## CSC 458/D58/2209 Course Project  Due Date: Sunday, November 25, 2018

RTT Estimation in Real World

# Dataset Statistics:

**General Statistic about the traffic:**

**a) Per-packet Statistics**

Total # of packets = 998830

Total # of bytes = 82925754

Statistics table for Link Layer:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Packet Type | Packet Count | Total # of Bytes | Packet Count Percentage | Packet Bytes(Size) Percentage |
| Ethernet | 998830 | 82925754 | 100% | 100% |

Statistic table for network layer:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Packet Type | Packet Count | Total # of Bytes | Packet Count Percentage | Packet Bytes(Size) Percentage |
| IPv4 | 892187 | 76337072 | 89.32% | 92.05% |
| Other | 61959 | 3717972 | 6.20% | 4.48% |
| ICMP | 24038 | 1450109 | 2.41% | 1.75% |
| IPv6 | 13 | 1248 | 0.001% | 0.001% |

Statistic table for transport layer:

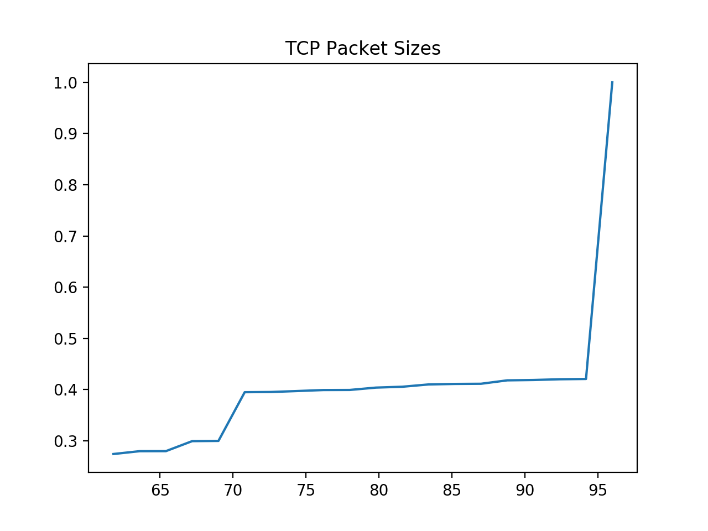
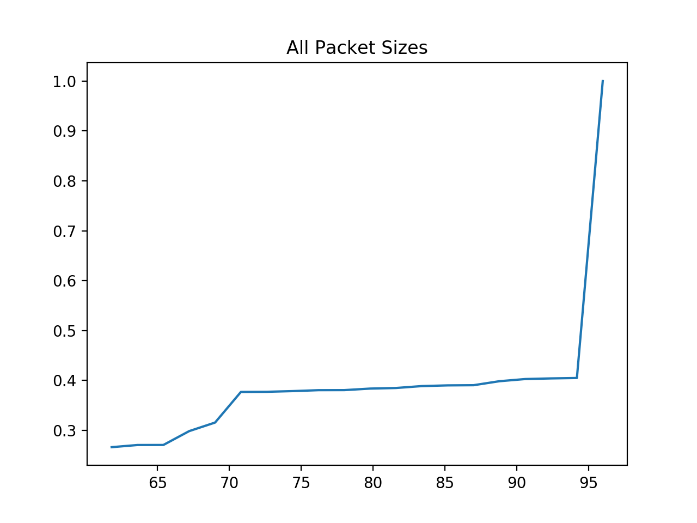
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Packet Type | Packet Count | Total # of Bytes | Packet Count Percentage | Packet Bytes(Size) Percentage |
| TCP | 697672 | 56378459 | 69.85% | 67.99% |
| UDP | 224516 | 20735303 | 22.48% | 25.00% |
| Other | 56009 | 4392639 | 5.61% | 5.30% |

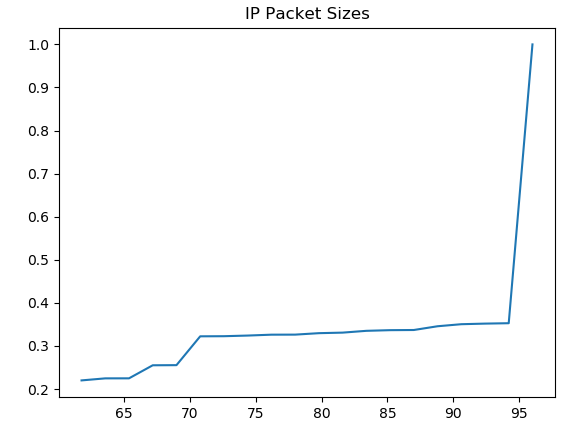
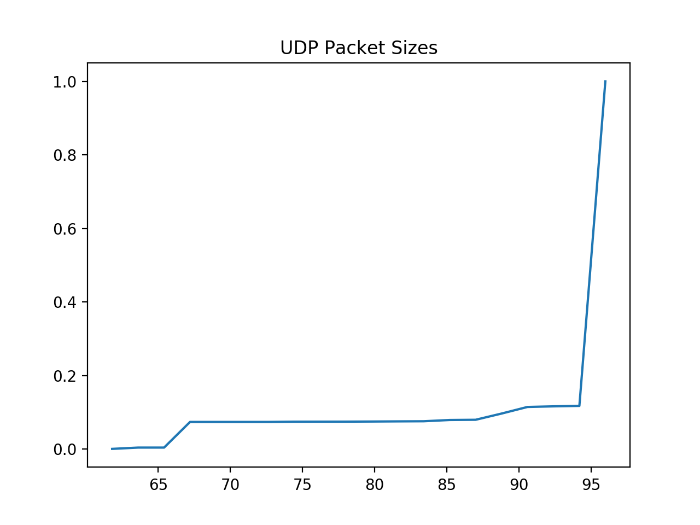
**b) CDF for Size of Packets**

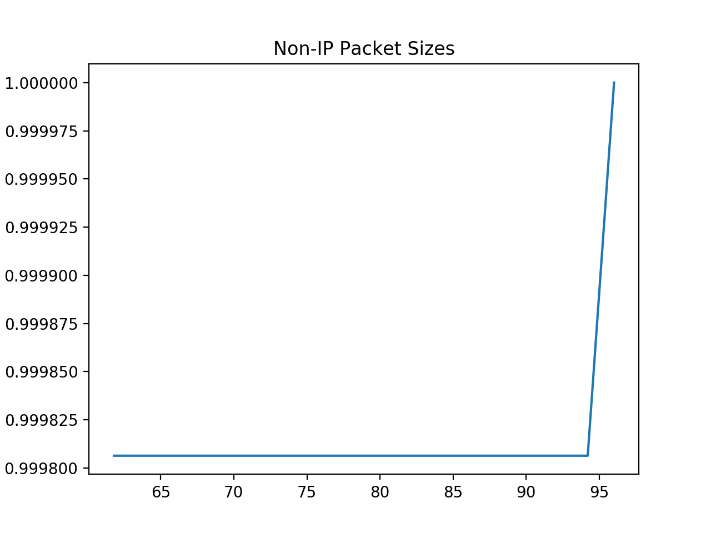
*Q: For many of the CDF charts, it is better to use logarithmic scale for the x‐axis. Explain why this is usually the case.*

There are two main reasons we usually use logarithmic scale for the x-axis in this case. The first reason is there are a few points are much larger or smaller than the bulk of the data. The second reason is the logarithmic scale can show percentage change clearly.

That being said, however, we decided against using the logarithmic scale for the following CDF’s because with the scaling we had already, the packet sizes at each point in the graph would be much clearer.

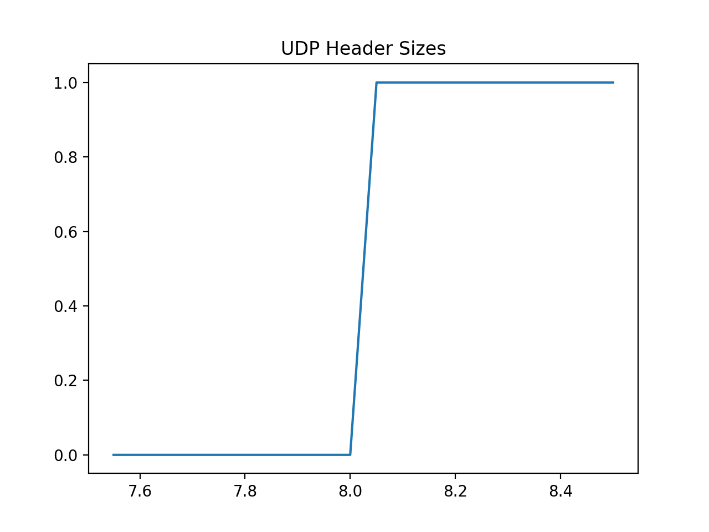
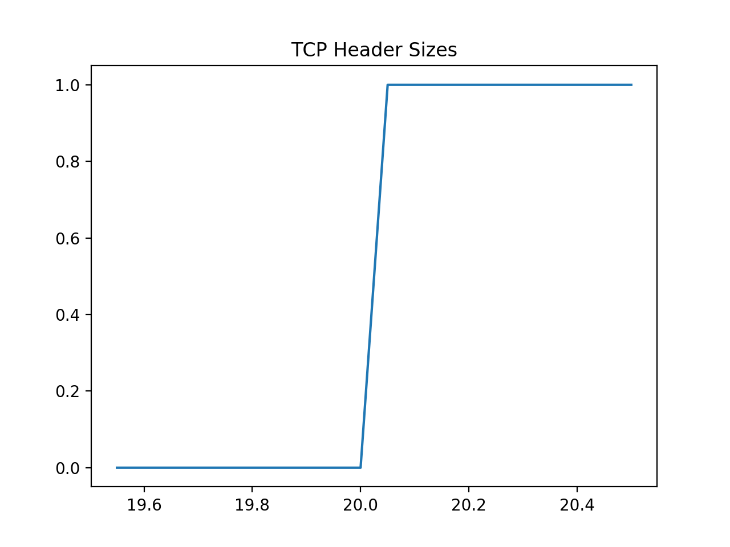
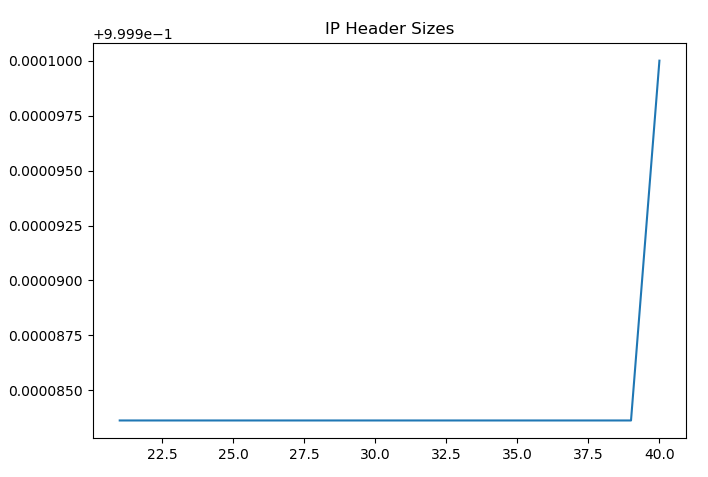






***Packet Size Analysis:***

* For all packet types, the vast majority of packets have size larger than 95 bytes.
* Additionally, only non-IP packets did not follow the same probability growth curve as the others – steady increases up until the 95 byte mark.
* Between UDP and TCP packets, TCP packets seem to generally have larger packet sizes than UDP.



***Header Size Analysis:***

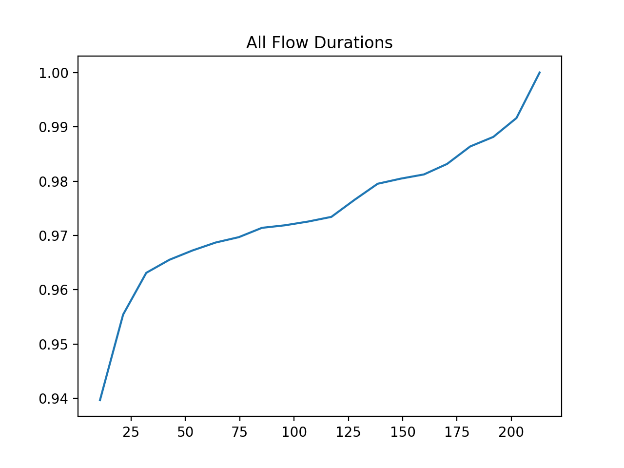
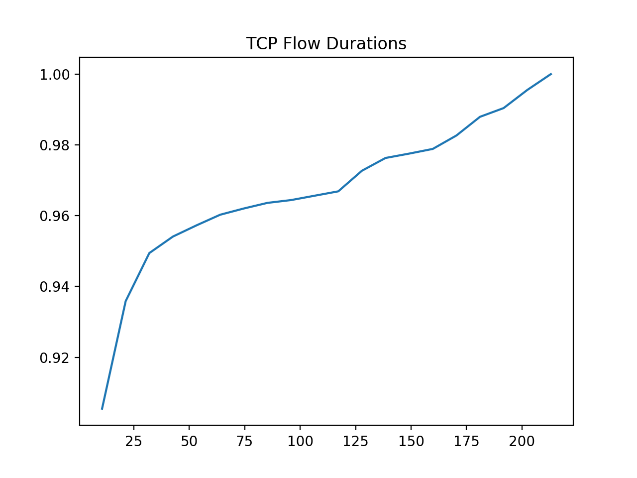
* The TCP and UDP header sizes match our expectations of their distribution. They were largely all 20/8 bytes large and that was represented by our graphs.
* The IP header size looks a little odd from the y-axis but this was because of the hyper-compressed results. There were only a few outliers with 40 byte headers and the vast majority were 20 bytes. As such, the graph was skewed to what is shown.

