Homework 4

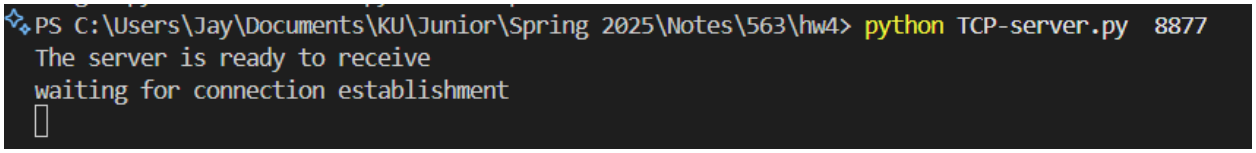
Questions and Deliverables.

(a) You need to submit your Python codes (4 .py files) along with the screenshots of output from your clients and servers. You should program your client and server to each print an informative statement whenever it takes an action (e.g., sends or receives a message, detects termination of input, etc.), so that you can see that your processes are working correctly (or not!). This also allows the GTA to also determine from this output if your processes are working correctly. – **python files are within the submitted zip folder. When the program is ran, the following screenshots shows the displayed outputs.**

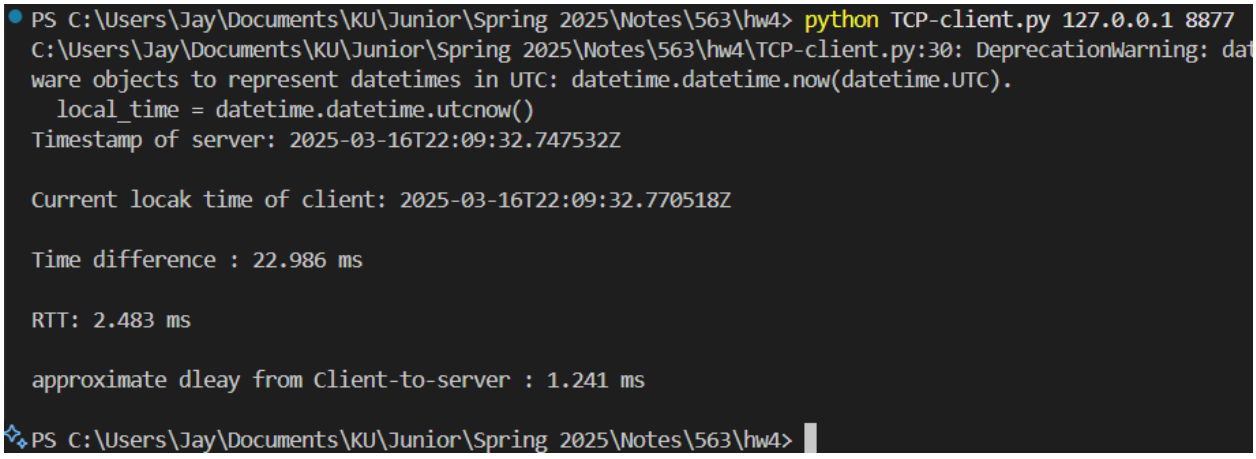
(b) You should hand in screenshots (or file content, if your process is writing to a file) of these informative messages as well as the required output of the client and server.

TCP client to server timestamps:

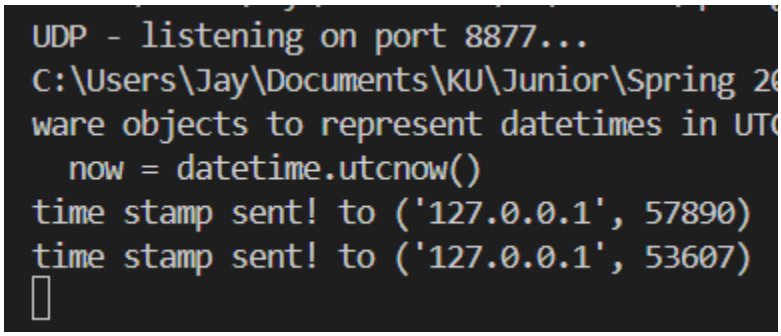
Server side:



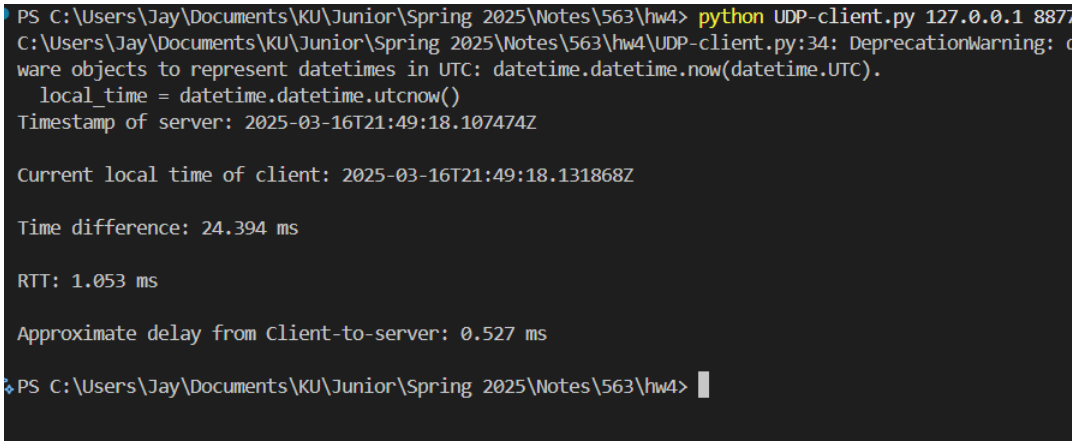
Client side:



UDP client-server timestamps:
Server side:



Client side:



(c) By comparing the TCP and UDP client-server applications, do you notice any differences in terms of delays? Explain your observations.
→ Yes. there were couple of instances where the delay was different for both applications. As I was creating these applications, the average time of delay was always around 1.3 - 1.5 ms. Since I was using my local machine for both client and server, there were a couple instances where the delay would be 0. I was shocked and though there is something wrong with my code. But, then realised that once the handshake is done, another request is achieved way faster.

While in UDP, this was never the case. The average time has always been around 1.4 - 1.5ms because there doesnt exist a handshake connection, there was never an instance of the delay time being 0 anywhere.

(d) Modify your client application to record the time at which the request is sent to the server and the time the response is received. Use this information to calculate the network round-trip time (RTT). Use the RTT to calculate the client-to-server delay. Are the network delay values the same for the client-to-server and server-to-client connections? Discuss your observations and provide possible explanations.

→ as in the provided screenshots, i calculated the time difference which is local_time - server_time.

For the RTTs -

- TCP:
    - RTT is approx 2-3ms when the initial connection is established between client and server .
    - persistently , the RTT along with client-to-server delay becomes 0.
    - Interestingly the time difference increases in some occurences. I think it could be because:
        - Since the server is processing the time stamp, there could be a potential processing delay as well as Queuing delay.
        - There can be fluctuations as well within the OSs' scheduling delay which can cause some small errors in python.
- UDP:
    - RTT is mostly around 1.2- 2ms.
    - persistently , the RTT holds it average value because the whole process is done again without any connection between server and client.
    - The time difference does not increase much but remains at an average.
- No, the delay are not always the same.
    - Because there is a possibility of network congestion along with routing differences between the server and client. I don't know what reason it could be if the client and server are the same.
    - Also, in TCP if there are packers that are longer to process for the server, it can have different transmission delay compared to the client.
    - And in UDP, because of the network congestion, the packets can be lost, which leads the server to create another response which causes the higher delay from server to client.

(e) Run your NTP client-server applications at various times of the day and when the client and server are located with physical distances from each other. Do you observe any changes in the delay values? Discuss your observations and provide possible explanations for the variations.

The following pictures shows the TCP and UDP applications running at 8:15pm.

The results are quite surprising.

For TCP:

- The RTT increased by double. Which is interesting as i thought it would take less than usual for an odd time like this. But, I figured that since i am on Jayhawk network at a residence hall, the traffic must have increased significantly compared to my house. And there also is possibility of routing inefficiencies.
- The reasons behind the increase in delay could be due to more queuing delays in routers of this public network.
- I couldn't think of the possibilities due to which the time difference is dropped significantly. I am not sure if that is due to the high powered network of some kind of better time synchronization.

TCP-

```
Timestamp of server: 2025-03-17T01:12:27.400816Z

Current locak time of client: 2025-03-17T01:12:27.408975Z

Time difference : 8.159 ms

RTT: 6.923 ms

approximate dleay from Client-to-server : 3.461 ms
```

For UDP:

- The RTT was less than the average time.
- The time difference here also dropped significantly from the previous sessions.
- The decrease in RTT as well as the client to server time could be mostly because of less congestion. Since because of break, the network is not being used by many users, resulting in less traffic and a more stable connection.
- The drop in the time difference here could be because of the different routing conditions of this public network but I am still not sure what affects the time difference most.

UDP-

```
Timestamp of server: 2025-03-17T01:14:38.083201Z

Current local time of client: 2025-03-17T01:14:38.092754Z

Time difference: 9.553 ms

RTT: 0.801 ms

Approximate delay from Client-to-server: 0.401 ms
```