CS349 Networks Lab - Assignment 1

Inderpreet Singh Chera (160101035)

Answer 1.

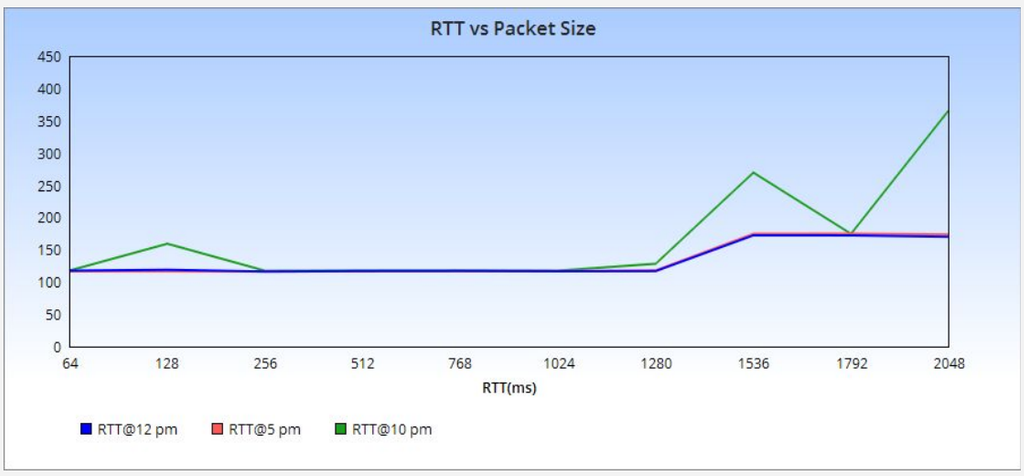
a) **-c count** option is required to specify the number of echo requests to send with ping command.
b) **-i interval** option required to set time interval (in seconds) between two successive ping requests.
c) Command to send ECHO_REQUEST packets to the destination one after another without waiting for a reply is **ping -l <preload> <hostname/ip>**. Normal users can send at max 3 such ECHO_REQUESTs.
d) **ping -s <packetsize> <hostname/ip>** is command to set packet size (in bytes). If packet size is 64 bytes, then total packet size will be 64 bytes + 8 bytes ICMP header + 20 bytes IP header = **92 bytes**.

Answer 2.

- Readings were taken at **12:00 pm**., **5:00 pm** and **11:00 pm**.
- Test PC was connected to **DIGITALOCEAN VPN** (Bangalore, India) while performing the experiment.
- '**iitg.ac.in**' was chosen for experimenting with packets of size from 64 bytes to 2048 bytes.

DESTINATION HOST ADDRESS	IP ADDRESS	GEOGRAPHIC LOCATION	Avg. RTT1 (ms)	Avg, RTT2 (ms)	Avg, RTT3 (ms)	Total Avg. RTT (ms)
iitg.ac.in	14.139.196.22	India	121.969	116.260	475.082	237.770
youtube.com	172.217.26.174	United States	66.872	64.823	66.988	66.228
bookmyshow.com	104.16.123.37	United States	66.586	81.290	64.614	70.83
libgen.io	93.174.95.87	Netherlands	259.150	204.062	335.457	266.223
yandex.ru	77.88.55.88	Russia	218.559	221.105	214.870	218.178

Size (bytes)	64	128	256	512	768	1024	1280	1536	1792	2048
Avg. RTT1(ms)	117.441	118.878	116.129	116.902	117.055	116.566	116.832	172.235	172.152	169.922
Avg. RTT2(ms)	116.443	116.372	116.331	116.436	116.754	116.939	118.005	174.641	174.828	173.665
Avg. RTT3(ms)	117.696	159.233	117.332	117.658	117.676	117.465	128.251	269.348	174.471	365.708



