0.0.0         255.255.0.0     U     256    0        0 wifi0
192.168.2.154   0.0.0.0         255.255.255.255 U     256    0        0 wifi0
192.168.255.255 0.0.0.0         255.255.255.255 U     256    0        0 wifi0
224.0.0.0       0.0.0.0         240.0.0.0       U     256    0        0 wifi0
255.255.255.255 0.0.0.0         255.255.255.255 U     256    0        0 wifi0
```

Method 3: Through the ip route command

Last but not least, here is the most recommended way of printing routing table information in Linux. Here is how to use this command:

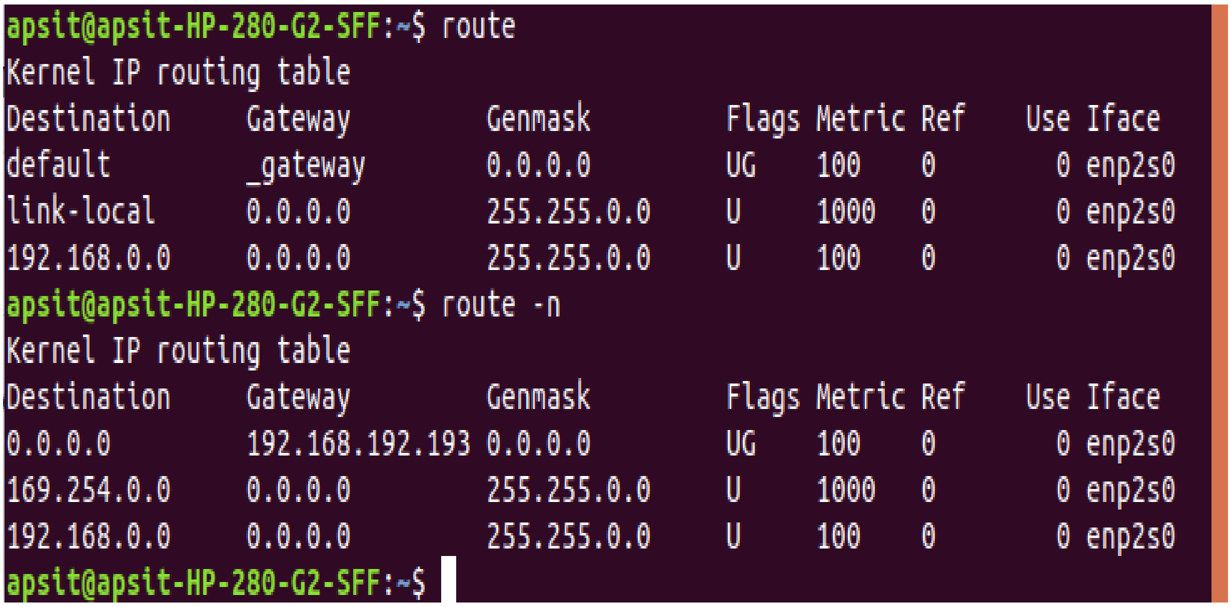
$ ip route



```
sofiya@LAPTOP-NT7PQD1K:~$ ip route
none 224.0.0.0/4 dev eth0 proto unspec metric 256
none 255.255.255.255 dev eth0 proto unspec metric 256
none 127.0.0.0/8 dev lo proto unspec metric 256
none 127.0.0.1 dev lo proto unspec metric 256
none 127.255.255.255 dev lo proto unspec metric 256
none 224.0.0.0/4 dev lo proto unspec metric 256
none 255.255.255.255 dev lo proto unspec metric 256
none default via 192.168.192.193 dev wifi0 proto unspec metric 0
none 192.168.0.0/16 dev wifi0 proto unspec metric 256
none 192.168.2.154 dev wifi0 proto unspec metric 256
none 192.168.255.255 dev wifi0 proto unspec metric 256
none 224.0.0.0/4 dev wifi0 proto unspec metric 256
none 255.255.255.255 dev wifi0 proto unspec metric 256
none 224.0.0.0/4 dev wifi1 proto unspec metric 256
none 255.255.255.255 dev wifi1 proto unspec metric 256
none 224.0.0.0/4 dev wifi2 proto unspec metric 256
none 255.255.255.255 dev wifi2 proto unspec metric 256
```

Though this information is not much reader-friendly as that of the previously mentioned commands, it is still enough for you to configure the router.
To add route in table—

First we will check current routing table



```
apsit@apsit-HP-280-G2-SFF:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 enp2s0
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 enp2s0
192.168.0.0     0.0.0.0         255.255.0.0     U     100    0        0 enp2s0
apsit@apsit-HP-280-G2-SFF:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.192.193 0.0.0.0         UG    100    0        0 enp2s0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 enp2s0
192.168.0.0     0.0.0.0         255.255.0.0     U     100    0        0 enp2s0
apsit@apsit-HP-280-G2-SFF:~$
```

Adding a static route using IP command

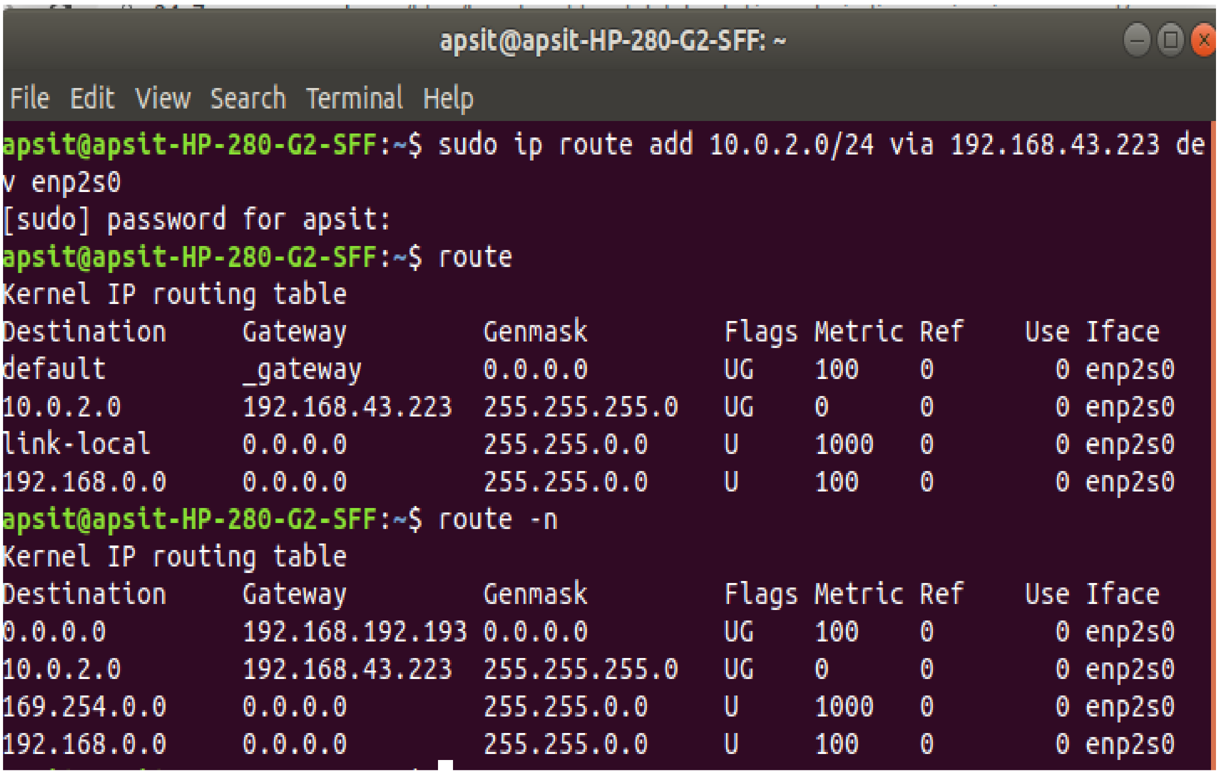
Suppose you want to take a backup of a Linux machine and push the backup file to another backup server in the subnet **10.0.2.0/24**. However, for one reason or the other, you cannot reach the backup server via the default gateway. In this case, you will have to create a new route for the backup server subnet via another IP, say **192.168.43.223** via the interface **enp0s3**.

The command for this will be

$ sudo IP route add 10.0.2.0 via 192.168.43.223 dev enp2s0

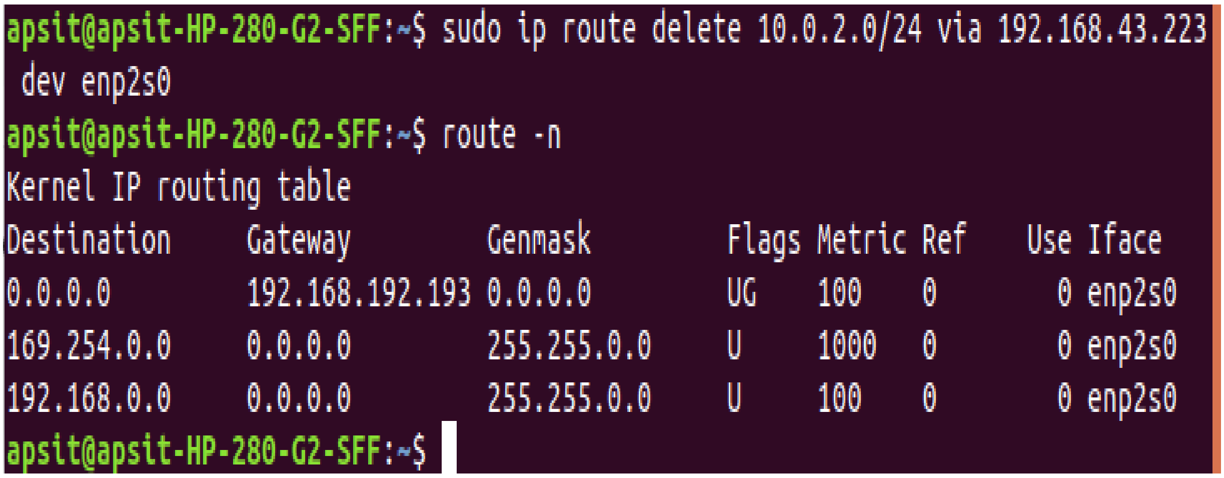
Where:

- 10.0.2.0 -> is the network you want to connect to
- /24 -> is the subnet mask
- 192.168.43.223 -> is the IP through which we will reach the server
- enp2s0 -> is the network interface


apsit@apsit-HP-280-G2-SFF: ~
File Edit View Search Terminal Help

```
apsit@apsit-HP-280-G2-SFF:~$ sudo ip route add 10.0.2.0/24 via 192.168.43.223 de
v enp2s0
[sudo] password for apsit:
apsit@apsit-HP-280-G2-SFF:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 enp2s0
10.0.2.0        192.168.43.223  255.255.255.0   UG    0      0        0 enp2s0
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 enp2s0
192.168.0.0     0.0.0.0         255.255.0.0     U     100    0        0 enp2s0
apsit@apsit-HP-280-G2-SFF:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.192.193 0.0.0.0         UG    100    0        0 enp2s0
10.0.2.0        192.168.43.223  255.255.255.0   UG    0      0        0 enp2s0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 enp2s0
192.168.0.0     0.0.0.0         255.255.0.0     U     100    0        0 enp2s0
```

Deleting a static route using IP command



```
apsit@apsit-HP-280-G2-SFF:~$ sudo ip route delete 10.0.2.0/24 via 192.168.43.223
 dev enp2s0
apsit@apsit-HP-280-G2-SFF:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.192.193 0.0.0.0         UG    100    0        0 enp2s0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 enp2s0
192.168.0.0     0.0.0.0         255.255.0.0     U     100    0        0 enp2s0
apsit@apsit-HP-280-G2-SFF:~$
```

To set default gateway-

ip command to set a default router to 192.168.1.254

Login as the root and type:

```
# ip route add default via 192.168.1.254
OR
$ sudo ip route add default via 192.168.1.254
```

route command to set a default router to 192.168.1.254

Login as the root and type:

```
# route add default gw 192.168.1.254
OR
$ sudo route add default gw 192.168.1.254
```

To perform packet filtering by enabling IP forwarding using IP tables in Linux

A user-space Unix utility that gives system administrators the ability to configure IP packet filtering rules implemented by the Kernel's net filter module. IP tables act as a firewall using packet filtering rules based on various criteria such as IP address, port, and protocols.

Iptables come pre-installed on Ubuntu and most Debian based distributions. Ubuntu also packages GUFW firewall, a graphical alternative you can use to work with iptables.

The Filter Tables

The filter table is a default table that contains chains used for network packet filtering. Some of the default chains in this table include:

Input	Iptables use this chain for any incoming packets to the system, i.e., packets going to local network sockets.
Output	Iptables use the output chain for locally generated packets, i.e., packets going out of the system.
Forward	This chain is what the Iptables use for packets routed or forwarded via the system.

1.To check current iptable rule-

```
iptables -L
```

2.To drop forward chain-(It drops packets for router)

```
sudo iptables -P FORWARD DROP
```

3.To drop packets incoming from specific ip address-

```
iptables -A INPUT -s 192.168.0.23 -j DROP
```

4.Consider the command below:To drop packets incoming from specific ip address-

```
sudo iptables -I INPUT -s 192.168.0.24 -j DROP
```

The command above tells the iptables to create a rule in the chain. The rule drops all the packets from the IP address 192.168.0.24.

Let us examine the command, line by line, to understand it better.

- The first command iptables calls the iptables command-line utility.
- Next is -I argument used for insertion. The insertion argument adds a rule at the beginning of the iptables chain and thus gets assigned a higher priority. To add a rule at a specific number in the chain, use the -I argument followed by the number where the rule should get assigned.
- The -s argument helps specify the source. Hence, we use the -s argument followed by the IP address.
- The -j parameter with iptables specifies the jump to a specific target. This option sets the action the Iptables shall perform once there’s a matching packet. Iptables offers four main targets by default, these include: ACCEPT, DROP, LOG(Use –log-level followed by a number to define the level of LOG provided by Iptables), and REJECT.

5.To drop packets from particular network-(For SMTP Port-25)

- `iptables -A INPUT -s 192.168.0.0/24 -p tcp - -destination-port 25 -j DROP`

6.To accept particular packets from specific network

- `iptables -A INPUT -s 192.168.0.66 -j ACCEPT`

Experiment: 06

Aim: To configure static routing in packet tracer (Simulation of router configuration).

Description: There are two types of routing available. Static routing and Dynamic routing.

Static Routing or Non-Adaptive Routing follows user-defined routing. Here, the routing table is not changed until the network administrator changes it. Static Routing uses simple routing algorithms and provides more security than dynamic routing.

Dynamic Routing or Adaptive Routing, as the name suggests, changes the routing table if there is any change in the network topology. During network change, dynamic routing sends a signal to the router, recalculates the routes and sends the updated routing information.

Difference between Static Routing and Dynamic Routing

The following table highlights the major differences between Static Routing and Dynamic Routing.

Key	**Static Routing**	**Dynamic Routing**
Routing pattern	In static routing, user-defined routes are used in the routing table.	In dynamic routing, routes are updated as per the changes in network.
Routing Algorithm	No complex algorithm used to figure out the shortest path.	Dynamic routing employs complex algorithms to find the shortest routes.
Security	Static routing provides higher security.	Dynamic routing is less secure.
Automation	Static routing is a manual process.	Dynamic routing is an automatic process.
Applicability	Static routing is used in smaller networks.	Dynamic routing is implemented in large networks.
Protocols	Static routing may not follow any specific protocol.	Dynamic routing follows protocols like BGP, RIP and EIGRP.

Additional Resources	Static routing does not require any additional resources.	Dynamic routing requires additional resources like memory, bandwidth etc.

Open Cisco packet tracer and create a network as per the following design .

1.Take two 1941 routers

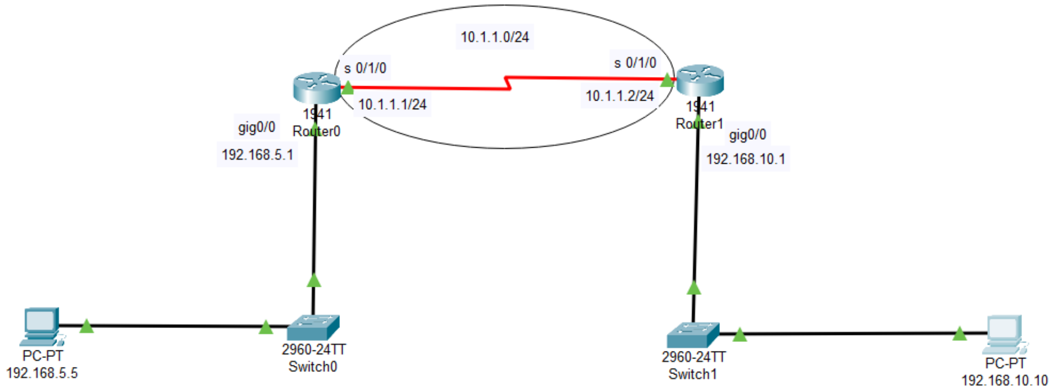
2.Take 2 2960 switches

3.Take two pc’s

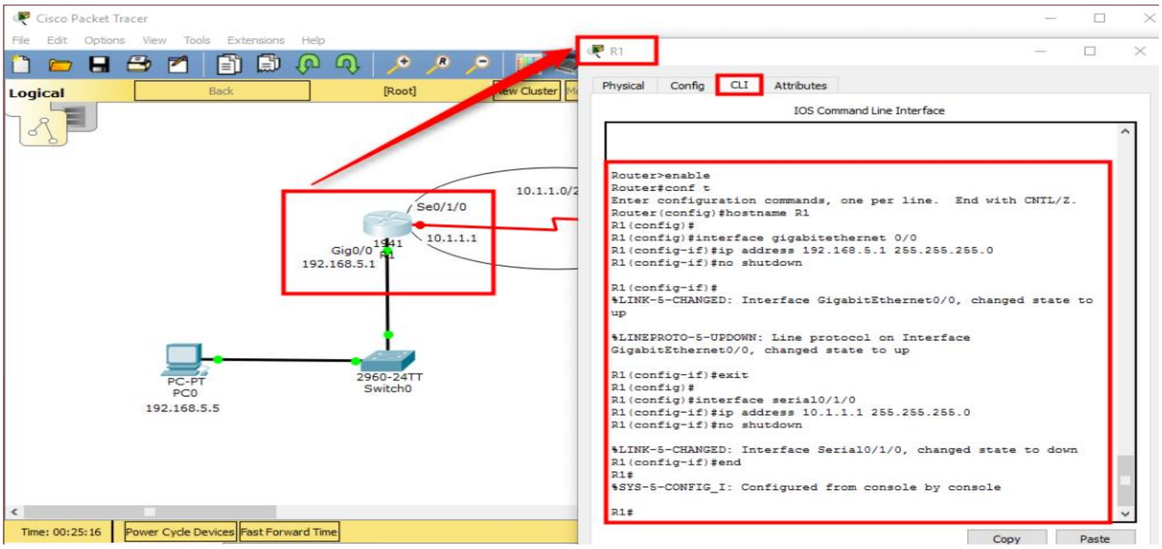
4.make a connection between all the devices .

5.Assign ip addresses to pcs

6.Configure Gigaethernet and serial interfaces as per the following instructions.

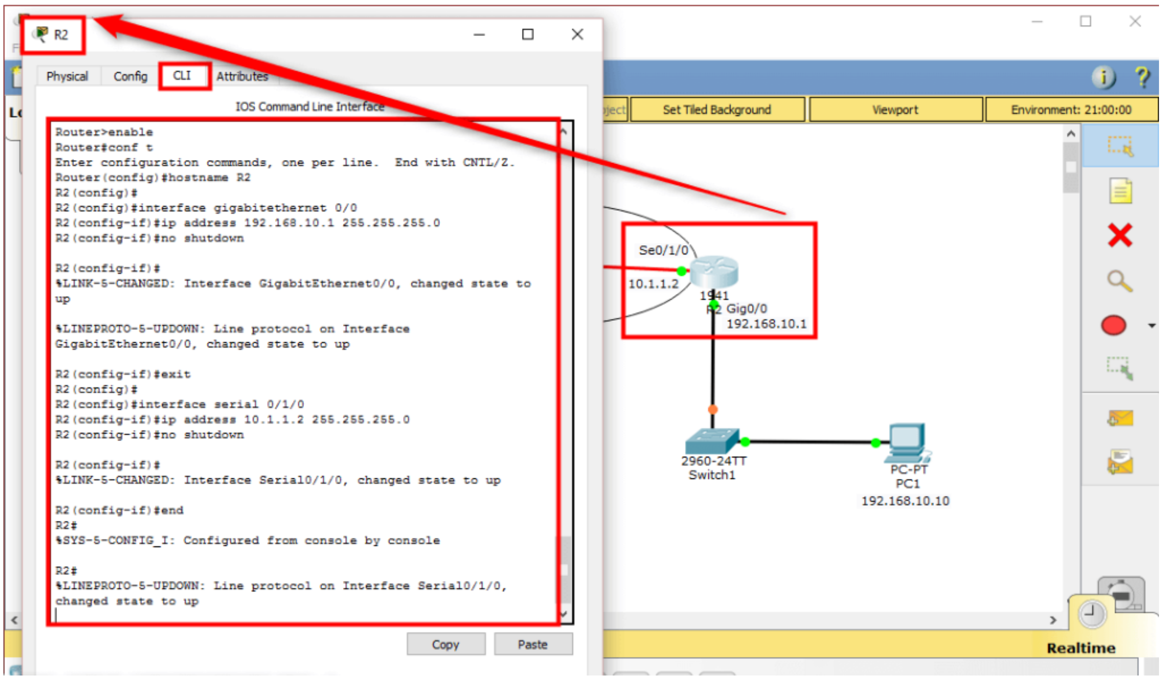


Router 1 Configuration



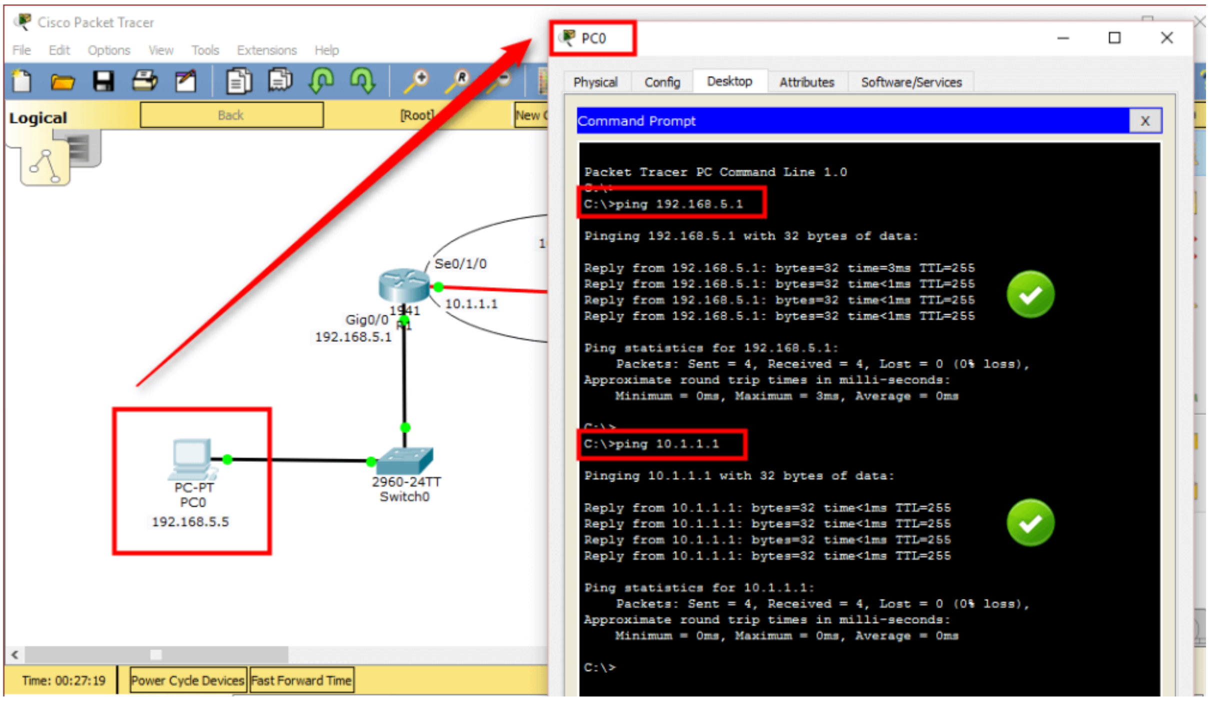
```
Router>enable
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R1
R1(config)#
R1(config)#interface gigabitethernet 0/0
R1(config-if)#ip address 192.168.5.1 255.255.255.0
R1(config-if)#no shutdown

