Networks Assignment 1

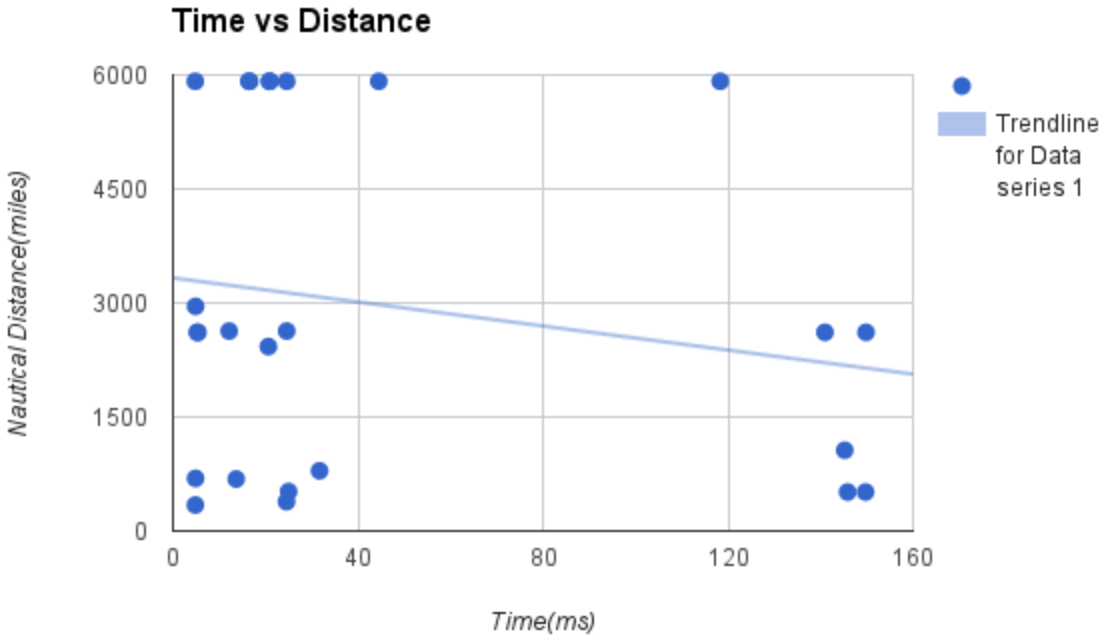
1.
a.) How does the traceroute program work?
It uses UDP/ICMP echo request packets and keeps pinging with increasing TTL (Time to Live). The time-to-live (TTL), also called hop limit, is used to determine intermediate routers which are being traversed towards the destination. The router which sets TTL to 0 sends back an ICMP Time to Live Limit Exceeded packet.

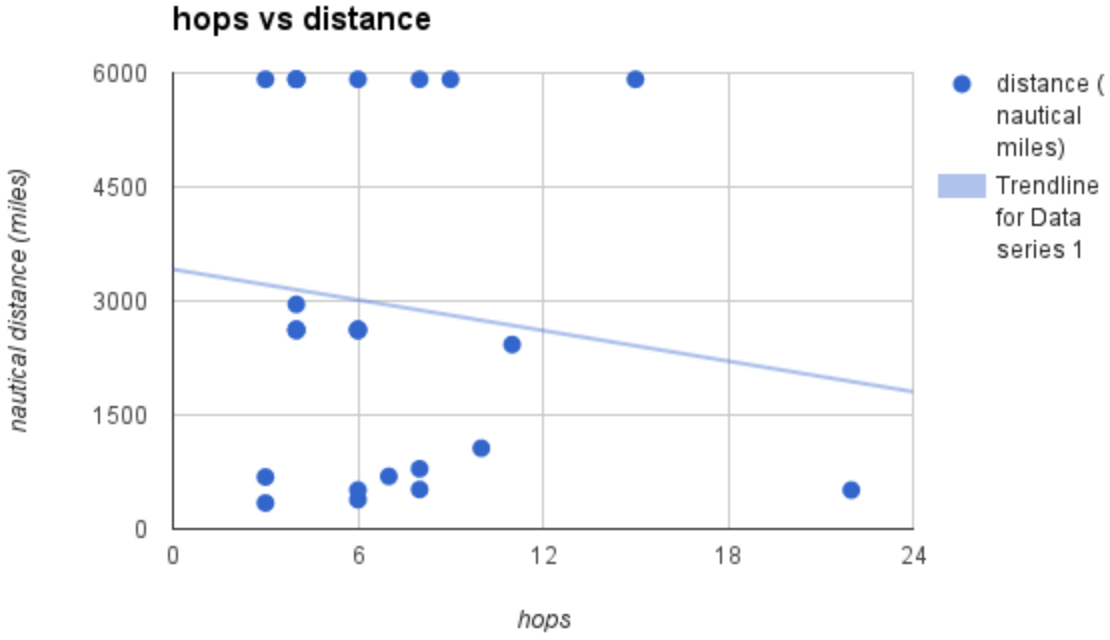
What are the parameters generated by the basic traceroute command?
There are two parameters in basic traceroute command - Host and PacketSize. Host specifies the destination host. It is written as hostname or IP number. This is a required parameter. And the other parameter is PacketSize, which specifies the probe datagram length.

