CSC458
Course Project

Due on Sunday, November 25, 2018

JooHyung Kim, Enhao Wu

November 26, 2018

Introduction

In the report, we will look into analysis of a captured PCAP file in three fields: Per Packet, Per flow, and RTT estimation. In addition RFC-6298 is used to estimate sample RTT. The goal is to observe the difficulties of reconstructing the state of a TCP connection in the network and to investigate how accurate and stable RTT estimation is in practice.

System and Tool Specs

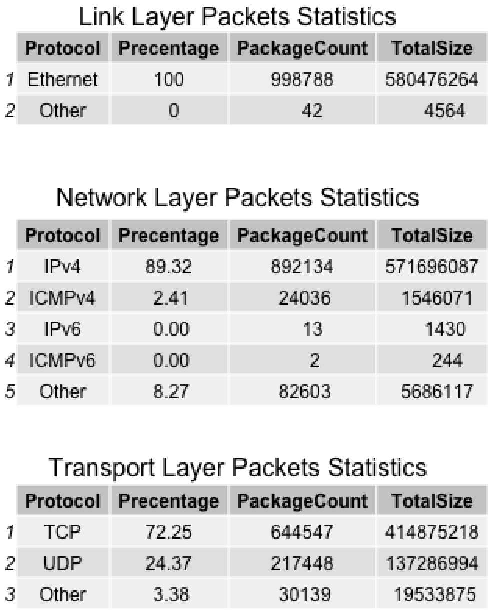
OS Environment	MacOS / Linux
Coding Language	Shell/Bash, R(version 3.4.4)
Required R Libraries	grid, gridExtra, ggplot2, varhandle, rlist
Network Analysising Tool	tShark(version 2.6.4), Wireshark

To analyze our trace file (univ1_pt9), we used the program Wireshark and tShark. R was used to create the codes to produce all of the CDF plots.

Wireshark and tShark was used to filter out and show only the necessary data, which then we exported it into a csv/tsv files. Afterwards, we used R to analyze these data files to create our CDF plots.

Per-Packet Statistics

Type of Packets:



Link Layer Packets Statistics

	Protocol	Precentage	PackageCount	TotalSize
1	Ethernet	100	998788	580476264
2	Other	0	42	4564

Network Layer Packets Statistics

	Protocol	Precentage	PackageCount	TotalSize
1	IPv4	89.32	892134	571696087
2	ICMPv4	2.41	24036	1546071
3	IPv6	0.00	13	1430
4	ICMPv6	0.00	2	244
5	Other	8.27	82603	5686117

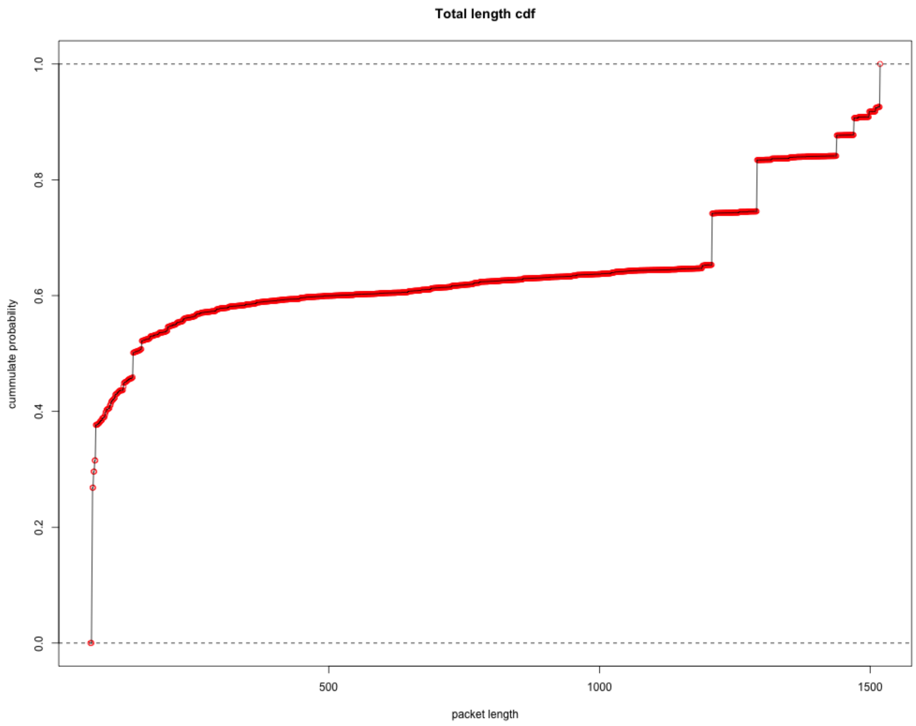
Transport Layer Packets Statistics

	Protocol	Precentage	PackageCount	TotalSize
1	TCP	72.25	644547	414875218
2	UDP	24.37	217448	137286994
3	Other	3.38	30139	19533875

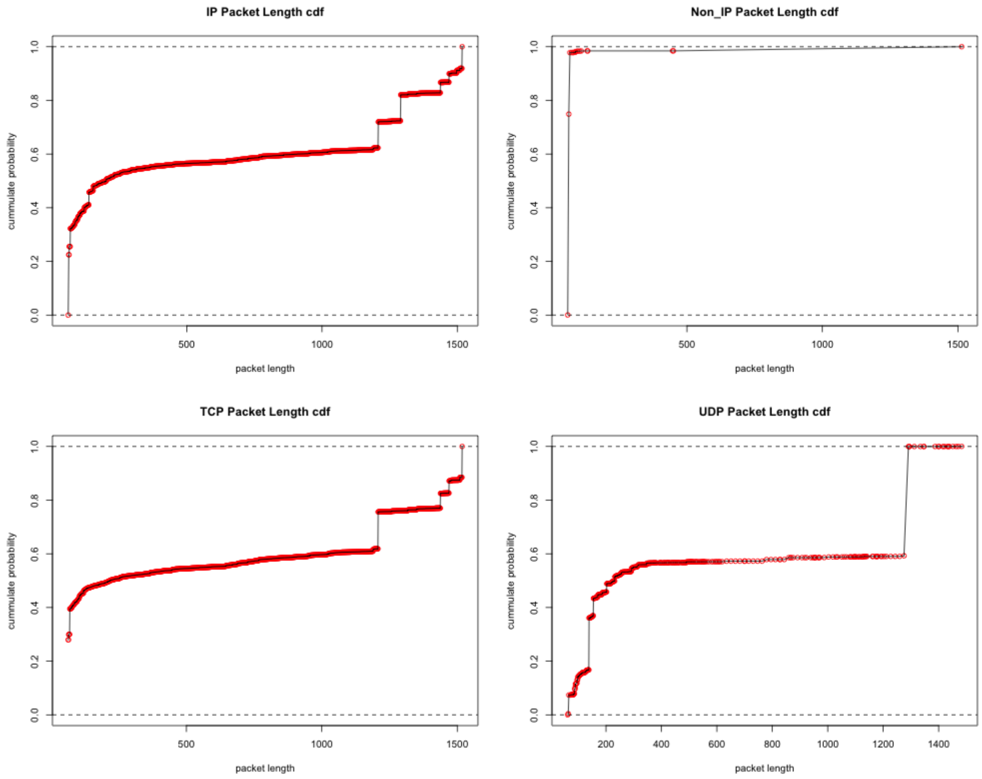
(a) Packet Statistics for different layers

Figure (a) shows the packet statistics for different layers in our trace file. Our Link layer is almost all (92.74%) Ethernet packets. In our Network Layer, it is mostly IP packets, IPv4 specifically. Lastly, as mentioned before, in our Transport Layer, it is mostly either TCP or UDP packets.

