Part 3: In this homework, I ran the Client/Server program on two machines: Machine A was my laptop running Ubuntu 16, Machine B was my desktop running Windows 10. I classified the required scenarios as shown below:

1. Machines A and B are on the *same machine*.
2. Machines A and B are on *different machines* within the *same local network*.
3. Machines A and B are on *different machines* within the *same wireless network*.
4. Machines A and B are on *different machines* at *different geographic locations*.

The code for this program can be found at:
https://github.com/dSouthard/ECEN5673_Fall2016/tree/master/UDPServerClient

In the first run, Machine A was the Client and Machine B was the server; in the second run that immediately followed, Machine A was the Server and Machine B was the Client.

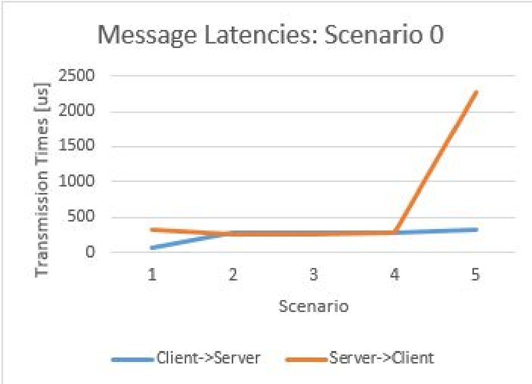
I ran into a difficulty when attempting to run Scenario 4. I was able to set up my home router to forward incoming ports to my home desktop (Machine B), allowing Machine B to receive messages from the Client and send replies. However, my different geographical location where Machine A was at this time was first the local library and then a Starbucks, with neither location allowing public reprogramming of their routers to allow for port forwarding. For this scenario then, I only have data from the first run. Additionally, since the laptop served as both client and server for both runs, the data from scenario 1 is identical for both runs.

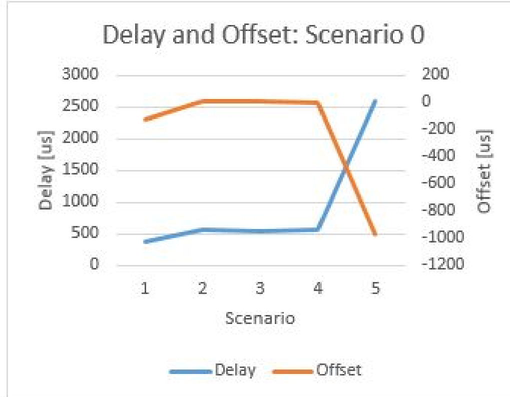
A second difficulty was when running scenario 2. For both runs in scenario 2 (same LAN), the clock times reported by Machine B on all message received/sent timestamps were exactly identical so far as the Windows OS was able to report. I attempted to use different time-reporting methods (from Python: *datetime.datetime.now()*, *datetime.datetime.utcnow()*, *time.time()*, *time.localtime()*) but all methods returned the same identical timestamps for this scenario alone. I concluded that the Windows OS might have something to do with this.

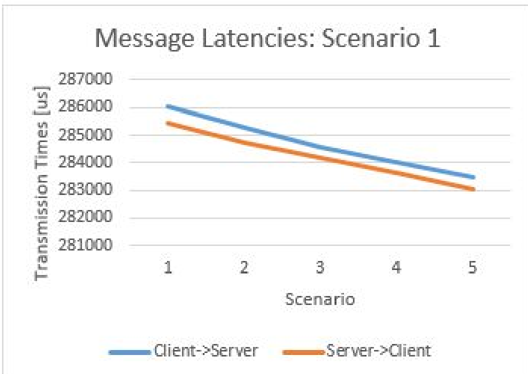
Please see the attached Excel spreadsheet to see the full calculations of messages latencies, averages, standard deviations, offsets, and delays. The plots of each are shown below.

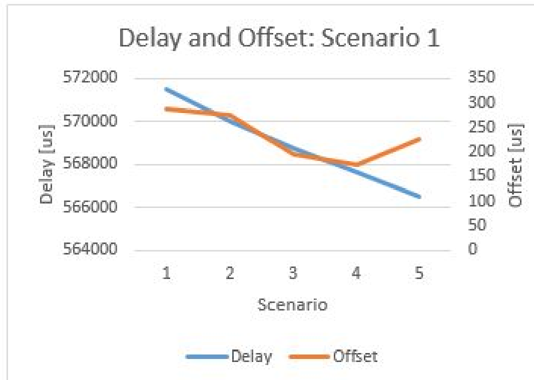
I would have expected the same machine latencies to be the smallest, since the message has the least amount to travel, and the different geographical locations to have the greatest latencies and variances, since those message would have to travel the furthest and would have the greatest chance of getting lost. In fact, that is what I did find. On average, the message latencies from the first scenario were miniscule compared to those from the other scenarios, with the last scenario having a dramatic increase in message latency times. The latency average does not grow significantly between scenario 2 and 3, where the machines are both on the same LAN and same WLAN respectively. This makes since because both machines are relatively close together and don't have to leave their local networks, wired or wireless. The dramatic increase in latency that occurred with scenario 4 can be explained by the messages having to travel across the internet some distances, greatly increasing their possibility of packet loss, packet collision, or packet delay.

