CS-349 NETWORKS LAB

ASSIGNMENT – 1

Anirudh Sharma

150101006

**ANS 1 .**

a . [ -c count ] , $ping -c count IP

[-c count ] by using this option ping will stop sending count ECHO\_REQUEST packet .

b. [ -i interval ] , $ping -i interval IP

wait ‘interval’ seconds between sending each packets . The default is to wait one second between each packet . Only super-user can set interval to values less 0.2 seconds .

c. [ -l preload ] . $ping -l preload IP

if preload is specified , ping send preload number of packet without waiting for reply . Only super-user

may select preload of more than 3 . Normal user could send upto 3 packets .

d. [-s packetsize ] , $ping -s packetsize IP

[-s packetsize ] will set buffer size of data to packetsize . Later In which 20 byte of IP header and 8

bytes of ICMP header are added .

If packetsize is set to 64 byte than total packet size will be **92(64 + 20(IP header) + 8(ICMP header))**

**ANS 2 .** Five host selected for the purpose :

1. www.iitd.ac.in

2. www.ucla.edu

3. www.iitkgp.ac.in

4. english.spbu.ru

5. web.mit.edu

RESULTS

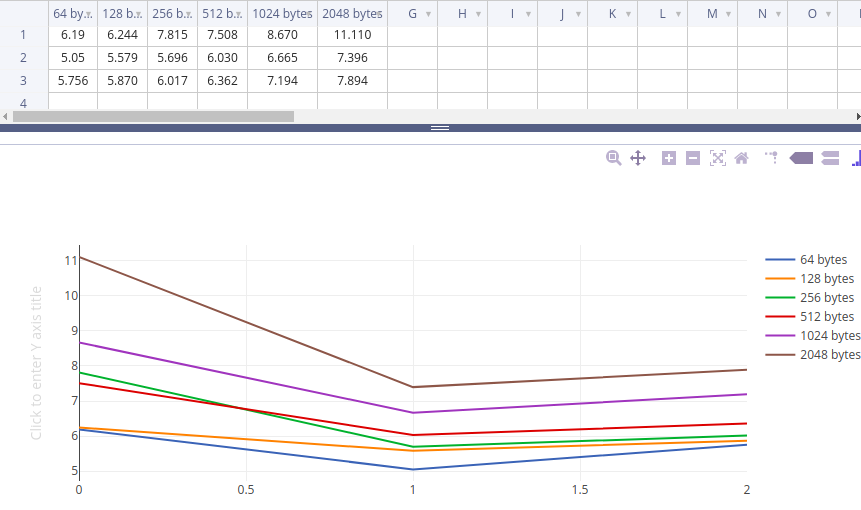
A. Ping Timing (evening ) :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | HOSTNAME | AVERAGE RTT  (noon) (ms) | AVERAGE RTT  (night) (ms) | AVERAGE RTT  (morning)(ms) | PACKET  LOSS | Distance  (Km) |
| 1 | www.iitd.ac.in | 176 | 213.834 | 213.564 | 0% | 11,828 |
| 2 | www.ucla.edu | 74.568 | 74.862 | 74.338 | 0% | 4643.92 |
| 3 | www.iitkgp.ac.in | 202 | 232.945 | 224.814 | 0% | 12,812 |
| 4 | english.spbu.ru | 116.534 | 116.707 | 115.965 | 0% | 6,962 |
| 5 | web.mit.edu | 5.466 | 5.817 | 5.748 | 0% | 5631.095 |

Clearly with increase in distance the average RTT is also increasing which is obvious by the fact that it take more time for packet to travel long distance due to dealy in propagation .

For next part of the question let the host be web.mit.edu . Packets are send from 64 bytes to 2048 bytes with inteval of 64 bytes .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BYTES | Avg RTT(morning) | Avg RTT (noon) | Avg RTT (ms)(night) | PACKET LOSS |
| 64 | 6.193 | 5.505 | 5.756 | 0 |
| 128 | 6.244 | 5.579 | 5.870 | 0 |
| 256 | 7.815 | 5.696 | 6.017 | 0 |
| 512 | 7.508 | 6.030 | 6.362 | 0 |
| 1024 | 8.670 | 6.665 | 7.194 | 0 |
| 2048 | 11.110 | 7.396 | 7.894 | 0 |

****

During noon time there might be less people using the server of the hosts hence leading ping to take less time . Increase in packet size will also increase average rtt as large packet will take more time in propagation.

