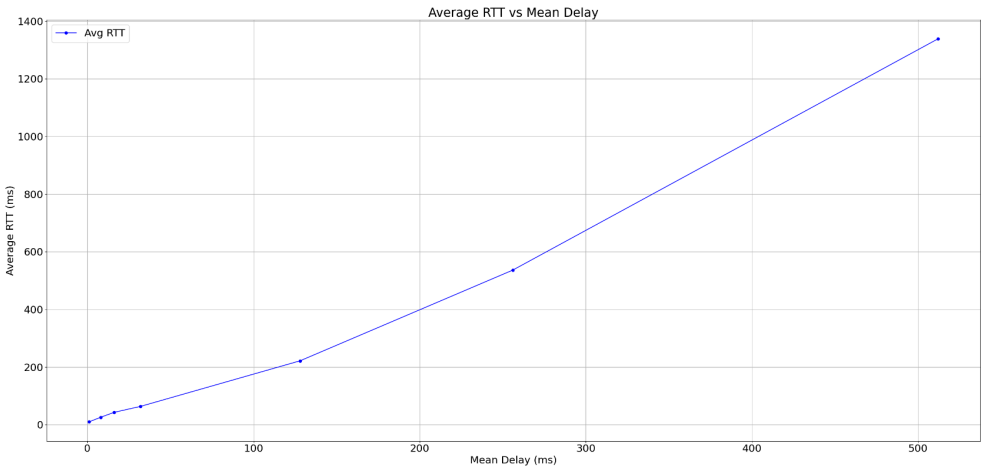
Random Packet Delays and Average RTT (Round-Trip Time)

In the first phase of our Network Security Project, we are tasked with adding random delays to package transfers between two environments, and a middle-man using a processor; then comparing the resulting mean delays with RTT (Round-Trip Time) averages. In this objective, I have chosen Python as my development language for the processor as I am more experienced in this language.

After setting up our environment, (through various tribulations such as some commands [./switch] resetting our environment without telling us). I added a pseudo randomized delay to every arriving packet in the processor. Since the pseudo-random generator I chose was using an exponential distribution with a mean that I could influence, I decided on testing our pings based on delays in ms (milliseconds), though with a floating point operation that would surely affect things. Regardless, I tested the pings with random millisecond delays of mean 1, 8, 16, 32, 128, 256 and 512. After parsing and analyzing the results, we saw a clear linear relationship between our delays and the average RTT.



Mean Delays (ms)	Average RTT (ms)
1	9.873
8	25.820
16	42.737
32	63.589
128	221.620
256	536.248
512	1338.701

As we can see in both the plot above and the table on the left. The chosen delays have a clear and linear affect on the Round-Trip Time (RTT). In fact, we can see that the RTTs almost perfectly increase with double the mean delay values, as we would expect; since a round trip is a simply a double trip with double the delays we added (as the processor is called on both ways of a ping). And it is worth mentioning that this relationship is true even though our “mean delays” have exponentially distributed randomizations and floating point uncertainties added on top of them, thus strengthening the indication of a relationship. As a result, we can clearly state that our environment is set up correctly, and the mean delays have a linear relationship with average RTTs.