CS349: NETWORKS LAB

ASSIGNMENT 1

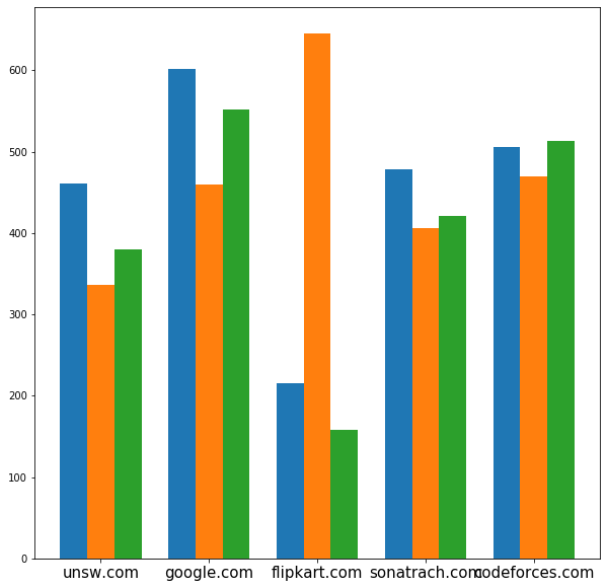
Kartik Sethi 170123057
18th January 2020

Q1

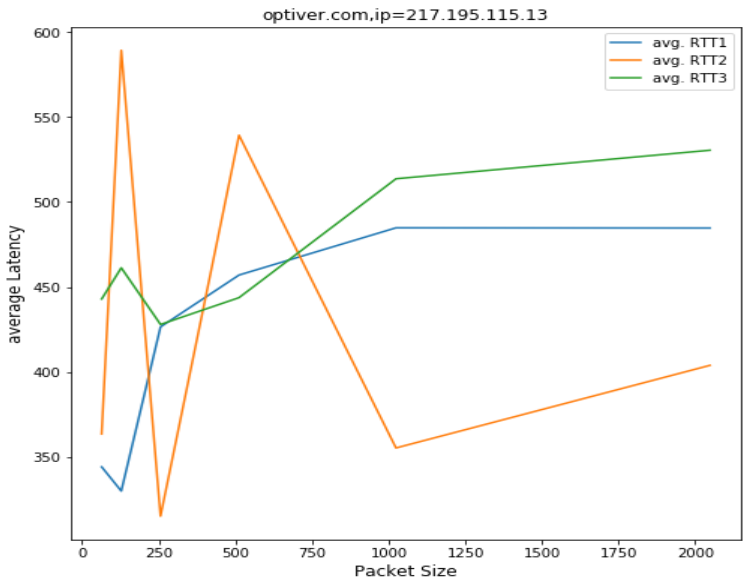
a) ping will continue to send ICMP packages until it receives an interrupt signal. To specify the number of Echo Request packages to be sent after which ping will exit, use the -c option followed by the number of the packages
b) You can set the time interval between two successive ping ECHO_REQUESTS by using the -i option followed by the new time interval.
c) Use -l this option to set the number of packets to send without waiting for a reply. For selecting a value more than 3, you need to be a super user.
d) Use -s this option to set the packet size.

Q2 PING

Hostname	Location	ip adress	avg. RTT 1	avg. RTT 2	avg. RTT3
unsw.com	Australia	202.58.60.194	460.458	335.831	380.013
google.com	United States	172.217.4.142	602.302	460.042	551.61
flipkart.com	India	163.53.78.128	215.156	645.291	157.522
sonatrach.com	Algeria	41.106.2.26	478.629	405.75	420.813
codeforces.com	Russia	81.27.240.126	505.981	469.571	513.697