Size of Packets:



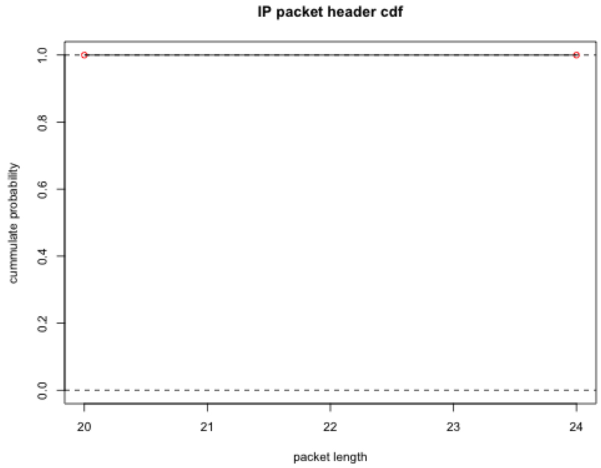
(a)

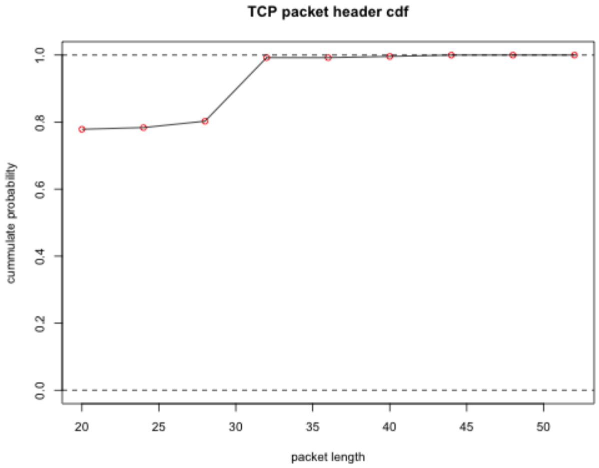


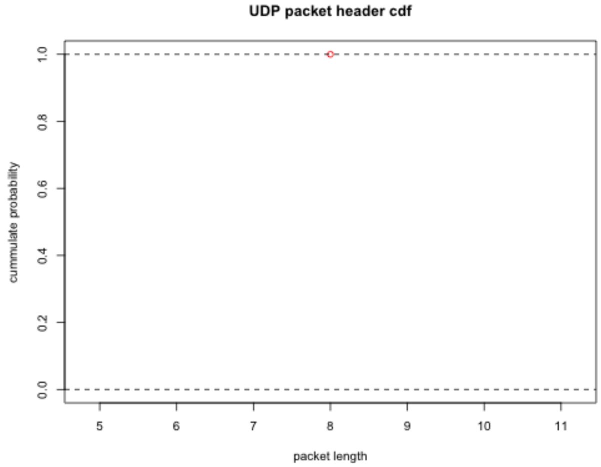
(b)

Comparing with figure(a) - CDF of total packets length with CDF of IP packet in packet in figure(b). We can see this two plots have similar shape. Therefore, we can conclude that most of packets are from IP packets.

From CDF of (TCP and UDP) packets length, we can see TCP packets typically has five levels of length: 100Bytes, 1250Bytes, 1400 Bytes, 1400 Bytes, 1480 Bytes, and 1550Bytes. While UDP has three levels of length, which are around 50-150 Bytes, 150-200 Bytes and 1300 Bytes. So we can say the size of TCP packets distribute uniformly in between 100 and 1550 Bytes, while UDP packets length are either very small($<$ 200 Bytes) or very big($>$ 1300 Bytes).



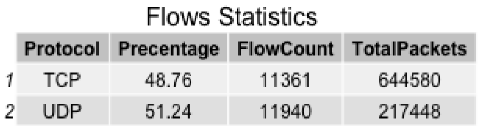




From above graph, we can see that IP header length(20 Bytes) and UDP header length(8 Bytes) are constant for almost all IP packets. Around 80% of TCP packets have 20 Bytes header length, the rest of TCP packets header size are in between 28 to 36 Bytes.

Per-Flow Statistics

Flow Type:



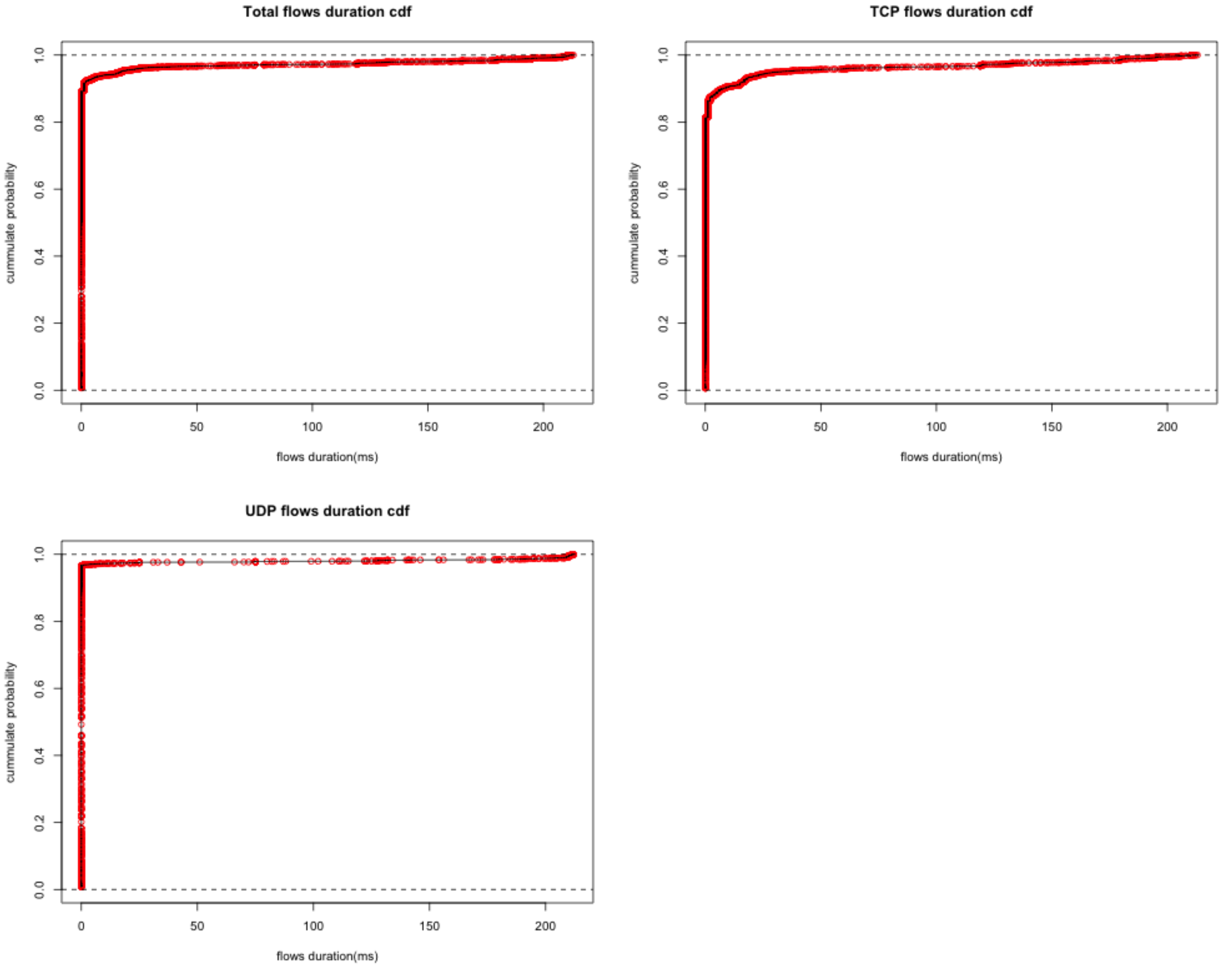
Flows Statistics

	Protocol	Precentage	FlowCount	TotalPackets
1	TCP	48.76	11361	644580
2	UDP	51.24	11940	217448

(a) Summary report about TCP and UDP flows

From our trace file, we have observed that there are slightly more UDP flows than TCP. However, TCP has the total number of packets is far greater.

Flow Duration:



(a) CDF of Flow Duration for all IP flows, TCP flows, and UDP flows

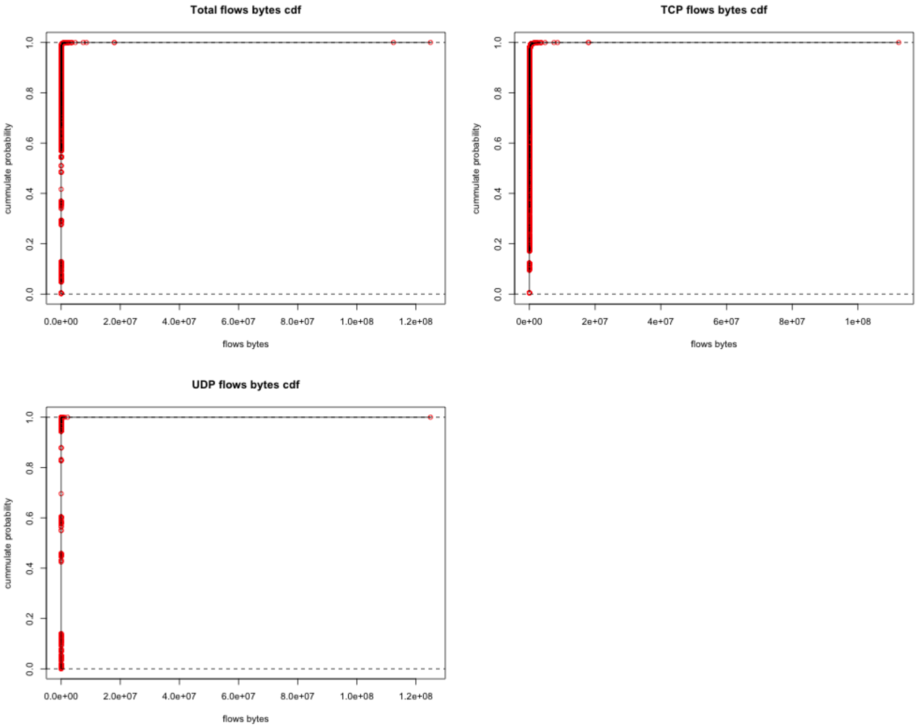
There is definitely a different between the TCP and UDP flows. It seems like the flow durations for UDP were mostly short compared to the TCP flows.

It looks like about 97% of the flow durations in UDP were really short and similar. And the rest seems to have take quite a long time, seeing how a huge portion is congregated at the end of the plot.

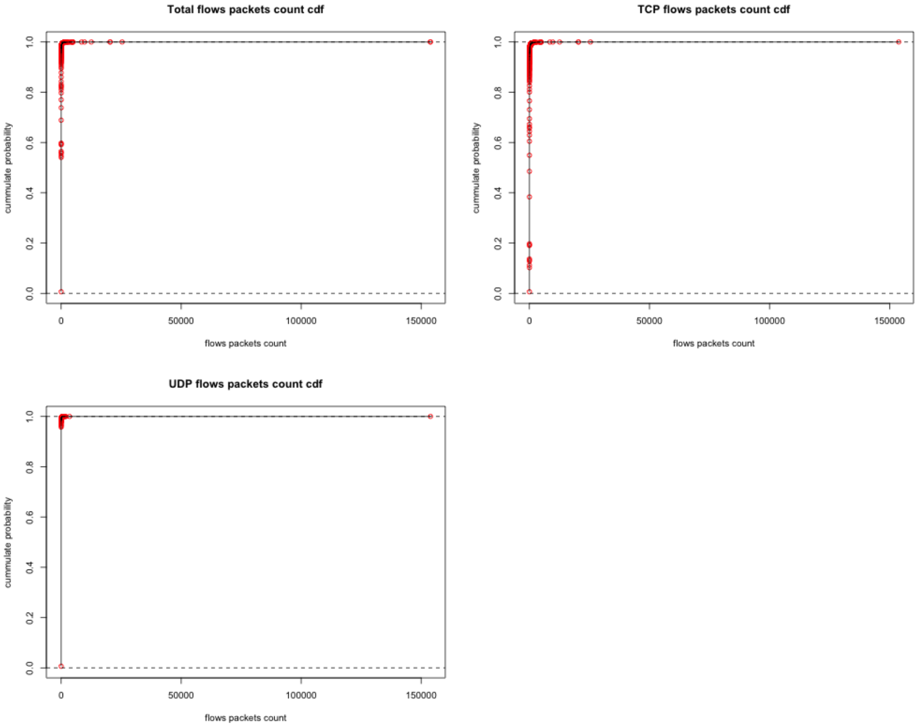
For TCP, about 85% of the flow durations were really short and similar in durations. and the rest took longer and had a wide variance, spread out equally compared to UDP flow durations.

We believe the TCP flows have more flows that took longer than UDP flows because there are much more packets that were transferred in each flow (as observed in the Flows Statistics chart).