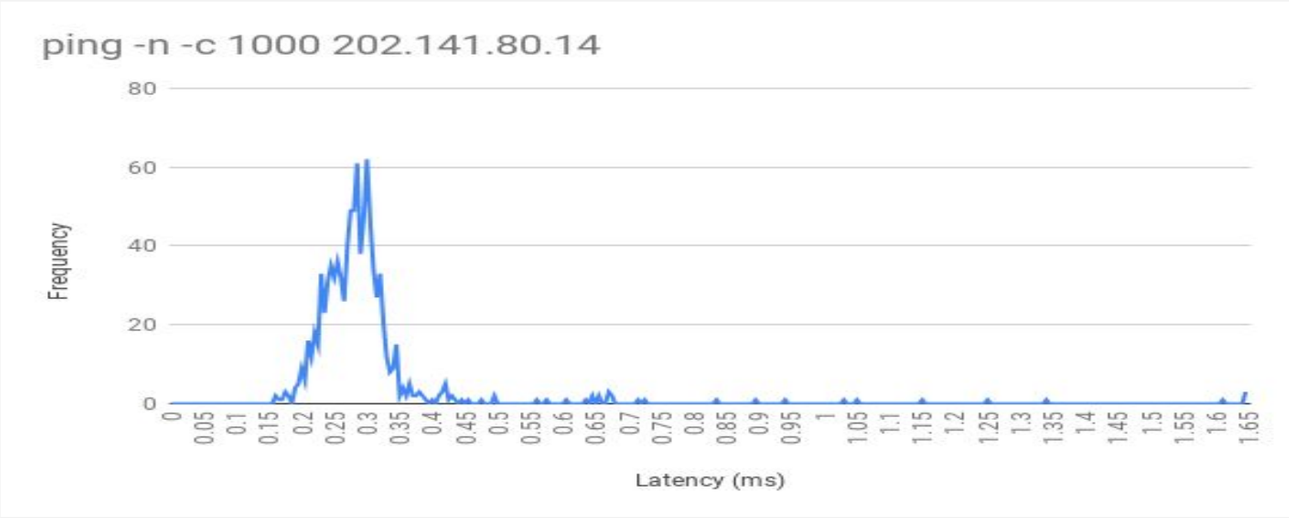
Packet Loss: There were some sites (for eg., netflix.com) that showed 100% packet loss. It can be because such sites are using **firewall to block ICMP packets** due to some past cases of DoS attack using ping. Other than this in each experiment **I got 0% packet loss**. But, in general packet loss can be greater than 0% because of **network congestion** and traffic. Some packets may collide with other packets in the network and result in packet loss. The ICMP Packets have **lower priority**. So they might take longer time to process in some destination server's queue.

- **Distance and avg. RTT**: From my experiments, I conclude that measured RTTs are **weakly correlated** with the geographical distance of the hosts. From above experiment, youtube.com should have more ping than iitg.ac.in, but it is not the case as server for youtube is in USA where iitg.ac.in is in India. Intuitively, they should be strongly correlated because larger the distance, longer it takes for a packets to propagate and also as the packets have to travel through more number of nodes, it increases the processing delay. But, as factors like network traffic and server capacities and many other also come into play, it becomes a weak correlation.

- **Time of Day**: We observe that RTT's vary with time of the day. At different times, the congestion in network is different. We observe that there is maximum congestion at around 10pm. 12 noon and 5pm have almost same network congestion.
- **Packet Size Effect:** From the above table, we can clearly observe that the Round Trip Time (RTT) is almost the same for packets upto size 1280 Bytes. After that, we observe a sudden jump in the RTT. This can be explained by the fact that Maximum Transmission Unit (MTU) is 1500 bytes by default. If the packet size is less than 1500 Bytes, then the data is padded to make the size 1500 bytes. Hence, for packets with size less than 1500 Bytes, the RTT is almost same. If packet size is more than 1500 bytes, then the packet is broken into two frames of size 1500 bytes. Hence we observe increase in the RTT for packets of size 1536 bytes to 2048 bytes.

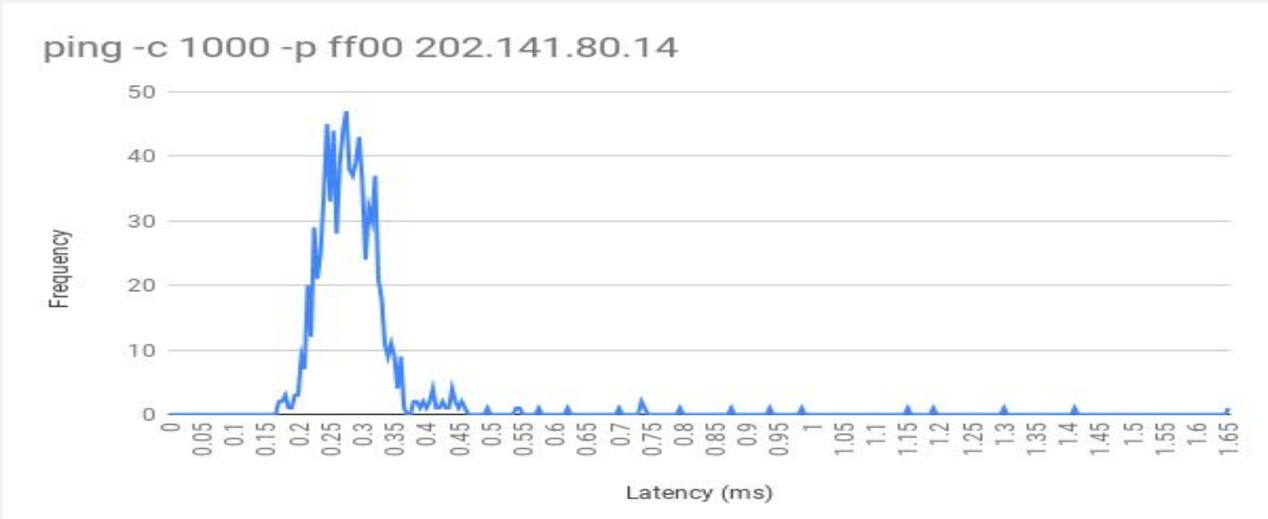
Answer 3.