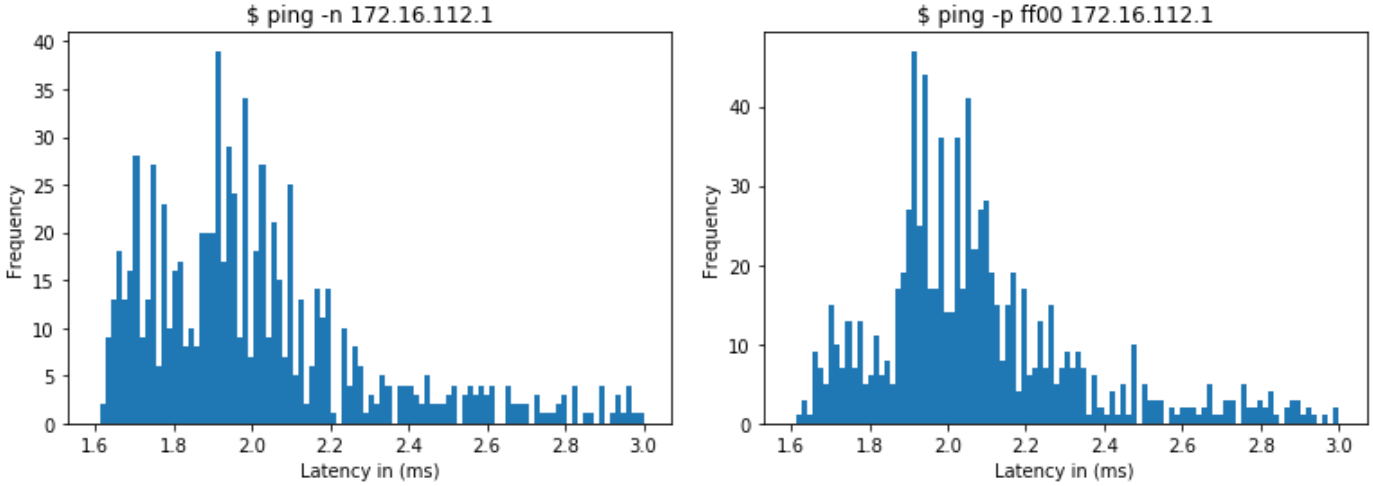


From the above table we can see that latency is weekly correlated with distance because of increased propagation delay and increased number of hops to the destination.



From what we can see on the left (RTT v/s packet size) the RTT increases with an increase in packet size generally.

Q3



From the shape of the histogram, we can see it resembles a log-normal distribution

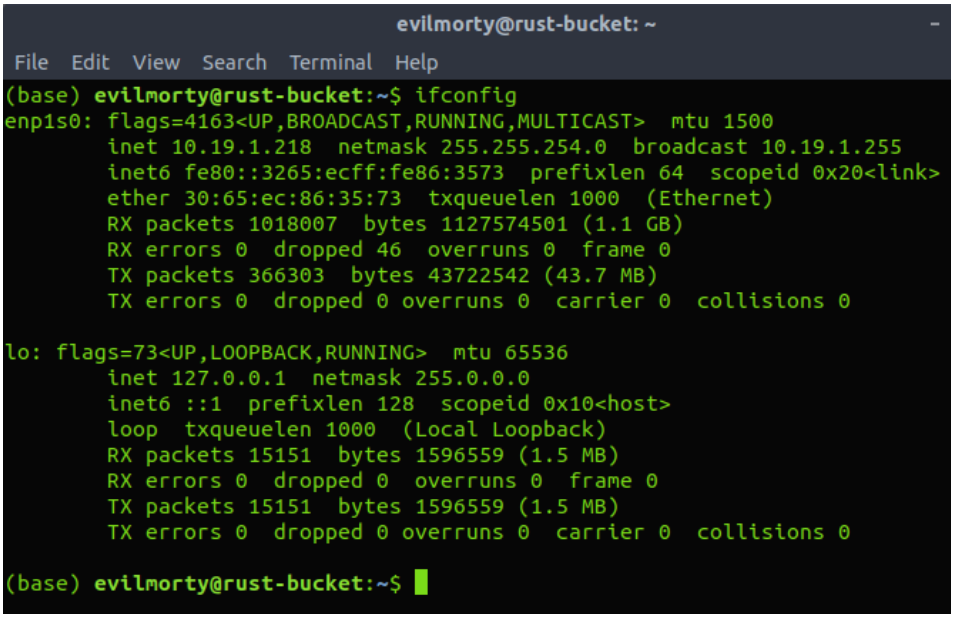
	packet loss%	min delay	max delay	median delay	mean delay
ping -n	0	1.625	124.342	2.025	4.581
ping -p ff00	0	1.595	104.196	2.06	2.686

Ping -n gives numeric output only. No attempt will be made to lookup symbolic names for host addresses.
ping -p allows you to specify up to 16 “pad" bytes to fill out the packet you send. This is useful for diagnosing data-dependent problems in a network.