Flow Size:

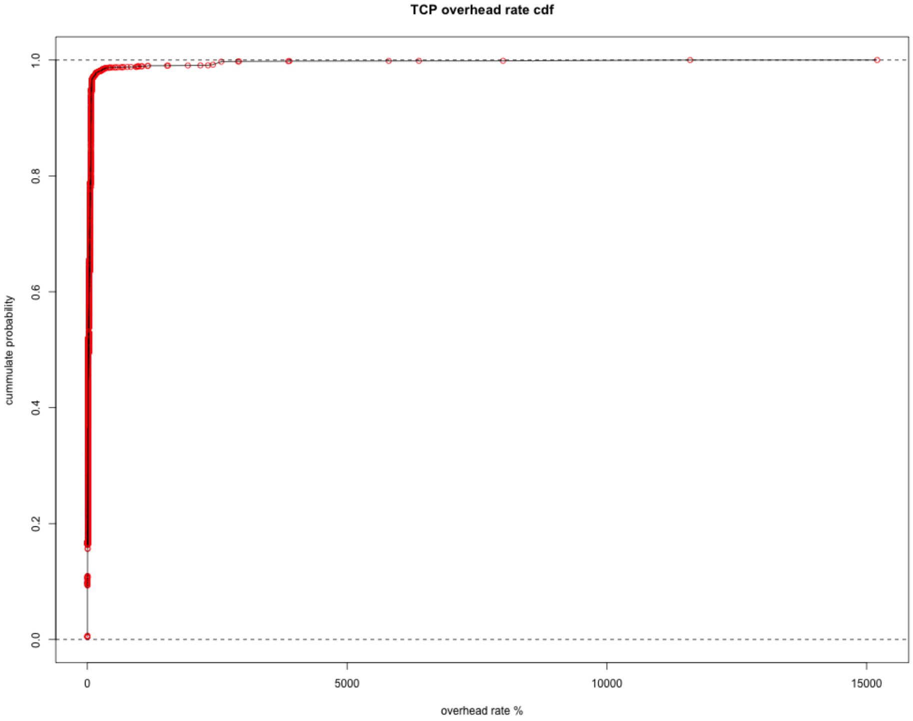


(a) Flow Size CDF /w Byte Sum (all IP, TCP, and UDP)



(b) Flow Size CDF /w Packet Count (all IP, TCP, and UDP)

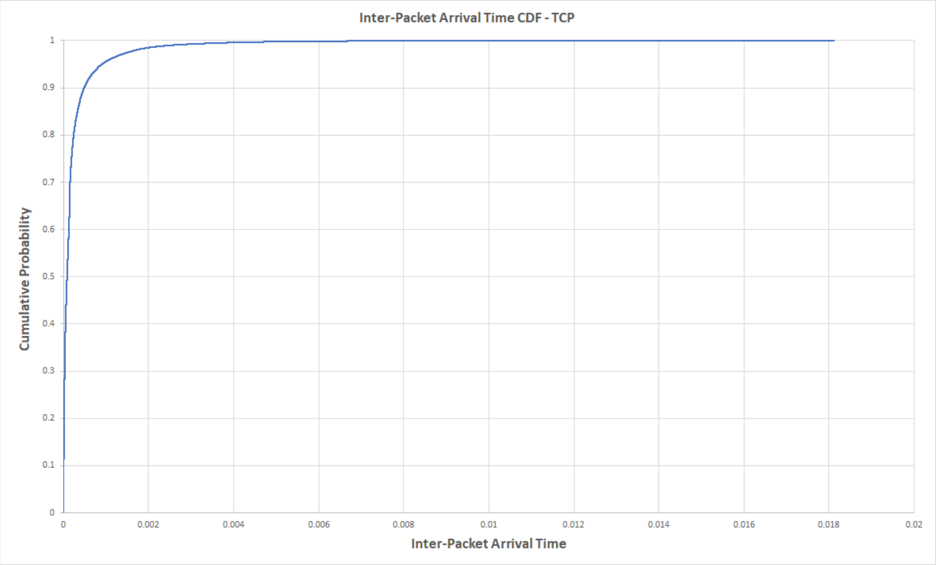
There is a huge difference in both CDF of flow size when creating the plots with either Packet Count or Byte Size. This is expected since in the Flows Statistics chart, it was shown that the total packets that were transferred in by TCP were triple the number than UDP packets.
One difference that can be seen from the Flow Size CDF with Byte Sum (a) is that there is more variance in the TCP flow sizes, in terms of Flow's byte sum, than the UDP flows.
Another difference when looking at the Flow Size with Packet Count (b) is that, almost all of the UDP flows has similar number of packets in each flow, whereas TCP flows has quite a variance in the number of packets in each flow.

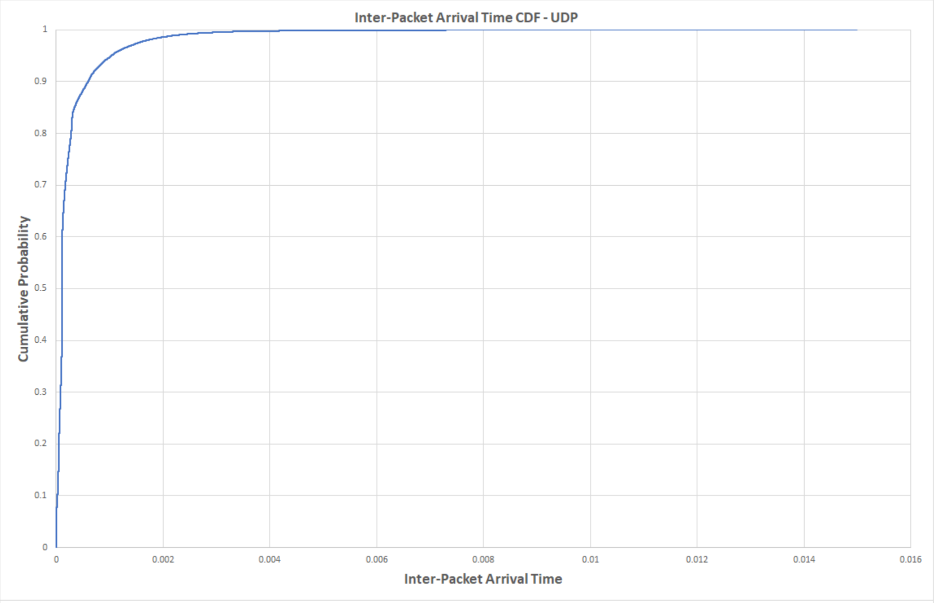


(a) CDF of overhead ratio

Based on the CDF of hit ratio that we have drawn up, we can say that there are overheads but not large for most of the TCP packets. Around 96% of the packets either have no overhead and very little. However, we can see from the plot that there are a few large TCP overheads. But still, most of all the TCP overheads are below the 5000.