R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to
up

%LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/0, changed state to up

R1(config-if)#exit
R1(config)#
R1(config)#interface serial0/1/0
R1(config-if)#ip address 10.1.1.1 255.255.255.0
R1(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/1/0, changed state to down
R1(config-if)#end
R1#
%SYS-5-CONFIG_I: Configured from console by console

R1#
```

Router 2 Configuration



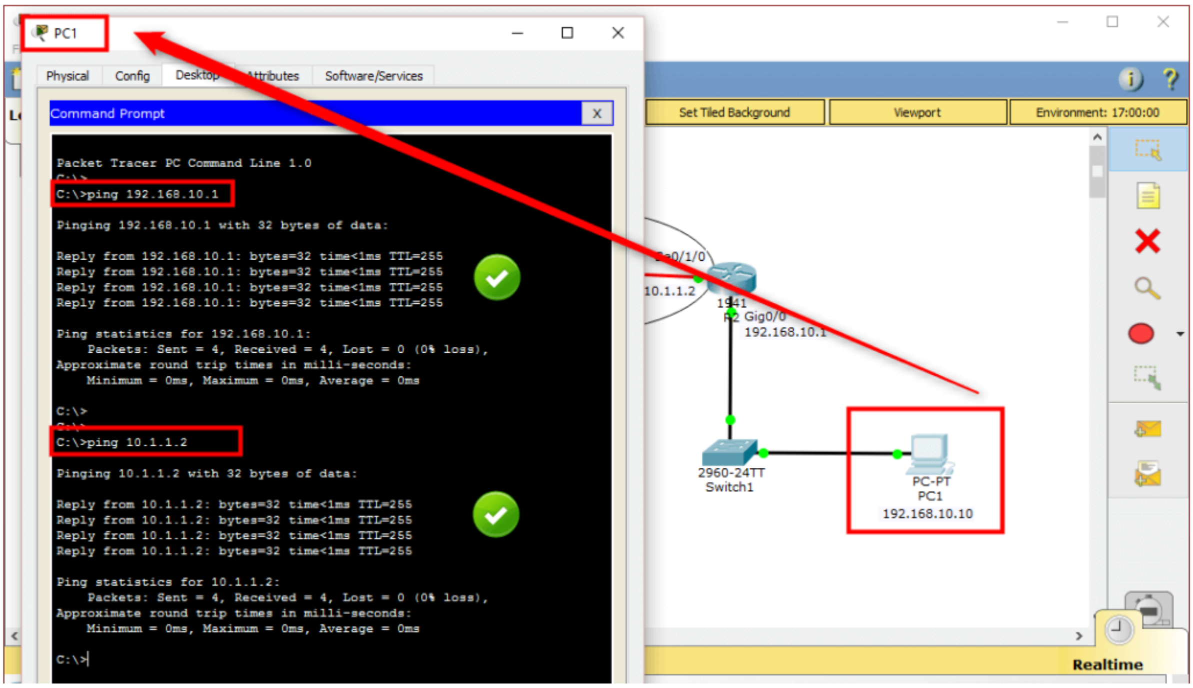
```
Router>enable
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R2
R2(config)#
R2(config)#interface gigabitethernet 0/0
R2(config-if)#ip address 192.168.10.1 255.255.255.0
R2(config-if)#no shutdown

R2(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to
up

%LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/0, changed state to up

R2(config-if)#exit
R2(config)#
R2(config)#interface serial 0/1/0
R2(config-if)#ip address 10.1.1.2 255.255.255.0
R2(config-if)#no shutdown

R2(config-if)#
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to up

R2(config-if)#end
R2#
%SYS-5-CONFIG_I: Configured from console by console

R2#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0,
changed state to up
```

pc0

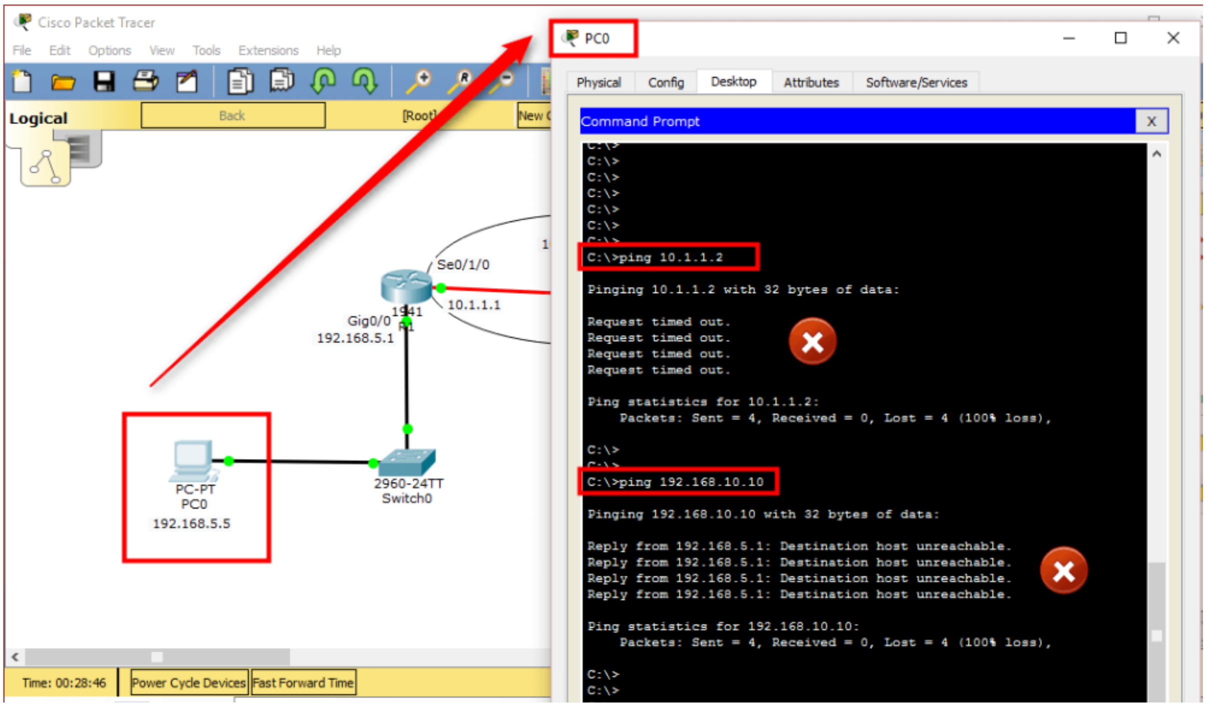


```
Packet Tracer PC Command Line 1.0
C:\>
C:\>ping 192.168.5.1

Pinging 192.168.5.1 with 32 bytes of data:

Reply from 192.168.5.1: bytes=32 time=3ms TTL=255
Reply from 192.168.5.1: bytes=32 time<1ms TTL=255
Reply from 192.168.5.1: bytes=32 time<1ms TTL=255
Reply from 192.168.5.1: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.5.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 3ms, Average = 0ms

C:\>
C:\>ping 10.1.1.1

Pinging 10.1.1.1 with 32 bytes of data:

Reply from 10.1.1.1: bytes=32 time<1ms TTL=255
Reply from 10.1.1.1: bytes=32 time<1ms TTL=255
Reply from 10.1.1.1: bytes=32 time<1ms TTL=255
Reply from 10.1.1.1: bytes=32 time<1ms TTL=255

