Fully Automatical Measuring and Plotting Tool of Estimated RTT & Time Interval

Zhihao Jin

Department of Computer Science, Truman State University

CS 470: Network

Dr. Chetan Jaiswal

April 1st, 2022

Introduction

This program is a fully automated tool coded in Python that is utilized to measure the Estimated Round Trip Time(Estimated RTT) and Timer Interval of specific IP addresses for 500 seconds on the Windows operating system. In order to fulfill the purpose of plotting graphs, the “matplotlib” library has been used to plot based on the corresponding data. Thus, the “matplotlib“ is necessary to be installed before running this tool.

The image guidance for using this tool on the Windows command line is shown below: The command line format is: $python EstRTT_and_TimeInter_Expe.py (IP address)

```
C:\Users\78302\Desktop\Project-2 CS-470>python EstRTT_and_TimeInter_Expe.py 166.111.4.100
```

Design:

The whole coding of this tool is composed of three major components: calculation part, command running part, and plotting graphs part.

Calculation part:

This segment of code is responsible for calculating the estimated RTT, deviation RTT, and time interval based on the previous two RTTs. What should be noticed is estimated RTT and deviation RTT calculation both are involved in computing the from the previous result, or recursion-like. The way that this program adopt is by using a loop and list to store and get the last element after each calculation. And all functions are perfectly dedicated to working on their own jobs as expected.

This part contains:

- def estimate_RTT_calculator(sampleRTT):
- def deviated_RTT_calculator(estimateRTTList, sampleRTTList):
- def timeInterval_calculator(estimateRTTList, devRTTLIst):

Command line part:

This segment of the code is designed to take the user's command-line arguments and pass them back to the command line then run it. After that, it initiates a for loop to keep sending the ping command 100 times, each for 5 seconds, 500 seconds in total. One of the possible circumstances that should be considered is that packets sent by the ping command may get lost. And in order to handle the aforementioned case, this tool has been configured to have an infinite loop that can keep sending the ping command until it receives the desired returned information or packets.

This part contains:

- def get_simple_cmd_output(cmd, stderr=STDOUT):
- def get_ping_time(host):
- def main():

Plotting part:

This segment is simply to plot the figure based on all the collected information. Neither a complex algorithm is involved nor tricky cases are expected.

This part contains:

- def plot_graph(sampleRTTList, estimateRTTList, timeIntervalList, host_address):

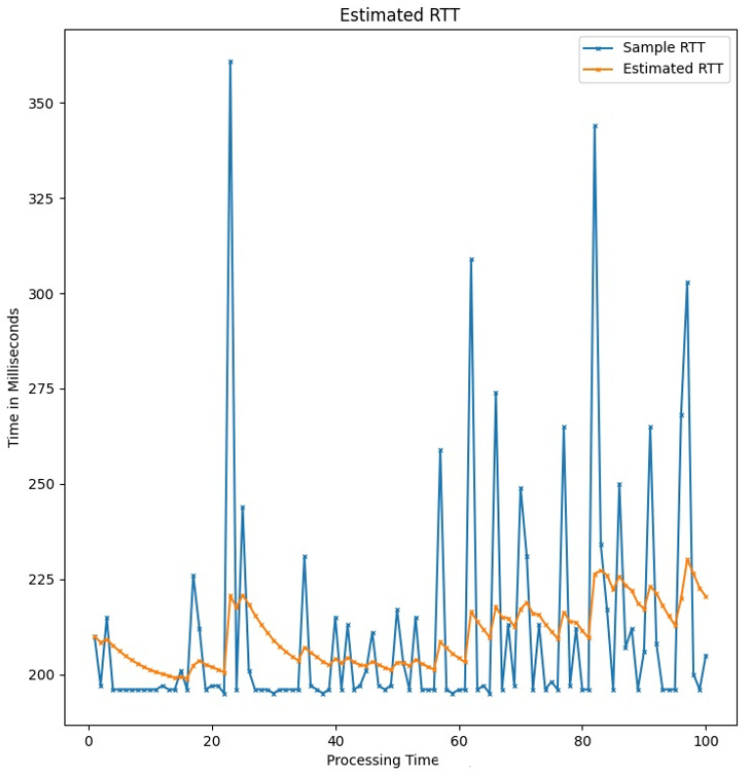
Following IP are the sample IP addresses that were used in this assignment. All of them have been carefully chosen from four different continents for testing purposes.

