CS349

Networks Lab

Assignment 1

Arkadeep DAS
150123053

1. a) **ping -c count** -- specifies the number of echo requests to send. [eg: ping -c 10]

b) **ping -i wait** -- wait is the number of seconds to wait between each ping. [eg: ping -i 2]

c) **ping -l “ip-address”** . Limit for sending ECHO_REQUEST packets by normal users : **3**

d) **ping -s <packet-size> <address>** .

Total packet size = 20 bytes IP Header + 8 bytes ICMP header + 64 bytes = **92 bytes**

2. Five hosts are **google.com**, **github.com**, **bitbucket.org**, **slack.com** and **stackexchange.com .**

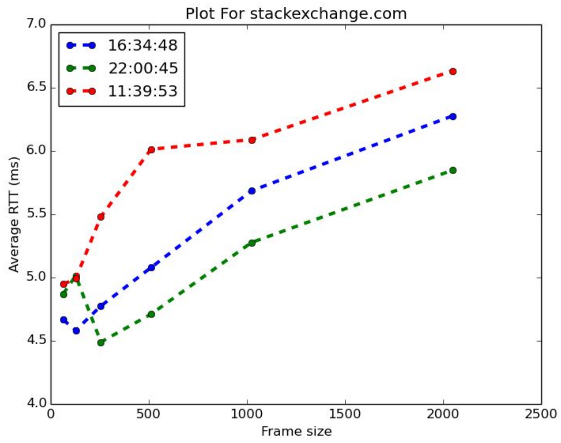
Packet size = 64 bytes . There exists no cases for packet loss greater than **0 %** .

Host	**Average RTT (ms)**	**Packet Loss**
google.com	5.569 5.082 4.444	0
github.com	19.17 18.38 19.446	0
bitbucket.org	11.997 11.96 12.140	0
slack.com	5.532 6.150 5.769	0
stackexchange.com	4.947 4.666 4.871	0

RTT is weakly correlated with the geographical distance of the nodes. Reason might be an increased hop count. The packets have to through more routers, at each router there may be a delay, hence the more routers, the longer the RTT.
Host : **stackexchange.com**

Packet size	**Average RTT (ms)**		
64	4.947	4.666	4.871
128	4.991	4.583	5.014
256	5.48	4.775	4.49
512	6.014	5.079	4.711
1024	6.09	5.686	5.277
2048	6.634	6.277	5.848

Increase in packet size increases average RTT . Time of the day generally does not affect average RTT but it can alter depending on network traffic. The plot below summarizes the effects at three different times of the day. Packet loss is mainly caused by network congestion. Larger packet sizes cause full loss because not all routes have the same MTU throughout thereby making the line unstable.

