

# **CNT5106C – COMPUTER NETWORKS**

## **FINAL PROJECT – P2P FILE SHARING**

### **Team Members :**

Siva Praneeth Reddy Papineni – 1887- 2370

Nikhitha Sudati – 9493-6209

Manoj Jampana – 1443-9156

### **Instructions to run :**

- We initially logged into the CISE Linux server by ssh from the storm server. Thereafter we go into the directory `cd CnSemproj/CNProject/P2Pfilesharing/src.`
- We used `ssh s.papineni@lin114 -00/01/02/03/04.cise.ufl.edu` for gaining access to corresponding Linux machines.
- We have to run all the 5 peers in the 5 different linux servers:

`java bitTorrents/peer_1001/PeerProcess1001 1001`

`java bitTorrents/peer_1002/PeerProcess1002 1002`

`java bitTorrents/peer_1003/PeerProcess1003 1003`

`java bitTorrents/peer_1004/PeerProcess1004 1004`

`java bitTorrents/peer_1005/PeerProcess1005 1005`

### **Team Contribution :**

- Siva Praneeth Reddy Papineni : He has contributed in Protocol Definition, Multithreading management, File Transfer, and In Video Recording of the output.
- Manoj Jampana: He has contributed in the connection establishment , and changing neighbors
- Nikhitha Sudati: She has contributed in the extraction of info from Common.cfg and PeerInfo.cfg and Logger Configuration.
- Everyone has Contributed equally in debugging and coding.

### **Link for the video explanation:**

[CN-P2P file sharing video explanation](#)

### **Description of the Protocol:**

The Common.cfg file consists of details regarding the size of the file which is initiated for peer-to-peer transfer. In addition, we also have defined the piece size which defines the size of individual bits of the file that are transferred through each peer, and the number of preferred neighbors for a peer to choke/unchoke its neighbors. Peerinfo.cfg consists of the port numbers and the corresponding UFL Linux machines through which we have to start the peers. We have demonstrated P2P file sharing for 5 peers namely peer1001, peer1002, peer1003, peer1004, peer1005. Peer1001 will be having the entire file at the beginning. When the protocol starts functioning, peers will connect, and HANDSHAKE messages will be exchanged upon successful connection establishment. Further, BITFIELD messages will be exchanged that specify the data indices of each peer. If say peer 1002 requires data from peer 1003 it sends an INTERESTED message. If peer 1002 has the data, it sends a NOT INTERESTED message to peer 1003. When a peer receives either a BITFIELD or HAVE message, the peer will send a REQUEST message with the specific PIECE number it is requesting to the corresponding peer. A peer will select its preferred k neighbors and changes them for every UNCHOKE interval. Peers will be transmitting data between unchoking intervals and the remaining peers will choke (CHOKe). Among the choked peers, one peer will be optimistically unchoked and receives messages from that peer. Eventually, when all peers get the file, the connection is terminated and the execution halts.