So ideally ping -n should be faster but this is not reflected in my output

Q4

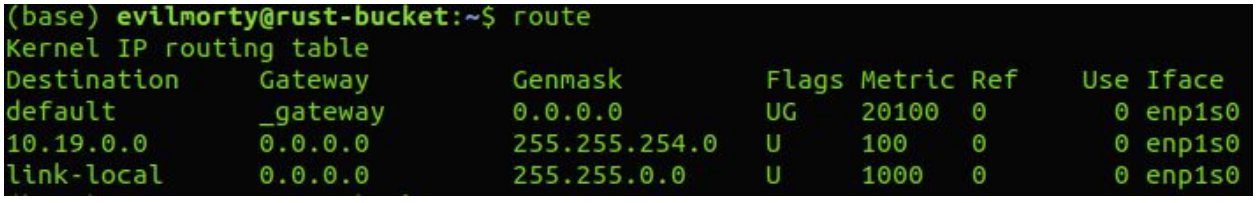
IFCONFIG



inet addr: indicates the machine IP address
BROADCAST: denotes that ethernet supports broadcasting
Broadcast: denotes broadcast denotes the broadcast address
UP: This flag indicates that the kernel modules related to the Ethernet interface have been loaded.
RUNNING: The interface is ready to accept data.
MULTICAST: denotes that multicasting is supported.
MTU: short form for the Maximum Transmission Unit is the size of each packet received by the Ethernet card.
RX Packets, **TX Packets**: show the total number of packets received and transmitted respectively. As you can see in the output, the total errors are 0, no packets are dropped and there are no overruns. If you find the errors or dropped value greater than zero, then it could mean that the Ethernet device is failing or there is some congestion in your network.
Txqueuelen: This denotes the length of the transmit queue of the device.

ifconfig options:
-a: display all interfaces which are currently available, even if down
-s: display a short list
-v: be more verbose for some error conditions.
mtu N: This parameter sets the Maximum Transfer Unit (MTU) of an interface.

ROUTE



The route command shows the routing tables.
Destination: this column gives the destination address
Gateway: the gateway column identifies the defined gateway. A (*) means no gateway is set.
Genmask: netmask for the routing table.
Iface: Network interface.
Uflag: It means the route is up.
Gflag: It means the specified gateway should be used for this route.
Metric: It is the distance to the target in hops.
Ref: It is the number of references to the route.
ROUTE -Options

-F: operate on the kernel's FIB (Forwarding Information Base) routing table. This is the default.
-C: operate on the kernel's routing cache.
-v: select verbose operation.
-n: show numerical addresses instead of trying to determine symbolic hostnames.

Q5 NETSTAT

(a) Netstat is used to print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.

(b) Netstat -at is used to print all established tcp connections

```
(base) evilmorty@rust-bucket:~$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 rust-bucket:domain      0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:57621           0.0.0.0:*               LISTEN
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:41883           0.0.0.0:*               LISTEN
tcp        0      0 rust-bucket:56680       maa03s23-in-f14.1:https ESTABLISHED
tcp        0      0 rust-bucket:60856       30.224.186.35.bc.:https ESTABLISHED
tcp        0      0 rust-bucket:47970       maa05s03-in-f2.1e:https ESTABLISHED
tcp        0      0 rust-bucket:59026       maa03s26-in-f14.1:https ESTABLISHED
tcp        0      0 rust-bucket:45528       maa05s10-in-f2.1e:https ESTABLISHED
tcp        0      0 rust-bucket:34130       maa05s06-in-f3.1e:https ESTABLISHED
tcp        0      0 rust-bucket:39038       maa03s31-in-f14.1:https ESTABLISHED
tcp        0      0 rust-bucket:39036       maa03s31-in-f14.1:https ESTABLISHED
tcp        0      0 rust-bucket:53714       221.240.199.104.b:https ESTABLISHED
tcp        0      0 rust-bucket:49928       47.224.186.35.bc.:https ESTABLISHED
```

Proto: indicates whether the socket listed is TCP or UDP
RecvQ and Send-Q: tell us how much data is in the queue waiting to be sent or received
The "Local Address" 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 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. The TCP protocol defines states, including "LISTEN" (wait for some external computer to contact us) and "ESTABLISHED" (ready for communication). The "CLOSE WAIT" state means that the foreign or remote machine has already closed the connection, but that the local program somehow hasn't followed suit.

