Assignment -1

CS 342: Networks Lab (September – November2020)

Name: Vaibhav Kumar Singh

Roll No.: 180101086

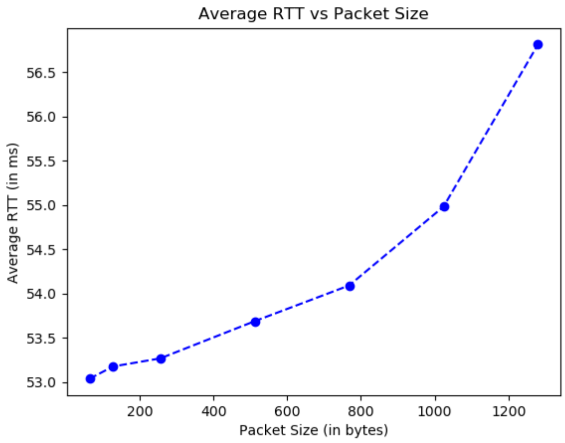
1. *ping* options:
   (a) `ping -c <count> <ip_address>`
   (b) `ping -i <interval> <ip_address>`
   (c) `ping -l <preload> <ip_address>` with a maximum value of 3 for normal users
   (d) `ping -s <payload_size> <ip_address>` and the total packet size for payload size of 32 bytes is 60 bytes (8 bytes for IP header and 20 bytes for ICMP header)
2. *ping*:

Host	Avg. RTT-1	Avg. RTT-2	Avg. RTT-3	Overall Avg. RTT	Packet loss %
Cloudflare	54.241	59.460	64.583	59.428	0%
Facebook	17.916	48.244	16.823	27.661	0%
Google	51.237	66.312	44.008	53.852	0%
Hackerrank	No RTT	No RTT	No RTT	No RTT	100%
Yahoo	331.452	341.744	340.988	338.061	0%
YouTube	42.277	58.345	24.634	41.752	0%

(a) Average RTT increases with increase in geographical distance.

(b) 100% packet loss occurs for Hackerrank. Possible reasons include restrictions on host ip address, network congestion and no network device attached to host ip address.

(c) I selected Cloudflare (1.1.1.1) for this experiment



Packet Size	64 bytes	128 bytes	256 bytes	512 bytes	768 bytes	1024 bytes	1280 bytes
Avg. RTT	53.037	53.176	53.267	53.687	54.090	54.980	56.810

(d) If all other conditions remain same, average RTT increases with increase in packet size because transmission delay is directly proportional to the packet size. Average RTT depends on the network congestion at that part of the day. More the traffic, greater the average RTT.

3. *ping -n* vs *ping -p ff00*:

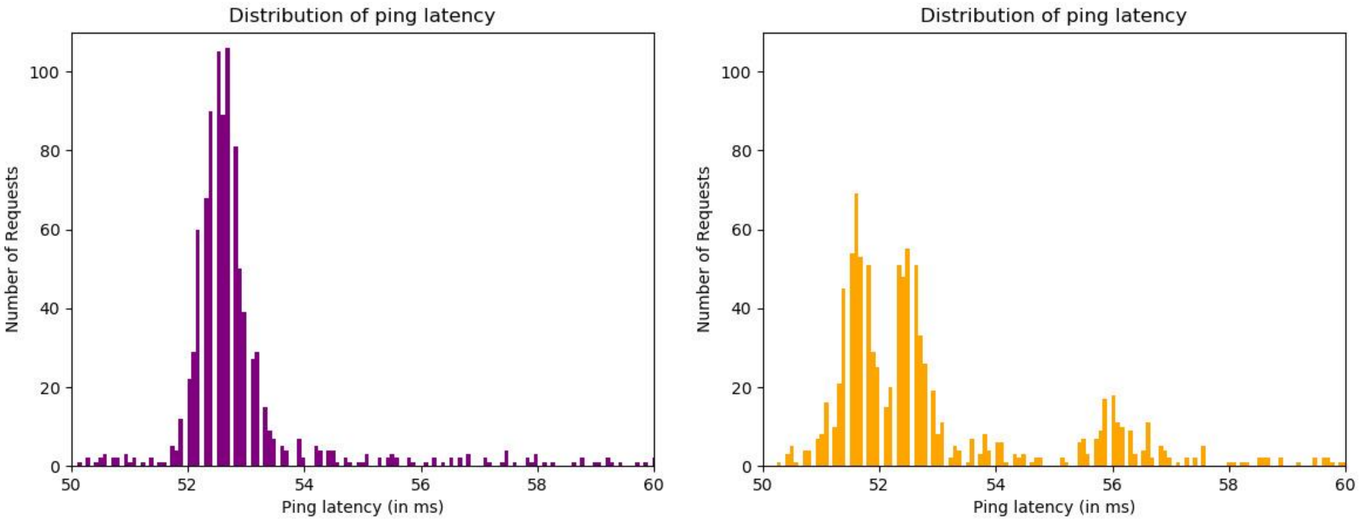
I selected Cloudflare (1.1.1.1) for this experiment