Command	Packets Sent	Packets Received	Packet Loss Rate	Minimum Latency	Maximum Latency	Mean Latency	Median Latency
ping -n -c 1000 202.141.80.14	1000	963	3%	0.159 ms	6.082 ms	0.301 ms	0.276 ms
ping -p ff00 -c 1000 202.141.80.14	1000	912	8%	0.166 ms	4.385 ms	0.303 ms	0.281 ms



We observe that both of the curves resemble the shape of Normal Distribution. They differ in 2 aspects

- Firstly, no attempt will be made to lookup symbolic names for host addresses when using '-n', hence it will be faster. So, the **mean latency is higher** in second case than the mean latency in the first case.
- Secondly, in case of '-p ff00' packet will be filled with '1111111100000000'. Now, this causes problems with synchronisation of clock because only one transition is present in the padding, from 1 to 0. Hence, the clocks are more likely to go out of synchronisation in second case and we observe that the **packet loss is higher** in the second case.



Answer 4.

The command ifconfig is used to configure the kernel-resident network interfaces. My machine has a **wired ethernet connection (eth0)** and a **loopback interface (lo)**.

```
root@root:~# ifconfig -a -v
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
       inet 172.16.114.107  netmask 255.255.255.128  broadcast 172.16.114.127
       inet6 fe80::b283:feff:fe8d:fd83  prefixlen 64  scopeid 0x20<link>
       ether b0:83:fe:8d:fd:83  txqueuelen 1000  (Ethernet)
       RX packets 538557  bytes 411386182 (392.3 MiB)
       RX errors 0  dropped 1721  overruns 0  frame 0
       TX packets 217993  bytes 45314015 (43.2 MiB)
       TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
       inet 127.0.0.1  netmask 255.0.0.0
       inet6 ::1  prefixlen 128  scopeid 0x10<host>
       loop  txqueuelen 1000  (Local Loopback)
       RX packets 56  bytes 2688 (2.6 KiB)
       RX errors 0  dropped 0  overruns 0  frame 0
       TX packets 56  bytes 2688 (2.6 KiB)
       TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- **Flags** - specify type of services that interface is entitled to.
- **UP** - This flag indicates that the kernel modules related to the interface has been loaded.
- **BROADCAST & MULTICAST** - Denotes that the device supports broadcasting - a necessary characteristic to obtain IP address via DHCP and multicasting respectively.
- **RUNNING** - The interface is ready to accept data.
- **MTU** - short form for Maximum Transmission Unit is the size of each packet received by the Ethernet card. The value of MTU for all Ethernet devices by default is set to 1500.
- **inet and inet6** - indicates IPv4 and IPv6 address assigned to the machine.
- **Broadcast and Netmask** - denotes the broadcast address and network mask of the interface.
- **Scopeid** - Gives id of scope and type of scope. Scope is scope of IPv6 address. It can be *link-local* or *global*. Link-local address is used in local area network and is not routable. Global address is routable.
- **Ether** - gives the MAC address which is unique to each Ethernet card which is manufactured.
- **txqueuelen** - This denotes the length of the transmit queue of the device.
- **RX packets, errors, dropped, overruns & frames** - It shows number of packets received, # damaged packets received, # dropped packets received, # received packets that experienced data overruns and # misaligned packets (i.e. frames with length not divisible by 8) respectively. As dropped value is greater than 0, it could mean interface is failing or there is some congestion in network.
- **TX packets, errors, dropped & overruns** - These are similar to TX equivalents where only difference is packets are transmitted instead of received.
- **TX carriers** is a number of packets that experienced loss of carriers.
- **RX Bytes, TX Bytes** - These indicate the total amount of data that has passed through the Ethernet interface either way.
- **collisions** - The value of this field should ideally be 0. If it has a value greater than 0, it could mean that the packets are colliding while traversing your network