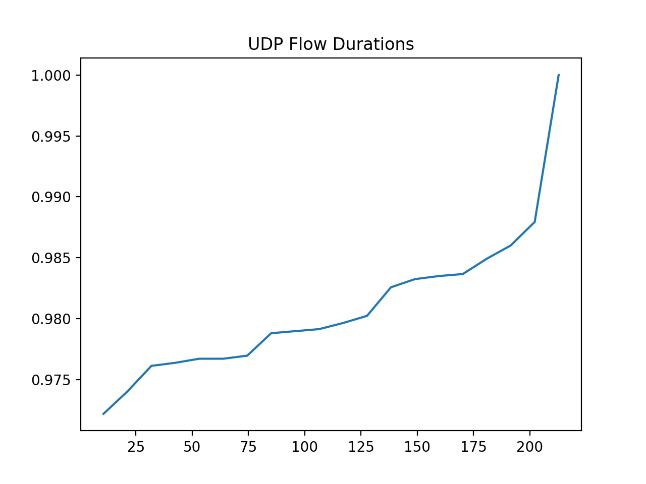
**Per-Flow Statistics**:

Flow Type:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Flow Type | Packet Count | Packet Size(bytes) | Packet Count Percentage | Packet Size Percentage |
| TCP | 11354 | 53190671 | 48.74% | 72.37% |
| UDP | 11940 | 20310119 | 51.26% | 27.63% |

Flow Duration CDFs:

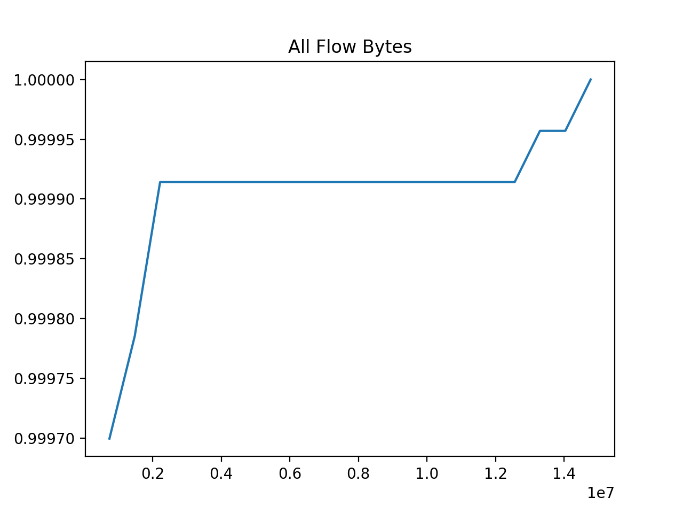
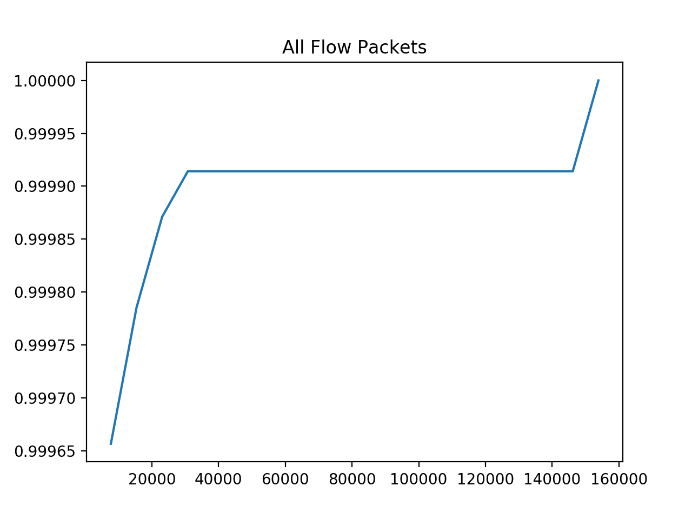
 

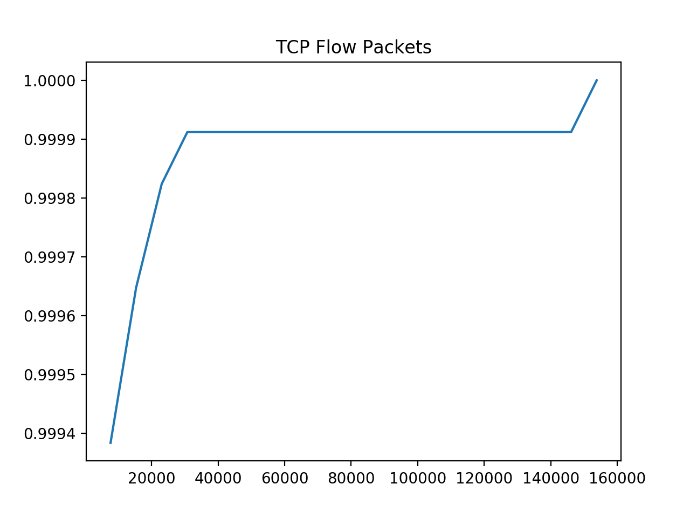
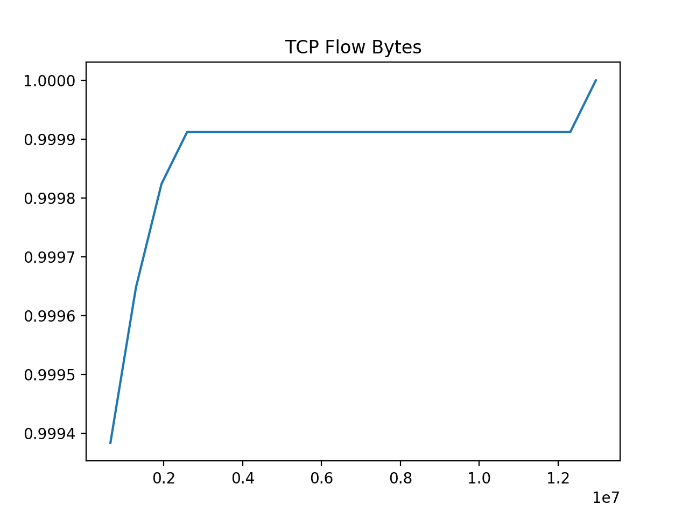


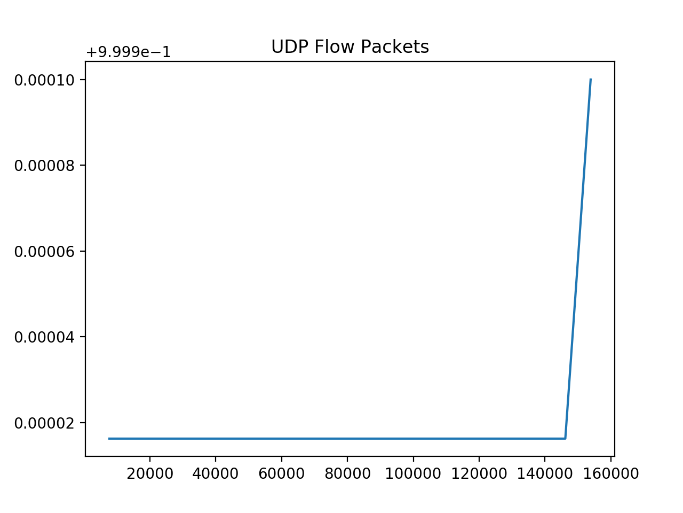
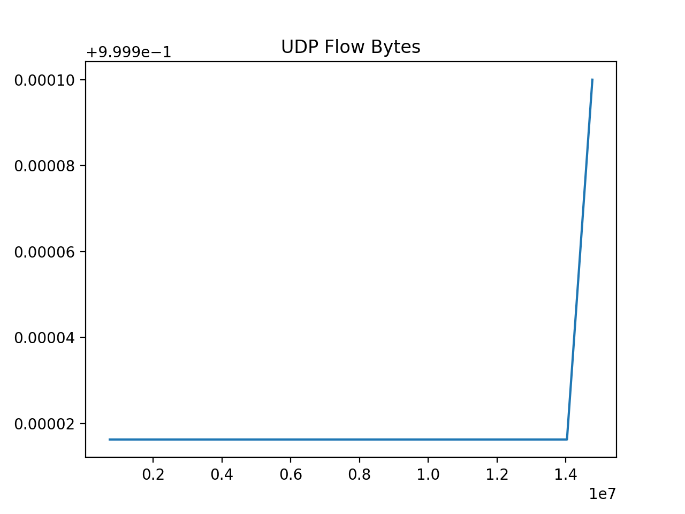
*Q: Is there any difference between TCP and UDP flows?*

There **is** a difference between the TCP and UDP flows but only for a small percentage of the data. For both protocols, most of the flows ran under 25 seconds. Past the point however, more of the TCP flows ran under 50 seconds whereas UDP flows ranged fairly even between 25 and 200+ seconds.

Flow Size:



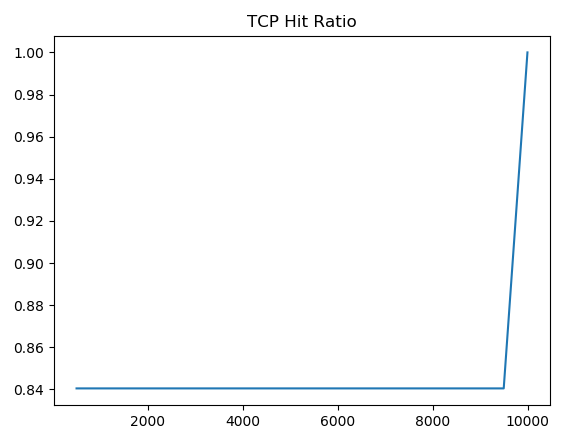
 

*Q: Do you see any difference between TCP and UDP? What about the case of using packet count vs. byte sum?*

The differences between the TCP and UDP flows were fairly large. UDP flows were mostly large flows, with a majority having number of packets larger than 140000. This difference between the two protocols held for bot packet count and byte sum.

Within each protocol however, the packet count and byte sum graphs looked fairly similar as we would expect.

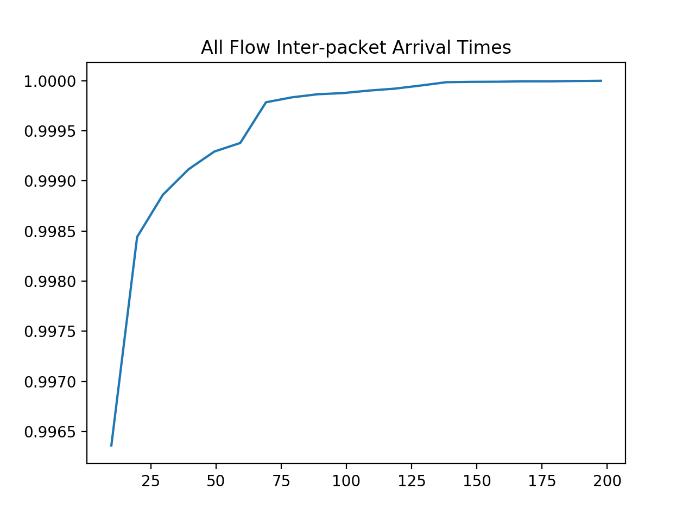
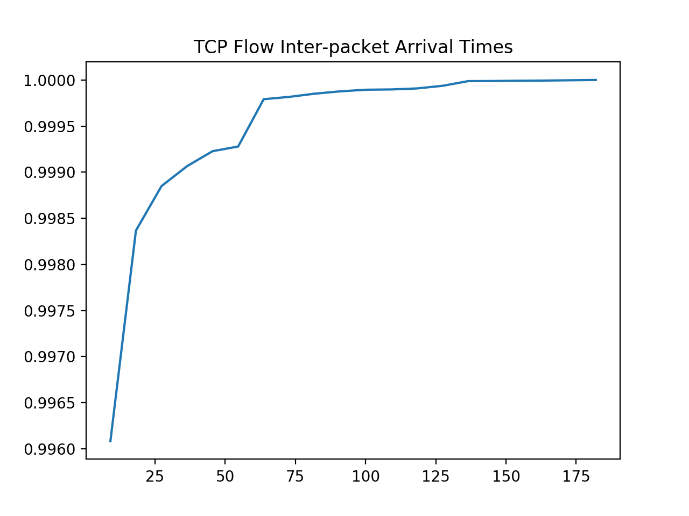
*TCP Hit Ratio:*

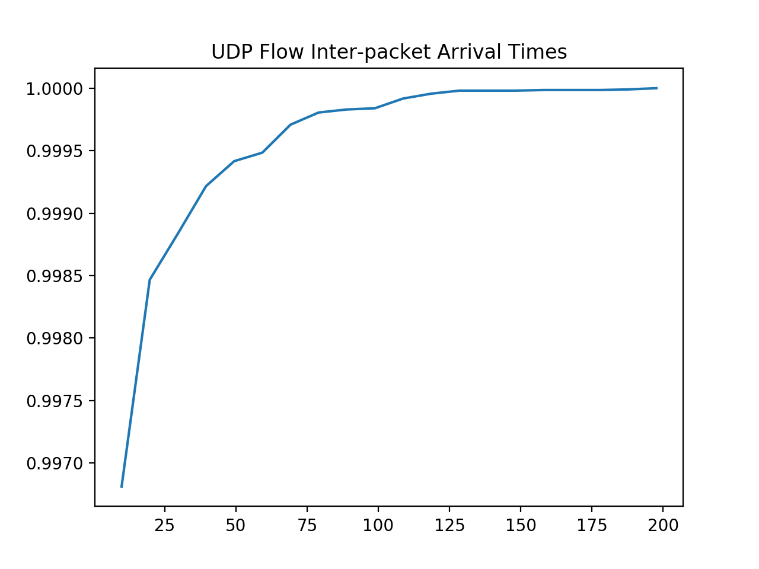


*Q: What can you say about TCP overhead base on this chart?*

It’s hard to tell from our graph but it seems that most of the hit ratios for our TCP packets are below the infinity marked ‘9999’. The packets all seem have larger TCP overheads as there is no evident sharp increase from values less than 1 to values over 1 in our graph.

Inter-arrival times:



*Qs:* *Is there any specific inter-arrival time that appears more commonly? If yes, is it present in all flows, TCP flows, or UDP flows? Do you see any difference between TCP and UDP flows?*

There is no specific inter-arrival time that appears more commonly, rather there is a most common range of times. A vast majority of the packets arrive sometime before 25 seconds according to our graph but looking at the data, most packets less than 1 second before the next. The pattern seems to extend to all the packets in our data set.

Between TCP and UDP flows, there doesn’t seem to be very big differences other than one noticeable spike in inter-arrival times between 50 and 75 seconds in TCP flows.

