P2P FILE SHARING APPLICATION USING PHYTON3

Group Members

Turgay Barkın Türkşen (1802548):
Designing and interpreting specifications into a project template

Gülce Külahçıoğlu: (1731380):
Research on Networking applications with Python3

Remaining workload of writing the actual code and testing it was shared equally by working together using Hamachi and Discord.

Development

The code was developed on *MacOS Catalina* and *Windows 10*, using *PyCharm Community Edition* and **Phyton3.8** as the interpreter. There have been a few challenges during the development process which we can list as:

- Some of the functions reaching common outer resources such as directories have caused errors if the different python projects were executed in a wrong order. We solved the problem by catching exceptions and waiting for the other files to get executed. Therefore unexpected exits were avoided.
- A lot of different combinations for which IPs were for which arguments were tried to get the UDP and TCP connections working properly.
- On the early stages, Windows Firewall has caused problems blocking the connections for security reasons. Problem was solved by disabling the Firewall on client devices running Windows.
- JSON formatting we created didn't have an extra room for the transferring file's extension. At first, the chunks were created by using only the name of the files and "_n" suffix combined. But after not being able to transfer the extensions in any other way and only being able to save the files without their extensions, using the file name combined with the extension and the suffix was decided.
- After mixing up which IPs to change for the Hamachi to run correctly, a global variable called *h_ip* at the beginning of the *service_listener.py* and *p2p_server.py* holding current Hamachi IP was added.
- The function call for the server.log was executed by a Thread, causing an infinite loop which was making the file size quite large. Problem was solved by rewriting the logging part of the code.

Notes

- service_listener.py and p2p_server.py files contains *h_ip* variable at the beginning of the code, which needs to be changed with the client's Hamachi IP before launching (for far-away testing).
- The opening order of the different Python files are set to be compatible with the specifications provided. Yet we have handled the exceptions and none of the files will give an error if not opened in the correct order.
- The project was tested on a network consisting of computers running Windows 7, Windows 8, Windows 10 and MacOS Catalina. The code was split in some parts to make it compatible with both operating systems. MacOS doesn't discover itself as a client but on Windows, it does.
- Application logs server and client data in two separate .log files. The available client and files to be downloaded are stored as a single key directory in a .txt file. If a far user deletes their chunks and another client tries to download those files, it can't. So the '*tags.txt*' needs to be reset if a client removes chunks.

References

Below are the some of the references used while creating the project:

- https://realpython.com/python-sockets/
- https://github.com/ppeddabbu/P2P-File-Sharing/
- https://www.python.org/doc/
- https://stackoverflow.com/questions/47317290/reading-data-from-a-file-python-3
- http://ebyerly.com/python-threading-examples.html
- CMP2204 Spring '20 Lecture Notes by Ece Gelal Soyak