```
root@root:~# route -n
Kernel IP routing table
Destination  Gateway      Genmask        Flags Metric Ref Use Iface
0.0.0.0      172.16.112.1 0.0.0.0          UG 100     0    0   eth0
172.16.112.1 0.0.0.0      255.255.255.255  UH 100     0    0   eth0
172.16.114.0 0.0.0.0      255.255.255.128   U        100     0    0   eth0
```

- **Route command** - show / manipulate the IP routing table.
- **Destination** column identifies the destination network or destination host.
- **Gateway**: The gateway address or '*' if none set.
- **Genmask**: The netmask for the destination net. '255.255.255.255' for a host destination and '0.0.0.0' for the default route.
- **Flags**: U indicates route is up and G indicates specified gateway is used & H is target is a host.
- **Metric & Ref**: The distance to the target (counted in hops) and number of references to this route resp..
- **Use**: Number of references to this route.
- **Iface**: Interface to which packets for this route will be sent.

Route command options:

-n: display the numerical IP address

-v: select verbose operation

-add: add a route

-host: specifies that the target is a host.

-C: list the kernel's routing cache information

-del: delete a route

-net: specifies that the target is a network

Answer 5.

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

netstat --tcp command is used to check all the tcp connections established.

```
root@root:~# netstat --tcp
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 root:39718              202.141.80.20:3128      ESTABLISHED
tcp        0      0 root:39708              202.141.80.20:3128      ESTABLISHED
tcp        0      0 root:39710              202.141.80.20:3128      ESTABLISHED
tcp        0      0 root:39800              202.141.80.20:3128      ESTABLISHED
tcp        0      0 root:39712              202.141.80.20:3128      ESTABLISHED
tcp        0      0 root:39740              202.141.80.20:3128      TIME_WAIT
tcp        0      0 root:39902              202.141.80.20:3128      ESTABLISHED
tcp        0      0 root:39888              202.141.80.20:3128      TIME_WAIT
tcp        0      0 root:39706              202.141.80.20:3128      ESTABLISHED
```

- **Proto** column tells us protocol (tcp, udp, udpl, raw) used by the socket.
- The **Recv-Q** and **Send-Q** columns tell us how much data is in the queue for that socket, waiting to be read or sent.
- **Local Address**: Address and port number of the local end of the socket.
- **Foreign Address**: Address and port number of the remote end of the socket.
- **State**: The state of the socket can be listen (waiting for an incoming connection), established (connections which are established), and time wait (the foreign or remote machine has already closed the connection, but that the local program somehow hasn't followed suit).

netstat -r shows the Kernel Routing Table of the Machine.

```
root@root:~# netstat -r
Kernel IP routing table
Destination  Gateway      Genmask           Flags   MSS Window  irtt Iface
default      _gateway     0.0.0.0             UG      0    0      0  eth0
_gateway     0.0.0.0      255.255.255.255     UH      0    0      0  eth0
172.16.114.0 0.0.0.0      255.255.255.128     U       0    0      0  eth0
```

- **Destination**: The destination network or destination host.
- **Gateway**: The gateway address or '*' if none set.
- **Genmask**: The netmask for the destination net; '255.255.255.255' for a host destination and '0.0.0.0' for the default route.
- The **Flags** column displays the flags that describe the route - G(route uses a gateway), U(interface is up), H(Only a single host can be reached through the route), D(route is dynamically created), M(route is set if the table entry was modified by an ICMP redirect message), !(route is a reject route and datagrams will be dropped).
- **MSS**: Default maximum segment size for TCP connections over this route.
- **Window**: Default maximum segment size for TCP connections over this route.
- **irtt**: Initial RTT (Round Trip Time). The kernel uses this to guess about the best TCP protocol parameters without waiting on (possibly slow) answers.
- **Iface**: Interface to which packets for this route will be sent.

netstat -i can be used to display network interface status.