- Peking University
    - IP address: 166.111.4.100 (AS)
- University of Cambridge
    - IP address: 128.232.132.8 (EU)
- University of Sydney:
    - IP address: 184.51.3.161 (AU)
- Universidade de São Paulo:
    - IP address: 200.144.248.54 (SA)

Observation

As we can observe from all four of the screenshots of the plotting graphics, sample RTT has some bursty peaks during 500 seconds transmission, which is approximately 3 or 4 times larger than average RTT. During most of the testing period, sample RTT is steady and the amplitude is kept within a certain range based on different IP addresses.

Below is the sample RTT(in blue) of Peking University:

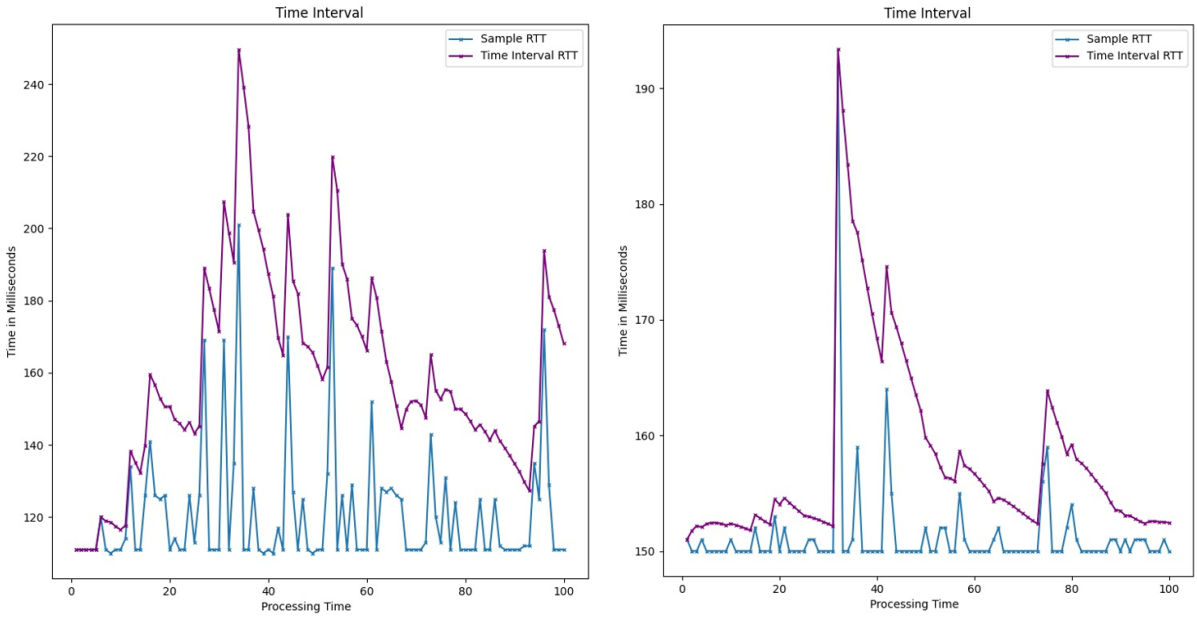


Due to the weighted calculation of estimated RTT, the sudden significant shift of sample RTT does not hugely affect the estimated RTT that can be found on the graph. However, the sudden change may lead estimated RTT rise and requires much more time to bring the value down to average. For example in the screenshot listed above, the blue line which represents sample RTT behaviors at around 25 sec, dramatically raised up to more than 375 from 180 milliseconds. And this rise did not lead the yellow line (estimated RTT) to increase as much as

sample RTT did due to the weighted calculation. This type of shift of estimated RTT is designed on intention since one bursty data is not supposed to affect the overall state transmission.