**ANS 3.**  IP used : 202.141.80.14

ping -n -c 1000 202.141.80.14 --- 1 ping -p ff00 -c 1000 202.141.80.14 :

ping -p ff00 -c 1000 202.141.80.14 --- 2

a. Packet loss rate : 1. 0.1% packet loss

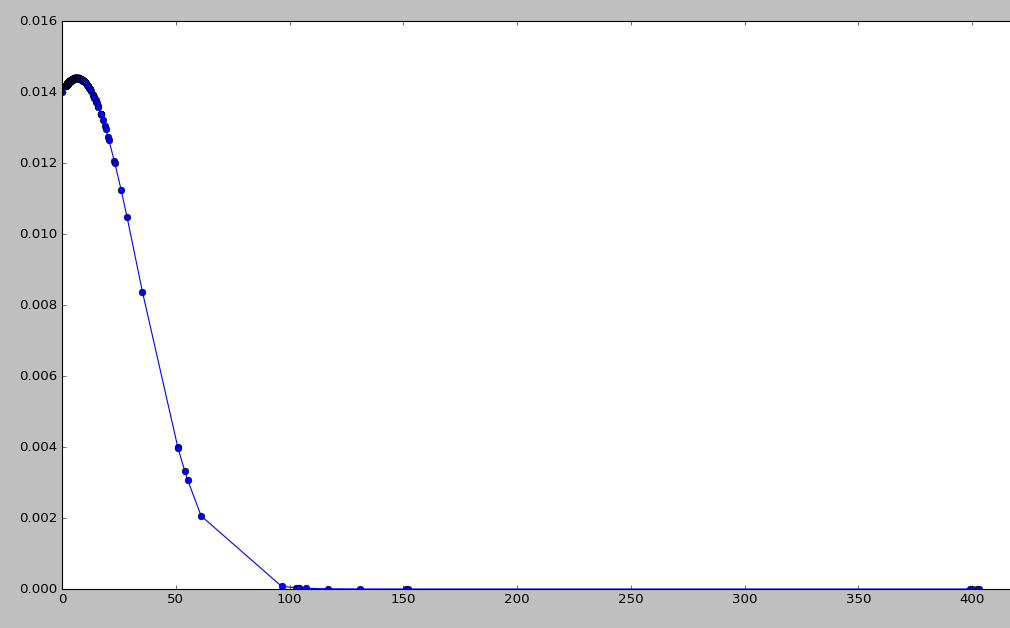
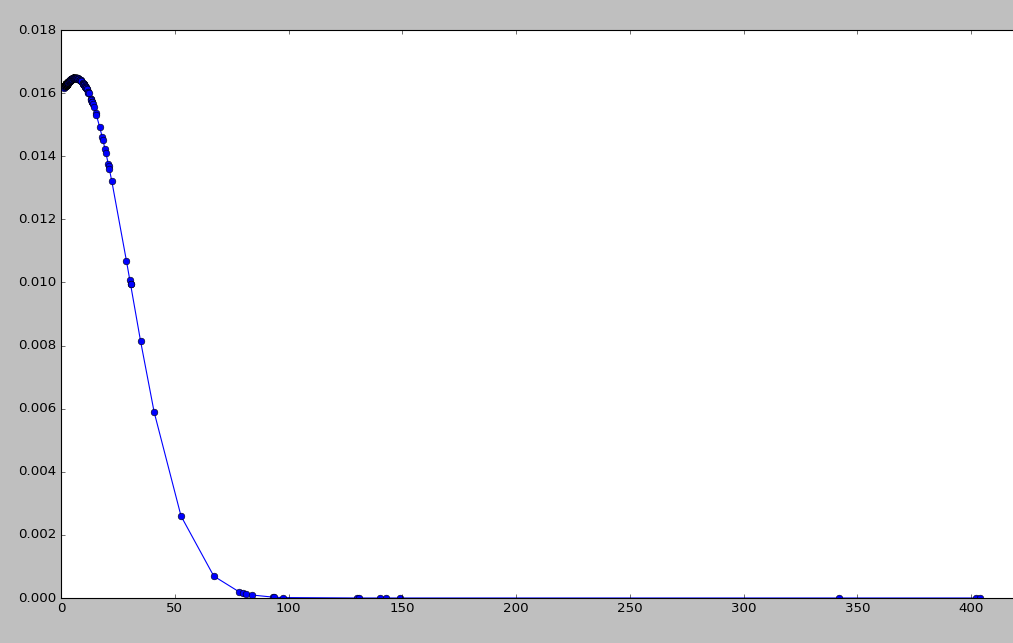
: 2. 0.2% packet loss