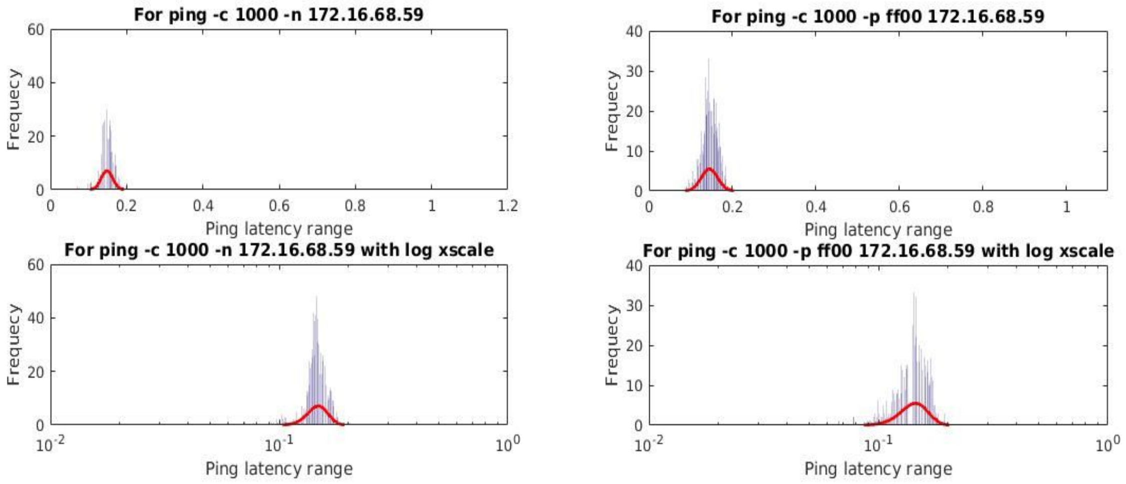


3. IP Address : **172.16.68.59**

a) Packet loss rate for each command was **0 %**

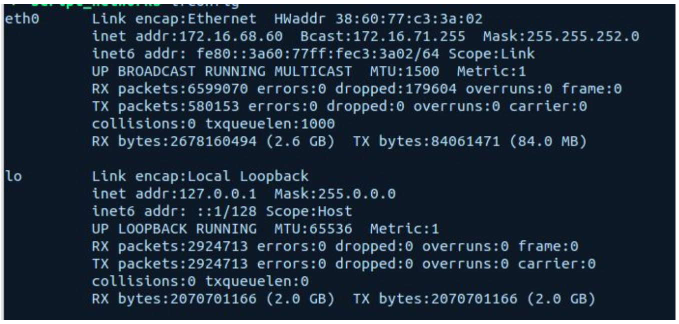
Command	**Minimum latency (ms)**	**Maximum latency (ms)**	**Mean latency (ms)**	**Median latency (ms)**
ping -n 172.16.68.59	0.071	0.204	0.148	0.147
Ping -p ff00 172.16.68.59	0.067	0.201	0.144	0.145

c)



d) In the second case, all the measured latencies turned out to be smaller than in the first case, **-n** just tells it to display IP addresses numerically instead of trying to resolve them to names. **-p** provides a data pattern to send in the packets, instead of the default in this case ff00.

4. **Output of ifconfig :**



```
eth0      Link encap:Ethernet  HWaddr 38:60:77:c3:3a:02
          inet addr:172.16.68.60  Bcast:172.16.71.255  Mask:255.255.252.0
          inet6 addr: fe80::3a60:77ff:fec3:3a02/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6599070 errors:0 dropped:179604 overruns:0 frame:0
          TX packets:580153 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2678160494 (2.6 GB)  TX bytes:84061471 (84.0 MB)

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:2924713 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2924713 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2070701166 (2.0 GB)  TX bytes:2070701166 (2.0 GB)
```

Link encap : This represents the frame type associated with this interface.
HWaddr : This represents the hardware address of the ethernet interface (MAC address),a global unique identifier for a network interface.
Inet addr : IPv4 address assigned to the interface
Bcast: Broadcast address of the network associated with the interface.
Mask : Network mask associated with the interface.
Up : This flag indicates that the network interface is configured to be enabled.
Broadcast : Indicates that the interface is configured to handle broadcast packets.
Running : Indicates that the network interface is operational and is ready to accept the data.
Multicast : Indicates that the interface is configured to handle multicast packets.
MTU : The maximum transmission unit for which the interface is configured.
Metric 1 : It tells the OS which interface a packet should be forwarded to, when multiple interfaces could be used to reach the packet's destination.
RX Packets : The number of packets received via the interface.
RX errors : The number of damaged packets received.
RX dropped : The number of dropped packets due to reception errors.
RX overrruns : The number of received packets that experienced data overruns.
RX frame : The number received packets that experienced frame errors.
TX Packets : The number of packets transmitted via the interface.
TX errors : The number of packets that experienced transmission error.
TX dropped : The number of dropped transmitted packets due to transmission errors.
TX overruns : The number of transmitted packets that experienced data overruns.
TX Carriers: The number received packets that experienced loss of carriers.
TX Collisions: 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.
RX bytes: The total bytes received over this interface.
TX bytes: The total bytes transmitted over this interface.
Interrupts: This field provides the information about the IRQ value assigned to this interface.
Base Address: The I/O base address associated with this interface.

Output of route :

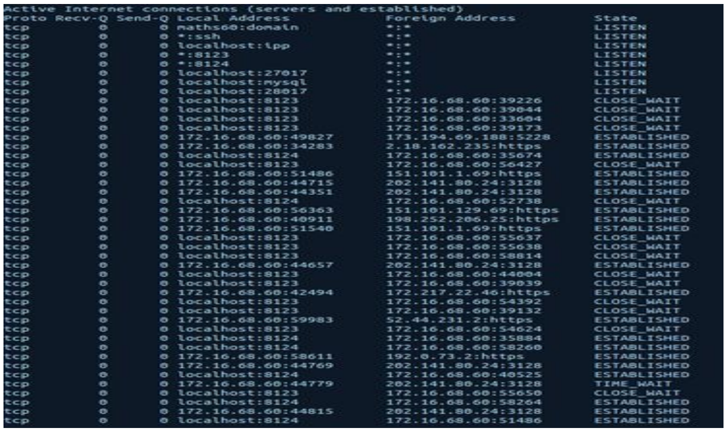
```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   Metric   Ref   Use   Iface
default         172.16.68.254   0.0.0.0          UG        0      0     0    eth0
172.16.68.0          *          255.255.252.0    U         1      0     0    eth0
```

Any traffic with destination 172.16.68.60 will not be using a gateway (thats the * on the line), will be using a 255.255.252.0 netmask, route is UP (that's the meaning of the U) and which interface the route uses. If any traffic does not fit the traffic defined on any other rules then use the default route.