TCP State:

|  |  |  |
| --- | --- | --- |
| State | Total | Percentage |
| Ongoing | 729 | 6.42% |
| Reset | 3288 | 28.96% |
| Finished | 7284 | 64.15% |
| Request | 53 | 0.47% |

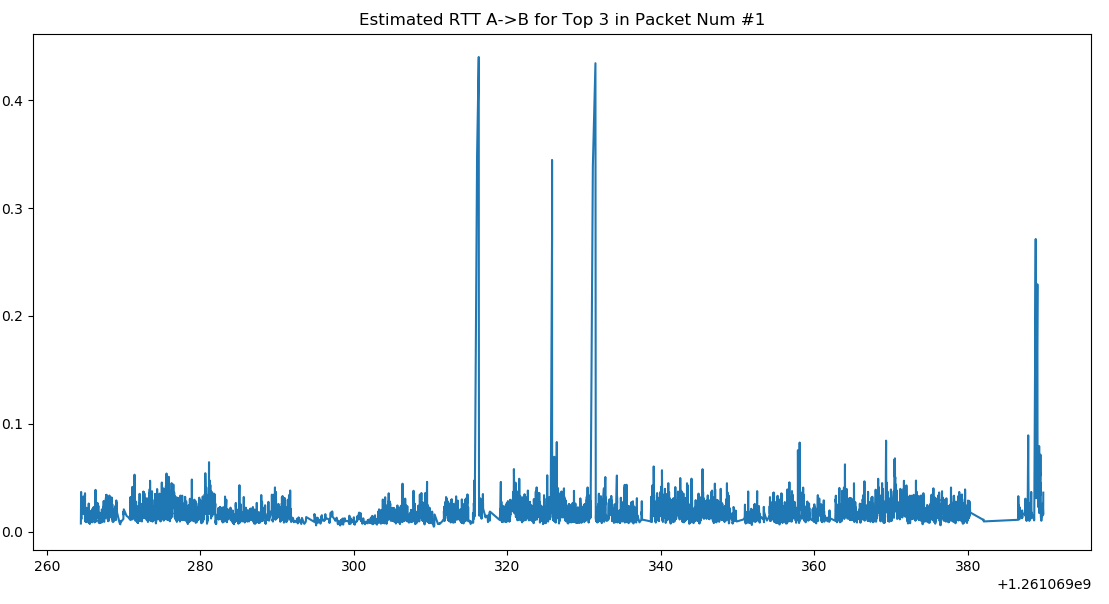
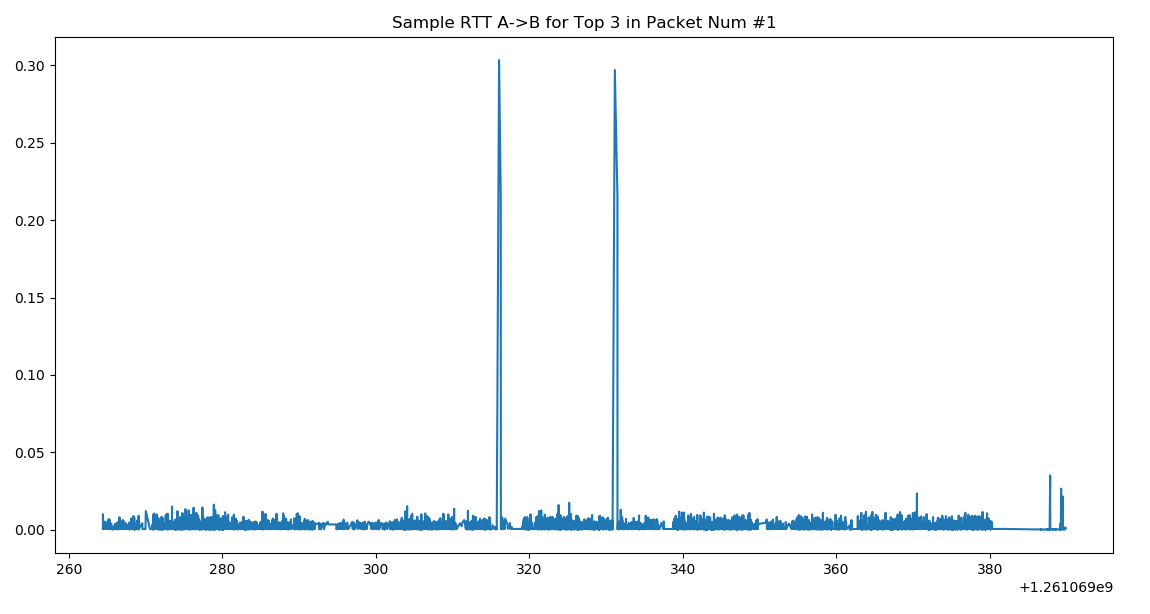
# RTT Estimation:

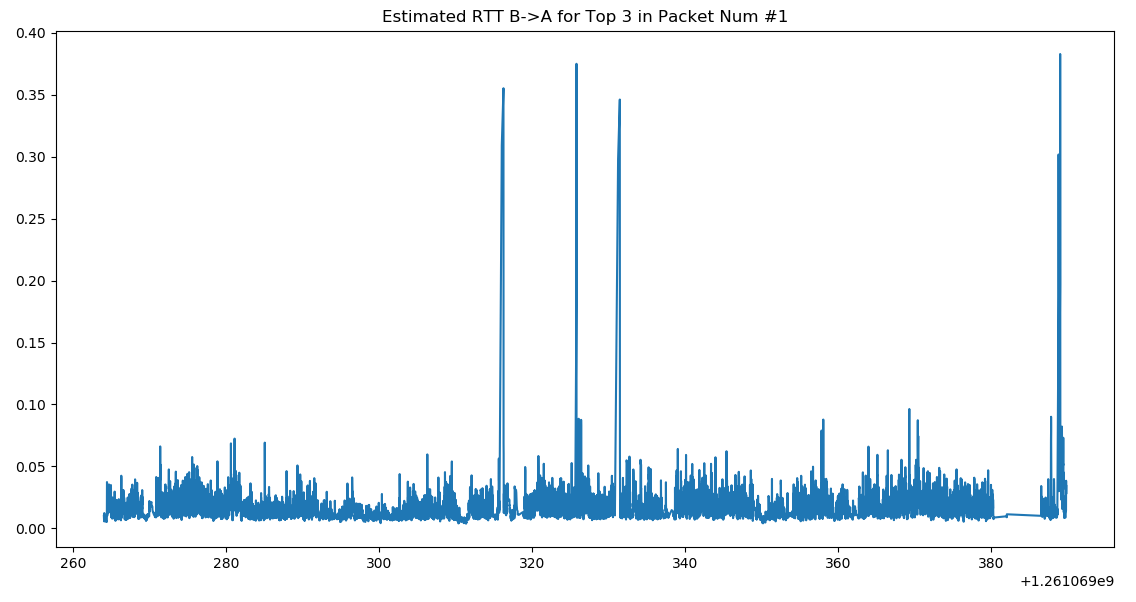
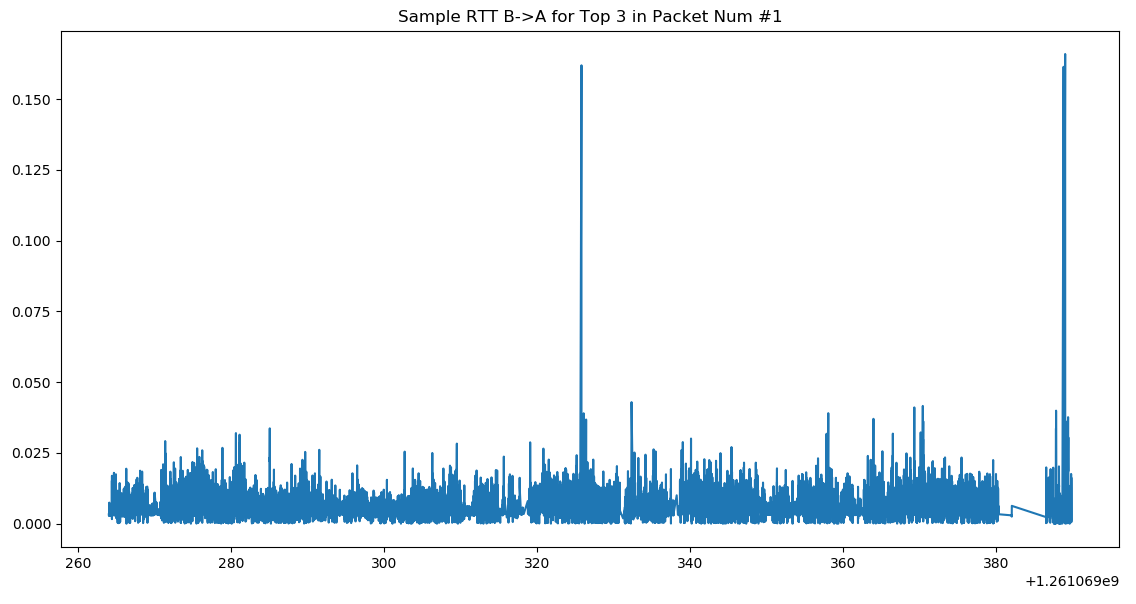
## *Implementation Notes:*

* For our RTT estimation, we saw that much of our sample RTTs fell below the 1s threshold specified in RFC6298 and so we decided to forego that requirement in order for our data to show some non-constant results.

## Top 3 TCP Flows in Terms of Packet Number:

**Flow #1 Charts:**

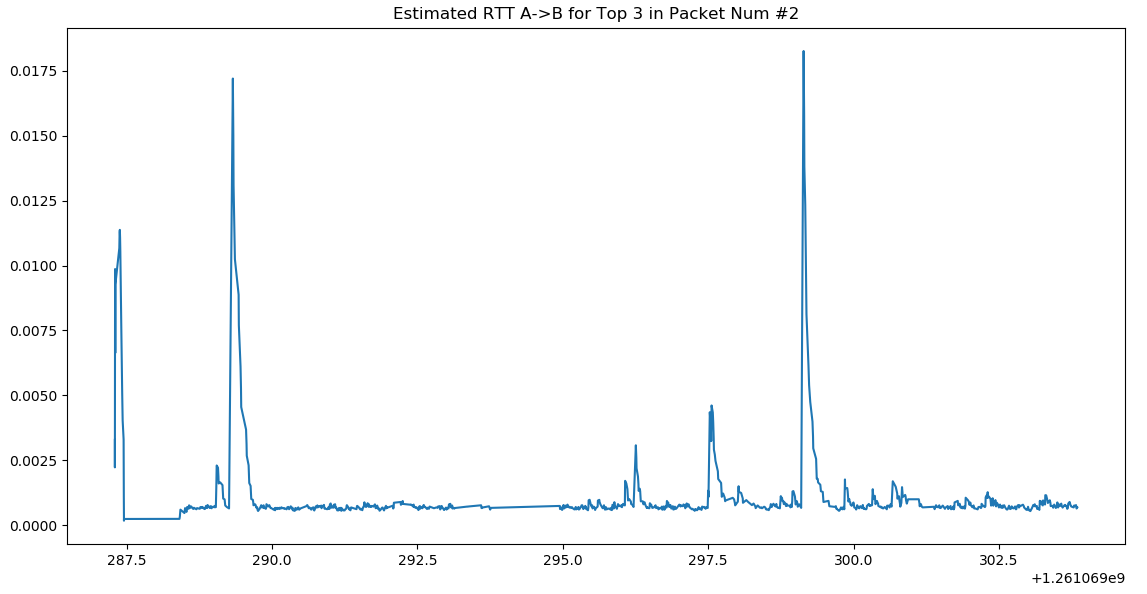
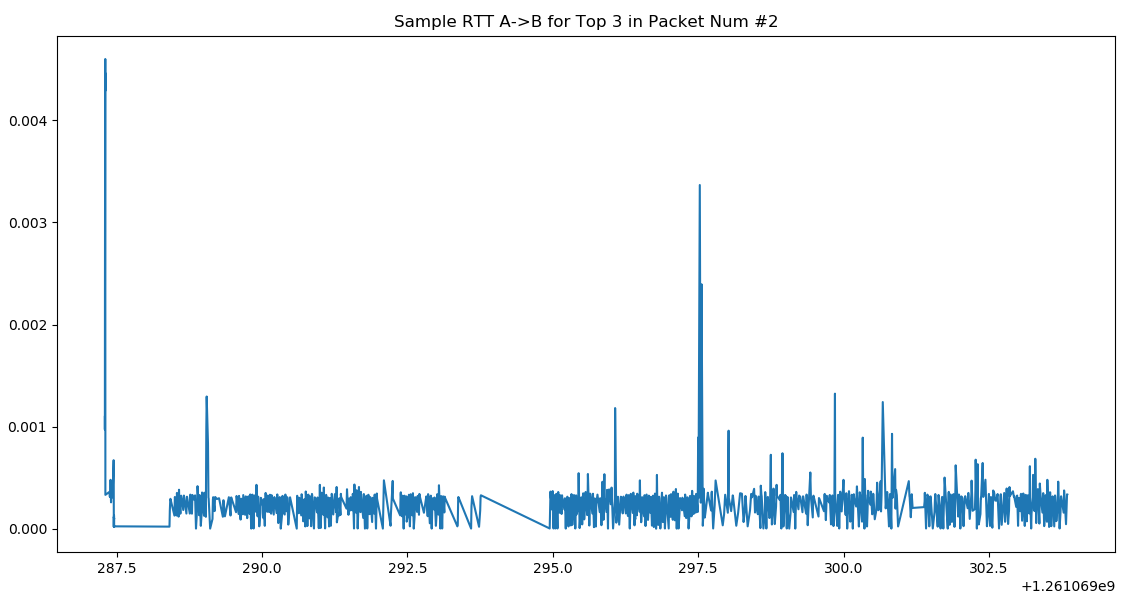


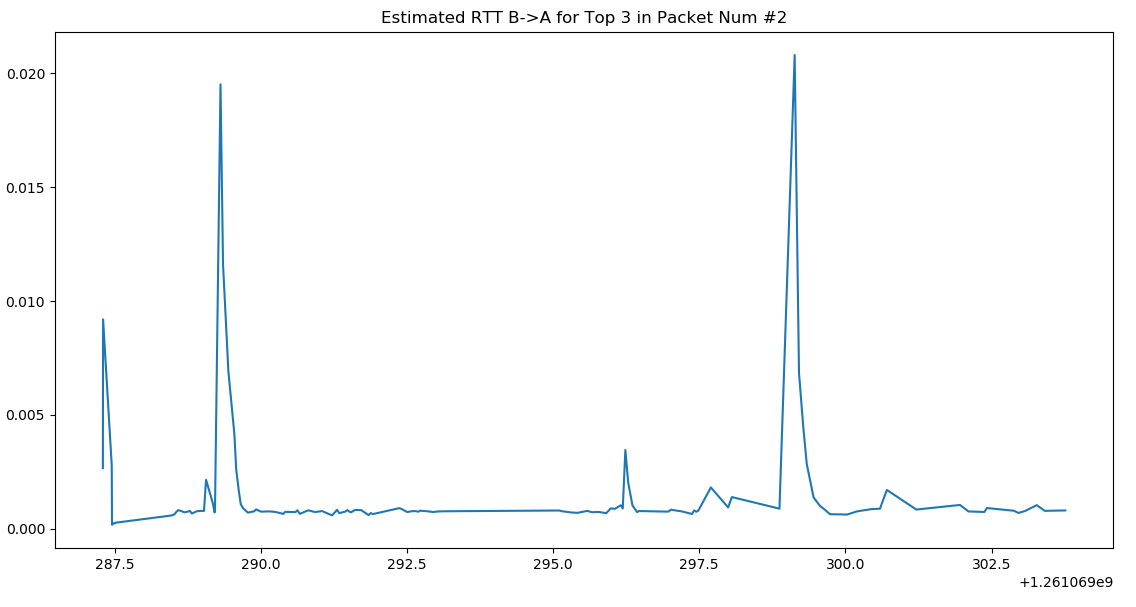
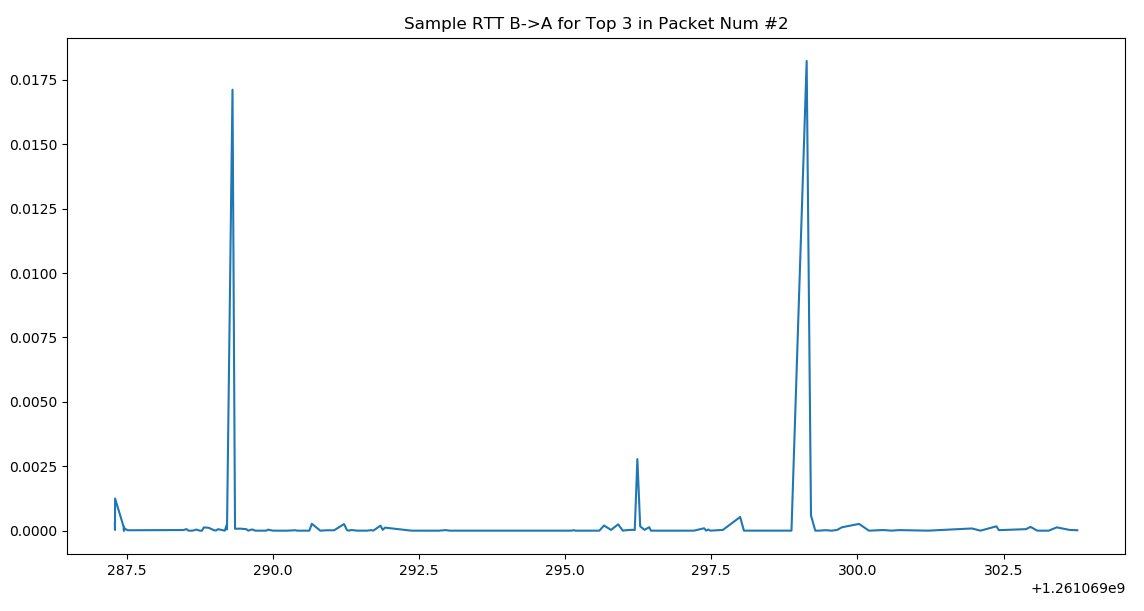


***Charts Analysis***

* In both directions of the flow, the appeared to be two to three main point of increased RTT in both the sample and estimated calculations. This would suggest some congestion at those points however given our inclusion of < 1s data, these variances could simply be small and expected variances in transfer rate
* Otherwise, the sample RTTs and estimated RTTs are relatively stable.