(a) Packet loss for first command is 0% whereas it is 0.2% for the second command.

(b)

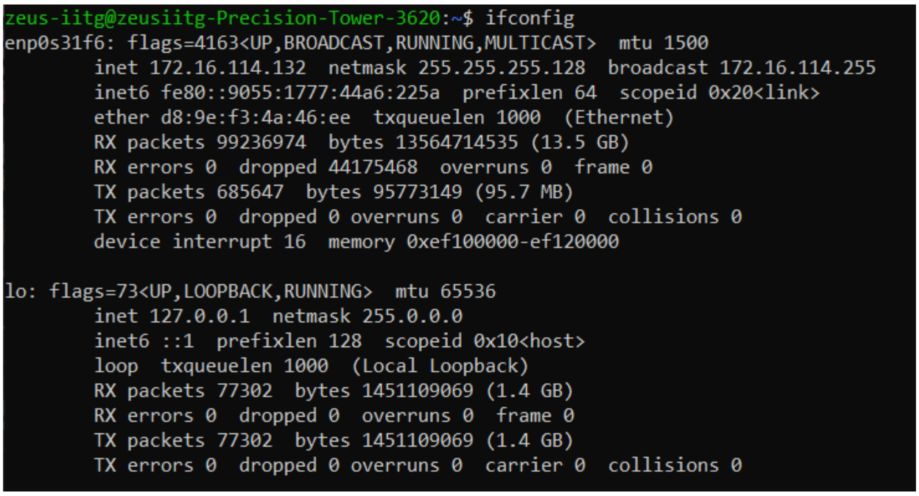
Command	Minimum latency	Maximum latency	Mean latency	Median latency
ping -n <IP_Address>	50.068	158.619	53.496	52.6
ping -p ff00 <IP_Address>	49.499	159.540	53.700	52.4

(c) Graph of Distribution of ping latencies for first and second command respectively.



(d) The '**-n**' option in ping gives numeric output only and no attempt is made to lookup symbolic names for host addresses. Because of this, the mean latency in first case (53.496 ms) is less than that in second (53.700 ms). The '**-p**' option is used to specify the content of the packet we send. This is useful for diagnosing data-dependent problems in a network. The pattern sent in the second case (ff00, i.e. 1111111100000000) has only one transition (from 1 to 0 at the 9th bit) and this is likely to cause synchronization problems between sender and receiver clocks.

