Bandwidth analysis of mobile networks

Rajwinder Singh, Narinder Pal Singh, Gyan Sagar Sinha, Ravee Malla
2008CS50223, 2008CS50218, 2008CS10170, 2008CS50224
Assignment 3, Computer Networks

8 October 2011

Abstract

We present below the findings of a series of experiements conducted to test the bandwidth and related parameters of a TCP and UDP connection on a Blackberry mobile phone using a Reliance connection. In the first part, we repeatedly established a TCP connection to a set of files, and report on the average latency, delay and downrate rate. For the second part, we use a customized UDP server using a new protocol to measure the UDP service parameters. We drew a heat map of the TCP rate plotted in IIT Delhi campus.

1 Introduction

In this assignment, we have written 2 clients on a Blackberry mobile phone using Eclipse editor. One client establishes a TCP connection to a server within IIT, and tries to download files of varying sizes. The other client establishes a UDP connection over EDGE/GPRS to a customized UDP listener which replies back with a protocol described later. These tests were conducted over a set of days so that we can average over spurious changes in the network.

2 Part 1

In these tests, we are required to set up a TCP connection over EDGE/GPRS to files of varying sizes residing on an IIT server. We first send a HTTP HEAD request so as to get the constant sized head reply from the server. Since a HEAD request involves an exchange of only one packet, we know that the time taken is the latency. Next we send an actual HTTP GET request for this file and note the time required. We repeat this at several locations to obtain the heatmap of IIT Delhi showing the TCP downlate rate.

2.1 Details of files

3 files were chosen of varying sizes and were queried repeatedly by a TCP connection over a persistent and a non-persistent HTTP connection. The 3 files chosen had sizes 10KB, 70KB and 5MB respectively.

2.2 Latency

Latency is defined as the round trip time of a packet between 2 end hosts. Hence, we computed this as the time elapsed between when an HTTP HEAD request was sent, and when a reply was received. The average latencies & standard deviation in the readings taken at various places.

- We find that persistent & non-persistent connections don't have much difference which is as expected since HEAD request is only a single packet transfer, and it doesn't matter if the HTTP connection were to remain alive after the inital packet transfer.
- HEAD request is used here as we only want a transfer of a single packet which would give an estimate of the round trip time, which is equal to the latency in this case.

Filename	Avg. Latency (ms)	Std. Deviation(ms)	Persistence
great.txt	322	108	Persistent
great.txt	364	152.2	Nonpersistent
paper-reading.pdf	365	64	Persistent
paper-reading.pdf	184	12	Nonpersistent
giving-talks.pdf	366	54	Persistent
giving-talks.pdf	403	30	Nonpersistent

Table 1: HTTP latency report

2.3 TCP Download Rate

We computed the average TCP download rate as the file size divided by the time taken to download the file. We note that there is a clear increase in the download speed as we are downloading larger files. This is because TCP gradually increases it's congestion window and the network can support such large bandwidth. As the file size increases, the TCP can over time utilize the full available bandwidth to it.

Filename	Avg. Rate (kbps)	Std. Deviation(kbps)
great.txt	6	4.2
paper-reading.pdf	356	135
giving-talks.pdf	700	200

Table 2: TCP download rates

2.4 Heat maps

Using the Heatmaps[1] API, we drew a heatmap to plot the average TCP download rate geotagged by the location measured by the GPS tracker on Blackberry. The map is presented below. We note that the service is better towards the hostel areas. The locations chosen were near to Aravali Hostel, Faculty Housing, GCL, Wind-T, etc.

[1] heatmapapi.com

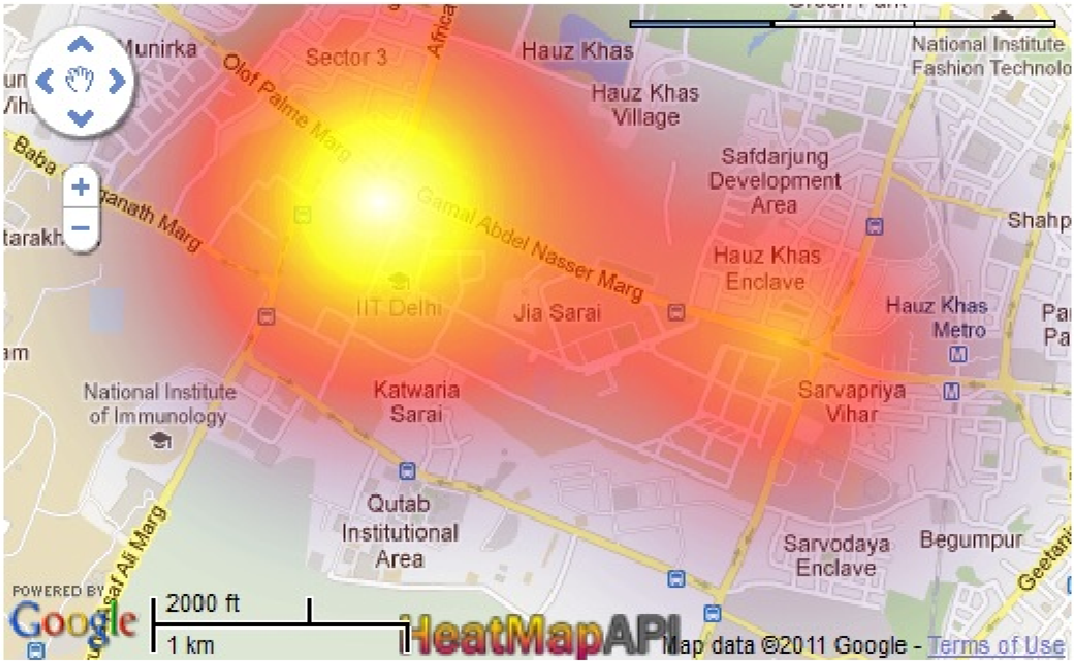

Figure 1: Heatmap for IIT Delhi as on October 10

3 Part 2

In these tests, we probed the bandwidth of the EDGE/GPRS cellular networks. We used a UDP client and communicated with a UDP server running at `124.124.247.5` on port `9010`.