Ping statistics for 10.1.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

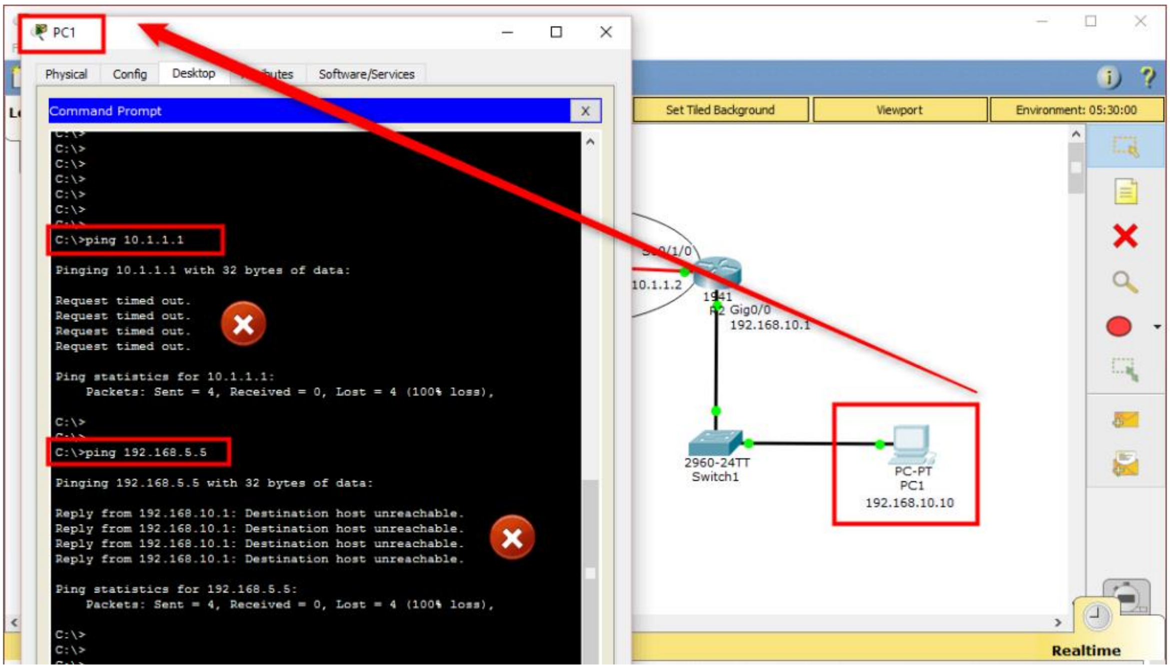
pc1



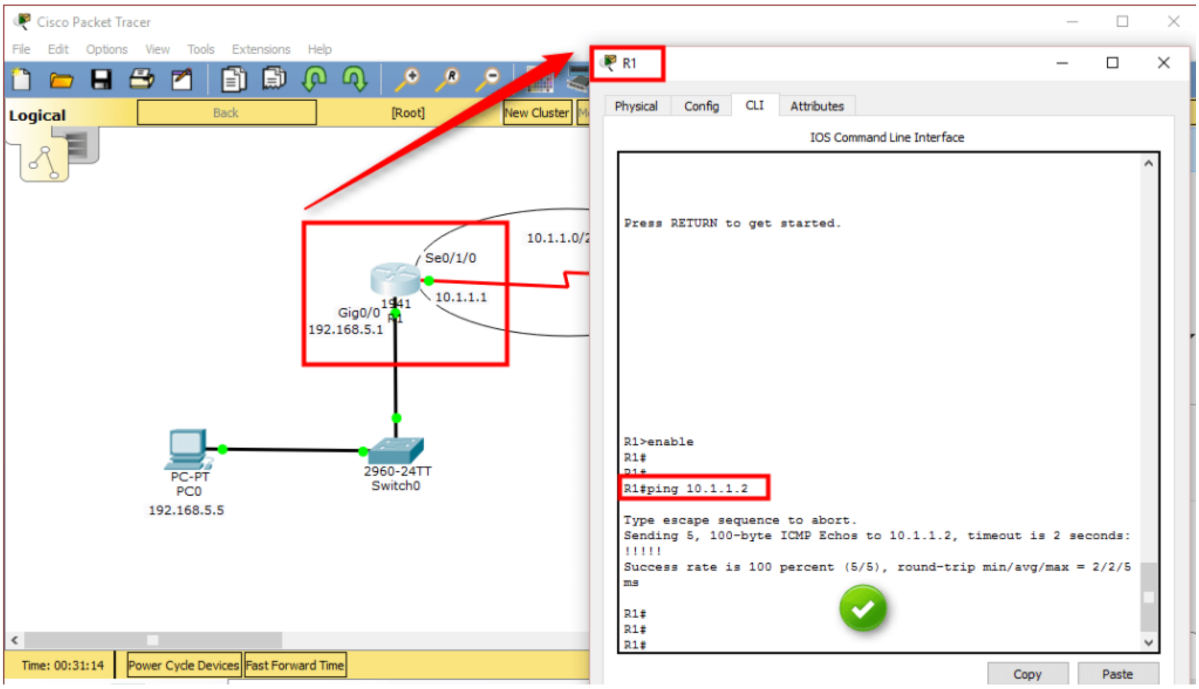
```
Packet Tracer PC Command Line 1.0
C:\>
C:\>ping 192.168.10.1

Pinging 192.168.10.1 with 32 bytes of data:

Reply from 192.168.10.1: bytes=32 time<1ms TTL=255
Reply from 192.168.10.1: bytes=32 time<1ms TTL=255
Reply from 192.168.10.1: bytes=32 time<1ms TTL=255
Reply from 192.168.10.1: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.10.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
C:\>
C:\>ping 10.1.1.2

Pinging 10.1.1.2 with 32 bytes of data:

Reply from 10.1.1.2: bytes=32 time<1ms TTL=255
Reply from 10.1.1.2: bytes=32 time<1ms TTL=255
Reply from 10.1.1.2: bytes=32 time<1ms TTL=255
Reply from 10.1.1.2: bytes=32 time<1ms TTL=255

Ping statistics for 10.1.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```



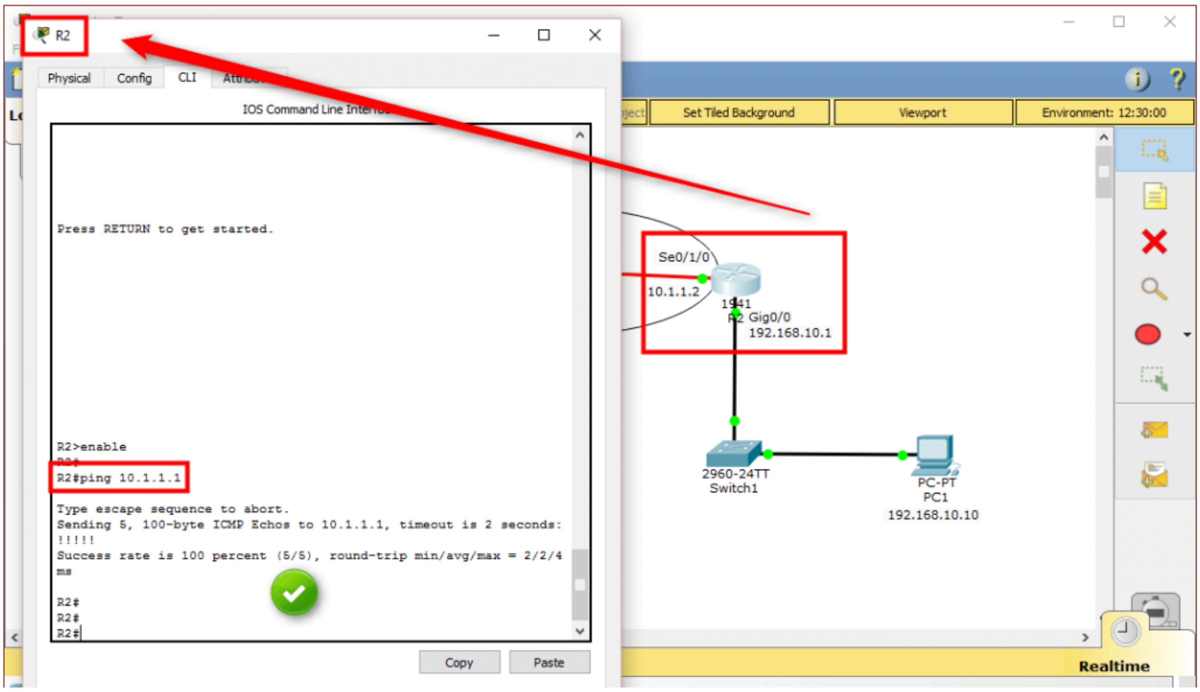
```
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>ping 10.1.1.2

Pinging 10.1.1.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
C:\>
C:\>ping 192.168.10.10

Pinging 192.168.10.10 with 32 bytes of data:

Reply from 192.168.5.1: Destination host unreachable.
Reply from 192.168.5.1: Destination host unreachable.
Reply from 192.168.5.1: Destination host unreachable.
Reply from 192.168.5.1: Destination host unreachable.

Ping statistics for 192.168.10.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
C:\>
```



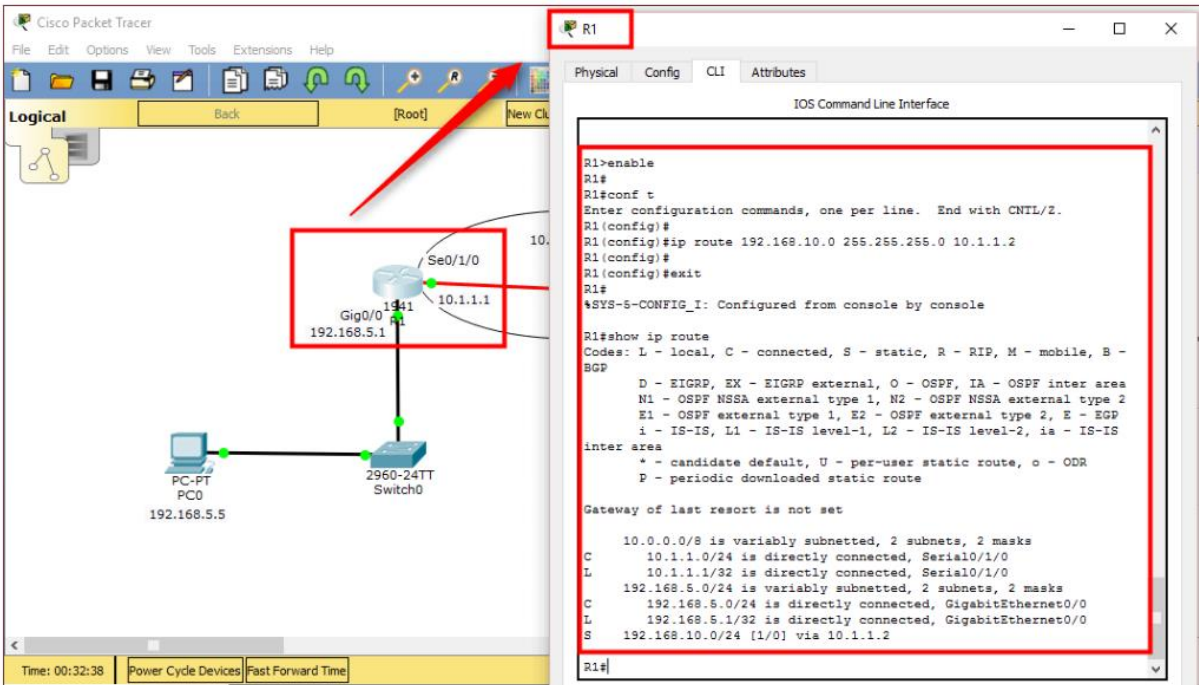
```
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>ping 10.1.1.1

Pinging 10.1.1.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.1.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
C:\>
C:\>ping 192.168.5.5

Pinging 192.168.5.5 with 32 bytes of data:

Reply from 192.168.10.1: Destination host unreachable.
Reply from 192.168.10.1: Destination host unreachable.
Reply from 192.168.10.1: Destination host unreachable.
Reply from 192.168.10.1: Destination host unreachable.

Ping statistics for 192.168.5.5:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
C:\>
C:\>
```



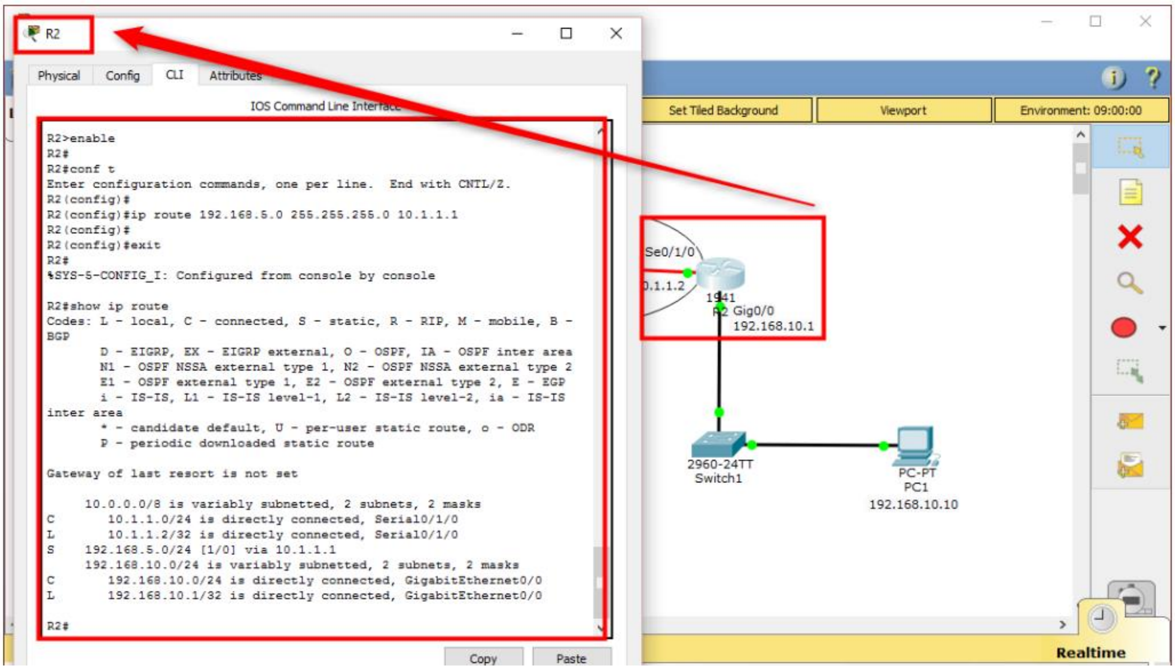
```
Press RETURN to get started.