A pattern that could be found from all four collected plotting graphics is sample RTT initially started at an average value of sample RTT and then began to climb rapidly. After passing the peak caused by the unnormal sample RTT records, the estimated RTT decreased slowly, and the whole process may occur six or seven times in all 500 seconds of testing time. And the pattern is more obvious and representative in the time interval plotting graphs. For example, in the time interval of the University of Cambridge, the purple line kept steady at the beginning for about 80 seconds, then a rapid jump occurred which was caused by sample RTT. Moreover, the time interval is always slightly higher than the sample RTT since it added estimated RTT and all four deviated RTT.

(Below are the Time Interval Plotting of the University of Cambridge(Left) and the Universidade de São Paulo(Right))



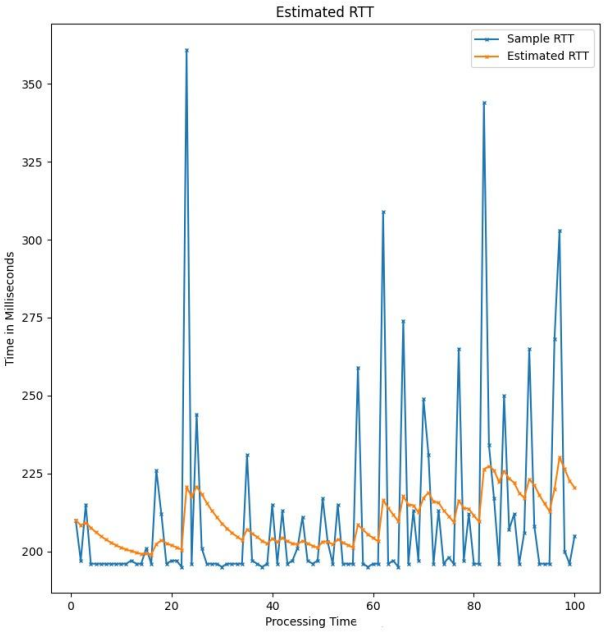
One of the most meaningful observations based on the collected figures is the amount of "peak" in all time intervals and estimated RTT plotting graphics are depended on the physical distance between the gateway router and the host servers. The far distance between the gateway

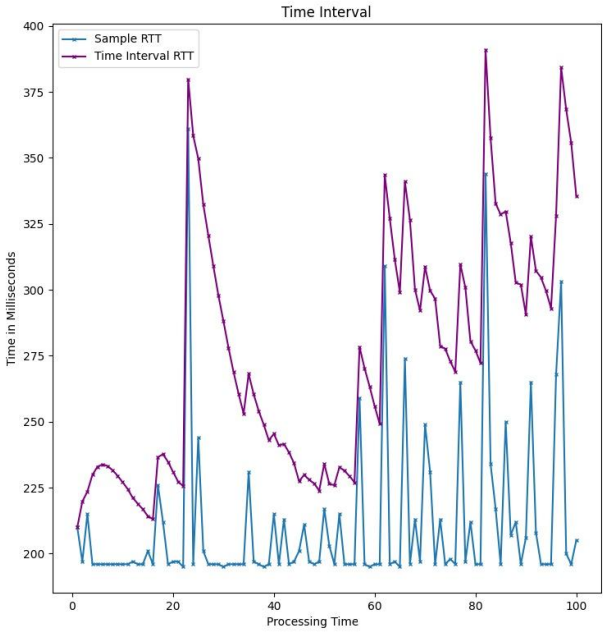
router and the server, the more “peaks” may occur during the same length of time for 500 seconds. For instance in the comparison of the two universities listed above, during the same amount of testing time, the ping command packet sent to Cambridge in the EU experiences more “peaks”, which leads the average time interval of that address to be certain large, and some of the intervals have been larger than 180 milliseconds. However, speaking with the Universidade de São Paulo which is physically located in Brazil and way closer to testing computers compared with Cambridge in the EU, the packets during the transmission have experienced fewer “peaks” shown on the plotting graph. At beginning of the transmission, we could see that only a few small amplitude peaks were found before about processing time 35, followed by a huge peak which is the highest interval in the whole testing time. Then the fluctuations of the sample RTT is decreasing, which makes the time intervals drop quickly toward the average value. And no other significant increase has been observed till the end of the testing.

This observation is not complicated to be drawn since the longer delivery, the higher possibility of packet loss could happen due to some unknown reasons, which could be congestion, delay, or dysfunctional routers.