(c) netstat -r prints the routing tables

```
(base) evilmorty@rust-bucket:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         _gateway        0.0.0.0         UG        0 0          0 enp1s0
10.19.0.0       0.0.0.0         255.255.254.0   U         0 0          0 enp1s0
10.42.0.0       0.0.0.0         255.255.255.0   U         0 0          0 wlp2s0
link-local      0.0.0.0         255.255.0.0     U         0 0          0 enp1s0
```

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.
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.

Genmask: it tells how many bits from the start of the IP address are used to identify the subnet but, 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.
Flags: This column 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.
MSS: This column 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.
Window: This column is like the MSS column in that it gives the option of altering a TCP parameter. In this case that parameter is the default window size, which indicates how many TCP packets can be sent before at least one of them has to be ACKnowledged.
Irtt: This column 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.
Iface: tells which network interface should be used for sending packets that match the destination. If your computer is connected to multiple subnets on multiple network cards, you may find that some lines have an Iface of eth0 and others have one of eth1.
(d)netstat -i is used to display status of all network interfaces.

```
(base) evilmorty@rust-bucket:~$ netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
enp1s0    1500  1833056      0     46 0        696735      0      0      0 BMRU
lo       65536    21134      0      0 0         21134      0      0      0 LRU
wlp2s0    1500   357920      0      7 0        604598      0      0      0 BMRU
```

My laptop has 3 interfaces
Enp1s0: ethernet interface
Lo: loopback device
Wlp2s0: wifi interface
(d) netstat -su is used list all stats of udp connections

```
(base) evilmorty@rust-bucket:~$ netstat -su
IcmpMsg:
    InType0: 4526
    InType3: 84
    OutType3: 1263
    OutType8: 4829
Udp:
    333190 packets received
    94 packets to unknown port received
    0 packet receive errors
    102513 packets sent
    0 receive buffer errors
    0 send buffer errors
    IgnoredMulti: 91796
```

(f) **loopback interface:** It is a logical, virtual interface in a Router. A loopback interface is not a physical interface like Fast Ethernet interface or Gigabit Ethernet interface.A loopback interface is a software interface that can be used to emulate a physical interface. By default, the router doesn’t have any loopback interfaces (loopback interfaces are not enabled by default), but they can easily be created. Loopback interfaces are treated similar to physical interfaces in a router and we can assign IP addresses to them. The command syntax to create a loopback interface is shown below.

Q6 TRACEROUTE

Number of Hops	unsw.com	google.com	flipkart.com	sonatrach.com	codeforces.com
#Hops1	17	8	>30 hops	>30 hops	>30 hops
#Hops2	18	18	>30 hops	>30 hops	>30 hops
#Hops3	12	14	>30 hops	>30 hops	>30 hops

The common hops besides my machine were *core22.fsn1.hetzner.com* which was common between google.com and unsw.com and *ashihnrtr01-vlan502.dc10.neustar.com* which was common between flipkart.com and sonatrach.com

(b) The route to the same host changes during different parts of the day this is because of varying network traffic. The path to a host depends on the load on the network at any given time.

(c) Traceroute is not always able to find a path to the host because the servers/hosts along the way have set up firewalls to block ICMP traffic.

(d) Ping is straight ICMP from point A to point B, that traverses networks via routing rules. Traceroute works very different, even though it uses ICMP. 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.

Q7 ARP

(a) $ arp -a is used to print arp table. The output from arp -a will list the network interface, target system and physical (MAC) address of each system.