There are many options in traceroute -
-f flow- The default value of this option is 0. It sets flow label field in the IPv6 packet header.
-m Max_ttl - The default value of this option is 30 hops. It sets the maximum TTL used in the outgoing probe packets.
-n - This option outputs the hop addresses numerically rather than symbolically & numerically.
-p Port - The default value of this option is 33434. It sets the base UDP port number used in the probes.
-q Nqueries - The default value of this option is 3 probes. It specifies the number of probes the traceroute command sends at each of the Max_ttl setting.
-r - This option sends the probe packet directly to a host on an attached network and an error is returned, if the particular host is not on a directly attached network.
-s SRC_Addr - This uses the next IP address in the numerical form as the source address in outgoing packets. This option can be used, on hosts with more than one IP address, to force source address to be something other than IP address of the interface on which the probe packet is sent.
-t TypeOfService - The default value of this option is 0. It sets the TypeOfService variable in the probe packets to an integer in the range of 0-255. This flag can be used to investigate whether different service types result in different paths.
-v - Receives packets other than PORT_UNREACHABLE & TIME_EXCEEDED.
-w WaitTime - The default value of this option is 3 seconds. It sets the time to wait for a response to a probe.

b.) Data set collected:-

S. No	Website	time (ms)	distance (nautical miles)	hops
1	www.google.com	5.379	2614.1	4
2	www.facebook.com	145.24	1062.5	10
3	www.myntra.com	24.594	5916.8	6
4	www.shopclues.com	4.815	5916.8	3
5	www.coursera.org	24.587	2630.5	6
6	www.bing.com	12.143	2630.5	4
7	www.nasa.gov	4.879	692.3	7
8	www.spacex.com	4.833	341.8	3
9	www.udacity.com	140.971	2614.1	6
10	www.bloomberg.com	25.001	519.7	8
11	www.mtv.ca	13.674	683.5	3
12	www.abdulkalam.com	118.32	5916.8	15
13	www.iisc.ac.in	20.917	5916.8	4
14	www.iitd.ac.in	44.496	5916.8	4
15	www.deutschland.de	4.887	2955.9	4
16	www.nottingham.ac.uk	20.625	2426.6	11
17	www.delhitourism.gov.in	16.394	5916.8	8
18	www.edx.org	24.552	385.4	6
19	www.twitter.com	5.256	2614.1	4
20	www.brahmakumaris.org	16.604	5916.8	9
21	ca.yahoo.com	31.69	791.9	8
22	www.stanford.edu	149.853	2614.1	6
23	www.ted.com	149.743	512.1	6
24	www.cyborgproject.com	145.862	512.1	22
25	www.india.gov.in	20.884	5916.8	4





Few interesting things that I observed here during experimentation are- Difference in number of hops even though the servers are at the same place. And few close servers reflect less one way time than servers which are far away. Hence, the one-way transit time is not directly proportional to distance and similarly the number of hops is also not directly proportional to distance.