R1>enable
R1#
R1#
R1#ping 10.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/5
ms

R1#
R1#
R1#
```



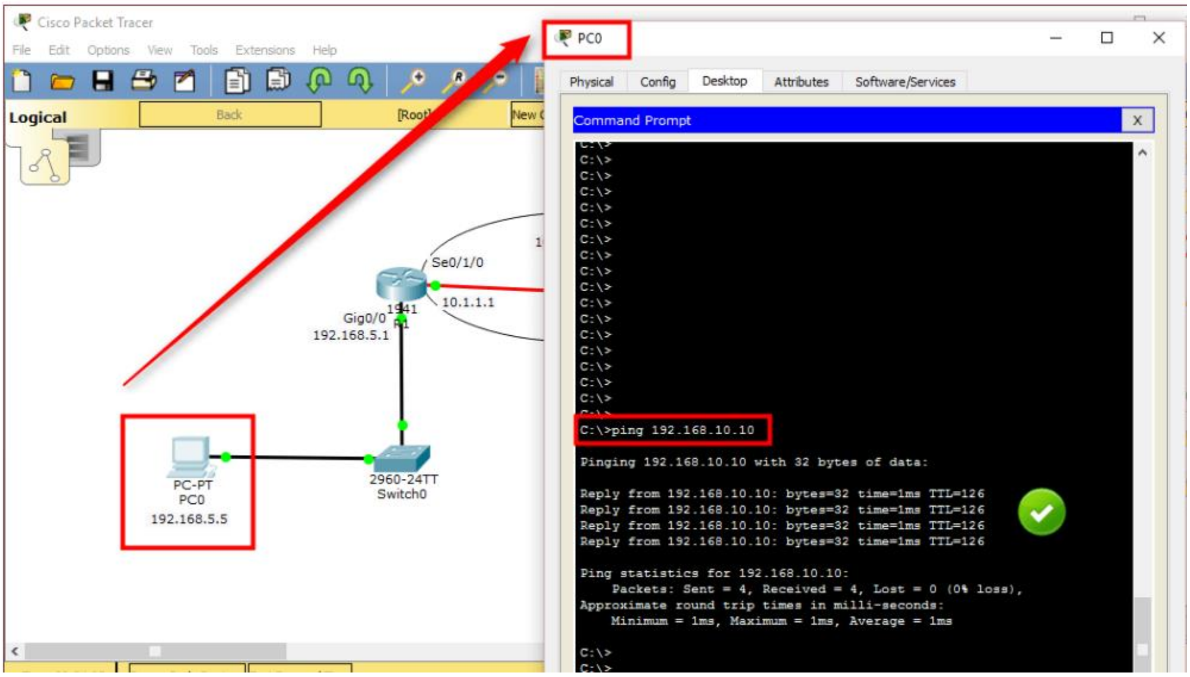
```
Press RETURN to get started.



R2>enable
R2#
R2#ping 10.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/4
ms

R2#
R2#
R2#
```



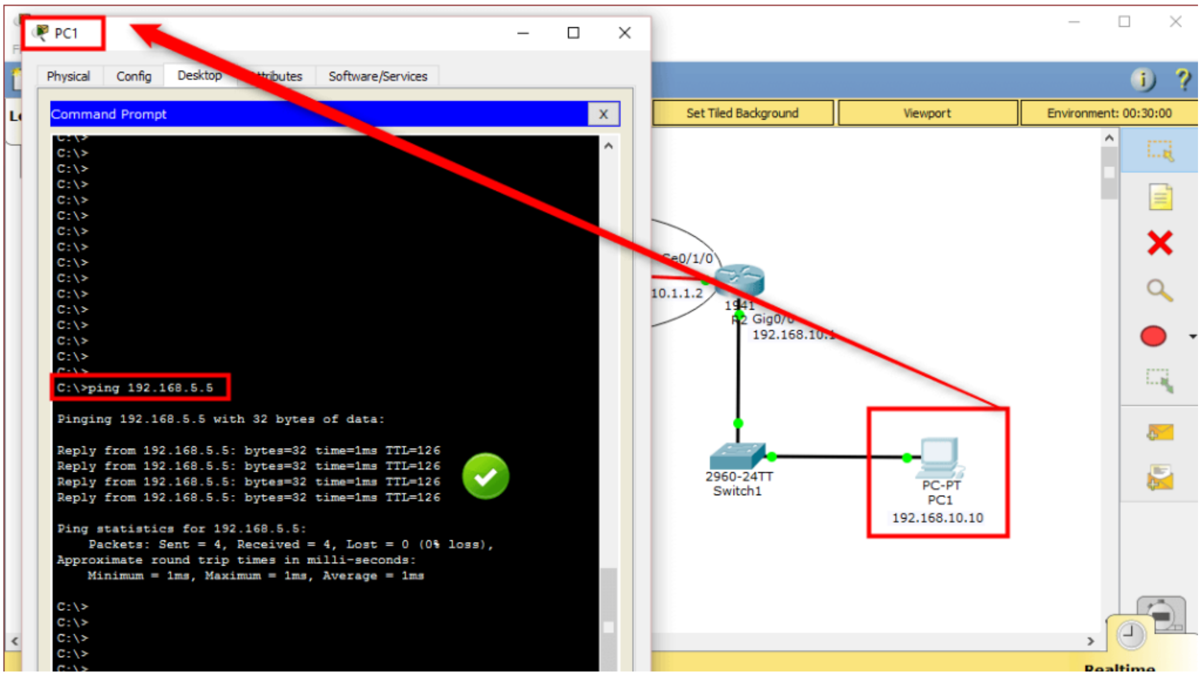
```
R1>enable
R1#
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#
R1(config)#ip route 192.168.10.0 255.255.255.0 10.1.1.2
R1(config)#
R1(config)#exit
R1#
%SYS-5-CONFIG_I: Configured from console by console

R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.1.1.0/24 is directly connected, Serial0/1/0
L       10.1.1.1/32 is directly connected, Serial0/1/0
     192.168.5.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.5.0/24 is directly connected, GigabitEthernet0/0
L       192.168.5.1/32 is directly connected, GigabitEthernet0/0
S    192.168.10.0/24 [1/0] via 10.1.1.2

R1#
```



```
R2>enable
R2#
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#
R2(config)#ip route 192.168.5.0 255.255.255.0 10.1.1.1
R2(config)#
R2(config)#exit
R2#
%SYS-5-CONFIG_I: Configured from console by console

R2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.1.1.0/24 is directly connected, Serial0/1/0
L       10.1.1.2/32 is directly connected, Serial0/1/0
S    192.168.5.0/24 [1/0] via 10.1.1.1
     192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.10.0/24 is directly connected, GigabitEthernet0/0
L       192.168.10.1/32 is directly connected, GigabitEthernet0/0

R2#
```



```
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>ping 192.168.10.10

Pinging 192.168.10.10 with 32 bytes of data:

Reply from 192.168.10.10: bytes=32 time=1ms TTL=126
Reply from 192.168.10.10: bytes=32 time=1ms TTL=126
Reply from 192.168.10.10: bytes=32 time=1ms TTL=126
Reply from 192.168.10.10: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.10.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\>
C:\>
```



```
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>ping 192.168.5.5

Pinging 192.168.5.5 with 32 bytes of data:

Reply from 192.168.5.5: bytes=32 time=1ms TTL=126
Reply from 192.168.5.5: bytes=32 time=1ms TTL=126
Reply from 192.168.5.5: bytes=32 time=1ms TTL=126
Reply from 192.168.5.5: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.5.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\>
C:\>
C:\>
C:\>
C:\>
```

Conclusion: Static routing is more suitable for small networks where a network administrator manages the routing tables. Static routing uses simple routing algorithms and provides better security than dynamic routing. Dynamic routing is used in extensive networks, as it allows routers to choose the best path based on the changes in the logical network layout in real-time.

Experiment: 07

Aim: Perform network discovery using discovery tools-Nmap

Description: Nmap stands for Network Mapper. It is an open source tool that is used widely for network discovery and security auditing. Nmap was originally designed to scan large networks, but it can work equally well for single hosts. Network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets to determine: • what hosts are available on the network, • what services those hosts are offering, • what operating systems they are running on, • what type of firewalls are in use, and other such characteristics. Nmap runs on all major computer operating systems such as Windows, Mac OS X, and Linux.

Basic Steps:

Before attacking a system, it is required that you know what operating system is hosting a website. Once a target OS is known, then it becomes easy to determine which vulnerabilities might be present to exploit the target system. Below is a simple nmap command which can be used to identify the operating system serving a website and all the opened ports associated with the domain name, i.e., the IP address. $nmap -O -v facebook.com It will show you the following sensitive information about the given domain name or IP address: Starting Nmap 5.51 (http://nmap.org) at 2015-10-04 09:57 CDT Initiating Parallel DNS resolution of 1 host. at 09:57 Completed Parallel DNS resolution of 1 host. at 09:57, 0.00s elapsed Initiating SYN Stealth Scan at 09:57

```
Scanning facebook.com (66.135.33.172) [1000 ports] Discovered open port 22/tcp on
66.135.33.172

Discovered open port 3306/tcp on 66.135.33.172

Discovered open port 80/tcp on 66.135.33.172 Discovered open port 443/tcp on
66.135.33.172
```

Completed SYN Stealth Scan at 09:57, 0.04s elapsed (1000 total ports) Initiating OS detection (try #1) against tutorialspoint.com (66.135.33.172) Retrying OS detection (try #2) against tutorialspoint.com (66.135.33.172) Retrying OS detection (try #3) against tutorialspoint.com (66.135.33.172) Retrying OS detection (try #4) against tutorialspoint.com (66.135.33.172) Retrying OS detection (try #5) against tutorialspoint.com (66.135.33.172) Nmap scan report for tutorialspoint.com (66.135.33.172) Host is up (0.000038s latency). Not shown: 996 closed ports

Port Scanning

We have just seen information given by the nmap command. This command lists down all the open ports on a given server.

```
22/tcpopen

ssh 80/tcpopen http

443/tcp open https

3306/tcpopen mysql
```

You can also check if a particular port is opened or not using the following command:

```
$nmap -sT -p 443 facebook.com
```

It will produce the following result:

```
Starting Nmap5.51 ( http://nmap.org ) at 2017-08-04 10:19 CDT Nmap scan report for facebook.com (66.135.33.172) [Assume]

Host is up (0.000067s latency).

PORT STATE SERVICE 443/tcpopen https

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
```

Once a hacker knows about open ports, then he can plan different attack techniques through the open ports.

Quick Fix: It is always recommended to check and close all the unwanted ports to safeguard the system from malicious attacks.

Ping Sweep:

A ping sweep is a network scanning technique that you can use to determine which IP address from a range of IP addresses map to live hosts. Ping Sweep is also known as ICMP sweep.

You can use fping command for ping sweep. This command is a ping-like program which uses the Internet Control Message Protocol (ICMP) echo request to determine if a host is up.

fping is different from ping in that you can specify any number of hosts on the command line, or specify a file containing the lists of hosts to ping. If a host does not respond within a certain time limit and/or retry limit, it will be considered unreachable.

Quick Fix To disable ping sweeps on a network, you can block ICMP ECHO requests from outside sources. This can be done using the following command which will create a firewall rule in iptable.