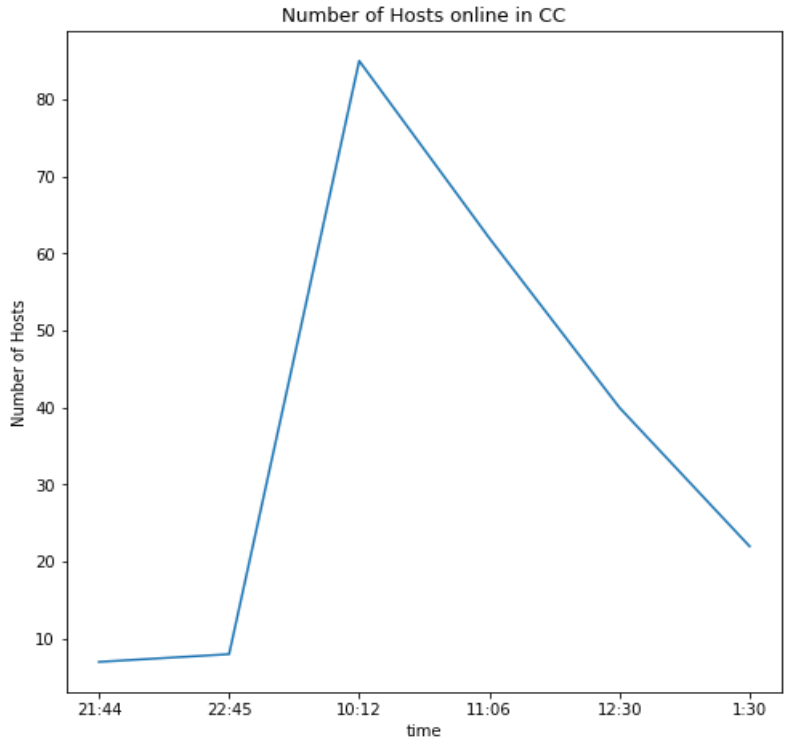
```
(base) evilmorty@rust-bucket:~/Documents/Semester VI/CS349_Networks_Lab$ arp -a
? (10.19.1.25) at 98:29:a6:35:f6:97 [ether] on enp1s0
_gateway (10.19.0.1) at ec:44:76:74:60:41 [ether] on enp1s0
? (10.19.1.225) at 04:d4:c4:79:9e:bd [ether] on enp1s0
? (10.19.1.89) at c8:d9:d2:8e:74:61 [ether] on enp1s0
? (10.42.0.42) at <incomplete> on wlp2s0
```

(b) $sudo arp -s <ip_addr> <mac_addr> is used to add hosts and
$sudo arp -d <ip_addr> to delete hosts

```
(base) evilmorty@rust-bucket:~$ sudo arp -s 10.19.1.213 00:0c:29:c0:94:bf
(base) evilmorty@rust-bucket:~$ sudo arp -s 10.19.1.211 00:0c:29:c0:94:bf
(base) evilmorty@rust-bucket:~$ sudo arp -s 10.19.1.212 00:0c:29:c0:94:bf
(base) evilmorty@rust-bucket:~$ sudo arp -s 10.19.1.219 00:0c:29:c0:94:bf
(base) evilmorty@rust-bucket:~$ arp -a
? (10.19.1.219) at 00:0c:29:c0:94:bf [ether] PERM on enp1s0
? (10.19.1.212) at 00:0c:29:c0:94:bf [ether] PERM on enp1s0
? (10.19.1.211) at 00:0c:29:c0:94:bf [ether] PERM on enp1s0
? (10.19.1.213) at 00:0c:29:c0:94:bf [ether] PERM on enp1s0
_gateway (10.19.0.1) at ec:44:76:74:60:41 [ether] on enp1s0
? (10.19.0.243) at e8:6a:64:8a:a4:1e [ether] on enp1s0
```

(c) You can get the time entries are cached in arp table by running
$ cat /proc/sys/net/ipv4/neigh/default/gc_stale_time
On my machine, the entries are cached for 60 seconds.
A trial and error method to check how long entries are cached is to temporarily add a host and check how long it is displayed in the arp table.
(d) two IP's can correspond to the same MAC address if the router connects two or more subnets. When communicating with machines on the same subnet range, MAC address is used for directing the packages. In the ARP Table, the IPs of the devices which are connected in the other subnet range have the MAC address as that of the Router which connects the two subnet ranges. ARP table is referred so as to convert these IP addresses to the MAC address and packets are sent to it. The router then uses it's routing table and sends the packet further to the correct device.

Q8 Number of Hosts



From the graph, we can see the number of hosts online is low at night and higher during the day