**Flow #2 Charts:**

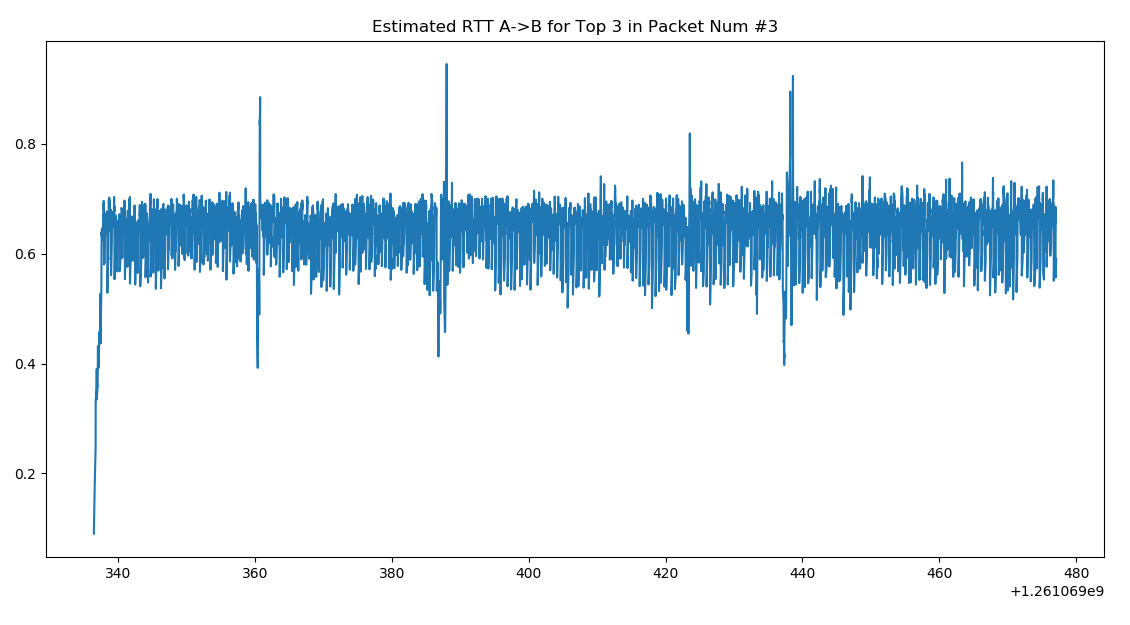
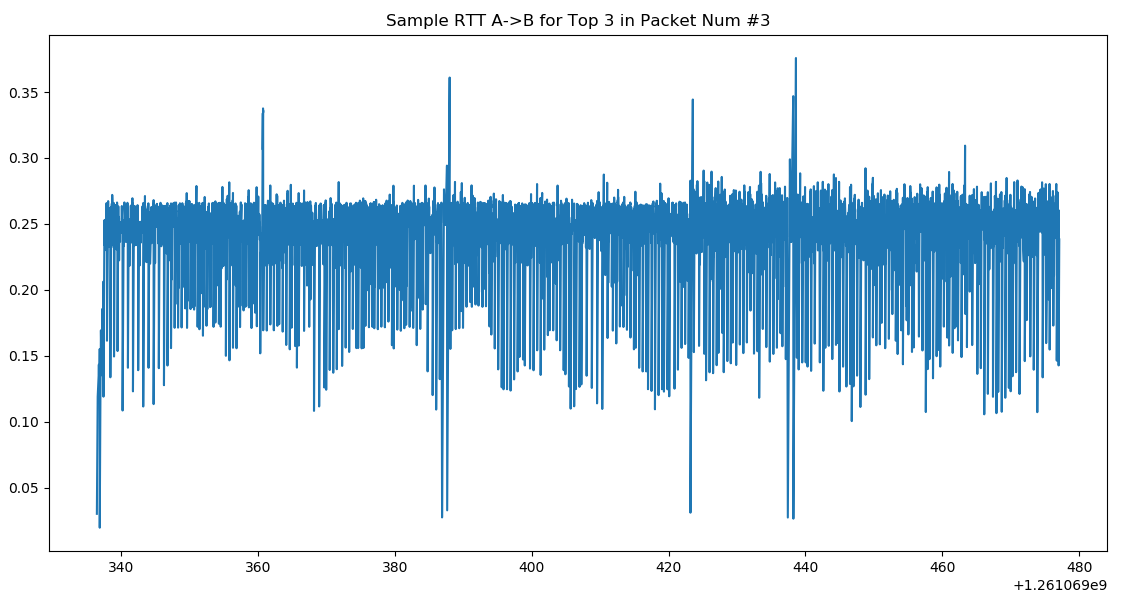


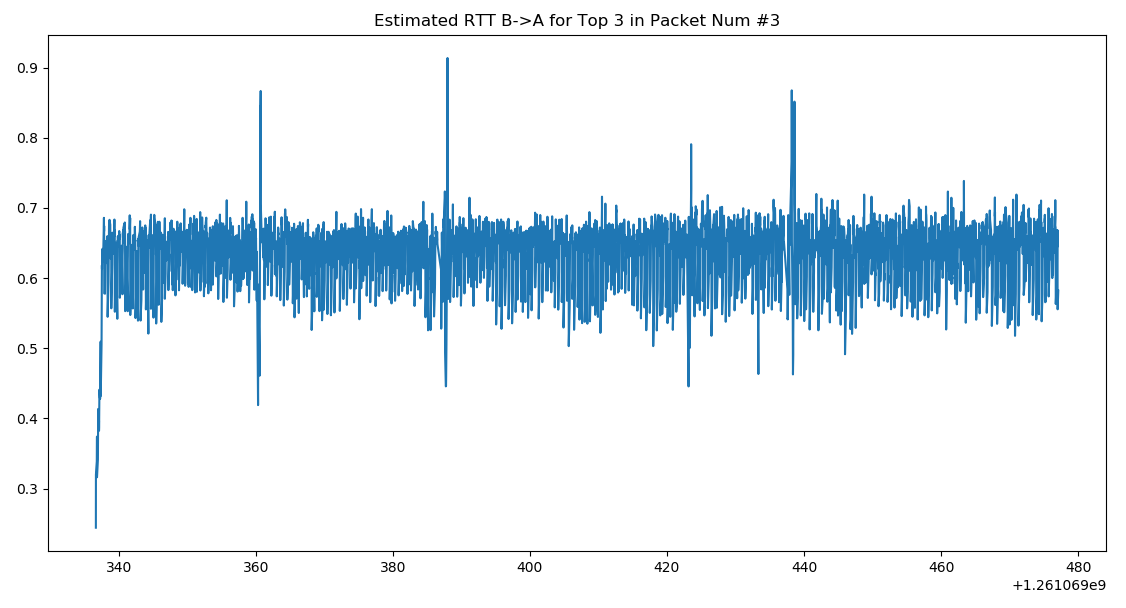
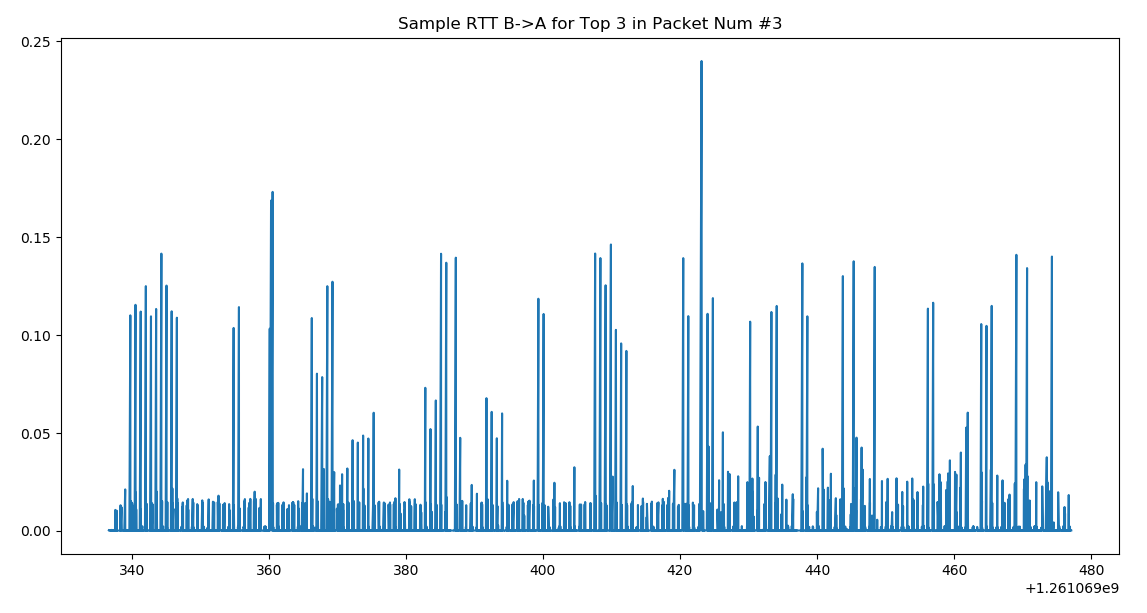


***Chart Analysis***

* This flow’s RTTs are once again fairly stable except for much clearer spikes than in flow #1
* This is likely because the actual RTT values in this flow are much lower compared to the former’s values. Again, this data could be attributed to normal variances in transfer rates or possibly some minor congestion

**Flow #3 Charts:**





***Chart Analysis:***

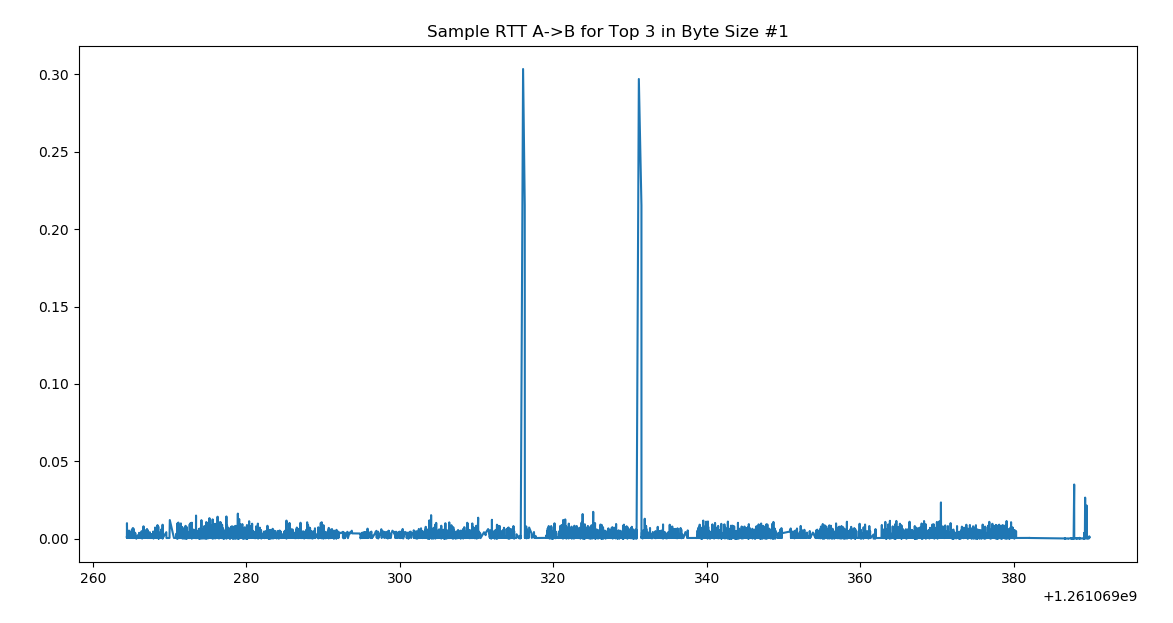
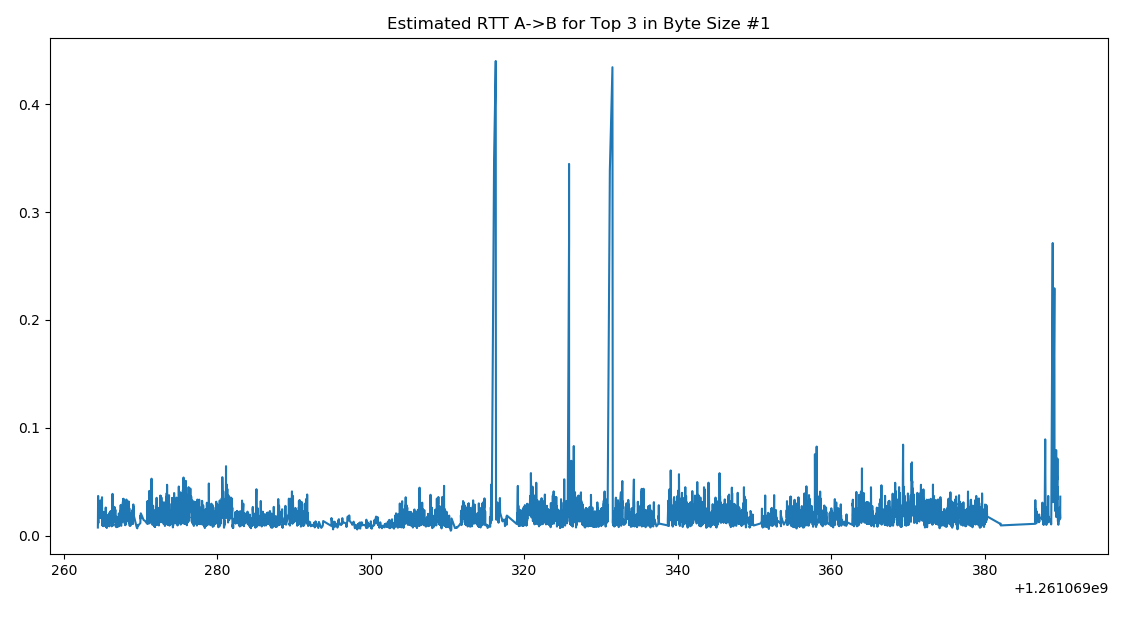
* This flow’s RTT values are much less stable than the others and also reach higher values.
* The A->B charts compared to B->A suggest that the RTT from A->B is generally slower than from B->A.
* And as previously, the more frequent spikes here can be indications of congestion or other transfer slow down issues.