```
$iptables -A OUTPUT -p icmp --icmp-type echo-request -j DROP
```

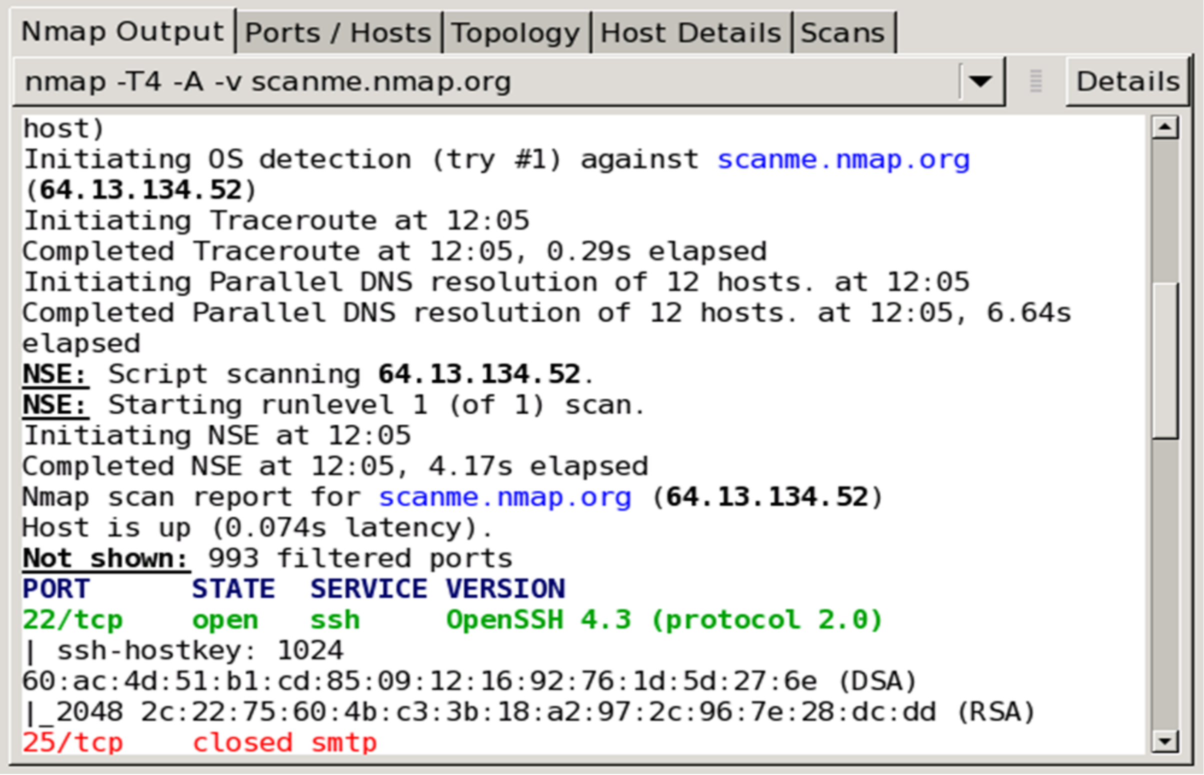
Interpreting Scan Results

Nmap's output is displayed during and after a scan. This output will be familiar to Nmap users. Except for Zenmap's color highlighting, this doesn't offer any visualization advantages over running Nmap in a terminal. However, other parts of Zenmap's interface interpret and aggregate the terminal output in a way that makes scan results easier to understand and use.

Scan Results Tabs

Each scan window contains five tabs which each display different aspects of the scan results. They are: “Nmap Output”, “Ports / Hosts”, “Topology”, “Host Details”, and “Scans”. Each of these are discussed in this section.

The “Nmap Output” tab

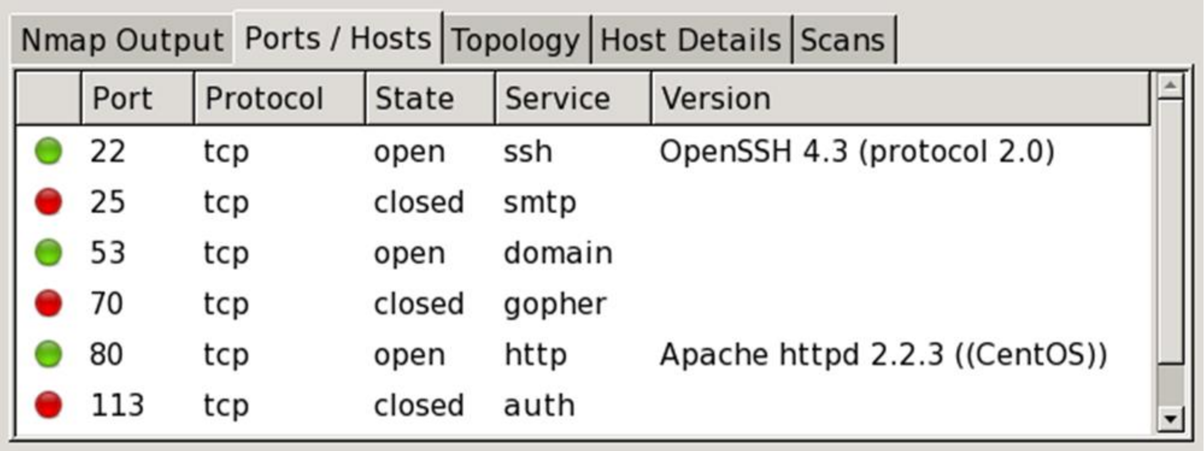


```
host)
Initiating OS detection (try #1) against scanme.nmap.org
(64.13.134.52)
Initiating Traceroute at 12:05
Completed Traceroute at 12:05, 0.29s elapsed
Initiating Parallel DNS resolution of 12 hosts. at 12:05
Completed Parallel DNS resolution of 12 hosts. at 12:05, 6.64s
elapsed
NSE: Script scanning 64.13.134.52.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 12:05
Completed NSE at 12:05, 4.17s elapsed
Nmap scan report for scanme.nmap.org (64.13.134.52)
Host is up (0.074s latency).
Not shown: 993 filtered ports
PORT      STATE  SERVICE VERSION
22/tcp    open   ssh     OpenSSH 4.3 (protocol 2.0)
| ssh-hostkey: 1024
60:ac:4d:51:b1:cd:85:09:12:16:92:76:1d:5d:27:6e (DSA)
|_2048 2c:22:75:60:4b:c3:3b:18:a2:97:2c:96:7e:28:dc:dd (RSA)
25/tcp    closed smtp
```

The “Nmap Output” tab is displayed by default when a scan is run. It shows the familiar Nmap terminal output. The display highlights parts of the output according to their meaning; for example, open and closed ports are displayed in different colors. Custom highlights can be configured in zenmap.conf.

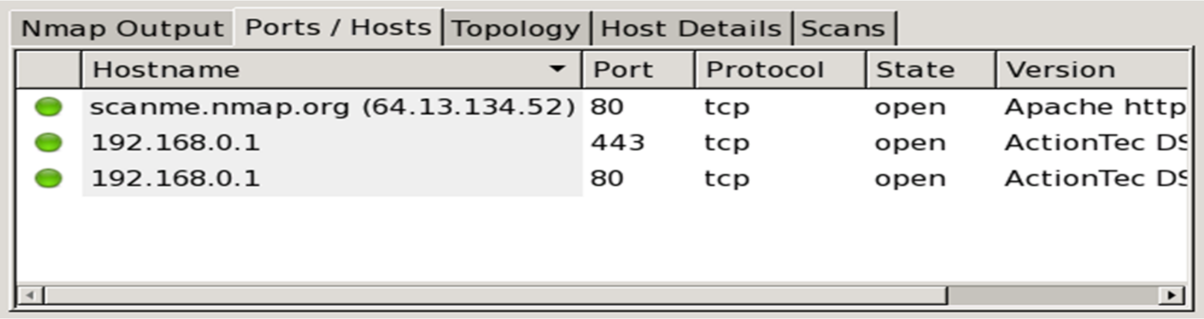
Recall that the results of more than one scan may be shown in a window. The drop-down combo box at the top of the tab allows you to select the scan to display. The “Details” button brings up a window showing miscellaneous information about the scan, such as timestamps, command-line options, and the Nmap version number used.

The “Ports / Hosts” tab



	Port	Protocol	State	Service	Version
🟢	22	tcp	open	ssh	OpenSSH 4.3 (protocol 2.0)
🔴	25	tcp	closed	smtp	
🟢	53	tcp	open	domain	
🔴	70	tcp	closed	gopher	
🟢	80	tcp	open	http	Apache httpd 2.2.3 ((CentOS))
🔴	113	tcp	closed	auth	

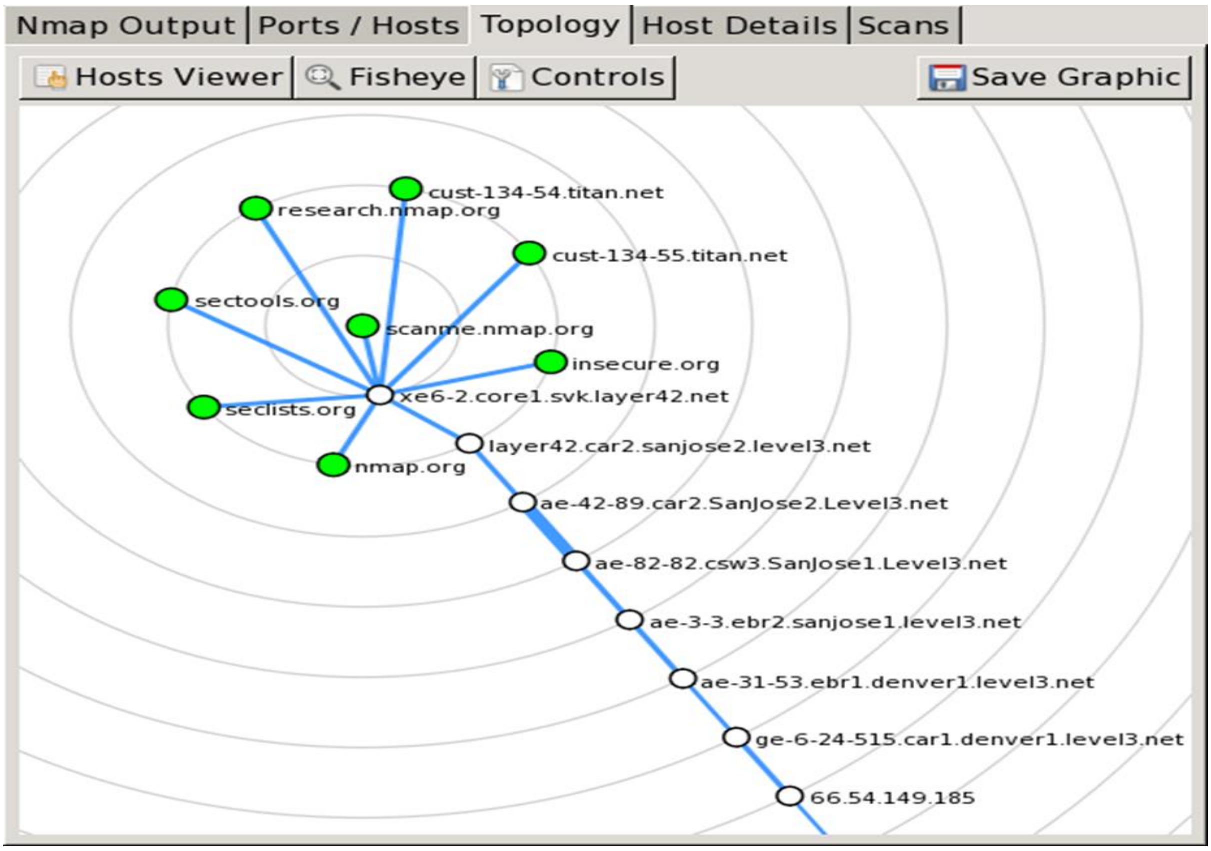
The “Ports / Hosts” tab's display differs depending on whether a host or a service is currently selected. When a host is selected, it shows all the interesting ports on that host, along with version information when available. Host selection is further described in the section called “Sorting by Host”.



	Hostname	Port	Protocol	State	Version
🟢	scanme.nmap.org (64.13.134.52)	80	tcp	open	Apache http
🟢	192.168.0.1	443	tcp	open	ActionTec DS
🟢	192.168.0.1	80	tcp	open	ActionTec DS

When a service is selected, the “Ports / Hosts” tab shows all the hosts which have that port open or filtered. This is a good way to quickly answer the question “What computers are running HTTP?” Service selection is further described in the section called “Sorting by Service”.

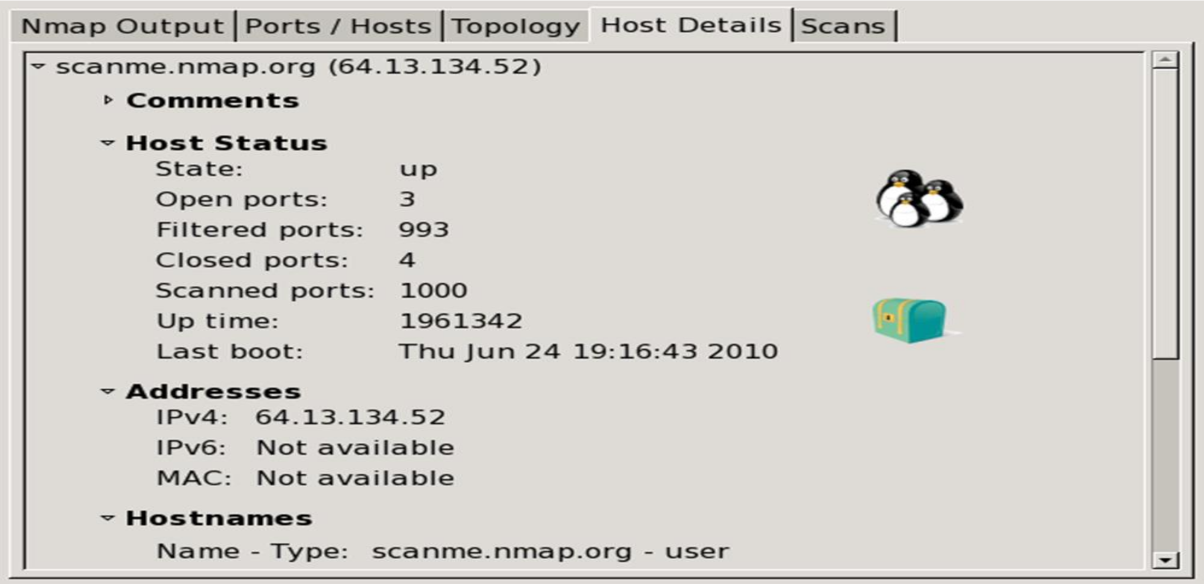
The “Topology” tab



Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Hosts Viewer
Fisheye
Controls
Save Graphic
cust-134-54.titan.net
research.nmap.org
cust-134-55.titan.net
sectools.org
scanme.nmap.org
insecure.org
xe6-2.core1.svk.layer42.net
seclists.org
layer42.car2.sanjose2.level3.net
nmap.org
ae-42-89.car2.SanJose2.Level3.net
ae-82-82.csw3.SanJose1.Level3.net
ae-3-3.ebr2.sanjose1.level3.net
ae-31-53.ebr1.denver1.level3.net
ge-6-24-515.car1.denver1.level3.net
66.54.149.185

The “Topology” tab is an interactive view of the connections between hosts in a network. Hosts are arranged in concentric rings. Each ring represents an additional network hop from the center node. Clicking on a node brings it to the center. Because it shows a representation of the network paths between hosts, the “Topology” tab benefits from the use of the --traceroute option. Topology view is discussed in more detail in the section called “Surfing the Network Topology”.

The “Host Details” tab