```
root@root:~# netstat -i
Kernel Interface table
Iface        MTU    RX-OK  RX-ERR RX-DRP RX-OVR      TX-OK  TX-ERR TX-DRP TX-OVR    Flg
eth0         1500   964798 0       3480   0          419100 0       0       0       BMRU
lo           65536  78     0       0      0          78     0       0       0       LRU
```

The **loopback** device is a special, virtual network interface that the computer uses to communicate with itself. It is used mainly for **diagnostics and troubleshooting**, and to connect to servers running on the local machine. The loopback interface does not represent any actual hardware, but exists so applications running on the computer can always connect to servers on the same machine. For example, if you run a web server, you have all your web documents and could examine them file by file on the local machine. For IPv4, the loopback interface is assigned all the IPs in the 127.0.0.0/8 address block (i.e.127.0.0.1 through 127.255.255.254). It can also perform functions like **device identification**, **routing information**, **packet filtering**.

Answer 6.

- The readings were taken at **12:00 pm, 5:00 pm and 11:00 pm**.
- Test PC was connected to **DIGITALOCEAN VPN** (Bangalore, India) while performing the experiment.

Hop count	iitg.ac.in	youtube.com	bookmyshow.com	libgen.io	yandex.ru
Hop count #1	11	12	10	10	12
Hop count #2	11	12	8	10	12
Hop count #3	11	12	8	10	12

- **10.8.0.1** and **139.59.64.253** (VPN service provider IP) were common hosts in all the tests. **138.197.249.18** was the common hop between youtube.com, bookmyshow.com, libgen.io and yandex.ru. Also, **202.56.198.57** was common between routes of libgen.io, bookmyshow.com and yandex.ru. 219.65.110.189 was a common hop at different time of day between youtube.com and iitg.ac.in.
- The route to the hosts **changes** at different times of the day in the experiments **because of network congestion**. The packets are redirected by the nodes to take a route having less traffic. The **load balancing** is done to reduce the load of congested path. In my experiment, bookmyshow.com showed 2 different number of hops at different time of day.
- Sometimes, traceroute **may not be** able to find complete path to the host. It may be because some servers/hosts may have not been configured to **respond to the ICMP Traffic** or may have set up **firewalls** which block the ICMP Traffic. However, they still **send the data to the next hop** as we still sometimes get the output of traceroute. This could be inferred by the fact that if send tcp packets instead of ICMP packets for the traceroute most of the servers respond correctly. Many network providers disable ICMP traffic if their network is under heavy load
- It **is possible** to find the route to certain hosts which fail to respond with ping experiment. Firstly, It is because of the reason that although ping and traceroute both use ICMP packets but they **follow slightly different protocols**. Ping is straight ICMP from point A to point B, that traverses networks via routing rules. Whereas, traceroute sends packets with TTL values that gradually increase from packet to packet. Routers decrement TTL values of packets by one and discard packets whose TTL value has reached zero, returning the ICMP error (ICMP Time Exceeded). Traceroute looks for the ICMP Time exceeded packet and not the ICMP Reply Packet, and that is why it might be possible. Secondly, traceroute can be forced to send **tcp packets** instead of ICMP packets hence we are guaranteed to have the path, but ping only works with ICMP packets and can not with tcp packets.

Answer 7.

arp displays and modifies entries in the Address Resolution Protocol (ARP) cache. Command to show full arp table is **arp** or **arp -v**.

```
joker@joker:~$ sudo arp -v
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   ec:44:76:74:60:41   C                     enp7s0
10.42.0.113              ether   00:1e:64:f7:20:dd   C                     wlp6s0
10.42.0.117              ether   00:1e:64:f4:f9:39   C                     wlp6s0
10.19.0.138              ether   fc:3f:db:34:61:6c   C                     enp7s0
Entries: 4       Skipped: 0   Found: 4

joker@joker:~$ sudo arp -sv 10.0.0.1 -i enp7s0 ff:ff:ff:ff:ff:ff
arp: SIOCSARP()

joker@joker:~$ sudo arp -sv 10.0.0.2 -i enp7s0 ff:ff:ff:ff:00:ff
arp: SIOCSARP()