b.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Minimum (ms) | Maximum (ms) | Average (ms) | Median (ms) |
| ping -n -c 1000 202.141.80.14 | 1.455 | 402.971 | 6.180 | 2.47 |
| ping -p ff00 -c 1000 202.141.80.14 | 1.565 | 403.603 | 6.466 | 2.58 |

Graph for ping latencies :

ping -n -c 1000 202.141.80.14: ping -p ff00 -c 1000 202.141.80.14 :



ping -f took more time than ping -n during the experiment reason being , ping -n avoid the time taken in DNS checking of the provided IP address taking less time than normal ping time . While -f pattern will check for pattern in the packets taking more time than normal ping . Therefore second command will take more time than first .

**ANS 4.**  IFCONFIG options :

a. $ ifconfig : without any option ifconfig displays active network interface on the computer .

b. $ ifconfig -a : display information of active or in-active network interface on the computer .

c. $ ifconfig [interface ] : will display about the specific network interface .

d. $ ifconfig [interface] up : will activate the [interface] network interface .

e. $ ifconfig [interface] down: will deactivates the specific network interface .

f. $ ifconfig [interface] IP : This command will assign IP to specified interface .

g. $ ifconfig [interface] netmask nm : This command will set netmask of interface to the specified value

h. $ ifconfig [interface] broadcast bd : This command will set broadcast of interface to bd .

i. $ ifconfig [interface] mtu value : will set mtu of interface to value .

Output of ‘ifconfig’ :

eno1 Link encap:Ethernet HWaddr b0:5a:da:d7:a4:8a

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:1000

RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

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:983 errors:0 dropped:0 overruns:0 frame:0

TX packets:983 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1

RX bytes:62445 (62.4 KB) TX bytes:62445 (62.4 KB)

wlo1 Link encap:Ethernet HWaddr 48:e2:44:01:63:a7

inet addr:192.168.1.101 Bcast:192.168.1.255 Mask:255.255.255.0

inet6 addr: fe80::3d20:19dd:66b3:cd91/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:24119 errors:0 dropped:0 overruns:0 frame:0

TX packets:8963 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:31087950 (31.0 MB) TX bytes:1280200 (1.2 MB)

a. lo :The loopback interface does not represent any actual hardware, but exists so applications running on your computer can always connect to servers on the same machine.

b. Link encap:Ethernet --> denotes that device is ehternet related interface

Link encap:Local Loopback --> denotes loopback interface

c. HWaddr b0:5a:da:d7:a4:8a --> this is hardware address or MAC address which is unique to each ethernet card . therefore loopback interface doesnt have any MAC address .

d. inet addr --> represent IPv4 address of the interface .

Bcast --> denotes the broadcast address

Mask --> Is the network mask

e. UP , BROADCAST , RUNNING ,MULTICAST , LOOPBACK are flags

UP --> this denotes that kernel modules related to network interface is loaded (it's enabled to transmit and receive network packets) .

BROADCAST -->Indicates that the interface is configured to handle broadcast packets . This is important for obtaining the IP address when hostname is given from DHCP server .

RUNNING -->Indicates that the network interface is operational and is ready to accept the data .

MULTICAST -->interface is configured to handle multicast packets .

LOOPBACK --> interface is a loopback interface

f. MTU : Maximum Transmission Unit for which the interface is configured .MTU is a link layer charcteristic which provides limit on the size of the Ethernet frame . if datagram from IP layer are larger then MTU then they are breaked into small part to have size lower than MTU .