4. *ifconfig* and *route*:



```
zeus-iitg@zeusiitg-Precision-Tower-3620:~$ ifconfig
enp0s31f6: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.114.132  netmask 255.255.255.128  broadcast 172.16.114.255
        inet6 fe80::9055:1777:44a6:225a  prefixlen 64  scopeid 0x20<link>
        ether d8:9e:f3:4a:46:ee  txqueuelen 1000  (Ethernet)
        RX packets 99236974  bytes 13564714535 (13.5 GB)
        RX errors 0  dropped 44175468  overruns 0  frame 0
        TX packets 685647  bytes 95773149 (95.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  memory 0xef100000-ef120000

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 77302  bytes 1451109069 (1.4 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 77302  bytes 1451109069 (1.4 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

(a) The following attributes belong to the enp0s31f6 (ethernet network peripheral) network interface: 1. **flags**: (a) **UP**: This flag indicates that the kernel modules related to the Ethernet interface has been loaded. (b) **BROADCAST**: This flag denotes that the Ethernet device supports broadcasting. (c) **RUNNING**: This indicates that the network interface is ready to accept data. (d) **MULTICAST**: This flag indicates that the Ethernet interface supports multicasting. 2. **mtu**: the maximum transmission unit is the size of the largest protocol data unit that can be communicated in a single network layer transaction. 3. **inet**: Indicates the machine IP address. 4. **netmask**: This is the network mask that shows how much of the address is routable, which determines whether the computer can connect directly to a device on the LAN or whether it needs to send the packet to a router. 5. **broadcast**: denotes the broadcast address. 6. **prefixlen**: specifies the prefix length of the network interface. 7. **txqueuelen**: This denotes the maximum number of packets in the transmission queue of the interface's device driver. 8. **RX Packets, TX Packets**: These denote the total number of packets received and transmitted respectively. 9. **RX Bytes, TX Bytes**: These denote the total amount of data received and transmitted respectively. 10. The output also has **RX and TX errors, drops and overruns**.