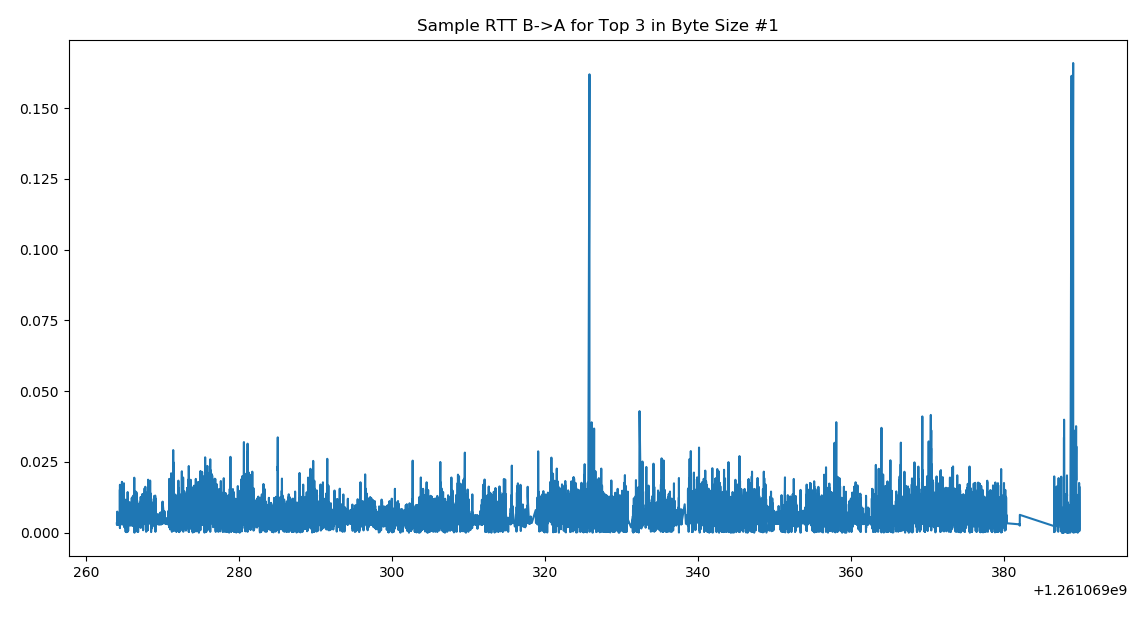
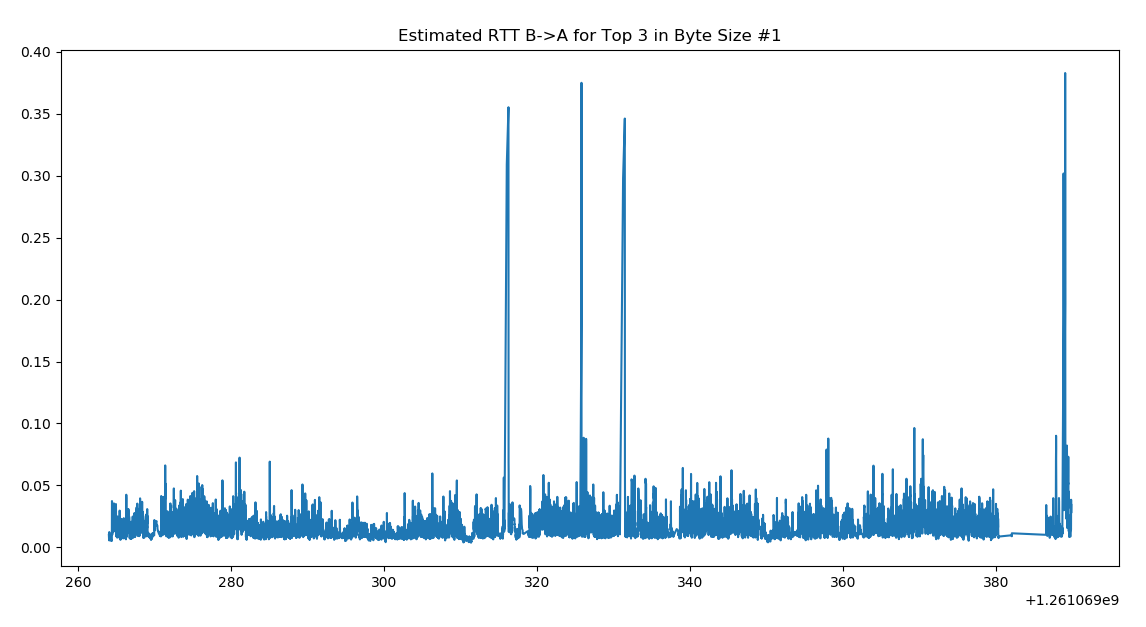
## Top 3 TCP Flows in Terms of Byte Size:

**Top Byte Size Flow Observations:**

* All three of the flows from the packet number ranking were the same and in the same order when it came to byte size. This is maybe not completely unexpected and we can see from this that there was no flows in our dataset with comparably larger packets.

**Flow #1 Charts:**

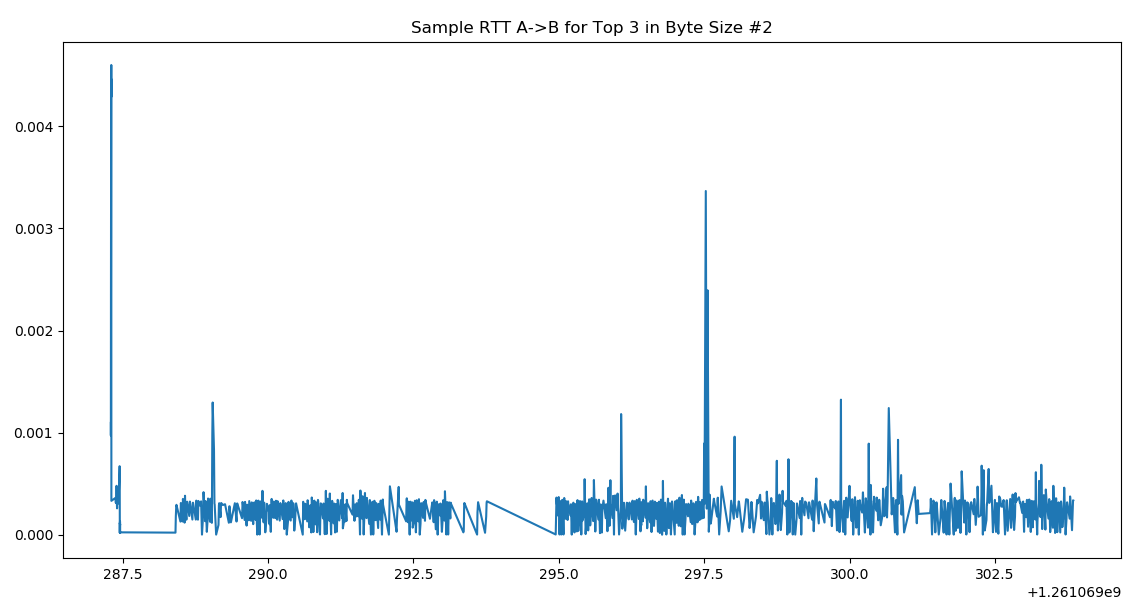
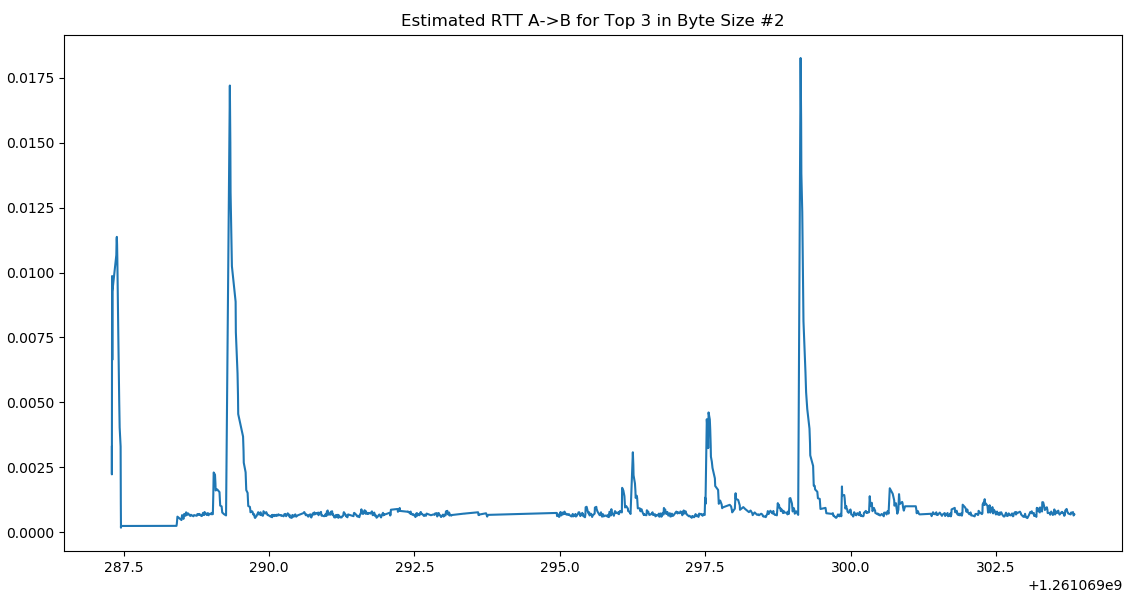
 

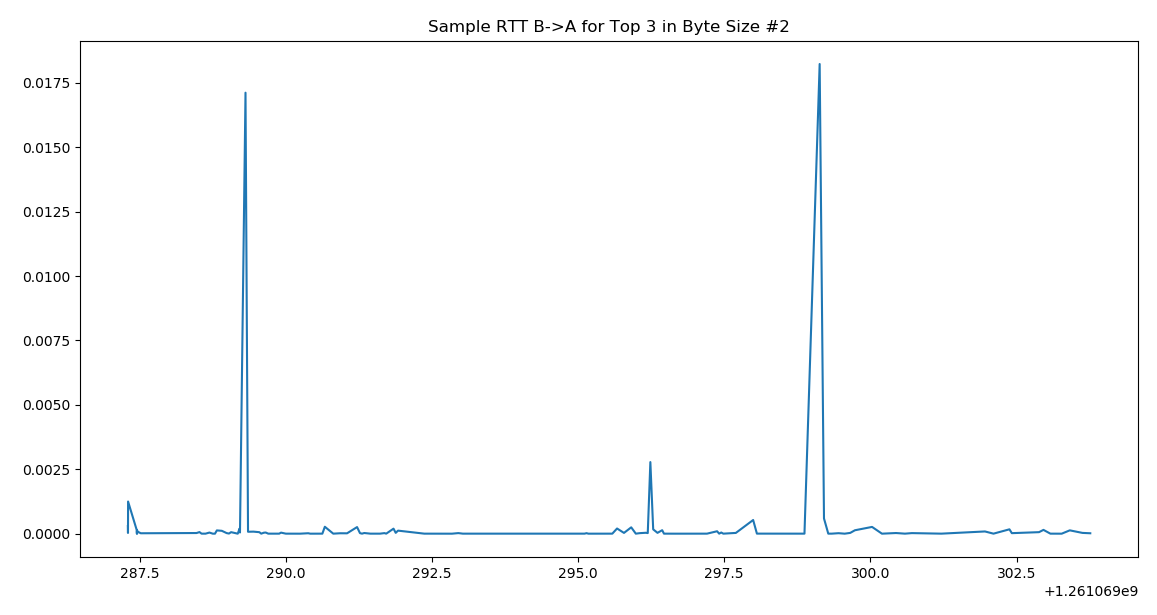
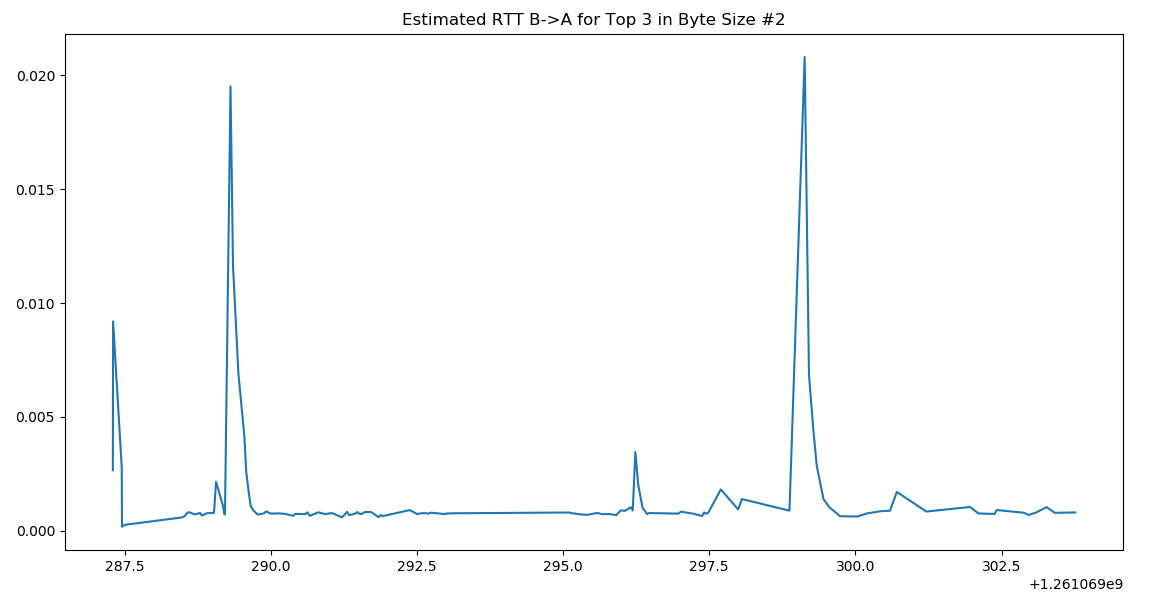
 

***Chart Analysis:***

* This flow is the same as #1 in “Top 3 in Term of Packet number” and as such the analysis is the same.