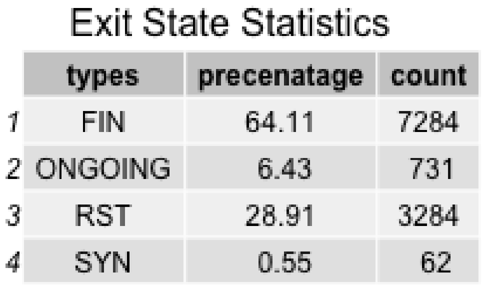
Inter-Packet Arrival Time:





From both CDF of inter-packet arrival time (TCP and UDP), we can see that most of the arrival time for packets are below 0.002 which happens most commonly, around 97% of the packets in the traces.
There aren't too big of a differences between the inter-packet arrival time for TCP and UDP flows. One thing that is noticable is that there are packets which arrive greater than 0.018 and in UDP there are no packets that takes longer than 0.018.

TCP State:

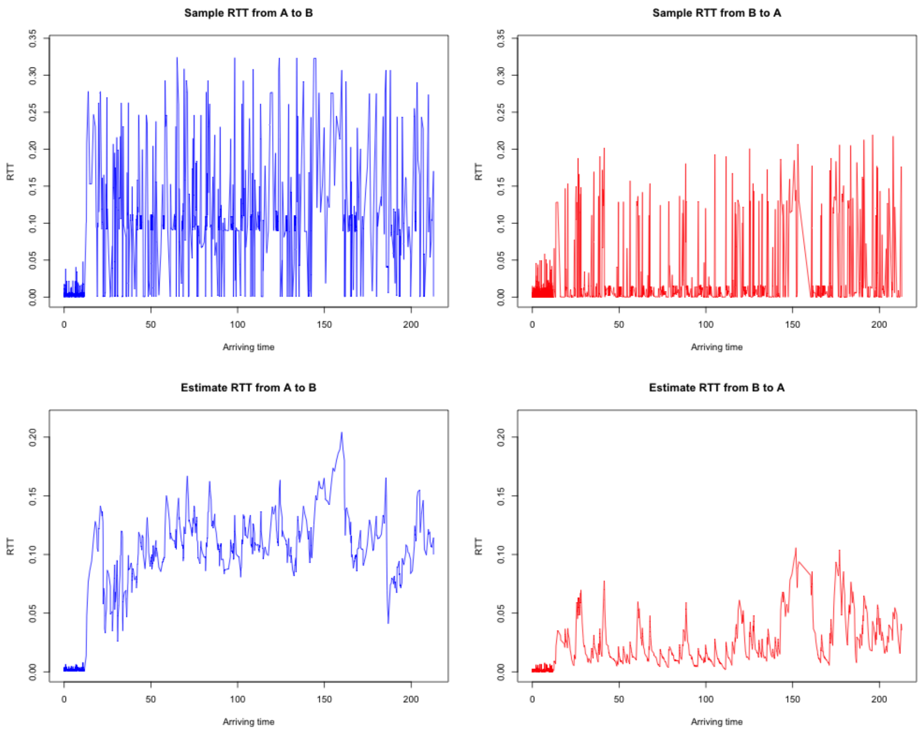


Exit State Statistics

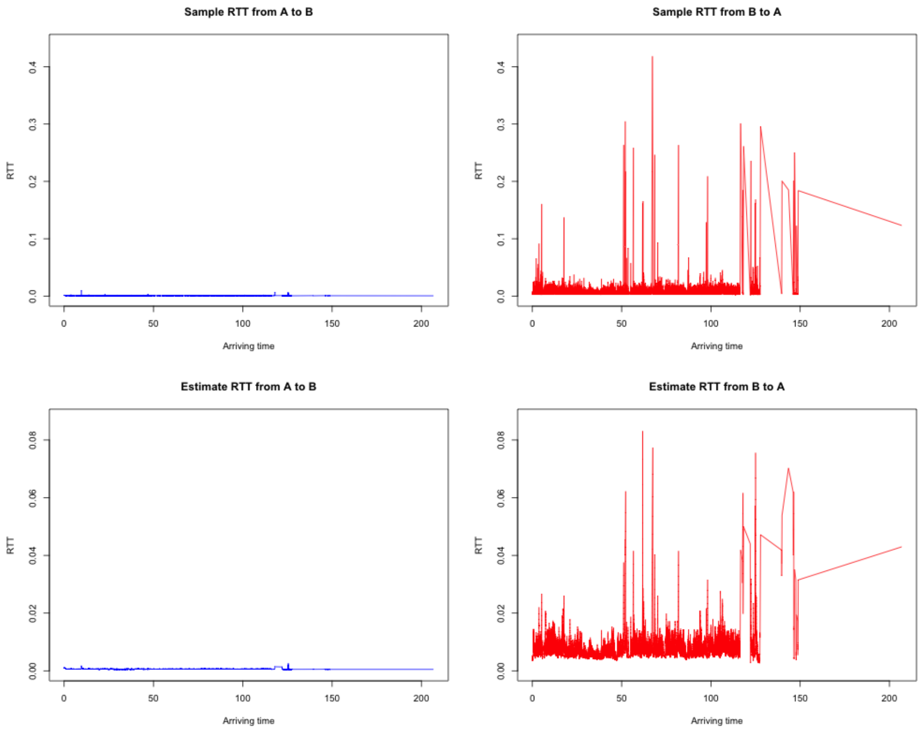
	types	precenatage	count
1	FIN	64.11	7284
2	ONGOING	6.43	731
3	RST	28.91	3284
4	SYN	0.55	62

From the Chart we can see that most of the TCP end states are either FIN or RST. FIN being the largest at 64.11% and RST being the second largest 28.91%.