Metric : it calculates the cost of transmitting data it help to decide when there are more than one interface to send packet to same location . The value of this property decides the priority of the device . if we have two Ethernet cards and we want to forcibly make your machine use one card over the other in sending the data. Then you can set the Metric value of the Ethernet card which you favor lower than that of the other Ethernet card

g. RX packet --> number of recieved packet via interface .

TX packet --> number of transmitted packet via interface .

error --> number of damaged packets recieved . dropped is the number of packet dropped due to reception error

overruns --> number of received packets that experienced data overruns

frames --> number received packets that have frame errors.

carrier --> packets that experience loss of carrier

h. collision --> The number of transmitted packets that experienced Ethernet collisions. A nonzero value of this field indicates possibility of network congestion.

Txqueuelen -->The field provides the information about the configured length of transmission queue.

RXByte --> total number of byte recieved over the interface .

TXByte --> total number of byte transmitted over the interface .

Output of ‘route’ :

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

default 192.168.1.1 0.0.0.0 UG 600 0 0 wlo1

link-local \* 255.255.0.0 U 1000 0 0 wlo1

192.168.1.0 \* 255.255.255.0 U 600 0 0 wlo1

Each Linux / UNIX / Windows or any computer that uses TCP/IP need to make routing decision. Routing table is used to control these decisions . Almost all computers and network devices connected to Internet use routing tables to compute the next hop for a packet

**a. “Destination”** indicates the pattern that the destination of a packet is compared to. When a packet has to be sent over the network, this table is examined top to bottom, and the first line with a matching destination is then used to determine where to send the packet .

**b.** **“Gateway”** tells the computer where to send a packet that matches the destination of the same line. An asterisk (\* ) here means “send locally”, because the destination is supposed to be on the same network.

**c. “Genmask”** it tells how many bits from the start of the ip address are used to identify the subnet . as a rule of thumb, it is 255 for any non-zero part of the destination and 0 for parts of the destination that are 0

**d. “Flags”** shows which flags apply to the current table line. “U” means Up, indicating that this is an active line. “G” means this line uses a Gateway.

**e. “MSS”** lists the value of the Maximum Segment Size for this line. The MSS is a TCP parameter and is used to split packets when the destination has indicated that it somehow can’t handle larger ones .

**f. “Windows”** indicates how many TCP packets can be sent before at least one of them has to be ACKnowledged .

**g. “irtt”** stands for Initial Round Trip Time and may be used by the kernel to guess about the best TCP parameters without waiting for slow replies .

**h. “Iface”** tells which network interface should be used for sending packets that match the destination.

**ANS 5 .** Netstat is a useful tool for checking network and Internet connections . Netstat displays network connections for the TCP/IP (both incoming and outgoing), routing table, and a number of network interface and network protocol statistics .

**Netstat -at** is used to display all the incoming and outgoing tcp connections . ‘Netstat -a -t’ output is :

Active Internet connections (servers and established)

Proto Recv-Q Send-Q Local Address Foreign Address State

tcp 0 0 localhost:mysql \*:\* LISTEN

tcp 0 0 localhost:5939 \*:\* LISTEN

tcp 0 0 ani:domain \*:\* LISTEN

tcp 0 0 localhost:ipp \*:\* LISTEN

tcp 0 0 192.168.1.101:52986 202.141.80.24:3128 ESTABLISHED

tcp 0 0 192.168.1.101:52584 202.141.80.24:3128 ESTABLISHED

tcp 1 0 192.168.1.101:52984 202.141.80.24:3128 CLOSE\_WAIT

tcp6 0 0 ip6-localhost:ipp [::]:\* LISTEN

a. **“Proto”** tells if the socket listed is TCP or UDP .

b. **“Recv-Q” & “Send-Q”** how much data is in the queue for that socket, waiting to be read (Recv-Q) or sent (Send-Q). In short: if this is 0, everything’s ok, if there are non-zero values anywhere, there may be trouble .

c. **“Local Address” & “Foreign Address”** tell to which hosts and ports the listed sockets are connected. The local end is always on the computer on which you’re running netstat and the foreign end is about the other computer (could be somewhere in the local network or somewhere on the internet).

d. **“State”** tells in which state the listed sockets are . Diffrent possible state are .