2. There is a tool named topostats, mentioned on the Center for Applied Internet Data Analysis website. So here is the brief description of the same. Internet topology and its study is a broad field. In the last few years there has been a tremendous increase in interests in the study and designing of network topologies. The ultimate motivation behind this study is the designing of better and optimal network topologies. And apart from these goals, comparing various network topologies is also a goal. There are many different sources from where get is collected to study network topologies. Hence there is need of tools which can help in study of various network topologies, and their designing. Topostats is a complete package of programs. It is used to calculate various statistics on the network topologies. Though it doesn't compute all the statistics, it's computed statistics are mentioned in paper 'Lessons from Three Views of the Internet Topology: Technical Report'. And moreover this package doesn't itself make any plots. There are many statistical parameters which are the output of this tool like- Average node degree. When the sum of degrees of each node is divided by the number of nodes, we get Average node degree. Number of nodes is the count of the unique nodes found in the graph. Number of edges is the count of the undirected edges between the nodes in graph. Maximum node degree is the largest degree found across all nodes. In the graph, exponent of a power law distribution which best fits distribution of degrees, is called Degree distribution exponent, which is a parameter output by this tool. Assortative coefficient is another parameter, which is the Pearson correlation coefficient. This indicates amount of relationship b/w nodes of dissimilar or similar degrees. The power law distribution's natural cutoff value that best fits the distribution of degrees in the graph is known as Power-law maximum degree. Local clustering summed over all the nodes and then divided by the number of the nodes is known as mean clustering. Out of all the connected node triplets in graph, the percentage of three cycles is the clustering coefficient. The nodes which remain completely connected to each other when sorted in decreasing order out of the largest set of nodes is known as Top clique size. The smallest coreness across all the nodes is the minimum node coreness. Similarly the average node coreness as names says, is the average of all the coreness of nodes divided by the number of nodes. The number of the nodes which have maximum coreness is known as core size. Similarly the largest coreness across all nodes is known as maximum node coreness. And the number of nodes that have minimum coreness is the Fringe size and among all the nodes with minimum coreness, the one with the largest coreness is defined as maximum degree in fringe. The standard deviation of distribution of hops b/w the pairs of nodes is defined as standard deviation of distance. The average eccentricity across the nodes is graph radius. Similarly, the average eccentricity is the average of all the eccentricities of the nodes divided by the number of nodes. Out of all the nodes that have minimum eccentricity, the one with the smallest degree is defined as minimum degree in graph center. Similar definition applies to maximum degree in graph periphery. And there are many similar parameters like minimum node betweenness, average node betweenness, maximum node betweenness, minimum edge betweenness, average edge betweenness and maximum edge betweenness. Here is an example output from the tool, which is taken from official caida Website- The main outputs of the tool are - topology_stats, which calculates many simple statistics. Then the second one is distance, that calculates statistics on distance b/w all nodes in graph. Betweenness is the third parameter, which calculates the statistics on betweenness of all nodes in graph. And the other parameter is

components, that is a utility for taking out the biggest connected component in a graph file and also viewing all connected components in graph.
topology_stats output:
Number of nodes: 9204
Number of edges: 28959
Avg node degree: 6.29269882659713
Max node degree: 2070
Degree dist exponent (via CCDF) [warning: can be inaccurate]: 2.13036587884151
Power-law maximum degree [warning: can be inaccurate]: 3212
Normalized avg avg neighbor degree: 0.0469423442421394
Normalized max avg neighbor degree: 0.0530352021237445
Assortative coefficient: -0.235624710744895
Mean clustering: 0.456656295004884
Clustering coefficient: 0.0257810954106928
Top clique size: 16
Min node coreness: 0
Avg node coreness: 2.22761842677097
Max node coreness: 27
Core size: 47
Min degree in core: 68
Fringe size: 2460
Max degree in fringe: 5
distance output:
loaded 9204 nodes, 28959 undirected links, 42352206 pairs
decreasing radius from 0 to 5 with node 1
raising diameter from 0 to 5 with node 1
raising diameter from 5 to 6 with node 9
decreasing radius from 5 to 4 with node 22
raising diameter from 6 to 7 with node 3287
decreasing radius from 4 to 1 with node 21372
average distance = 3.115
std deviation of distance = 0.635
average eccentricity = 5.108
graph radius = 1
graph diameter = 7
min degree in graph center = 1
max degree in graph periphery = 1
betweenness output:
loaded 9204 nodes, 28959 undirected links, 42352206 pairs
min node betweenness = 0.0000e+00
average node betweenness = 2.2997e-04
max node betweenness = 2.4114e-01
min edge betweenness = 2.3617e-08

average edge betweenness = 1.0760e-04
max edge betweenness = 8.6025e-03
components output:
loaded 9204 nodes, 28959 undirected links
component at node 1: 9200 nodes, 28957 undirected links
component at node 21372: 2 nodes, 1 undirected links
component at node 21437: 2 nodes, 1 undirected links
3 components; largest at node ID 1

3. Given: Speed s = 2.5 x 10^8 metres/sec
Bandwidth R = 28 Kbps
Size of packet L = 100 bits
Distance = m
1 byte = 8 bits
1 Kbps = 1024 bps
Propagation delay = distance d / speed s
Transmission delay = size of packet L / Bandwidth B

Since it's given that Transmission Delay = Propagation Delay
So, L/R = m/s

Distance m = ((size of packet L)*speed S)/ Bandwidth
= (100*2.5*10^8)/28*1024
= 8.719*10^5 metres

So distance should be = 8.719*10^5 metres to make Transmission delay equal to Propagation delay.

