

Alexandru Cioanca

Gabriel Molina

Dustin Phan

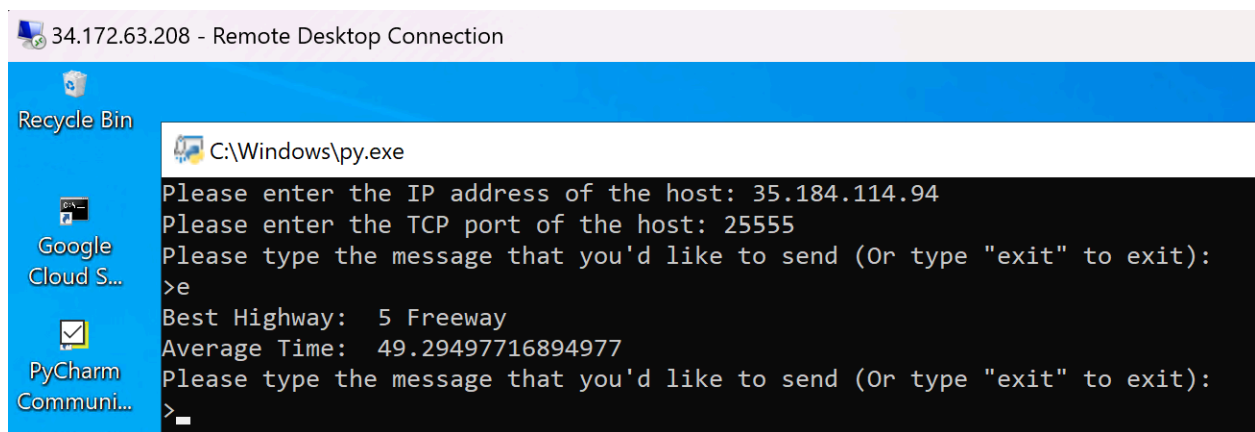
Assignment 8 Report

This demo has been a fun and interactive assignment overall. Personally writing this, I feel there was the right amount of challenge mixed in with the knowledge that we have gained throughout the semester regarding networks to make this a good assignment. That said, there were a couple of unique parts about this assignment that I'll highlight. Afterward, I'll go over our experience with Dataniz briefly as well.

Firstly, our approach. The way the challenge is presented in the assignment feels like it is immediately pointing to a singular, simple solution, especially considering the scale at which we were working. In order to get the average time to go through traffic from our sensors, we simply designed a function called "SortSensors()" to go through the database and assign all the times to their corresponding sensors in a new dictionary. Using this method, we then used another function "BestHighway" to go through the dictionary and return the one with the lowest average time. Though the logic behind it wasn't hard to process at all, there was one considerable challenge we faced when doing this. In order to return the freeway name that the average is associated with, we had to query another mongo database collection, which required making some extra edits to our MongoDB program. Eventually, we were able to query our metadata collection in search for the freeway whose sensor matches the id of the sensor that gave us our average data.

Next, our Dataniz experience. Personally, my experience using Dataniz has been nothing special. I feel that the best way to know that you're using good software is how often you notice that you are using it, and in this case, other than the login, the software was clean and concise, making connecting it to our database easy. I have seen other students complaining about login issues and other glitches, but I have not experienced such. Neat stuff.

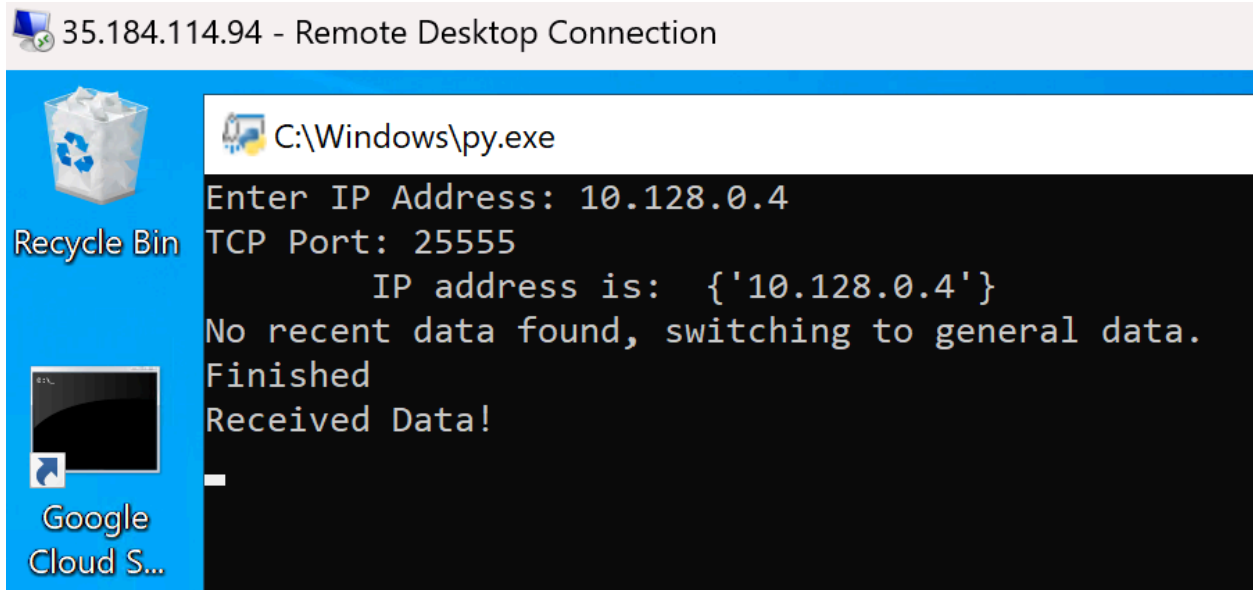
Finally, I'll quickly go over our screenshots.



The screenshot shows a Remote Desktop Connection window titled "34.172.63.208 - Remote Desktop Connection". The desktop background is blue. On the left side, there is a taskbar with icons for "Recycle Bin", "Google Cloud S...", and "PyCharm Communi...". In the center of the desktop, a command prompt window is open, displaying the following text:

```
C:\Windows\py.exe
Please enter the IP address of the host: 35.184.114.94
Please enter the TCP port of the host: 25555
Please type the message that you'd like to send (Or type "exit" to exit):
>e
Best Highway: 5 Freeway
Average Time: 49.29497716894977
Please type the message that you'd like to send (Or type "exit" to exit):
>
```

Here is our client program, it connects to the tcp port and allows the user to enter characters as input. Now, aside from the case that the user types "exit," after receiving input from the user, the program takes the data, whatever it may be, encodes it, and then sends it to the server. The contents of the message do not matter to the server, and it will just treat it as a ping.



On our backend side, the server receives the ping and then runs our two functions to get the best highway, querying the database for the most recent documents (which if unfound will prompt the server to simply use all the documents,) and then return the results to the client.

Overall, the assignment has been a fun thought experiment and exercise. We look forward to bringing this knowledge to our future workplaces.