**Flow #2 Charts:**

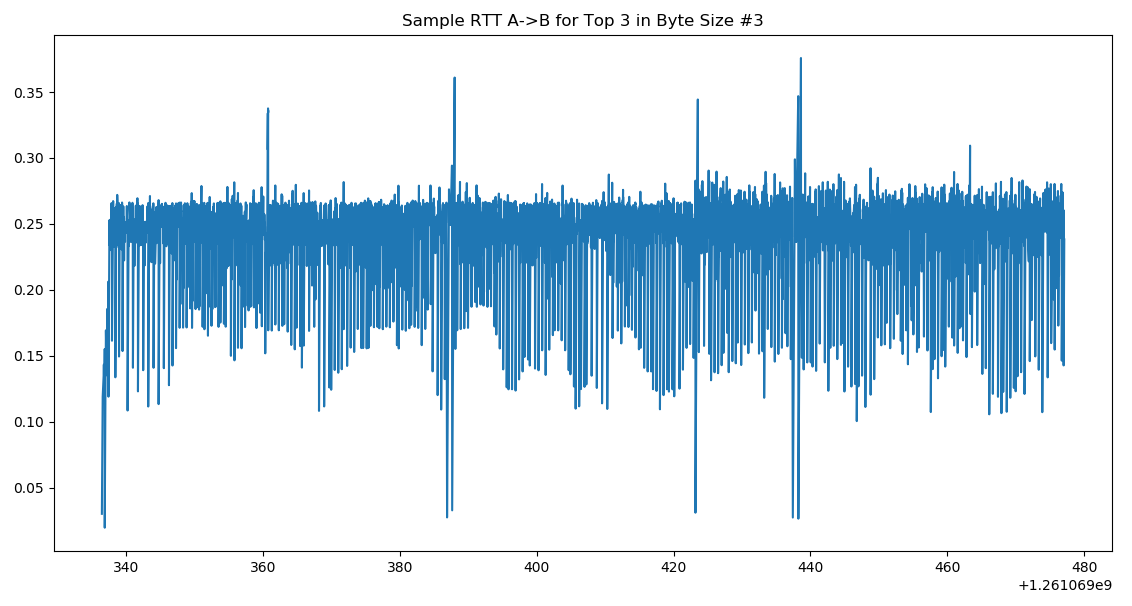
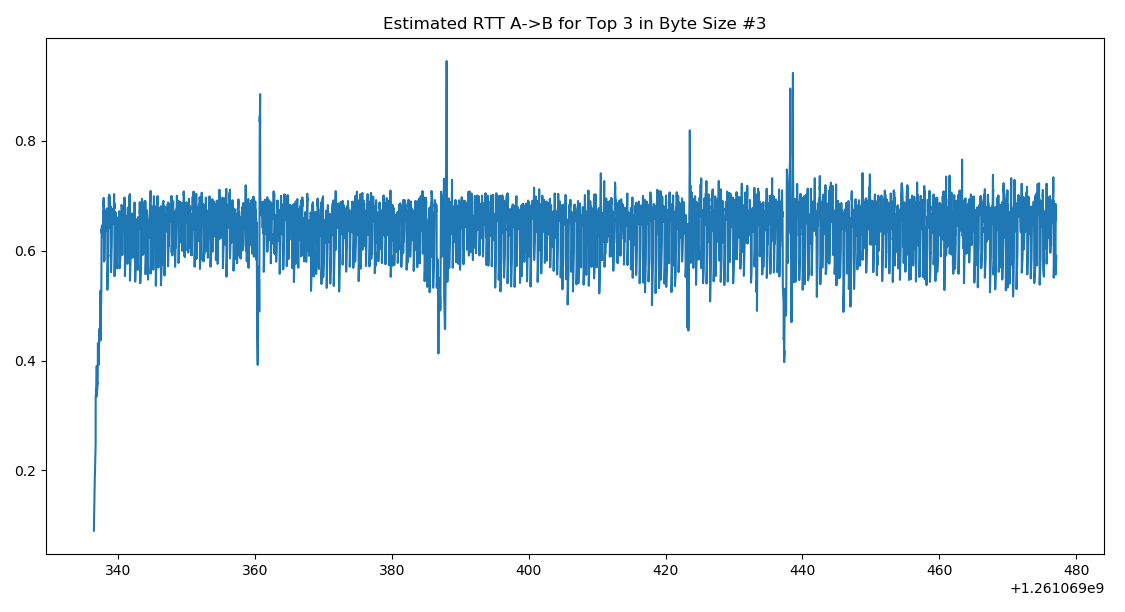
 

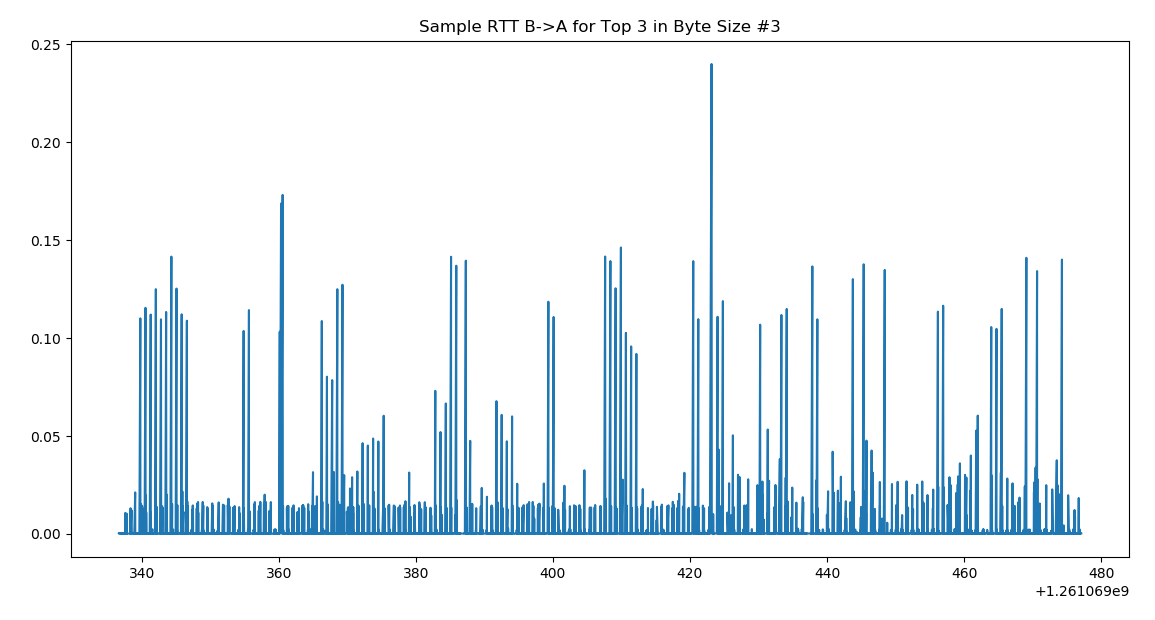
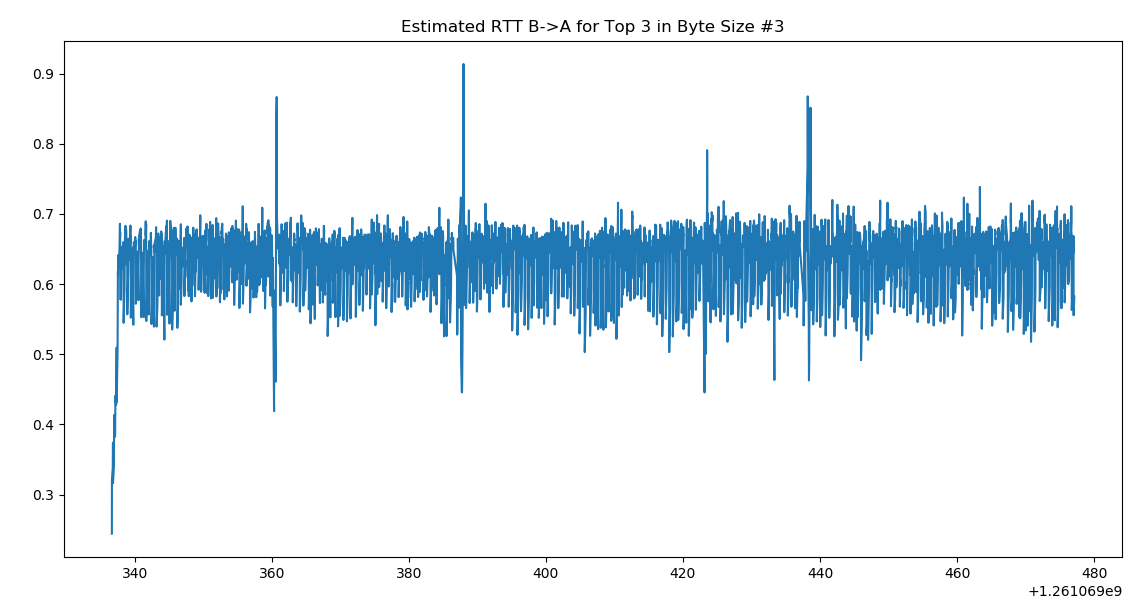
 

***Chart Analysis:***

* This flow is the same as #2 in “Top 3 in Term of Packet number” and as such the analysis is the same.

**Flow #3 Charts:**

***Chart Analysis:***

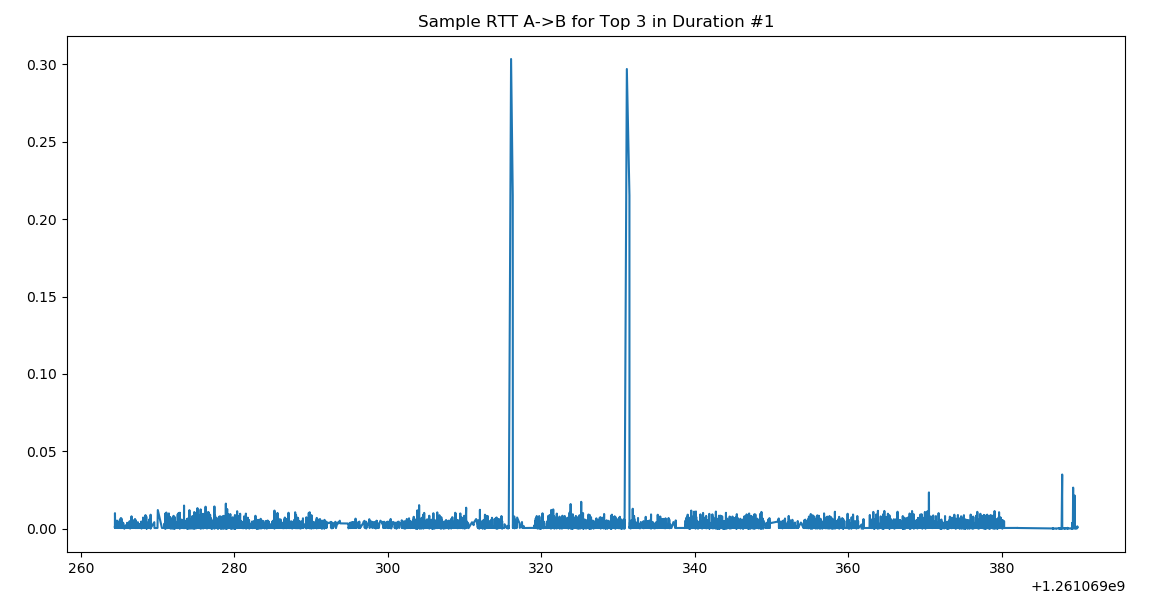
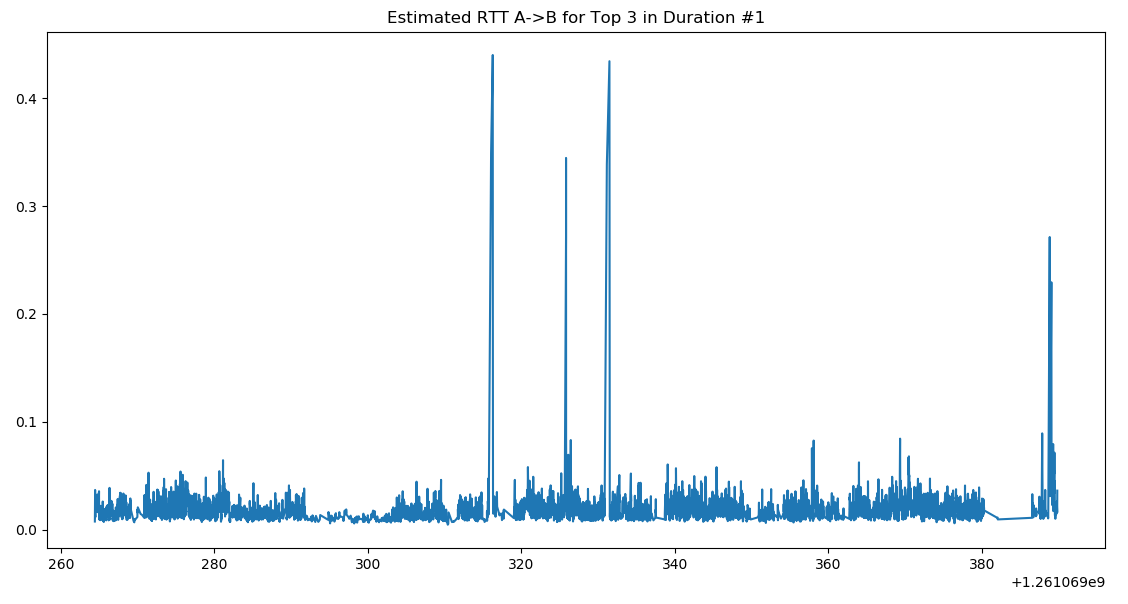
* This flow is the same as #3 in “Top 3 in Term of Packet number” and as such the analysis is the same.

