

Team Members:

Name	UFID
Ashish Nautiyal	6351-6173
Gagan Sharma	9134-9422
Sandeep Dommaraju	2519-3611

How to Run:

There is a shell script present in the zip folder. Running the script will open 3 Peers in the network, which will look for the other active peers through the data present in “Common.cfg” and “PeerInfo.cfg” and instantiate handshake respectively.

In order to run the script, simply type the following command:

“sh run.sh”

In order to check whether it ran properly, one can look for the log files created for the active peers and logs present in them for the operations performed during the execution.

An alternative way to execute is to manually start the peers based on the info available in the “PeerInfo.cfg”

Note: Currently the simulation has been performed by running different peers on the same machine by using different ports. However, if we have available machines, we can run the same program on different machines.

Current Progress:

Initially at this checkpoint, we have completed the following tasks:

Class	Functions
Peer	This class will create instances of Peer for the peer to peer communication.
BitField	This class maintains the record of various pieces of files available by a Peer
P2PLogger	This class is used to log different events. Every Peer instance will have an object of Logger.
Configuration	This class is used to parse the data from "Common.cfg" and "PeerInfo.cfg".
HandShake	This class is used to send and receive handshake message between various peers.

At the current phase, we can instantiate peers by parsing the data from Config files. As the peers instantiate they will send handshake message to establish communication with other active peers in the network.

As the peers are created and establish communication with other peers, the event will be logged for each peer with the time at which the event occur.

Future Progress:

The following tasks remains to be completed:

1. Declare interest in communicating with other peers
2. Request particular piece of file from other peers
3. Implement choke and unchoke operations.

We use threads to establish peer to peer communication. We need to make sure that at the end when all the peers have received files, there are no remaining zombie threads present in the network.