(b)

(i) **[-] arp** Enable or disable the use of the ARP protocol on this interface.

(ii) **mtu N** This parameter sets the Maximum Transfer Unit (MTU) of an interface.

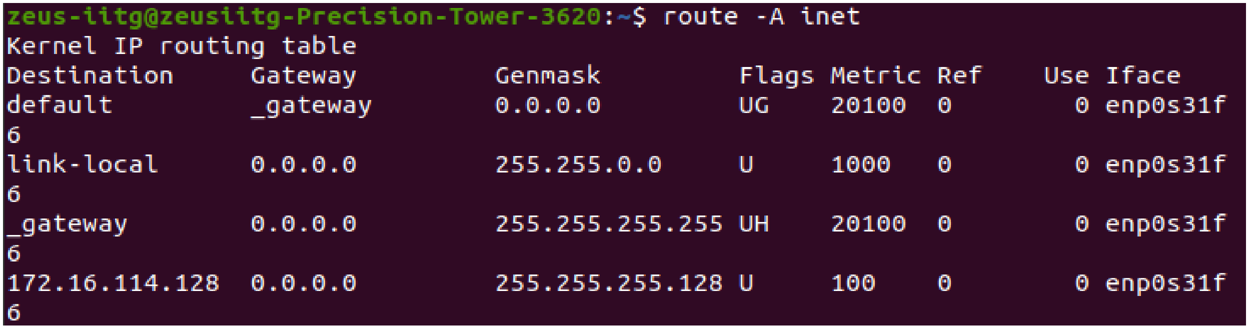
(iii) **[-]allmulti** Enable or disable **all-multicast** mode. If selected, all multicast packets on the network will be received by the interface.

(iv) **up** This flag causes the interface to be activated. It is implicitly specified if an address is assigned to the interface.

(c) Important attributes of the output of route command: 1. **Destination**: Contains the destination host. 2. **Gateway**: Contains the gateway address or '*' if not specified. 3. **Genmask**: Contains the netmask of destination net for a host destination. 4. **Iface**: Contains the interface of the destination of the packets of this route.

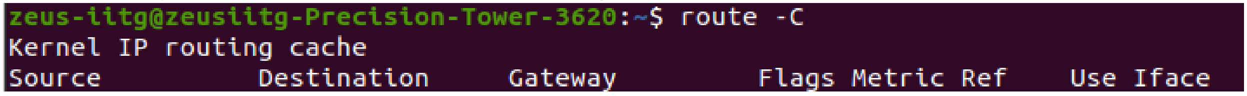
(d)

(i) **-A family** use the specified address family (eg. inet)



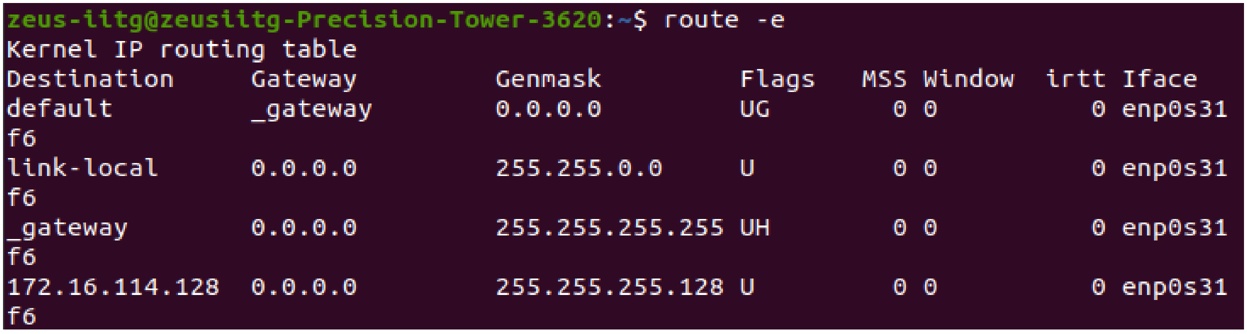
```
zeus-iitg@zeusiitg-Precision-Tower-3620:~$ route -A inet
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    20100  0        0 enp0s31f
6
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 enp0s31f
6
_gateway        0.0.0.0         255.255.255.255 UH    20100  0        0 enp0s31f
6
172.16.114.128  0.0.0.0         255.255.255.128 U     100    0        0 enp0s31f
6
```

(ii) **-C** operate on the kernel's routing cache



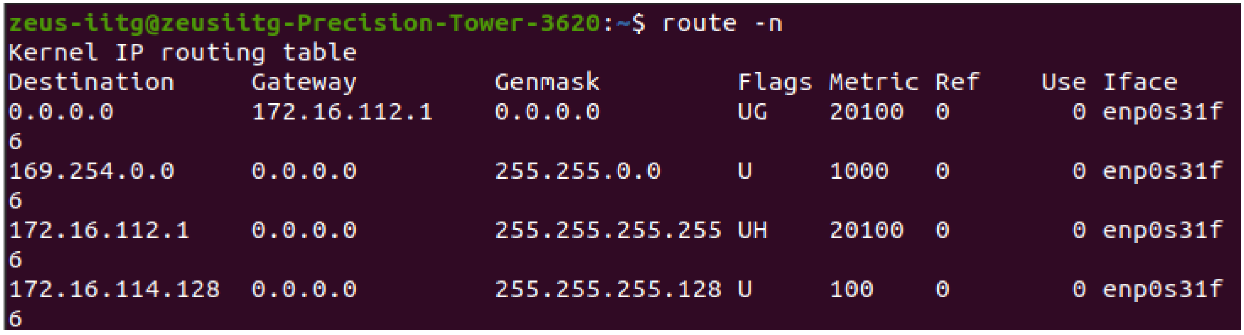
```
zeus-iitg@zeusiitg-Precision-Tower-3620:~$ route -C
Kernel IP routing cache
Source          Destination     Gateway         Flags Metric Ref    Use Iface
```

(iii) **-e** use *netstat(8)*-format for displaying the routing table. **-ee** generates a very long line with all parameters from the routing table.



