John Marrion

Yiang Shen

Binkang Yu

Brenda Solis

Reflection on the Term Project

Defined as a group of computers that are linked together with equal permission and responsibilities for processing data. The peer-to-peer network has quickly dominated the 21st century's workplace environment, reshaped entire industries and vastly improved connectivity and access to shared resources. The task at hand required us to design a program which allowed for two or more computers to synchronize files across a local area network, with each computer being able to run its own local instance of the client software, and the file. Implementing such a feat would be quite the challenge.

At the heart of all peer to peer networks there are an equal number of peer nodes that work as both clients and servers typically implementing an overlay network. When deciding whether to implement either an unstructured or structed network we took into consideration that while unstructured networks are said to be easy to build, they offer poor search success.

For a group with relatively little experience designing networks, let alone working with distributive systems this project allowed us to acquire a substantiated grasp on the content material as well as the innovative features of GitHub, Visual Studio 2019, and Java 11. For example, as discussed in class no two computers inherently operate with the same standardized time, we attempted to establish our own local time. We tried implementing super peers, that way we could make it responsible for mapping clients on the network. We would soon have to abandon that idea as we concluded that we were too ambitious. We are incredibly grateful for the revert feature as we experienced a situation in which a merger broke the code, and thankfully we were able to return to a prior version of the project. Visuals Studio’s Live share allowed us to experience real time collaboration a la Google Docs. It was during one of these coding sessions that we learned of the enhanced switch statement.

A large bulk of this endeavor was spent reading the documentation regarding sockets and hash maps, as well as variety of file handling. Unfortunately, even with the countless hours spent reading documentation we still found ourselves having to rework our code because our HashMap wasn't properly storing our client or receiving a file. It felt like we were just going in circles. The amount of frustration experienced has been immeasurable, at one point we were trying to for like three hours. Until it finally occurred to us, we were using the wrong key. If we had more time, we really think we could have been able to add more than one client without the code breaking. Overall, we really appreciated the opportunity.