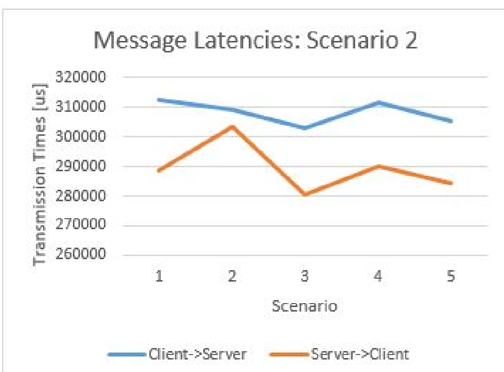
Run 1: Laptop = Client

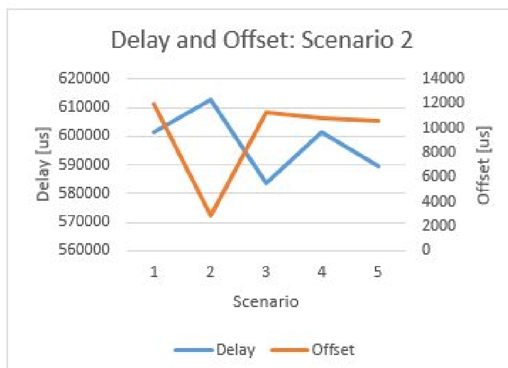


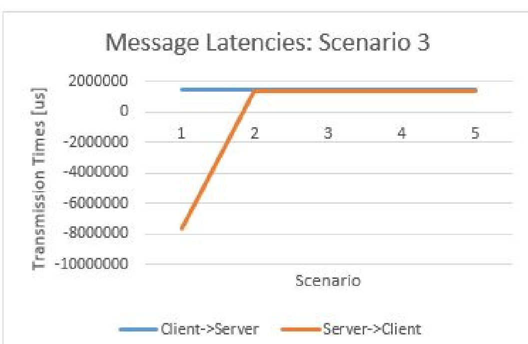


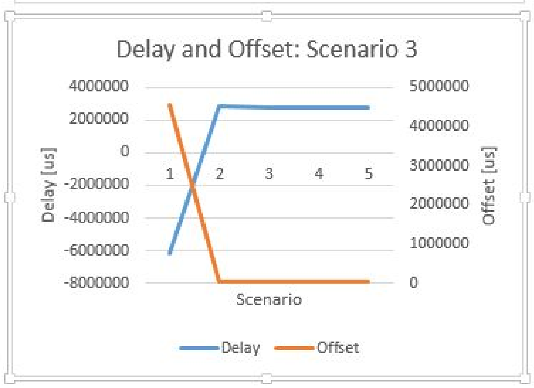


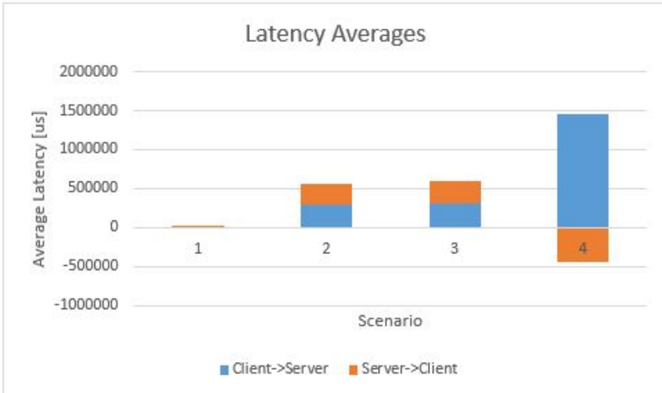








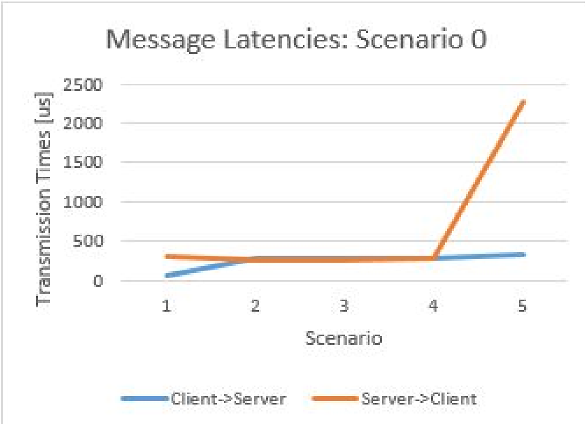


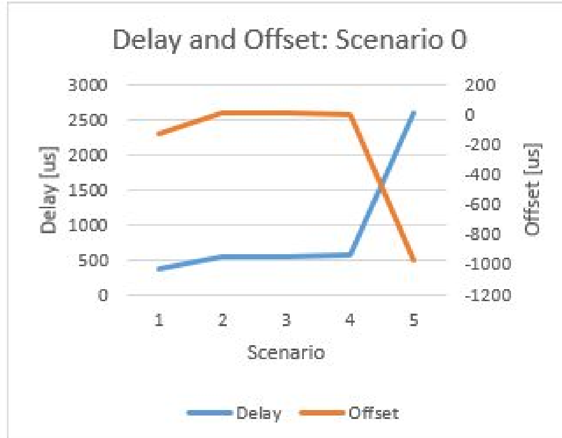


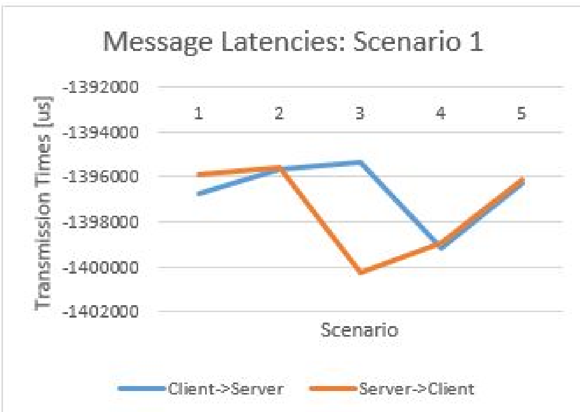
Q3. For each scenario, I was to decide which of the five calculated measurements of offset and delay provided the best accuracy.

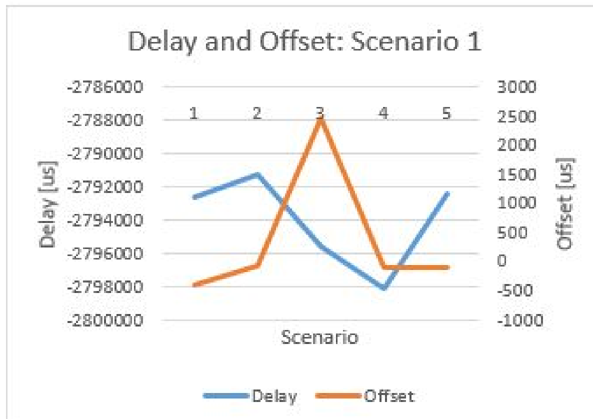
- Scenario 1 (Same Machine): The calculations from the 4th message would be the most inaccurate, as the message latencies increased significantly for the 5th message. For this scenario, I would say that the calculations from the 2nd messages were the most accurate. They resulted in no great increase in the message latencies.
- Scenario 2 (Same LAN): For each message, the calculated latencies decreased in time when compared to the times of the message before. Because of this trend, I would say that each newly calculated offset/delay pair was more accurate than the pair before. This would mean that the last pair, calculated from message 5, would be the most accurate of the group. However, looking at the final calculated pair, it looks as though the calculated offset from message 5 veers off the pattern set by the previous calculations. It may have resulted in increased latencies for any following messages, but there were none recorded for this experiment.
- Scenario 3 (Same WLAN): The message latencies decreased steadily between messages 1 and 3, and then again between messages 4 and 5 for client->server messages. The message latencies also decreased in between messages 2 and 3 and messages 4 and 5 for server->client replies. The greatest rate of decrease was between messages 2 and 3 for server->client replies. Because of this, I would say that the most accurate NTP calculation would be from message 2.
- Scenario 4 (Different geographical locations): In this scenario, the message latencies are relatively constant in between messages 2 and 5. There was a large change between message 1 and 2 for server->client replies, with the jump going from an extreme difference to a relatively constant transmission time. Because of this, I could say that the calculations from message 1 were the most accurate for the first correction, correcting the time differences well enough to leave the rest of the calculations as minor corrections in comparison.