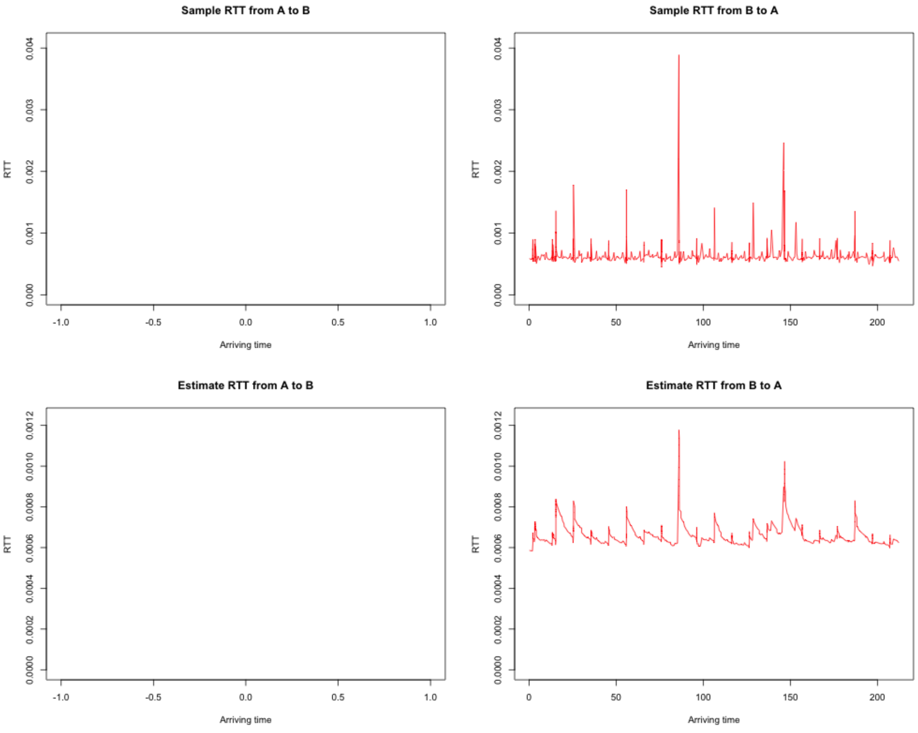
RTT Estimation



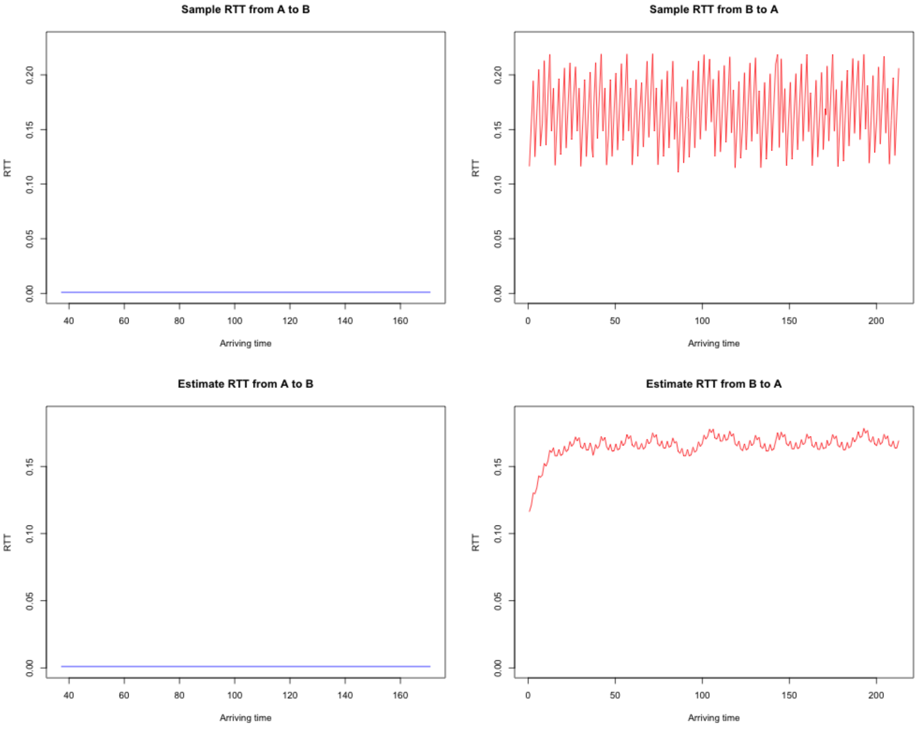
(a) stream#: 0



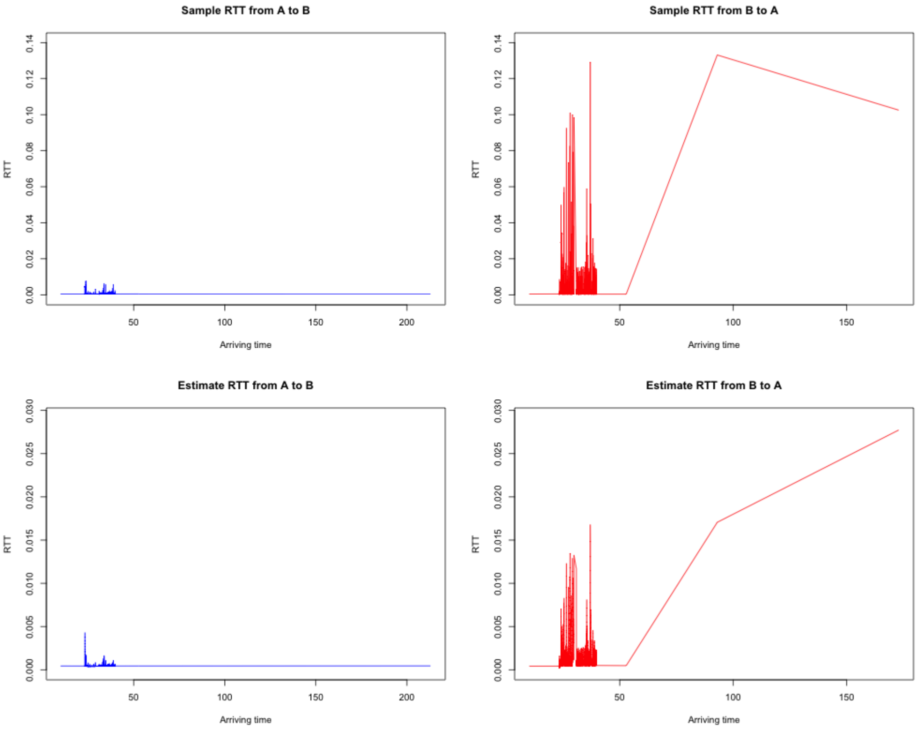
(b) stream#: 4



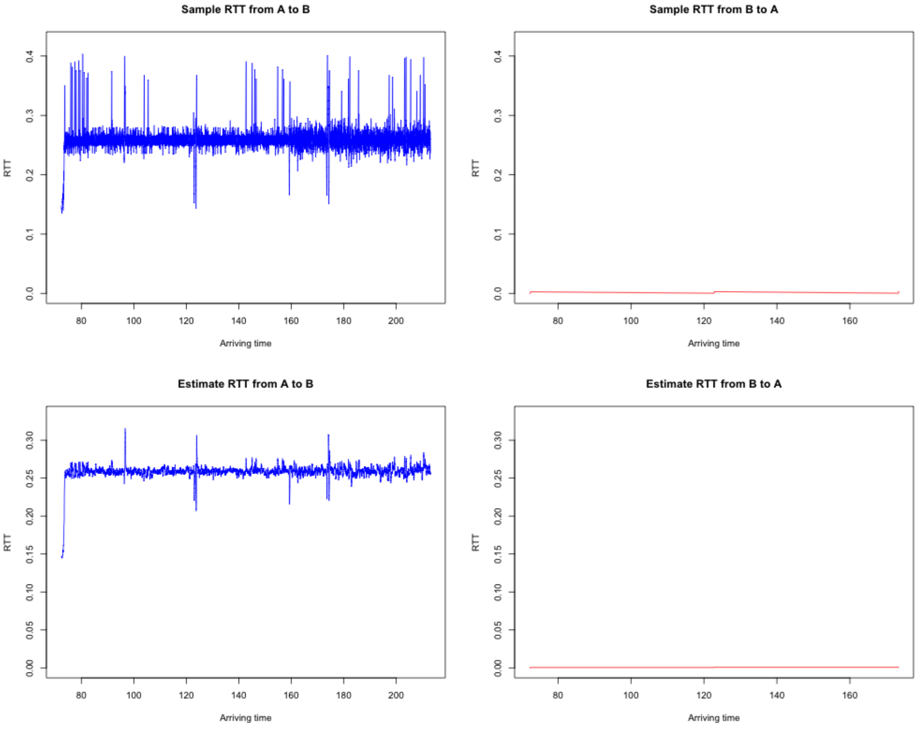
(a) stream#: 63



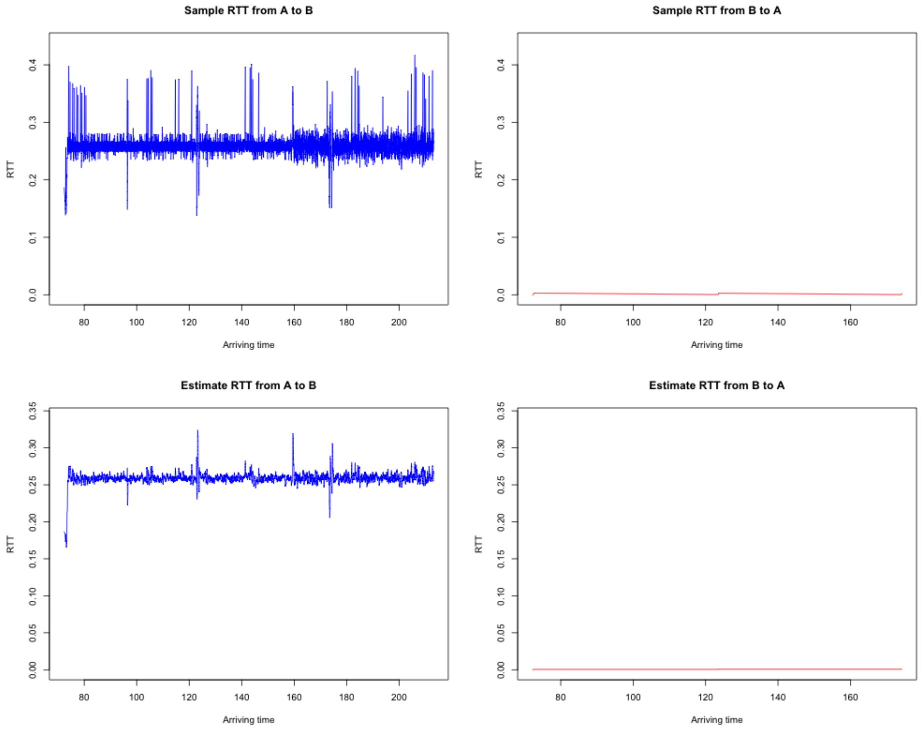
(b) stream#: 117



(a) stream#: 847



(b) stream#: 3963



(a) stream#: 3965

Sample RTT and Estimation RTT Analysis

NOTE: We labeled our TCP flows with different stream number from 0-11361. Within all TCP flows, we found that the top three TCP stream connections with largest package count are #4, #847, #3963; #4, #3963, #3965 are the ones with the largest Bytes sum ; #0, #117, #63 for the longest duration. Above 7 plots are for the corresponding to stream number of TCP flows.

Q: Is estimated RTT relatively stable?
From looking at the estimated RTT of our connections, we can see that they are not relatively stable. There are lots of increase and decreases (spikes) during the connection.

For example, Stream#847 RTT from B to A is very unstable for the beginning, then it only keeps on increasing till the end of the connection.