```
▾ scanme.nmap.org (64.13.134.52)
    ▸ Comments
    ▾ Host Status
        State:            up
        Open ports:       3
        Filtered ports:   993
        Closed ports:     4
        Scanned ports:    1000
        Up time:          1961342
        Last boot:        Thu Jun 24 19:16:43 2010
    ▾ Addresses
        IPv4:  64.13.134.52
        IPv6:  Not available
        MAC:   Not available
    ▾ Hostnames
        Name - Type:  scanme.nmap.org - user
```

The “Host Details” tab breaks all the information about a single host into a hierarchical display. Shown are the host's names and addresses, its state (up or down), and the number and status of scanned ports. The host's uptime, operating system, OS, and other associated details are shown when available. When no exact OS match is found, the closest matches are displayed. There is also a collapsible text field for storing a comment about the host which will be saved when the scan is saved to a file. Each host has an icon that provides a very rough “vulnerability” estimate, which is based solely on the number of open ports. The icons and the numbers of open ports they correspond to are

0–2 open ports,

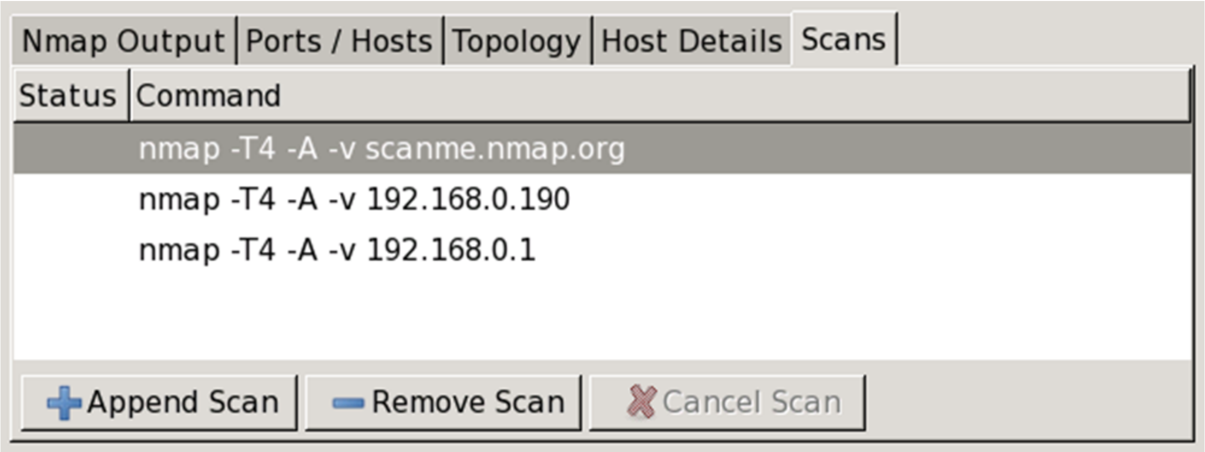
3–4 open ports,

5–6 open ports,

7–8 open ports, and

9 or more open ports.

The “Scans” tab



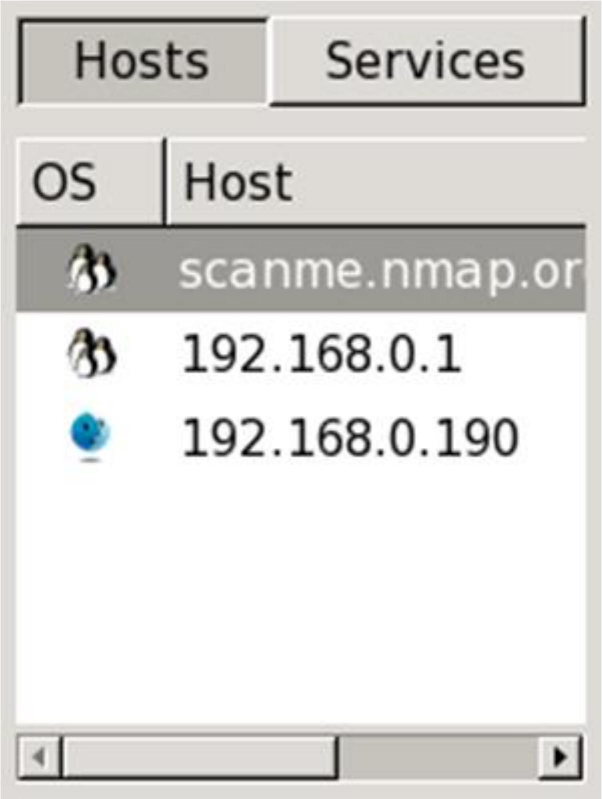
Status	Command
	nmap -T4 -A -v scanme.nmap.org
	nmap -T4 -A -v 192.168.0.190
	nmap -T4 -A -v 192.168.0.1

The “Scans” tab shows all the scans that are aggregated to make up the network inventory. From this tab you can add scans (from a file or directory) and remove scans.

While a scan is executing and not yet complete, its status is “Running”. You may cancel a running scan by clicking the “Cancel Scan” button.

Sorting by Host

Figure Host selection



OS	Host
	scanme.nmap.or
	192.168.0.1
	192.168.0.190

On the left side of Zenmap's main window is a column headed by two buttons labeled “Hosts” and “Services”. Clicking the “Hosts” button will bring up a list of all hosts that were scanned, as in Figure. Commonly this contains just a single host, but it can contain thousands in a large scan. The host list can be sorted by OS or host name/IP address by clicking the headers at the top of the list. Selecting a host will cause the “Ports / Hosts” tab to display the interesting ports on that host.

Each host is labeled with its host name or IP address and has an icon indicating the operating system that was detected for that host. The icon is meaningful only if OS detection (-O) was performed. Otherwise, the icon will be a default one indicating that the OS is unknown. Figure shows all possible icons. Note that Nmap's OS detection cannot always provide the level of specificity implied by the icons; for example a Red Hat Linux host will often be displayed with the generic Linux icon.

Conclusion:

Network scanning provides a wealth of information about the target network, which is valuable regardless of whether you're trying to attack the network or protect it from attack. While performing a basic scan is a simple matter, the network scanners covered in this experiment provide a wide array of options to tweak your scan to achieve the best results. Nmap is used to detect IP spoofing and port scanning.

Experiment: 08

Aim: Socket programming using TCP.
Design TCP iterative Client and Server application to reverse the given input sentence.
Theory:

Description: Most interprocess communication uses the client server model. These terms refer to the two processes which will be communicating with each other. One of the two processes, the client, connects to the other process, the server, typically to make a request for information. A good analogy is a person who makes a phone call to another person.
The system calls for establishing a connection are somewhat different for the client and the server, but both involve the basic construct of a socket. A socket is one end of an interprocess communication channel. The two processes each establish their own socket.
The steps involved in establishing a socket on the client side are as follows:
1. Create a socket with the socket () system call
2. Connect the socket to the address of the server using the connect () system call
3. Send and receive data. There are a number of ways to do this, but the simplest is to use the Read () and write () system calls.
The steps involved in establishing a socket on the server side are as follows:
1. Create a socket with the socket() system call
2. Bind the socket to an address using the bind() system call. For a server socket on the Internet, an address consists of a port number on the host machine.
3. Listen for connection s with the listen() system call
4. Accept a connection with the accept() system call. This call typically blocks until a client connects with the server.
5. Send and receive data
6. When a socket is created, the program has to specify the address domain and the socket type. Two processes can communicate with each other only if their sockets are of the same type and in the same domain. There are two widely used address domains, the unix domain, in which two processes which share a common file system communicate, and the Internet do-main, in which two processes running on any two hosts on the Internet communicate. Each of these has its own address format.
7. The address of a socket in the Unix domain is a character string which is basically an entry in the file system.
8. The address of a socket in the Internet domain consists of the Internet address of the host machine (every computer on the Internet has a unique 32 bit address, often referred to as its IP address). In addition, each socket needs a port number on that host. Port numbers are 16 bit unsigned integers. The lower numbers are reserved in Unix for standard services. For example, the port number for the FTP server is 21. It is important that standard services be at
the same port on all computers so that clients will know their addresses. However, port numbers above 2000 are generally available.
9. There are two widely used socket types, stream sockets, and datagram sockets. Stream sockets treat communications as a continuous stream of characters, while datagram sockets

have to read entire messages at once. Each uses its own communications protocol. Stream sockets use TCP (Transmission Control Protocol), which is a reliable, stream oriented protocol, and data-gram sockets use UDP (Unix Datagram Protocol), which is unreliable and message oriented.

Conclusion:

Studied TCP Client and Server application to reverse the given input sentence.

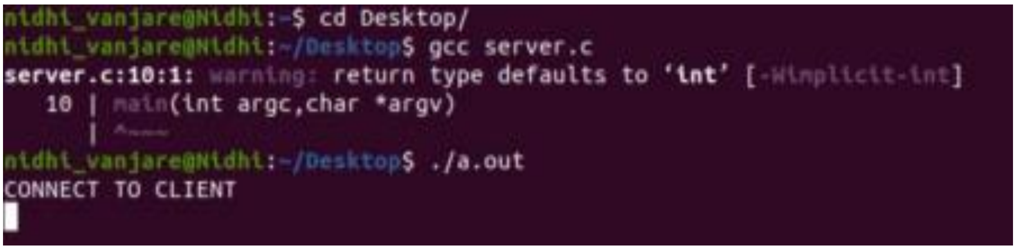
1 . server.c