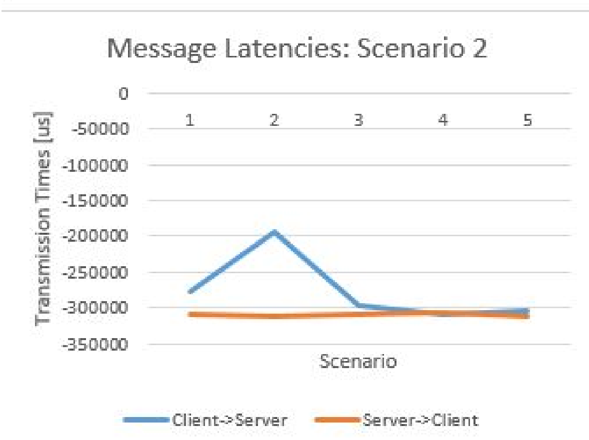
Run 2: Laptop = Server

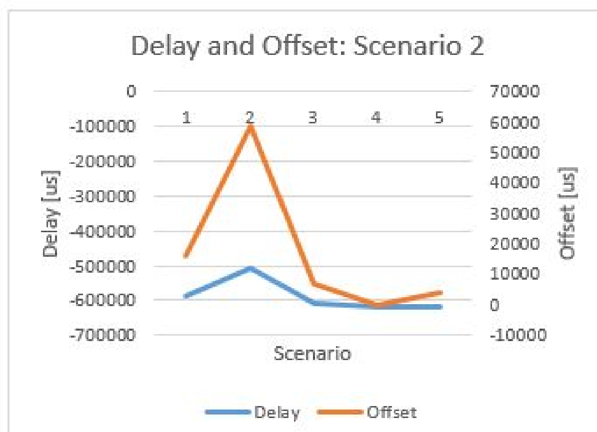


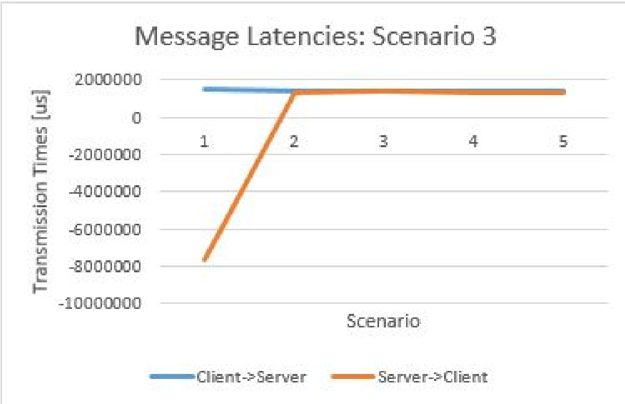


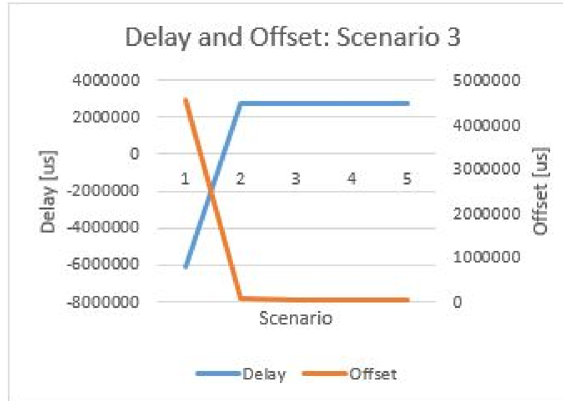


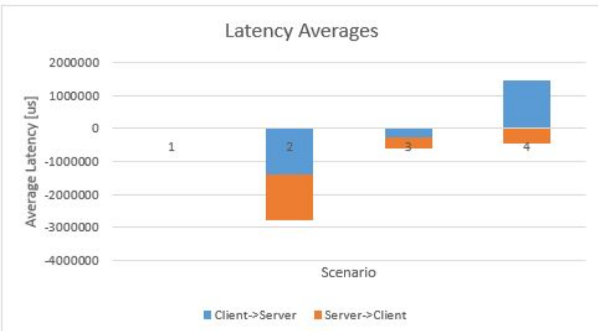












Q3. For each scenario, I was to decide which of the five calculated measurements of offset and delay provided the best accuracy. This is for the 2nd run.

- Scenario 1 (Same Machine): *Same as Run 1*
- Scenario 2 (Same LAN): The message latencies trended towards 0 most quickly in between messages 4 and 5. Also, the latencies between client->server and server->client messages most closely matched each other in that same period. Because of this, I would say that the NTP calculations from message 4 were the most accurate. The calculations from message 5 might also be of the same level of accuracy, but we do not have the results from a message 6 to see how the latencies trend.
- Scenario 3 (Same WLAN): The message latencies reach a relative consistency after message 3. From then on, there is only relatively minor changes in the latency times. The calculated delay and offset are also relatively similar from messages 3 to 5. However, there is a slight decrease in the magnitude of the latency from client->server in between messages 4 and 5. Because of this, I would say that the NTP calculations from message 4 are the most accurate.
- Scenario 4 (Different geographical locations): *Same as Run 1*