3.1 Packet pair tests

Our model has the following parameters.

- S : Time gap at client
- R : Times to repeat
- N : Time gap at server
- ID : ID to be assigned to the UDP packet

We take these values from the user, and repeat the experiments.

S-actual	**S**	**N**	**S**	**N**	**S**	**N**	**S**	**N**
10	9	11	11	5.5	13	24	14	32
20	21	1	19	19.5	21	27	22	34
100	110	98	102	82.5	99	92	106	91
200	225	200	235	181	205	183	220	205
500	503	454	522	500	520	514	502	494
1000	1003	983	1038	1031	1007	979	1020	1020

Table 3: The time difference between packets at client and server

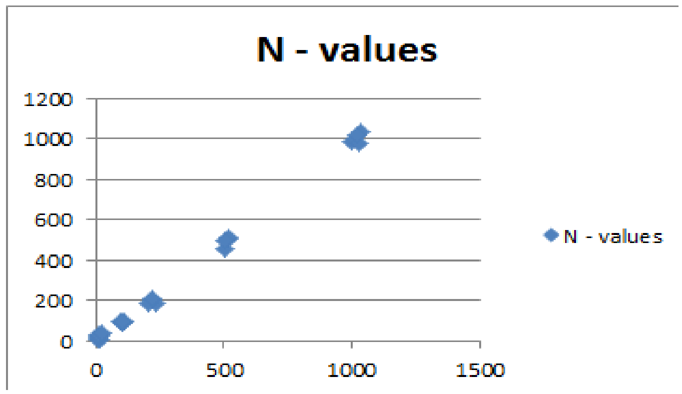

Figure 2: Time lag at server for various time differences at client

Available bandwidth is $\frac{|N-S|}{S} * C$. In presence of a single bottleneck link, $\frac{|N-S|}{S}$ is a measure of the utilization of this link by virtue of the delay it causes in packets from one source. We found out the **average advertised bandwidth** to be around *200 kbps*. Comparing this to our measured values of around 100kbps upwards gives reassurance.

We find that in many cases the time difference decreases between the 2 packets at the server. This means that subsequent packets are routed faster by the network, since if packet 1 took δ to reach server, packet 2 took less than δ. This points towards the adaptive nature of this network, such that it routes subsequent packets faster.

In the way we have interpreted the variables S & N, the formula given in the assignment doesn't actually make sense as the available bandwidth. For one, N can be less than S by the following argument. If we suppose UDP packets are routed independently, then both packets can take two different times transfer times. Hence, the time difference of arrival at the server can actually decrease just as well. Hence $\frac{N-S}{S}$ becomes a negative quantity.

To be able to probe the downlink, we need an additional ECHOTrain protocol with which a client can request for in general a train of packets from the server which can test the downlink. No, the available bandwidth may appear *greater* for UDP tests since UDP is more aggressive and doesn't care about congestion control or fairness.

3.2 Packet train tests

In these tests, we send two small packets separated by a train of large packets. The spacing between the two packets will suggest the time taken to process the train packets. As number of train packets increase, the time difference increases. However after a certain threshold, the difference become constant since remaining train packets get dropped. We can also measure the number of packets dropped by querying STAT on these packets. All the results from these experiments are reported below.

For the sake of simplicity, we send first packet of the pair and the packet train, then wait for the specified sleep time and subsequently send the second packet of the pair. We adjusted N and S, and found that *24 packets* is the **threshold limit on packet queue size** at the gateway after which packets start getting dropped.

From the scatter plot, we see that we get the best service at a train size of 16 packets, after which the delay between 2 packets of the pair becomes constant. Additionally, we found time delay of 100ms and packet size of 1KB to work.

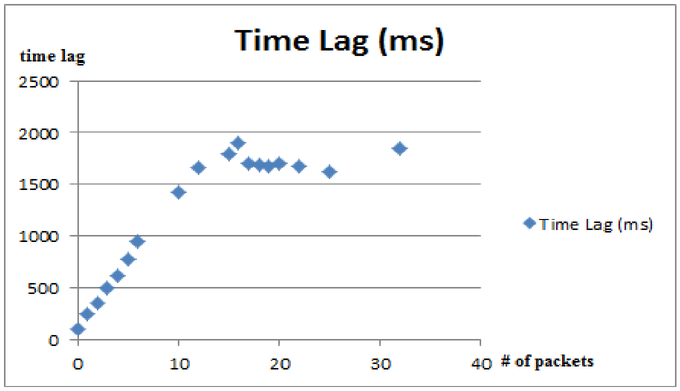

Figure 3: Time lag between packet pairs for various train sizes

3.3 Bonus : Downlink tests

To be able to measure the downlink bandwidth, we should have

- A new control message ECHO2(id) which replies back with when did the server start replying the file packets
- Modify ECHO so as to specify both the length of bytes requested and the number of packets it should be divided into
- We could also have an option to specify at what time the server should send each packet, so that we can model realistic traffic situations

4 Conclusion

We find Reliance provides good network connectivity. We ran a set of bandwidth tests, and found that there is very little congestion on Reliance mobile network. This is due to good service of the provider as well as the powerful radio of Blackberry device.