Stream#0, 4, 3963, 3965 shows unstable estimated RTT as well. however stream#63 and 117 shows a constant pattern of increase and decrease estimated RTT throughout the connection which is a bit more reliable. However for Streams# 3963, 3965. the Estimated RTT from B to A is stable, seen from the constant linear line, with no increase or decrease at all from the looks of it.

Q: Is sample RTT relatively stable?
As for sample RTTs, they are very unstable for Stream#0, 4, 847, 3963, 3965. Showing a similar unstable RTTs throughout the connection as the Estimated RTTs but with in a larger difference.

For stream#63 and 117, they show a constant pattern of increase and decrease, similar to the estimated RTTs. However, the patterns are at a much bigger scale and the difference in the decrease and increase is much larger and so it is very unstable.

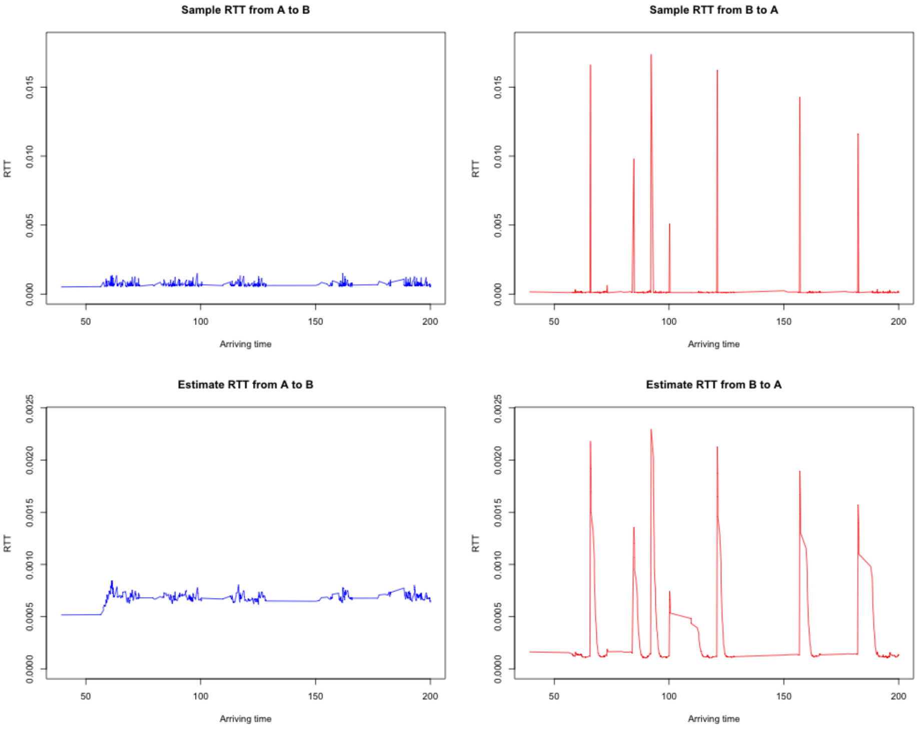
However for Streams# 3963, 3965. the Sample RTT from B to A is stable, seen from the constant linear lines. with either a slight increase in RTT or non at all.