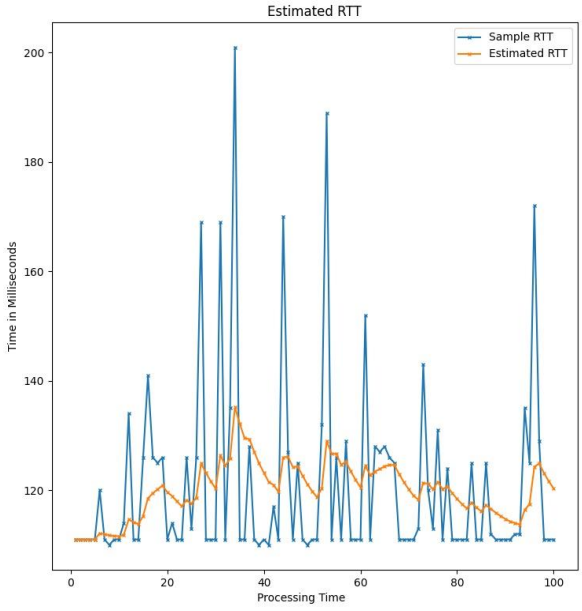
Following are the screenshots of the results of each IP address plotting graph：

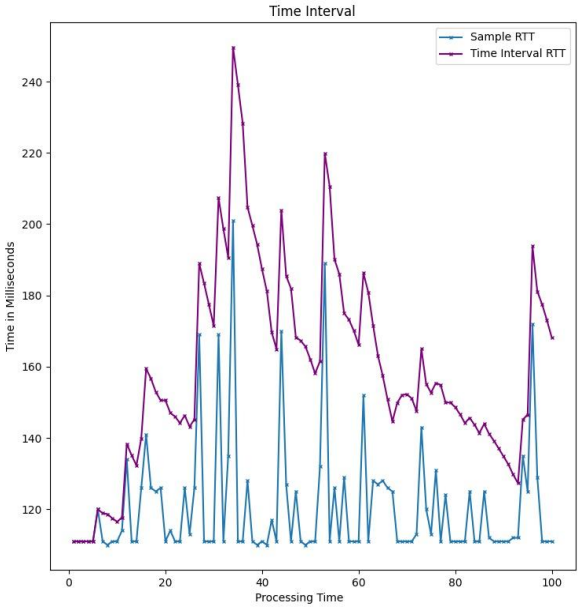
- **Peking University:**



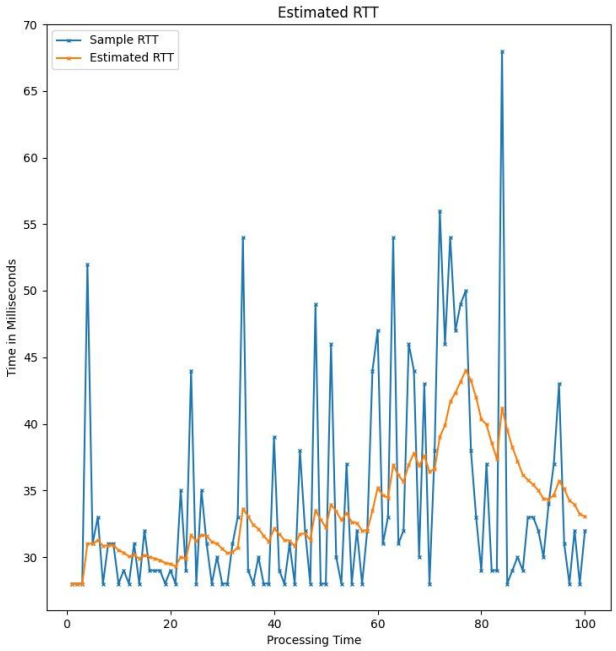


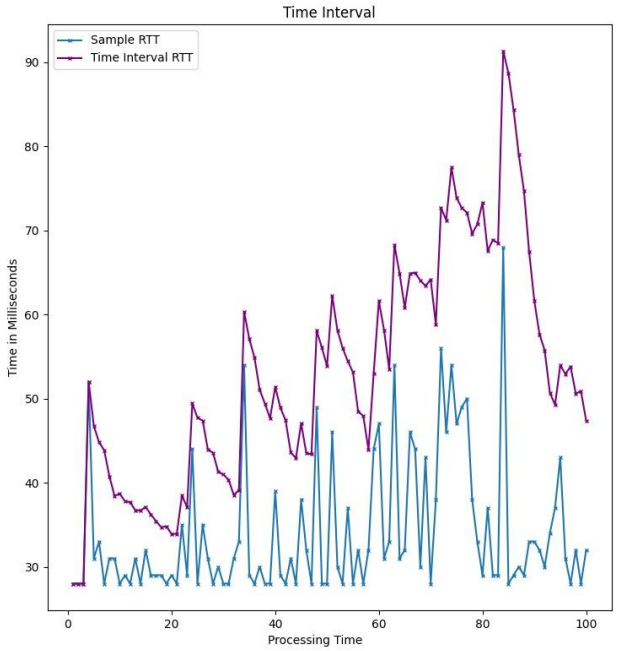
- **University of Cambridge**：



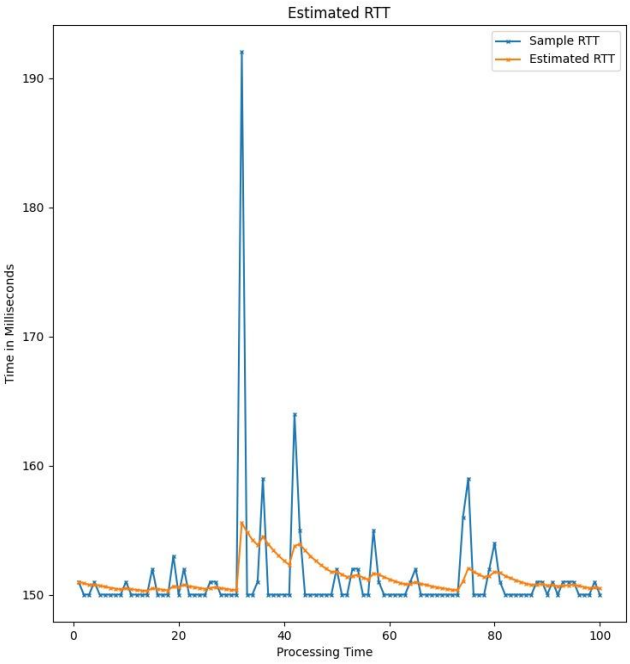