4.
a.) Given: Packet size = F bits
Header size = h bits
Number of links = Q
Bandwidth = R bps
VC Setup time = t_s sec

Time to send packet (Ignoring queuing, propagation and any other processing delay) = Transmission Delay + VC Setup Time
= ((Size of packet + Header size) / Bandwidth)*Number of links) + t_s
= ((F+h)/R)*Q)+t_s
So time to send the packet will be ((F+h)/R)*Q)+ t_s seconds

b.) Given: Header size= 2h bits
Time to send packet = Transmission delay
= ((size of packet + Header Size)/ Bandwidth)* Number of links
= ((F+2h)/R)*Q
So time to send the packet over datagram packet switched network will be ((F+2h)/R)*Q seconds

5. Given: Bandwidth R = 2 Mbps
Speed s = 2.5 * 10^8 meters/sec
Distance d = 20,000 km
Length of packet L = 800,000 bits
1 Mbps = 1024*1024 bps

Time taken to send file = Transmission delay + Propagation delay
= L/R + d/s
= ((800000/(2*1024*1024)) + ((20000*1000)/(2.5*10^8)))
= 0.3815 + 0.08
= 0.4615 sec

So time to send the file as one big message will be 0.46 sec

Acknowledgement time = 100 ms= 0.1 s
Now we have 20 packets. Since each packet contains 40,000 bits. Therefore, time taken by one packet to travel will be = Transmission delay + Propagation Delay + Acknowledgement time
= L/R + d/s + 0.1
= ((40,000/(2*1024*1024)) + ((20000*1000)/(2.5*10^8))) + 0.1
= 0.001907 + 0.08 + 0.1
= 0.181907 sec

So time to send twenty packets = 0.181907*20 = 3.63814 sec ≈ 3.64 sec

6. Given:
File Size L = 1.5 MB
RTT = 80 ms so Propagation time will be RTT/2 = 40 ms
Handshake Time = 2*RTT = 160 ms

a. Bandwidth B = 10 Mbps
Time taken to send file = Handshake time + Transmission delay + Propagation delay
= Handshake time + L/B + Propagation delay
= ($160*10^{-3}$) + ((1.5*8 Mbits)/(10*1024 bits/sec)) + ($40*10^{-3}$)
= 0.16 + 1.2 + 0.04
= 1.4 sec

Hence time to send the file is 1.4 sec.

b. Since there is a wait of 1 RTT after sending each packet. So overall propagation time is 40+80=120ms
Time taken to send file = Handshake time + Transmission delay + (Propagation delay * #packets)
= Handshake time + L/B + Propagation delay
= $(160*10^{-3})$ + ((1.5*8 Mbits)/(10*1024 bits/sec)) + $(120*10^{-3}$*(1.5MB/1KB))
= 0.16 + 1.2 + 184.32
= 185.68 sec
Hence time to send the file is 185.68 sec.

c. Propagation time is zero as given. So the only time required will be the wait time, which is 1 RTT after every 20 packets.
Hence Time taken to send file = Handshake time + Transmission delay + Wait Time
= Handshake time + L/B + Wait Time
= $(160*10^{-3})$ + ((1.5*8 Mbits)/(10*1024 bits/sec)) + $(80*10^{-3}$*(1.5MB/1KB/20))
= 0.16 + 1.2 + 6.144
= 7.504 sec
Hence time to send the file is 7.51 sec.

d. Propagation time is again zero as given. In this case it follows a G.P. Suppose we have to wait for n times (1 RTT for each time). So total number of packets as per question will be-
1+2+4+8+.....
Which is same as $2^0+2^1+2^2+2^3+.....2^{n-1}$ and this is equal to number of packets calculated from total file size/size of one packet i.e. 1.5MB/1KB

So $2^0+2^1+2^2+2^3+.....2^{n-1}$ = 1.5MB/1KB
Applying sum of G.P formula $a(r^n-1)/(r-1)$ and solving it further-
$2^n-1=1.5*1024$
$n=\log_2 1537$
n=10.5859

So Time taken to send file = Handshake time + Transmission delay + Wait Time
= Handshake time + L/B + Wait Time
= $(160*10^{-3})$ + ((1.5*8 Mbits)/(10*1024 bits/sec)) + $(80*10^{-3}$*n)
= $(160*10^{-3})$ + ((1.5*8 Mbits)/(10*1024 bits/sec)) + $(80*10^{-3}$*10.5859)
= 0.16 + 1.2 + 0.846872
= 2.206872 sec
Hence time to send the file is 2.21 sec.

7. Assuming each time a new message is floated (i.e no reply is initiated)

a) s,r11,S,D,50000,21,Data

b) r14,r21,S,D,50000,21,Data
c) r22,r34,S,D,50000,21,Data
d) r31,d,S,D,50000,21,Data
e) r11,s,D,S,51000,21,data
f) r21,r14,D,S,51000,21,data
g) r34,r22,D,S,51000,21,Data
h) r14,r21,S,D,52000,22,Data
i) r22,r34,S,D,52000,22,Data
j) r31,d,S,D,52000,22,Data
k) r11,s,D,S,53000,22,Data
l) r21,r14,D,S,53000,22,Data