```
zeus-iitg@zeusiitg-Precision-Tower-3620:~$ route -e
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         _gateway        0.0.0.0         UG        0 0          0 enp0s31
f6
link-local      0.0.0.0         255.255.0.0     U         0 0          0 enp0s31
f6
_gateway        0.0.0.0         255.255.255.255 UH        0 0          0 enp0s31
f6
172.16.114.128  0.0.0.0         255.255.255.128 U         0 0          0 enp0s31
f6
```

(iv) **-n** show numerical addresses instead of trying to determine symbolic host names. This is useful if you are trying to determine why the route to your nameserver has vanished.

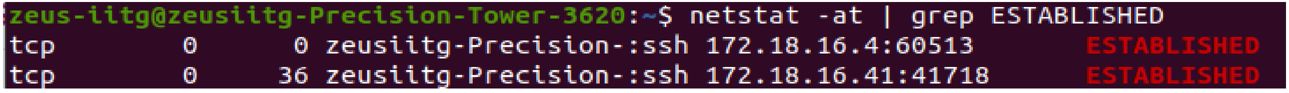


```
zeus-iitg@zeusiitg-Precision-Tower-3620:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.16.112.1    0.0.0.0         UG    20100  0        0 enp0s31f
6
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 enp0s31f
6
172.16.112.1    0.0.0.0         255.255.255.255 UH    20100  0        0 enp0s31f
6
172.16.114.128  0.0.0.0         255.255.255.128 U     100    0        0 enp0s31f
6
```

5. *netstat*:

(a) In computing, **netstat** (***net**work **stat**istics*) is a command-line network utility that displays network connections for Transmission Control Protocol (both incoming and outgoing), routing tables, and a number of network interface (network interface controller or software-defined network interface) and network protocol statistics.

(b) For listing only **TCP** (**Transmission Control Protocol**) port connections we use "**netstat -at**" command. To get all ESTABLISHED connections, we use "**netstat -at | grep ESTABLISHED**".

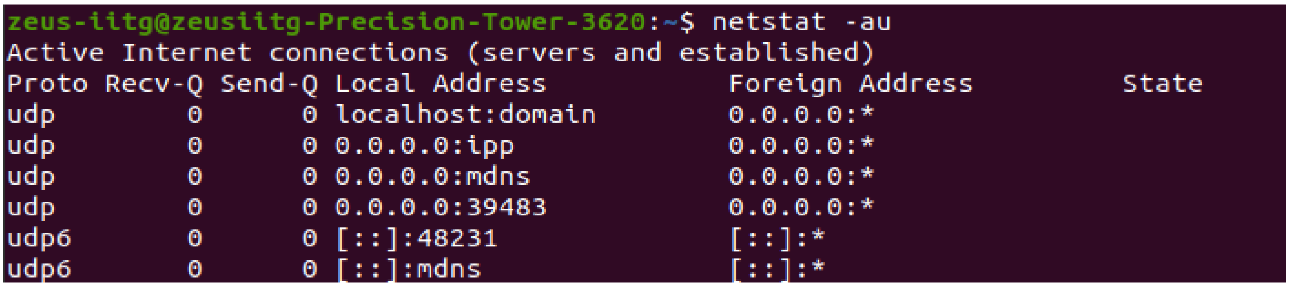


```
zeus-iitg@zeusiitg-Precision-Tower-3620:~$ netstat -at | grep ESTABLISHED
tcp        0      0 zeusiitg-Precision-:ssh 172.18.16.4:60513       ESTABLISHED
tcp        0     36 zeusiitg-Precision-:ssh 172.18.16.41:41718      ESTABLISHED
```

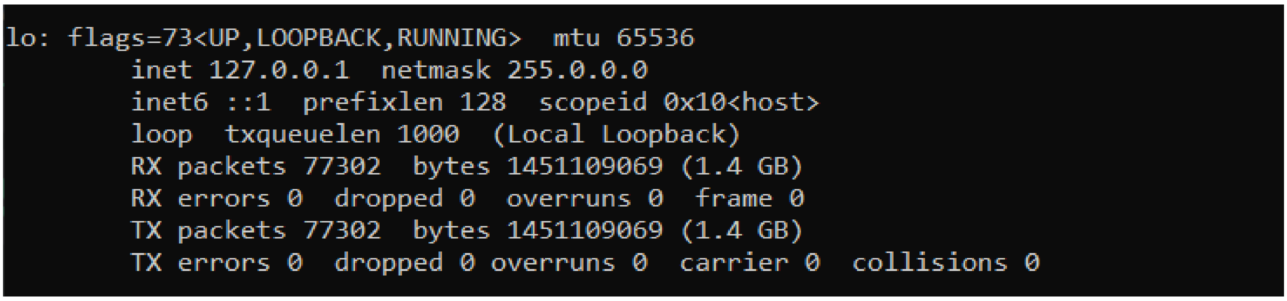
(c) **"netstat -r"** is used to get the kernel routing information. 1. **Destination**: Contains the destination network/host. 2. **Gateway**: Contains the gateway address or '*' if not specified. 3. **Genmask**: Contains the netmask of destination net. 4. **Flags**: Contains various flags: G(Route uses a gateway) etc. 5. **MSS**: Contains the maximum size segment of TCP of this route. 6. **Window**: Contains the default window size of this route. 7. **IRTT**: Contains the initial RTT. 8. **Iface**: Contains the interface of the destination of the packets of this route.

(d) "**netstat -i**" is used to display network interface status. "**netstat -i | wc -l**" gives (<number_of_interfaces> + 2) as the output. On my computer, the output was 4 which means there are 2 network interfaces.

(e) For Listing only **UDP** (**User Datagram Protocol)** port connections we use "**netstat -au**".



```
zeus-iitg@zeusiitg-Precision-Tower-3620:~$ netstat -au
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 localhost:domain        0.0.0.0:*
udp        0      0 0.0.0.0:ipp             0.0.0.0:*
udp        0      0 0.0.0.0:mdns            0.0.0.0:*
udp        0      0 0.0.0.0:39483           0.0.0.0:*
udp6       0      0 [::]:48231              [::]:*
udp6       0      0 [::]:mdns               [::]:*
```



```
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 77302  bytes 1451109069 (1.4 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 77302  bytes 1451109069 (1.4 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

(f) Loopback interface allows a device to communicate with itself through virtual interface. It is used mainly for diagnostics and troubleshooting, to connect to servers running on the local machine and it is a logical, virtual interface, which a machine uses to communicate to itself.