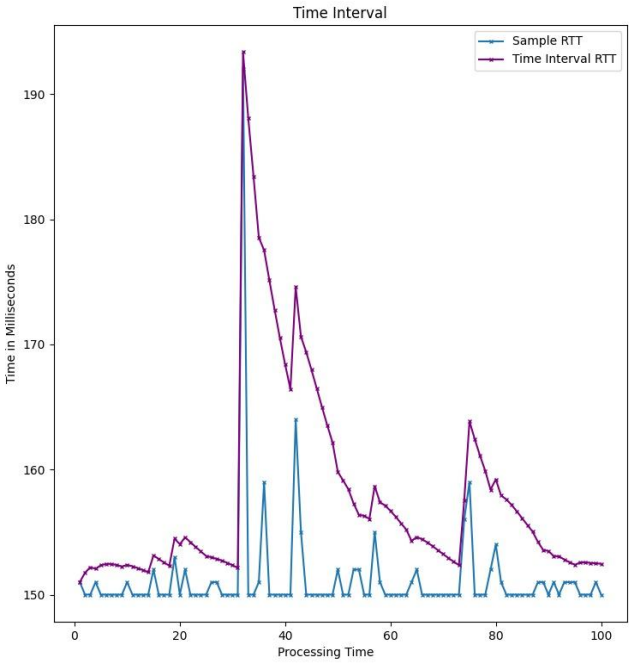
- **University of Sydney:**





- **Universidade de São Paulo:**





(All original size plotting graphics can be found in the submitted folder)

Work Reference

- https://stackoverflow.com/questions/2525312/measuring-ping-latency-of-a-server-python
- https://www.geeksforgeeks.org/matplotlib-pyplot-subplots-in-python/