Destination : The destination network or destination host.
Gateway : The gateway address or '*' if none set.
Genmask : The netmask for the destination net
Flags: U (route is up)
Iface : Interface to which packets for this route will be sent.
Metric : The 'distance' to the target (usually counted in hops).
Ref : Number of references to this route. (Not used in Linux kernel.)
Use : **C** -- Count of lookups for the route.

5. **netstat ("network statistics")** is a command-line tool that displays network connections (both incoming and outgoing), routing tables, and a number of network interface (network interface controller or software-defined network interface) and network protocol statistics.
It is used for finding problems in the network and to determine the amount of traffic on the network as a performance measurement.

-at:



```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 maths60:domain          *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 localhost:ipp           *:*                     LISTEN
tcp        0      0 *:8123                  *:*                     LISTEN
tcp        0      0 *:8124                  *:*                     LISTEN
tcp        0      0 localhost:27017         *:*                     LISTEN
tcp        0      0 localhost:mysql         *:*                     LISTEN
tcp        0      0 localhost:28017         *:*                     LISTEN
tcp        0      0 localhost:8123          172.16.68.60:39226      CLOSE_WAIT
tcp        0      0 localhost:8123          172.16.68.60:39044      CLOSE_WAIT
tcp        0      0 localhost:8123          172.16.68.60:33604      CLOSE_WAIT
tcp        0      0 localhost:8123          172.16.68.60:39173      CLOSE_WAIT
tcp        0      0 172.16.68.60:49827      173.194.69.188:5228     ESTABLISHED
tcp        0      0 172.16.68.60:34283      2.18.162.235:https      ESTABLISHED
tcp        0      0 localhost:8124          172.16.68.60:35674      ESTABLISHED
tcp        0      0 localhost:8123          172.16.68.60:56427      CLOSE_WAIT
tcp        0      0 172.16.68.60:51486      151.101.1.69:https      ESTABLISHED
tcp        0      0 172.16.68.60:44715      202.141.80.24:3128      ESTABLISHED
tcp        0      0 172.16.68.60:44351      202.141.80.24:3128      ESTABLISHED
tcp        0      0 localhost:8124          172.16.68.60:52738      CLOSE_WAIT
tcp        0      0 172.16.68.60:56363      151.101.129.69:https    ESTABLISHED
tcp        0      0 172.16.68.60:40911      198.252.206.25:https    ESTABLISHED
tcp        0      0 172.16.68.60:51540      151.101.1.69:https      ESTABLISHED
tcp        0      0 localhost:8123          172.16.68.60:55637      CLOSE_WAIT
tcp        0      0 localhost:8123          172.16.68.60:55638      CLOSE_WAIT
tcp        0      0 localhost:8123          172.16.68.60:58814      CLOSE_WAIT
tcp        0      0 172.16.68.60:44657      202.141.80.24:3128      ESTABLISHED
tcp        0      0 localhost:8123          172.16.68.60:44004      CLOSE_WAIT
tcp        0      0 localhost:8123          172.16.68.60:39039      CLOSE_WAIT
tcp        0      0 172.16.68.60:42494      172.217.22.46:https     ESTABLISHED
tcp        0      0 localhost:8123          172.16.68.60:54392      CLOSE_WAIT
tcp        0      0 localhost:8123          172.16.68.60:39132      CLOSE_WAIT
tcp        0      0 172.16.68.60:59983      52.44.231.2:https       ESTABLISHED
tcp        0      0 localhost:8123          172.16.68.60:54624      CLOSE_WAIT
tcp        0      0 localhost:8124          172.16.68.60:35884      ESTABLISHED
tcp        0      0 localhost:8124          172.16.68.60:58260      ESTABLISHED
tcp        0      0 172.16.68.60:58611      192.0.73.2:https        ESTABLISHED
tcp        0      0 172.16.68.60:44769      202.141.80.24:3128      ESTABLISHED
tcp        0      0 localhost:8124          172.16.68.60:40525      ESTABLISHED
tcp        0      0 172.16.68.60:44779      202.141.80.24:3128      TIME_WAIT
tcp        0      0 localhost:8123          172.16.68.60:55650      CLOSE_WAIT
tcp        0      0 localhost:8124          172.16.68.60:58264      ESTABLISHED
tcp        0      0 172.16.68.60:44815      202.141.80.24:3128      ESTABLISHED
tcp        0      0 localhost:8124          172.16.68.60:51486      ESTABLISHED
```