6. *traceroute*:
   (a) Traceroute is a network diagnostic tool used to track in real-time the pathway taken by a packet on an IP network from source to destination, reporting the IP addresses of all the routers it pinged in between. Traceroute also records the time taken for each hop the packet makes during its route to the destination.
   (b) The first 4 hops are same for all the routes as all the packets that are generated by my machine go through the same devices initially (home router and ISP).

Host	Hop count -1	Hop count -2	Hop count -3
Cloudflare	12	12	12
Facebook	10	11	11
Google	11	13	15
Hackerrank	64 (max)	64 (max)	64 (max)
Yahoo	21	17	16
YouTube	14	12	14

   (c) The path shown by traceroute vary at different times of day. This is mainly caused because the routing algorithms depend on the network traffic at that instant. This is variable over different times of the day. Also, a change in route can be caused by some router not functioning during some requests while functioning in the others.
   (d) Yes, traceroute for Hackerrank did not find complete route to the hosts as it exceeds the max hops (64) allowed. The reason maybe that Firewall of that host might be blocking our IP or there may be packet loss between various routers in the path.
   (e) Traceroute uses UDP packets with an incrementing TTL field to map the hops to the final destination whereas Ping uses ICMP. Some networks block ICMP by default so both PING and tracert from a Windows machine will fail but a traceroute from a Linux device may still work. Reference

7. *arp*:
   (a) To access the complete ARP table, we use the "***arp -e***" command. The output of this command has 5 columns. The first column **Address** shows the IP address of the corresponding device. The second column **HWtype** shows hardware type. The third column **HWaddress** shows hardware address (MAC address). The fourth column **Flags Mask** indicates if the MAC address has been learned, manually set, published (announced by another node than the requested) or is incomplete. Complete entries are marked with C flag.

Permanent entries are marked with M flag and published entries are marked with P flag. The fifth column **Iface** represents the network interface.

(b) “`sudo arp -s <IP_Address> <MAC_Address>`” is used to add an entry in the ARP table. “`sudo arp -d <IP_Address>`” is used to delete an entry from the ARP table

(c) ARP only works between devices in the same IP subnet. It is not a routed protocol which limits its usage to local networks. Hence, there cannot be an entry for any IP from a different subnet in the ARP table.

(d) No reply packet is received from the target IP Address. Hence, ping timeout occurs.

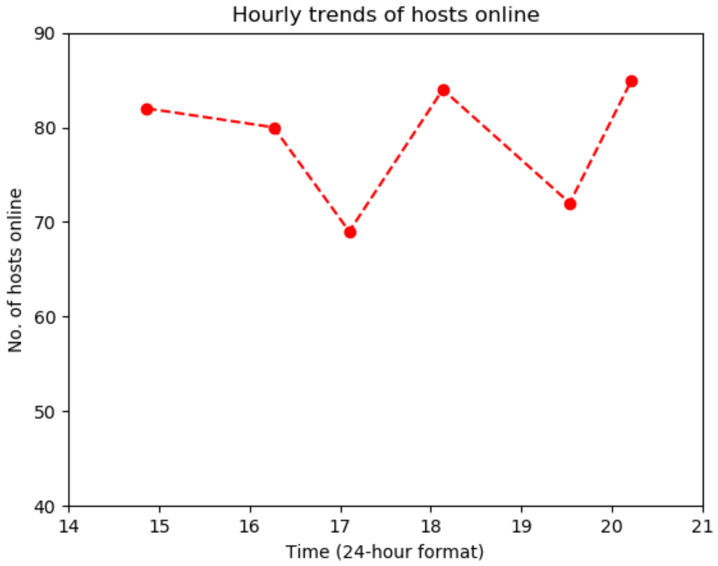
8. *nmap*:

(a) nmap -sn 172.16.114.*

(b) nmap -sA <your_ip_address>

(c)

Time	14:52	16:16	17:06	18:08	19:32	20:13
No. of hosts online	82	80	69	84	72	85


Hourly trends of hosts online
No. of hosts online
Time (24-hour format)