Q: Do you see any increase or decrease in RTT? If yes, what can be the reason that the RTT is changing during the life of a connection?
Now, since these charts are created with conversations with large amounts of packet numbers, total byte sizes, longest duration. There could be numerous factors that changes the RTT during the life of these connections.

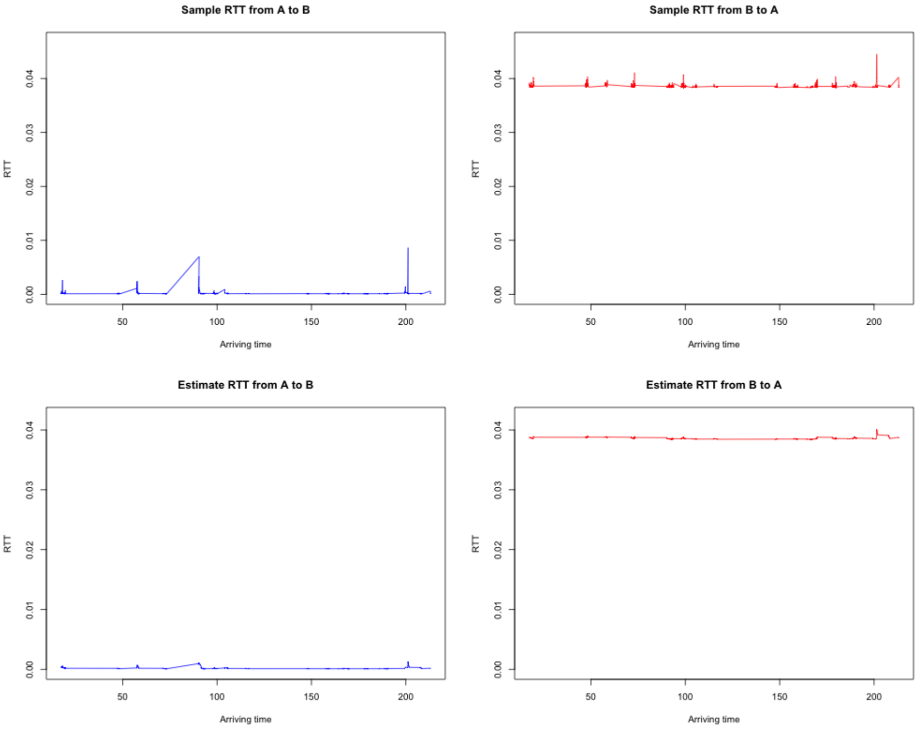
As stated in the Course Project, there are RTT differences between A to B and B to A because we're collecting the RTT between the hosts and the collection point. Now, since we don't know the physical distance between the collection point and the hosts. We can assume that the difference in distances has a factor in the different RTTs. Since there is a difference in distances, there could be difference in the routes, hops, it takes to reach the collection point, which also affects the RTT.

Now, for the changing of RTT during the life of the connection could be from maybe the collection point is being congested by other connections. The collection point could also be congested by the packets from the hosts by sending large size packets fast. Another cause could be the transmission medium is wireless and therefore the frequency could have been interfered during the connection which caused the unstable RTT.

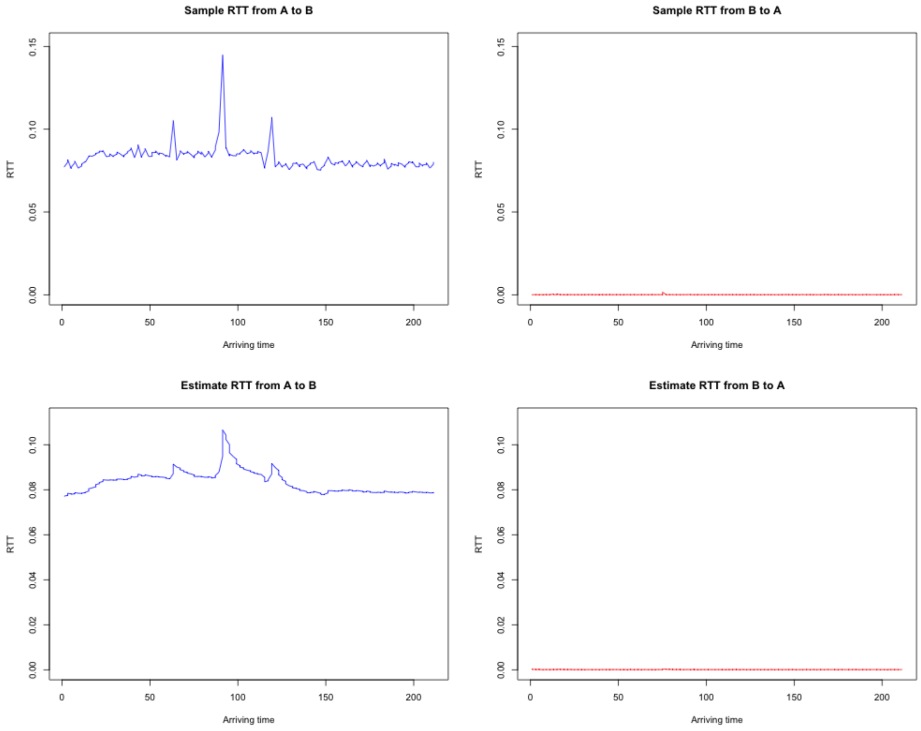
Analysis of Top 3 pair of hosts with highest TCP connections



(a) Chart 1



(b) Chart 2



(a) Chart 3

Analysis of Top 3 Median Estimated RTT

NOTE: After comparing all TCP flows, the top 3 pair of hosts that have the highest number of TCP connections are:

Top 1(chart1)	*244.3.160.239*	*244.3.48.41*
Top 2(chart2)	*94.2.202.197*	*244.3.160.239*
Top 3(chart3)	*244.3.160.254*	*244.3.160.204*

The top 3 pairs can be found in the log.txt file

Q: Comparing sample RTTs and estimate RTTs of 3 different hosts pairs. Analyzing and reasoning the patterns you find.

There aren't any specific patterns for charts 2 and 3, there are few huge spikes in the RTT during the connection, but it looks very irregular.

However in Chart 1, we do see a pattern in increasing very high then dropping back down to the 'normal' constant RTT throughout the connection. Some of the drops take longer than other drops but it does seem constant.

The reason for these spikes could be from the connection point being congested at those moments of the spikes, causing the RTT to take significantly longer than usual. Afterwards, the connection point realizes the congestion and handles it, which then brings the RTT down back to 'normal' time.

When comparing the 3 charts, we can see that the Chart 1 has the fastest RTT even with the spikes, then chart 2, then lastly chart 3. Chart 2 and 3 does has spikes in the RTT however they are different from the way the spikes look in chart 1. Almost all the spikes in Chart 1 seems to increase then decrease as soon as possible.

However, Chart 2 and 3 seems to spike up, then slowly decreases to 'normal' RTT. The reason for this difference could be from the collection point using a different kinds of congestion handling algorithms. Moreover, the difference in the RTT between charts could be from the size of the packets that are being transferred between on average.