Proto : tell us if the socket listed is TCP or UDP. Those are network protocols.
Recv-Q and Send-Q : tell us how much data is in the queue for that socket, waiting to be read (Recv-Q) or sent (Send-Q)
Local and 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 (in the example, the computer is called "Trafalgar"), and the foreign end is about the other computer (could be somewhere in the local network or somewhere on the internet).
State : tells in which state the listed sockets are.

-r : Displays the contents of the IP routing table. Same as route print command.

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags     MSS   Window  irtt    Iface
default         172.16.68.254     0.0.0.0         UG        0       0         0     eth0
172.16.68.0      *                255.255.252.0     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
Iface : Interface to which packets for this route will be sent.
MSS : default Maximum Segment Size for TCP connections over this route.
Window : the default window size for TCP connections over this route.
Irtt : the Initial Round Trip Time for this route.

-i : Used to display network interface status.
netstat -i | tail -n+3 | awk '{print $1}' : to display the interfaces on machine.

The loopback device is a special, virtual network interface that your computer uses to communicate with itself. It is used mainly for diagnostics and troubleshooting, and to connect to servers running on the local machine.
ifconfig 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.
It is also helpful when a server offering a resource you need is running on your own machine.

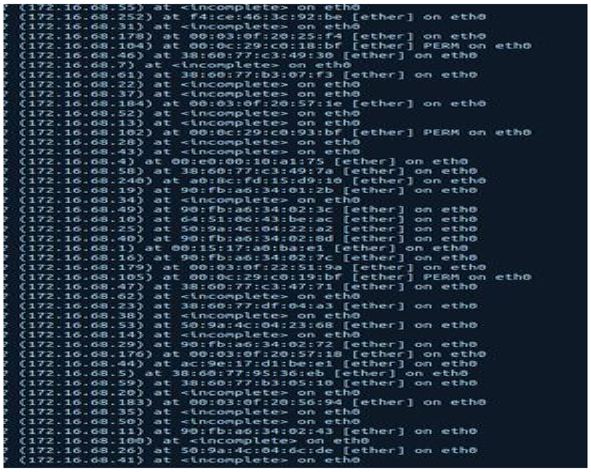
6.

Host	**Hop Count**
google.com	7
github.com	10
bitbucket.org	12
slack.com	9
stackexchange.com	7

1. Almost all hosts have common hops between two routes even at different times of the day.
Hosts slack.com and bitbucket.org only have one different hop between two routes.
2. Yes routes to same hosts changes for some because routes can vary in the Internet as the paths are redundant. Actually that's how the internet works.
3. This can indicate the hosts have a network/firewall issue that is preventing me from reaching their server .
4. Yes it can happen. Traceroute 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.
Ping implementation uses ICMP echo request packets. Some hosts may be configured to work as a router
traffic through it (which includes sending ICMP TTL EXCEEDED control messages for failures) but to block traffic to it, specifically to not reply to ICMP ECHO REQ. So, ping might get timed out but traceroute works.

7. **arp** command is used to show the full arp table of my machine.
Address: The hostname or ip address of the host.
HWaddr : This represents the hardware address of the ethernet interface (MAC address),a global unique identifier for a network interface.
HWtype : The type of network hardware . ether stands for Ethernet.
Flags: C - Cloning—A new route is cloned from this entry when it is used
Mask : Network mask associated with the interface.
Iface : Interface to which packets for this route will be sent.