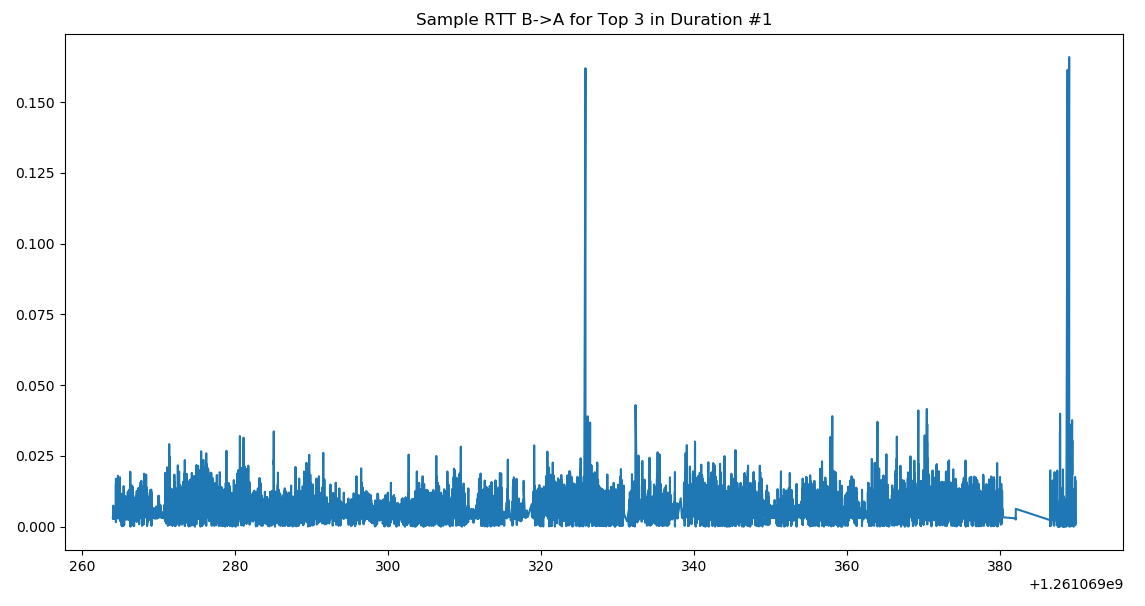
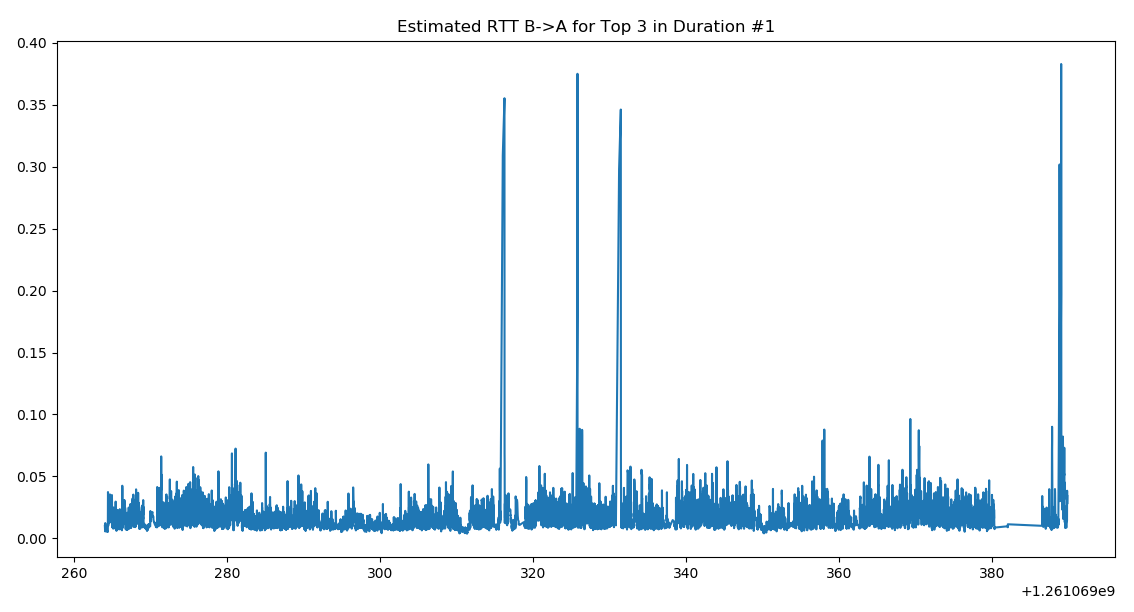
## Top 3 TCP Flows in Terms of Duration:

**Top Duration Flow Observations:**

* The top duration flows turned out the same way as both the top packet number and top byte size flows, and such the analysis for these ones will be, once again, the same.
* We can also observe from this that the top flows in our data have a correlation between all three attributes: packet number, byte size, and duration.

**Flow #1 Charts:**

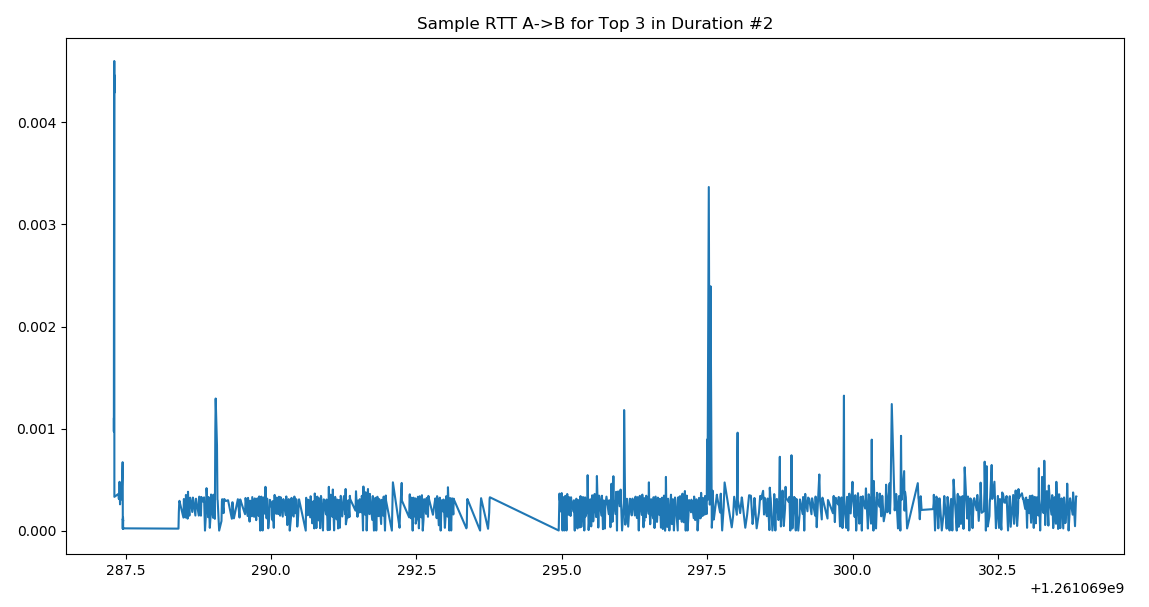
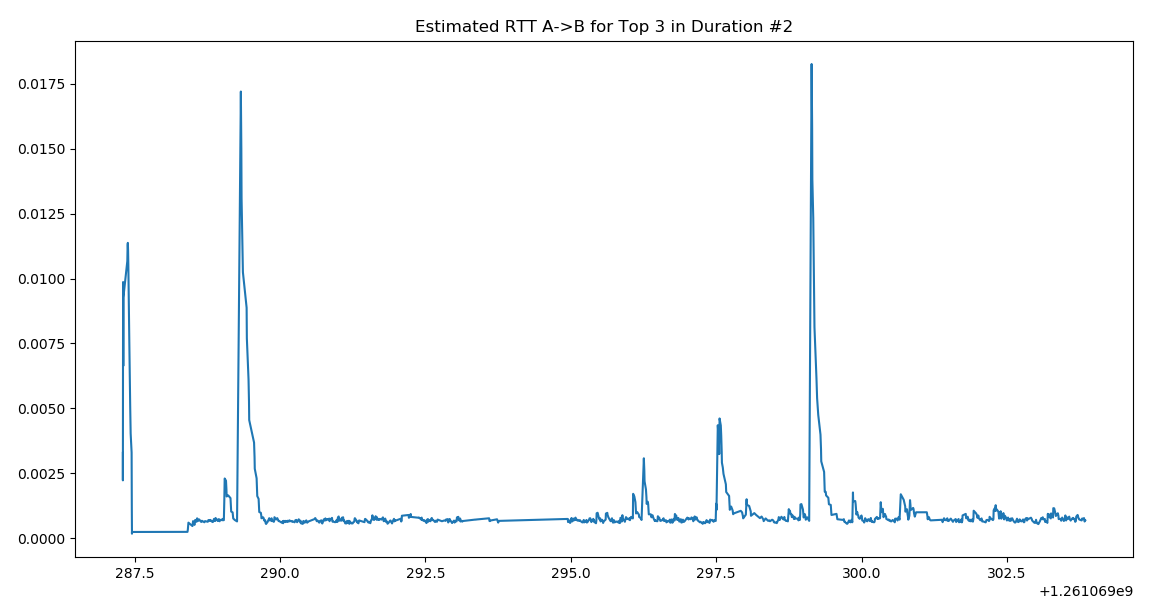
 

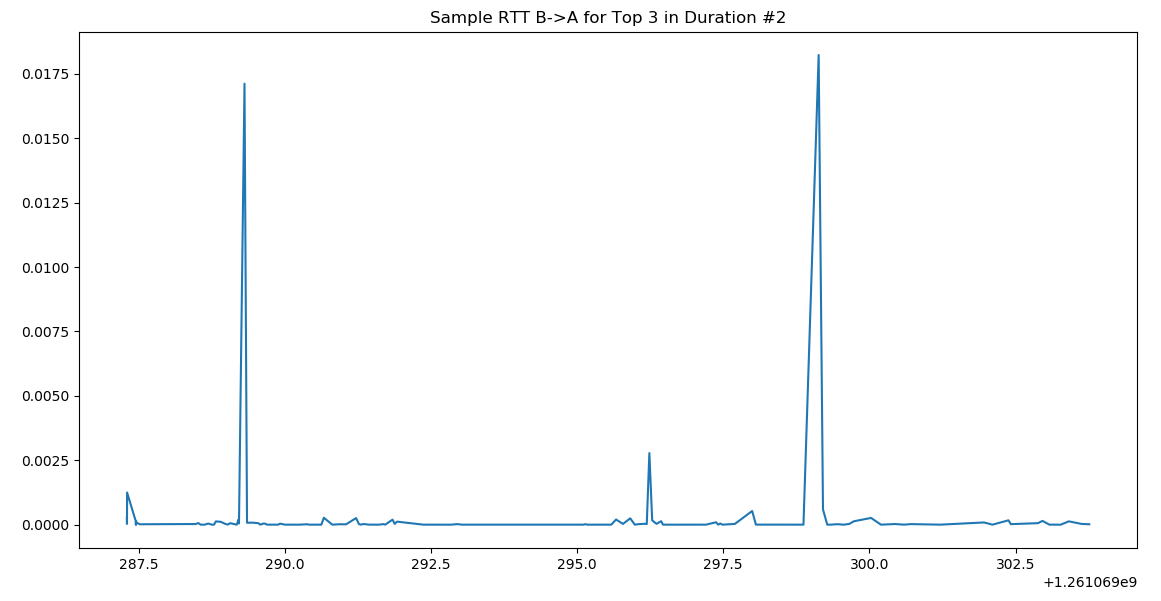
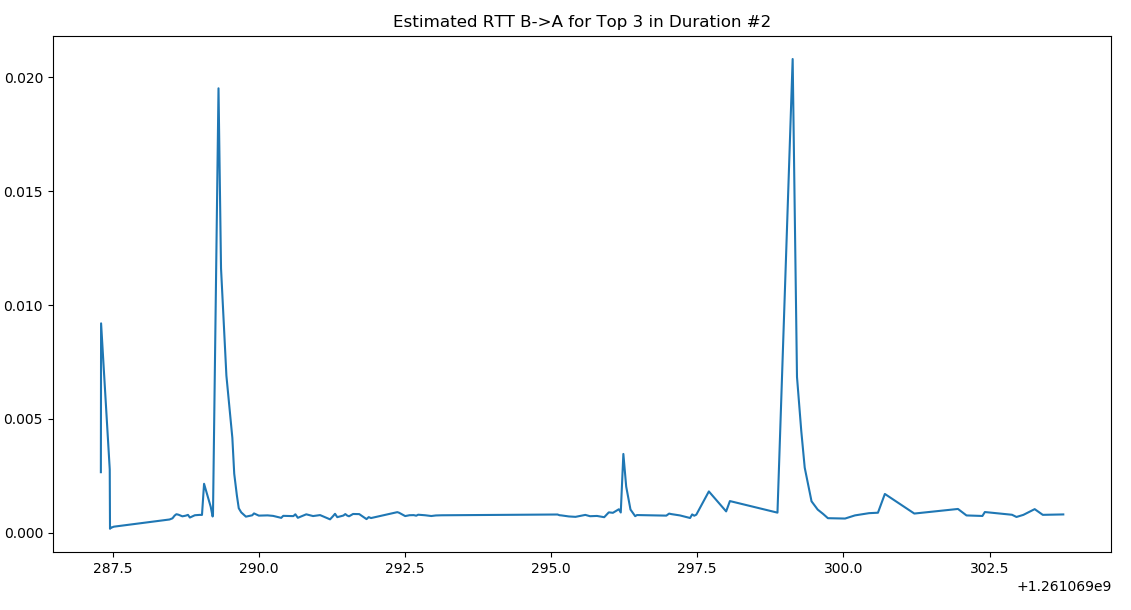
 

***Chart Analysis:***

* This flow is the same as #1 in “Top 3 in Term of Packet number” and as such the analysis is the same.

**Flow #2 Charts:**

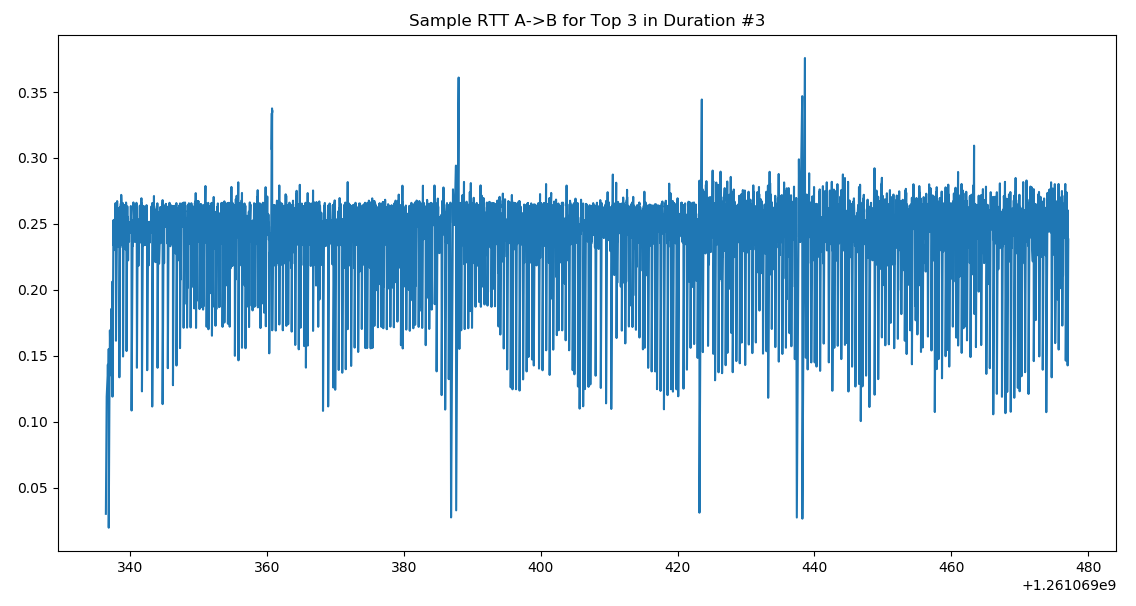
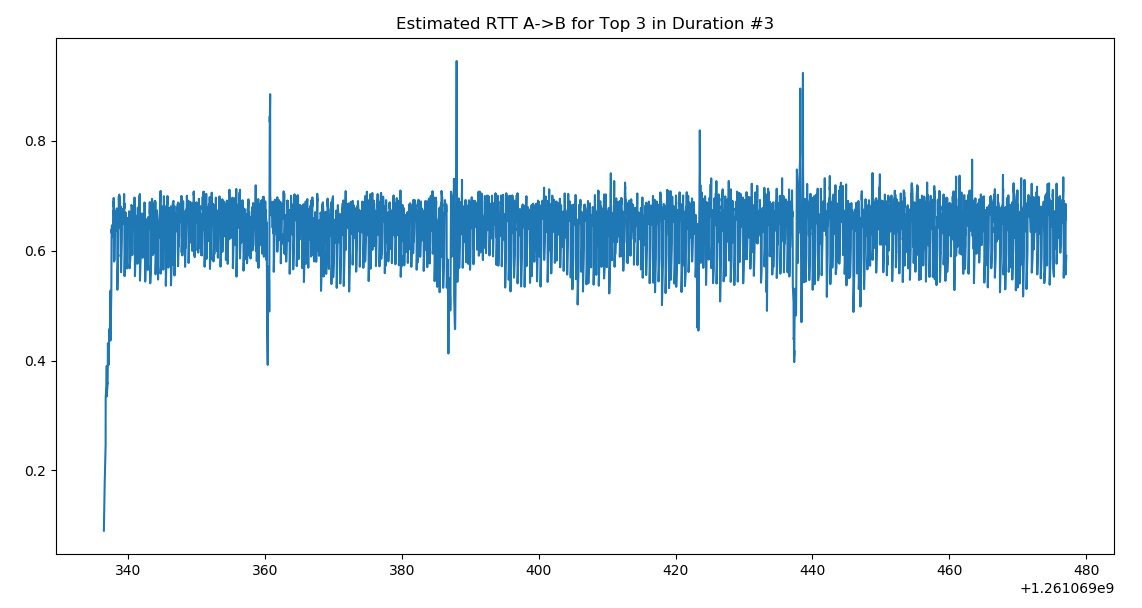
 