SYN\_SEND Indicates active open

SYN\_RECEIVED Server just received SYN from the client.

ESTABLISHED Client received server's SYN and session is established.   
 LISTEN Server is ready to accept connection.

FIN\_WAIT\_1 Indicates active close.

TIMED\_WAIT Client enters this state after active close.

CLOSE\_WAIT Indicates passive close. Server just received first FIN from a client.

FIN\_WAIT\_2 Client just received acknowledgment of its first FIN from the server.

LAST\_ACK Server is in this state when it sends its own FIN.

CLOSED Server received ACK from client and connection is closed.

**Netstat -r**  is used to display routing table . Kernel IP routing table

Destination Gateway Genmask Flags MSS Window irtt Iface

default 192.168.43.1 0.0.0.0 UG 0 0 0 wlo1

link-local \* 255.255.0.0 U 0 0 0 wlo1

192.168.43.0 \* 255.255.255.0 U 0 0 0 wlo1

**a. “Destination”** indicates the pattern that the destination of a packet is compared to. When a packet has to be sent over the network, this table is examined top to bottom, and the first line with a matching destination is then used to determine where to send the packet .

**b.** **“Gateway”** tells the computer where to send a packet that matches the destination of the same line. An asterisk (\*) here means “send locally”, because the destination is supposed to be on the same network.

**c. “Genmask”** it tells how many bits from the start of the ip address are used to identify the subnet . as a rule of thumb, it is 255 for any non-zero part of the destination and 0 for parts of the destination that are 0

**d. “Flags”** shows which flags apply to the current table line. “U” means Up, indicating that this is an active line. “G” means this line uses a Gateway.

**e. “MSS”** lists the value of the Maximum Segment Size for this line. The MSS is a TCP parameter and is used to split packets when the destination has indicated that it somehow can’t handle larger ones .

**f. “Windows”** indicates how many TCP packets can be sent before at least one of them has to be ACKnowledged .

**g. “irtt”** stands for Initial Round Trip Time and may be used by the kernel to guess about the best TCP parameters without waiting for slow replies .

**h. “Iface”** tells which network interface should be used for sending packets that match the destination.

**Netstat -i** can be used to display the status of network interface . And Netstat -a -i will show the interface

**LOOPBACK INTERFACE :**

The loopback interface does not represent any actual hardware, but exists so applications running on your computer can always connect to servers on the same machine. it is helpful when a server offering a resource you need is running on your own machine. For IPv4, the loopback interface is assigned all the IPs in the 127.0.0.0/8 address block. That is, 127.0.0.1 through 127.255.255.254 all represent your computer. For most purposes, though, it is only necessary to use one IP address, and that is 127.0.0.1 .

**ANS 6 .**  Five host selected are same question 2 .Trace-out experiment :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | HOSTNAME | Hop-Count(morning) | Hop-Count(After-noon) | Hop-Count(Midnight) |
| 1 | www.iitd.ac.in | 15 | 18 | 15 |
| 2 | [www.ucla.edu](http://www.ucla.edu/) | 13 | 13 | 13 |
| 3 | www.iitkgp.ac.in | 8 | 8 | 8 |
| 4 | english.spbu.ru | 12 | 12 | 12 |
| 5 | web.mit.edu | 6 | 4 | 6 |

For IIT-d and IIT -Kgp inital four hops were sames ( 207.86.208.17 , 207.88.13.122 , 207.88.14.189 ,206.111.5.34), denoting same route till india then having diffrent route. Mit and st. Petersburg university also have inital two same hops (67.219.148.9 , 184.105.25.73) .

the ttl value which is subracted per hop may have been reduced to 0 many people block ICMP/ping for security reasons, like preventing hackers from getting information about open ports and staving off denial of service attacks. When ping is blocked, the server doesn’t respond at all, resulting in “request timed out” messages that prevent traceroute from ever being able to map the path to the final destination .

Tracert works by targeting the final hop, but limiting the TTL and waiting for a time exceeded message, and then increasing it by one for the next iteration. Therefore, the response it gets is not an ICMP echo reply to the ICMP echo request from the host along the way, but a time exceeded message from that host - so even though it is using ICMP, it is using it in a very different way.