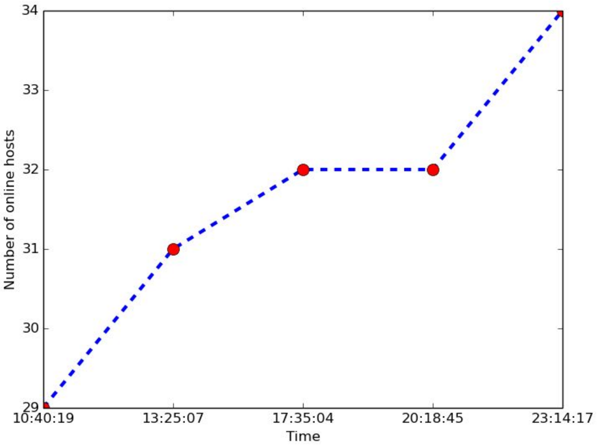
sudo arp -s HOSTNAME MAC-Address : to add entry to arp table.
sudo arp -d HOSTNAME : to delete entry from arp table.
It does not completely delete the entry but marks it incomplete since deleting things from caches is hard and expensive.



```
? (172.16.68.55) at <incomplete> on eth0
? (172.16.68.252) at f4:ce:46:3c:92:be [ether] on eth0
? (172.16.68.31) at <incomplete> on eth0
? (172.16.68.178) at 00:03:0f:20:25:f4 [ether] on eth0
? (172.16.68.104) at 00:0c:29:c0:18:bf [ether] PERM on eth0
? (172.16.68.46) at 38:60:77:c3:49:30 [ether] on eth0
? (172.16.68.7) at <incomplete> on eth0
? (172.16.68.61) at 38:60:77:b3:07:f3 [ether] on eth0
? (172.16.68.22) at <incomplete> on eth0
? (172.16.68.37) at <incomplete> on eth0
? (172.16.68.184) at 00:03:0f:20:57:1e [ether] on eth0
? (172.16.68.52) at <incomplete> on eth0
? (172.16.68.13) at <incomplete> on eth0
? (172.16.68.102) at 00:0c:29:c0:93:bf [ether] PERM on eth0
? (172.16.68.28) at <incomplete> on eth0
? (172.16.68.43) at <incomplete> on eth0
? (172.16.68.4) at 00:e0:00:10:a1:75 [ether] on eth0
? (172.16.68.58) at 38:60:77:c3:49:7a [ether] on eth0
? (172.16.68.240) at a0:8c:fd:15:d9:10 [ether] on eth0
? (172.16.68.19) at 90:fb:a6:34:01:2b [ether] on eth0
? (172.16.68.34) at <incomplete> on eth0
? (172.16.68.49) at 90:fb:a6:34:02:3c [ether] on eth0
? (172.16.68.10) at d4:51:06:43:be:ac [ether] on eth0
? (172.16.68.25) at 50:9a:4c:04:22:a2 [ether] on eth0
? (172.16.68.40) at 90:fb:a6:34:02:8d [ether] on eth0
? (172.16.68.1) at 00:15:17:a0:ba:e1 [ether] on eth0
? (172.16.68.16) at 90:fb:a6:34:02:7c [ether] on eth0
? (172.16.68.179) at 00:03:0f:22:51:9a [ether] on eth0
? (172.16.68.105) at 00:0c:29:c0:19:bf [ether] PERM on eth0
? (172.16.68.47) at 38:60:77:c3:47:71 [ether] on eth0
? (172.16.68.62) at <incomplete> on eth0
? (172.16.68.23) at 38:60:77:df:04:a3 [ether] on eth0
? (172.16.68.38) at <incomplete> on eth0
? (172.16.68.53) at 50:9a:4c:04:23:68 [ether] on eth0
? (172.16.68.14) at <incomplete> on eth0
? (172.16.68.29) at 90:fb:a6:34:02:72 [ether] on eth0
? (172.16.68.176) at 00:03:0f:20:57:18 [ether] on eth0
? (172.16.68.44) at ac:9e:17:d1:be:e1 [ether] on eth0
? (172.16.68.5) at 38:60:77:95:36:eb [ether] on eth0
? (172.16.68.59) at 38:60:77:b3:05:10 [ether] on eth0
? (172.16.68.20) at <incomplete> on eth0
? (172.16.68.183) at 00:03:0f:20:56:94 [ether] on eth0
? (172.16.68.35) at <incomplete> on eth0
? (172.16.68.50) at <incomplete> on eth0
? (172.16.68.11) at 90:fb:a6:34:02:43 [ether] on eth0
? (172.16.68.100) at <incomplete> on eth0
? (172.16.68.26) at 50:9a:4c:04:6c:de [ether] on eth0
? (172.16.68.41) at <incomplete> on eth0
```

cat /proc/sys/net/ipv4/neigh/default/gc_stale_time
Will provide the default arp cache timeout , in my case it is 60 s
We'll check every 20 seconds if the entry is still cached, when it ceases to exist, that time is the approximate time.
There is nothing wrong with having multiple IP addresses with the same MAC address.

8. Command used : **nmap -n -sP 172.16.68.59/26**



The number of online hosts show a very slow change with increase in number as the day progressed.