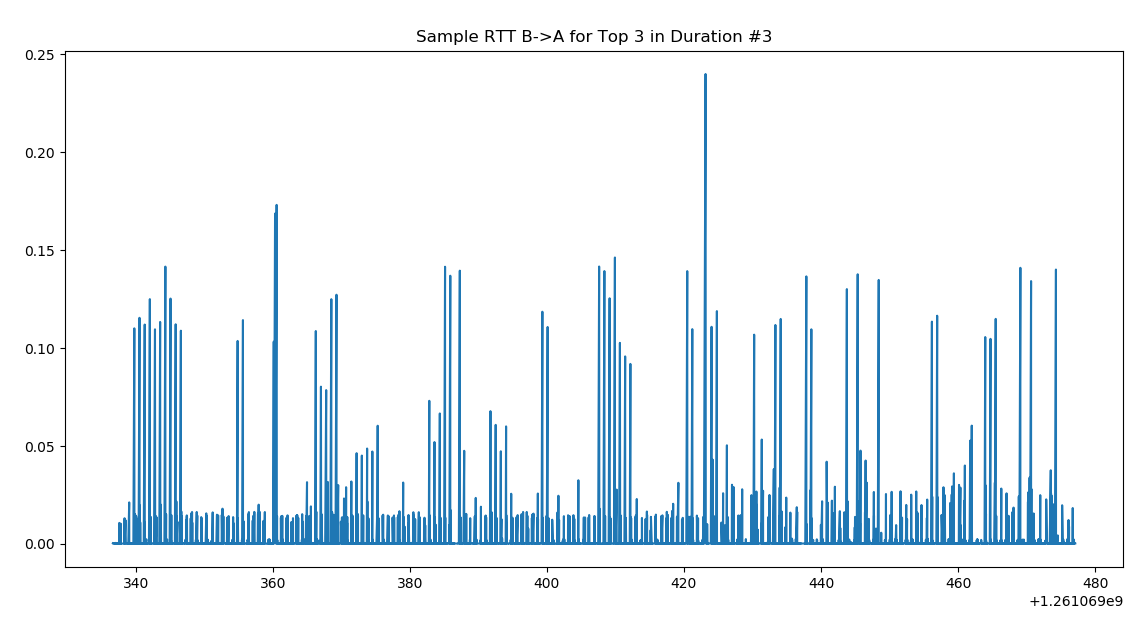
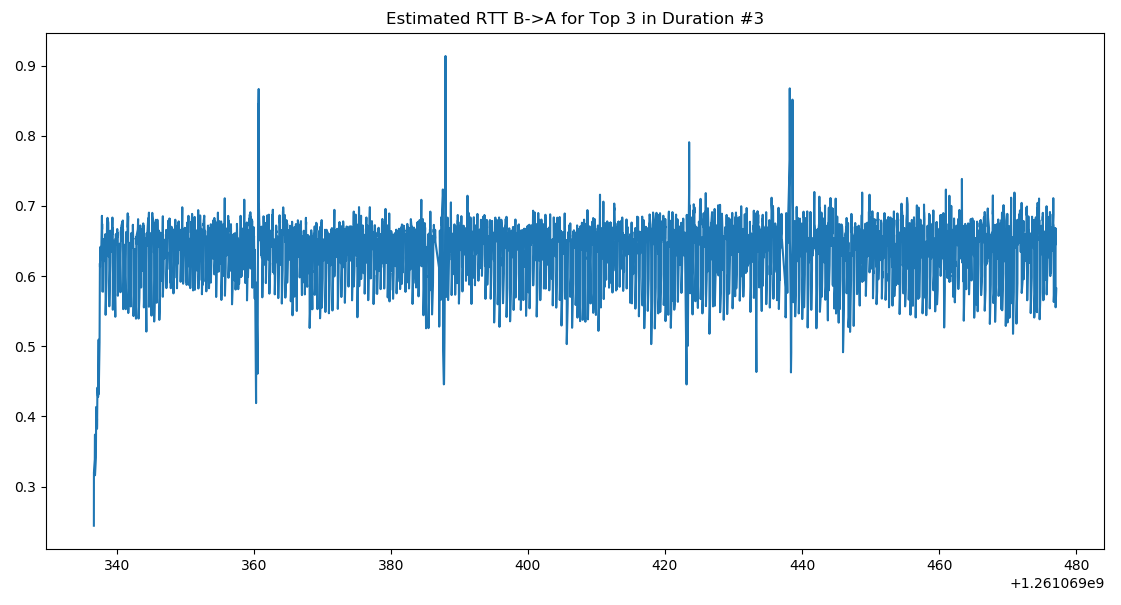
 

***Chart Analysis:***

* This flow is the same as #2 in “Top 3 in Term of Packet number” and as such the analysis is the same.

**Flow #3 Charts:**

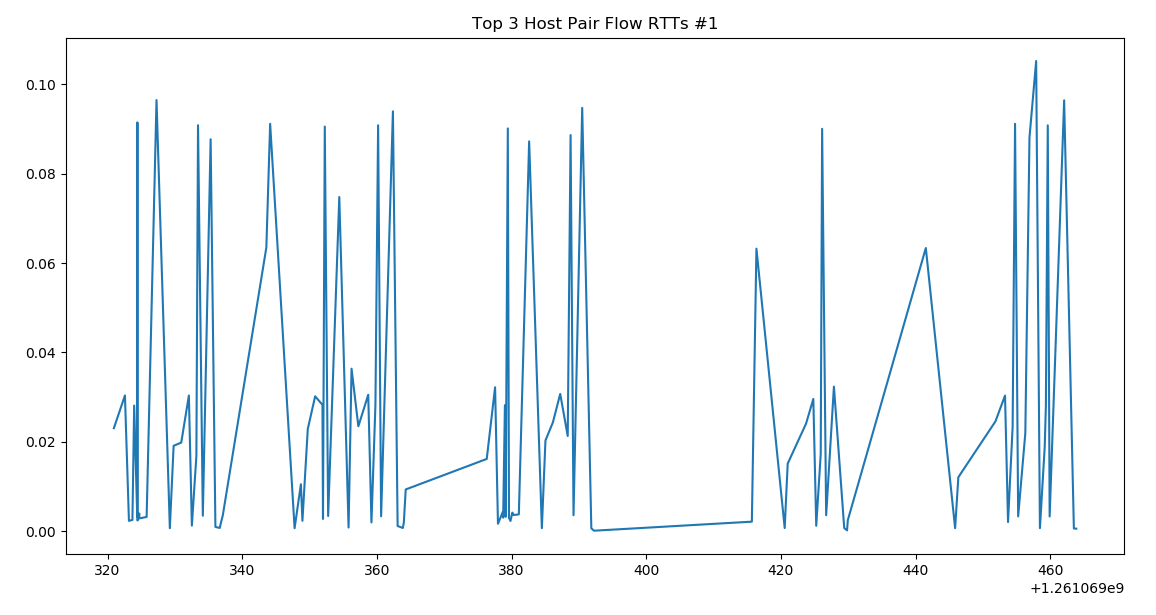
 

***Chart Analysis:***

* This flow is the same as #3 in “Top 3 in Term of Packet number” and as such the analysis is the same.

## Top 3 Host Pairs with most TCP Connections:

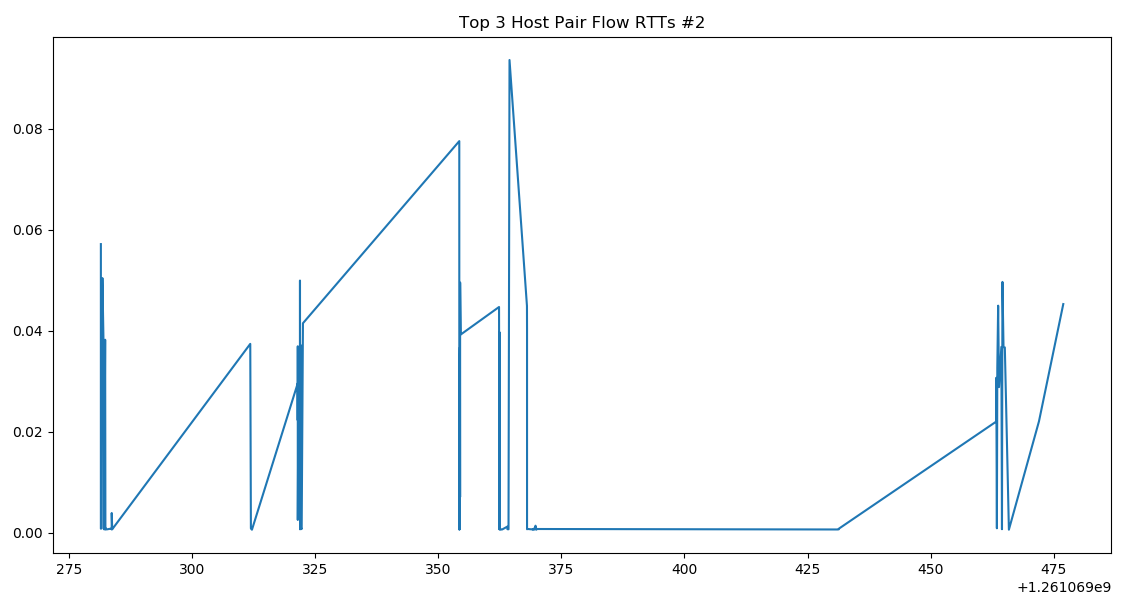
**Host Pair #1:**



***Analysis:***

* For this host-pair, the flows seems to occur fairly frequently and are in a relatively stable pattern (very stable lows and highs) in terms of RTT, meaning that the flows between the hosts operate relatively similarly throughout the dataset.
* This seems to suggest that the connection between the two hosts operates in a relatively stable manner at worst and at best.

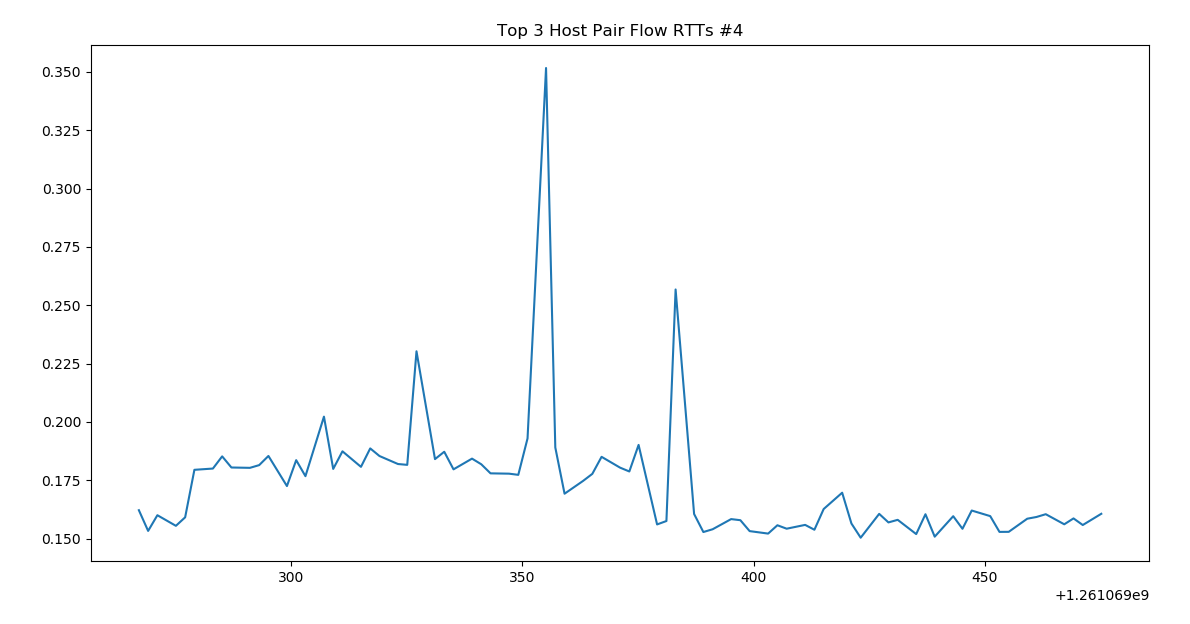
**Host Pair #2:**



***Analysis:***

* This host pair seems to be fairly infrequently used in terms of time blocks but given that the number of flows is second to most, it would seem that either the density of the flow are high at certain time blocks or that many of the connections never returned acknowledgments.
* This second case could be possible as our charts ignored flows that did not have any map-able acknowledgements.

**Host Pair #4:**



***Analysis:***

* First thing to note is that this is the host-pair with the 4th most TCP flows associated with it. This was the case because, as mentioned above, we ignored cases in which we could not map acknowledgments to packets and **all flows** for the third most host-pair had all packets in which they could not be mapped.
* This host-pair exhibits more evenly distributed time block of connections than the other two.
* As well, this host-pair does not have the same consistency in terms of RTT than the other two in that there are very clear spikes in RTT here.
* This could mean that this connection is more susceptible to congestion than the other two.

# Source Attribution

We used a number of tools and example code to complete our project and this section will be used to cite these sources:

1. Wireshark – A PCAP file analyzer that we used to verify our findings
2. DPKT Python package – this was the main contributor to our project with libraries allowing us to analyze the PCAP formatted file
   1. Additionally, we followed examples and subsequently took code bits from DPKT related sources like: dpkt.readthedocs.io (noted in our source code)
3. Python CDF plotting – also noted in our source code is our usage of code that allowed us to plot CDF plots using matlabplot: <http://stanford.edu/~raejoon/blog/2017/05/16/python-recipes-for-cdfs.html>