joker@joker:~$ sudo arp -v
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   ec:44:76:74:60:41   C                     enp7s0
10.42.0.113              ether   00:1e:64:f7:20:dd   C                     wlp6s0
10.0.0.2                 ether   ff:ff:ff:ff:00:ff   CM                    enp7s0
10.0.0.1                 ether   ff:ff:ff:ff:ff:ff   CM                    enp7s0
10.42.0.143              ether   b4:ef:fa:52:14:5b   C                     wlp6s0
10.19.1.111              ether   fc:3f:db:8c:51:f1   C                     enp7s0
10.42.0.117              ether   00:1e:64:f4:f9:39   C                     wlp6s0
10.19.0.138              ether   fc:3f:db:34:61:6c   C                     enp7s0
Entries: 8       Skipped: 0   Found: 8
```

arp command shows **ip addresses** of the pc, correspondings **hardware addresses** and **hardware type** for neighbouring devices. It also show **flags** mask where each complete entry in the ARP cache will be marked with the C flag. Permanent entries are marked with M and published entries have the P flag. **Hardware Interface** shows the network interface through which local user is connected to this subnet.

To delete an entry for a host from the arp cache we use **arp -d <ip-address>**. To add an entry in arp cache we use **arp -s <ip-address> <mac-address>**. **-i enp7so** is added with above mentioned command to specify the interface to be added. If not, added then it showed error: Network is unreachable in my pc. The above command adds permanent entry. To add a temporary entry add **temp** at the last of above command.

By default, stale entries stay in cache for about **60 seconds**. We can confirm it by running following command in terminal: `cat /proc/sys/net/ipv4/neigh/default/gc_stale_time`.

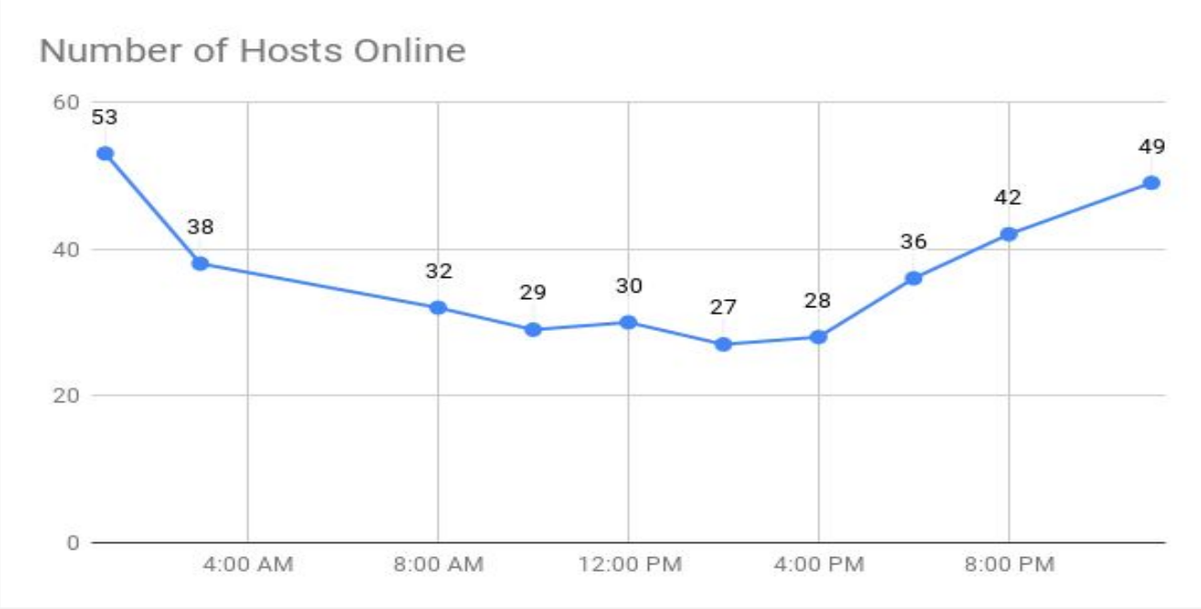
Trial and Error method to discover the timeout value could be to run command at interval of 5000 ms for 2 minutes and then for 30 seconds until 5 minutes. Then start **doubling** the wait time till we know time when the entry is deleted. After we know the time gap in which the entry was deleted, we can do **binary search** in that time.

There are 2 scenarios to map two IP addresses map to the same Ethernet address. In the first scenario if both the devices (in one of them arp entry is added) are in **same subnet range**, then it won't pose any problem as devices in same subnet range communicate with the help of hardware address. Whereas, if both devices are in **different subnet range** then they will require IP address to communicate and hence error will be popped.

Answer 8.

nmap has been done for Dihing and Siang Hostel at 10 different times of day.

```
joker@joker:~$ nmap -n -sP 10.1.0-3.0-255
```



From the above graph it can be observed that hosts are **most active at night around 1AM** and then no of hosts start decreasing. Then this number remains constant from about 8pm to 5pm and then again it starts increasing at night.