**ANS 7.**  ‘arp -a’ and ‘arp -e’ comands will show the entire arp table of the machine .

a. ‘Address’ colum contains IP address of the connection from which we are connected , for ex in my case 192.168.1.1 is the ip address of the wireless router .

b. ‘HWType’ : tells which type of network like ethrent , ARCNET ,LocalNet etc .

c. ‘Hwaddress’ : shows hardware address of the network interface

arp -d address will delete the table entrie with address .

arp -s address Hwaddress will set up new entire in arp table with address and HWaddress .

to add two new Host to arp table :sudo arp -s 192.168.1.2 98:de:d0:4b:65:a0 , sudo arp -s 192.168.1.3 98:de:d0:4b:65:a0

Address HWtype HWaddress Flags Mask Iface

192.168.1.3 ether 98:de:d0:4b:65:a0 CM wlo1

192.168.1.1 ether 98:de:d0:4b:65:a0 C wlo1

192.168.1.2 ether 98:de:d0:4b:65:a0 CM wlo1

When a host sends an ARP request to another host and a reply is received the sender caches the received information is a table for later use.

We can use trail and error method to find out the cache time out . We could assume it is 60 min and fast clock by 60 min to check if the neighbour entries are still there or not .

time out of cache could be seen by : cat /proc/sys/net/ipv4/neigh/default/gc\_stale\_time

In my pc it shows value 60 which means 60 second .

Lets make a situation where two ip address have same Hwaddress . I added an IP 192.168.1.2 with same HWaddress as of the router with ip 192.168.1.1 and then ping 192.168.1.2

ping 192.168.1.2

PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.

From 192.168.1.1: icmp\_seq=2 Redirect Host(New nexthop: 192.168.1.2)

From 192.168.1.1: icmp\_seq=3 Redirect Host(New nexthop: 192.168.1.2)

From 192.168.1.1: icmp\_seq=4 Redirect Host(New nexthop: 192.168.1.2)

--- 192.168.1.2 ping statistics ---

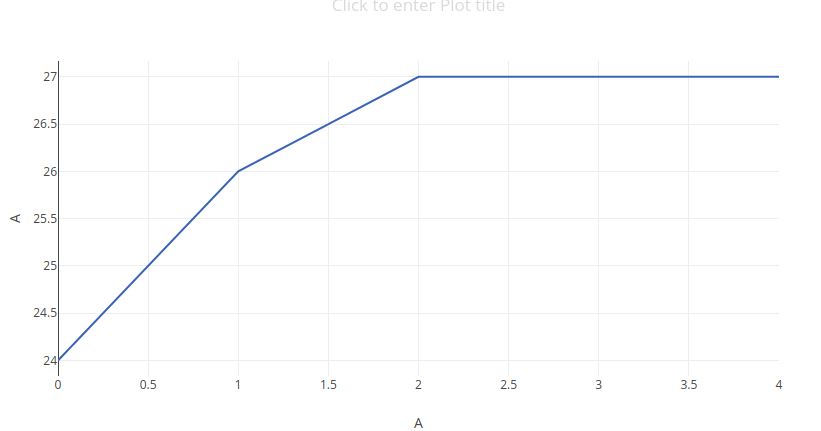
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3002ms

means that packets send to 192.168.1.2 are sent through 192.168.1.1 which later cant find 192.168.1.2 and hence lead in 100% packet loss . So if we have a router which is connected to pc with two diffrent ip

ping will be send to one and afterward will hop to the second ip address .

**ANS 8 .** nmap -n –sP 172.16.112.0/26

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Timing | 7:00 AM | 11:00 AM | 3:00PM | 8:00 PM | 12:00PM |
| No of Computers up | 24 hosts up | 26 hosts up | 27 hosts up | 27 hosts up | 27 hosts up |



there is almost a constant number of computers active all the time in day . So two situation may happen 1. people in the lab are doing some machine learning task therefore they have left there PCs active .

2. otherwise they might have just left opened pc to use it as squid server and acess internet in hostel during restricted hours .