```
                #include<stdio.h>

                #include<unistd.h>

                #include<string.h>

                #include<sys/socket.h>

                #include<stdlib.h>

                #include<netinet/in.h>

                #include<sys/types.h>

                #define MAXLINE 20

                #define SERV_PORT 5777 main(int argc,char *argv) {
                int i,j;
                ssize_t n;
                char line[MAXLINE],revline[MAXLINE];
                int listenfd,connfd,clilen;
                struct sockaddr_in servaddr,cliaddr;

                listenfd=socket(AF_INET,SOCK_STREAM,0);

                bzero(&servaddr,sizeof(servaddr));

                servaddr.sin_family=AF_INET;

                servaddr.sin_port=htons(SERV_PORT);
bind(listenfd,(struct sockaddr*)&servaddr,sizeof(servaddr));

                listen(listenfd,1);
```

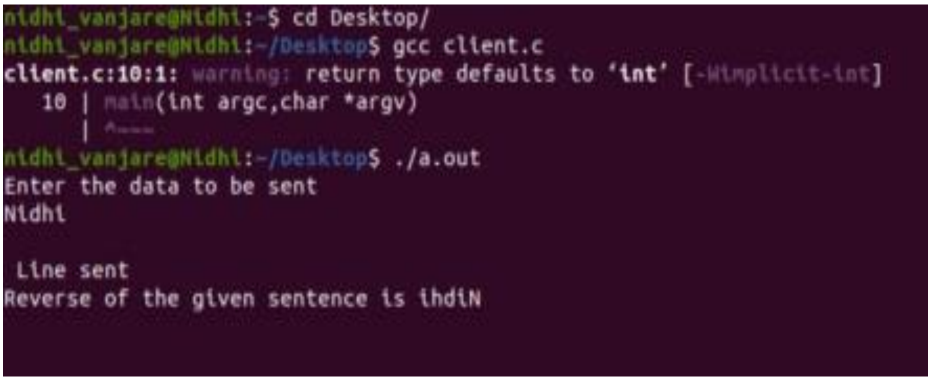
```
                    for(;;)
                    {
                    clilen=sizeof(cliaddr);
connfd=accept(listenfd,(struct sockaddr*)&cliaddr,&clilen);

                    printf("CONNECT TO CLIENT\n");
                    while(1)
                    { if((n=read(connfd,line,MAXLINE))==0)
                    break;
                    line[n-1]='\0';
                    j=0;
                    for(i=n-2;i>=0;i--)
                    revline[j++]=line[i];
                    revline[j]='\0';
                    write(connfd,revline,n);}
                    }

                    }
```

2. client.c

```
#include<stdio.h>
#include<unistd.h>
#include<string.h>
#include<sys/socket.h>
#include<stdlib.h>
#include<netinet/in.h> #include<sys/types.h>
#define MAXLINE 20
#define SERV_PORT 5777
main(int argc,char *argv)
{ char sendline[MAXLINE],revline[MAXLINE]; int sockfd;

struct sockaddr_in servaddr; sockfd=socket(AF_INET,SOCK_STREAM,0);
bzero(&servaddr,sizeof(servaddr));
servaddr.sin_family=AF_INET;servaddr.sin_port=ntohs(SERV_PORT);
connect(sockfd,(struct          sockaddr*)&servaddr,sizeof(servaddr));
printf("Enter the data to be sent\n");
while(fgets(sendline,MAXLINE,stdin)!=NULL)

{
write(sockfd,sendline,strlen(sendline));
printf("\n Line sent");
read(sockfd,revline,MAXLINE);
printf("\nReverse of the given sentence is %s",revline);
printf("\n");
}

exit(0); }
```



```
nidhi_vanjare@Nidhi:~$ cd Desktop/
nidhi_vanjare@Nidhi:~/Desktop$ gcc server.c
server.c:10:1: warning: return type defaults to 'int' [-Wimplicit-int]
   10 | main(int argc,char *argv)
      | ^~~~
nidhi_vanjare@Nidhi:~/Desktop$ ./a.out
CONNECT TO CLIENT
```

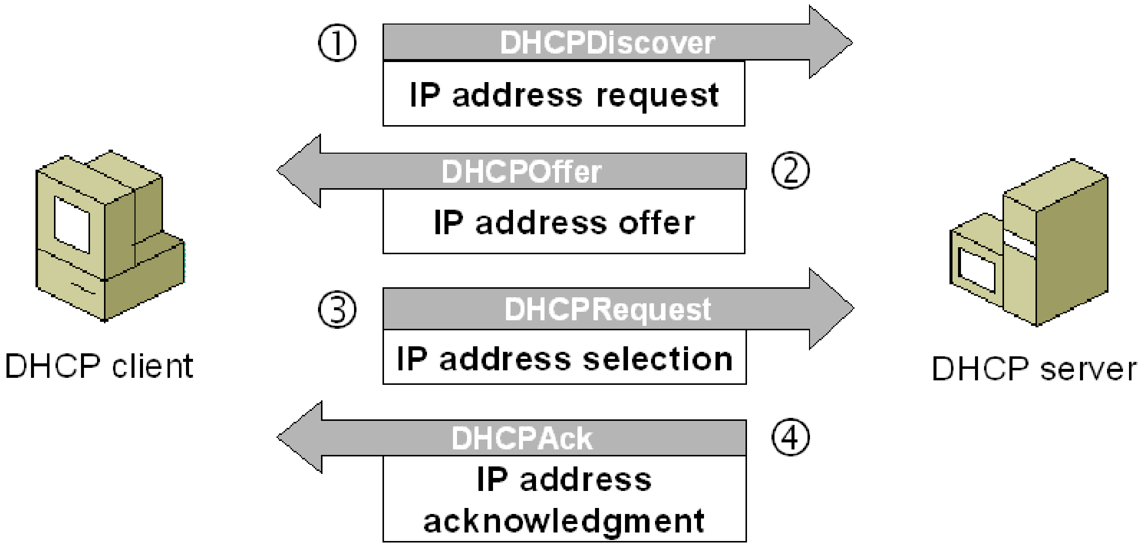


```
nidhi_vanjare@Nidhi:~$ cd Desktop/
nidhi_vanjare@Nidhi:~/Desktop$ gcc client.c
client.c:10:1: warning: return type defaults to 'int' [-Wimplicit-int]
   10 | main(int argc,char *argv)
      | ^~~~
nidhi_vanjare@Nidhi:~/Desktop$ ./a.out
Enter the data to be sent
Nidhi

 Line sent
Reverse of the given sentence is ihdiN
```

Experiment: 09

Aim: Implementation of DHCP using packet tracer.

Description: DHCP is a service. It allows devices to acquire their IP configuration dynamically. It is defined in RFC 2131 and 2939. It works in the server/client model. The server offers and delivers IP configurations. Clients request and acquire their IP configurations.



Discover-The DHCP client broadcasts this message to find a DHCP server.

Offer-The DHCP server broadcasts this message to lease an IP configuration to the DHCP client.

Request-The DHCP client uses this message to notify the DHCP server whether it accepts the proposed IP configuration or not.

Acknowledgement- The DHCP server uses this message to confirm the DHCP client that it can use the offered IP configuration.

Static allocation

In this method, the administrator configures an allocation table on the DHCP server. In this table, the administrator fills the MAC addresses of all clients and assigns an IP configuration to each client.

The DHCP server uses the allocation table to provide IP configurations. When a client requests an IP configuration, the DHCP server checks the table and finds a match. If the DHCP server finds a match, the DHCP server offers the IP configuration that is associated with the MAC address of the client in the match.

Dynamic allocation

In this method, the administrator configures a range of IP addresses on the DHCP server. The DHCP server assigns an IP configuration from the configured range to each client that requests an IP configuration.

In this method, the DHCP offers the IP configuration only for a specific time. This specific time is known as the *lease*. The IP configuration remains valid until the lease duration is over. Once the lease duration is over, the client is required to obtain a new IP configuration from the server.

Automatic allocation

Same as the dynamic method, in this method, the administrator also configures a range of IP addresses on the DHCP server and the DHCP server assigns an IP configuration from the configured range to each client that requests an IP configuration.

Unlike the dynamic method, in this method, the DHCP server assigns the IP configuration permanently. To assign an IP configuration permanently, the DHCP server sets the lease duration to infinite. As a result, once the DHCP server chooses an IP configuration from the pool and assigns the IP configuration to a client, the IP configuration remains with that same client indefinitely.

Conclusion:

Using Cisco packet tracer demonstrated the working of DHCP service.

Experiment: 10

Aim: Configuration of email services (SMTP & POP) using packet tracer.

Description: For sending and receiving messages, we use two protocols one is SMTP (Simple Mail Transfer Protocol) and another is POP3 (Post Office Protocol version 3). They are also called PUSH and POP protocols respectively. They are agents, Message Transfer Agent, and Message Access Agent respectively to send and retrieve the messages.

Difference between SMTP and POP3:

S.N	SMTP	POP3
1.	SMTP stands for Simple Mail Transfer Protocol.	POP3 stands for Post Office Protocol version 3.
2.	It is used for sending messages.	It is used for accessing messages.
3.	The port number of SMTP is 25, 465, and 587 for secured connection (TLS connection).	The port number of POP3 is 110 or port 995 for SSL/TLS connection.
4.	It is an MTA (Message Transfer Agent) for sending the message to the receiver.	It is an MAA (Message Access Agent) for accessing the messages from mailboxes.
5.	It has two MTAs one is client MTA (Message Transfer Agent) and second one is server MTA (Message Transfer Agent).	It has also two MAAs one is client MAA (Message Access Agent) and another is server MAA(Message Access Agent).
6.	SMTP is also known as the PUSH protocol.	POP3 is also known as POP protocol.

7.	SMTP transfers the mail from the sender's computer to the mailbox present on the receiver's mail server.	POP3 allows you to retrieve and organize mails from the mailbox on the receiver mail server to the receiver's computer.
8.	It is implied between sender mail server and receiver mail server.	It is implied between receiver and receiver mail server.

Conclusion:

Configured server for email services using SMTP and POP3 protocol in Cisco packet tracer.