ARJUN ASHOK, **18PW03**, **M.Sc. Software Systems**

18XW46 - **COMPUTER NETWORKS AND TCP/IP LAB** PACKAGE

BitTorrent Client

BitTorrent is a **protocol** for **downloading** and **distributing** files across the Internet. In contrast with the traditional **client/server** relationship, in which downloaders connect to a central server, participants in the BitTorrent network, called **peers**, download pieces of files from each other—this is what makes it a **peer-to-peer protocol**.

The original specification given at the links:
https://www.bittorrent.org/beps/bep_0003.html
https://wiki.theory.org/index.php/BitTorrentSpecification
will be implemented in **Python 3.**

1. The torrent client will be able to open a **.torrent** file, **parse** the torrent file to identify the files pointed to by the torrent and the **URL of the trackers** given in the torrent.

2. When this is done, the client will be able to send **GET** requests to the **trackers** with the parameters specified in the specification, to get back a **list of other users**(**seeders and peers**) in the **swarm** for this particular torrent. **Seeders** are users that have a **complete copy** of the file, and **peers** are users that have a **partial copy** of the file and are still downloading the other parts from other peers/seeders.

3. The client connects to peers using **TCP sockets**. The connections are made by **handshaking**, then sends an **'interested'** message, to which the peer responds by **unchoking** the client. There onwards, the peer **responds to requests** made by the client to obtain specific **pieces** of files that the peer possesses. Whenever a piece is received, the **hash** of the piece is checked with the **hash** in the torrent file to verify its **integrity**, failing which it has to request the piece again. With the concept of **multi-threading** in place, **multiple pieces** will be **downloaded** from **multiple peers simultaneously**.

4. Once the client has at least one piece, it can now **respond to requests from other peers**, and thus starts **uploading the pieces** that it has to the requesting peers, greatly improving the **overall speed of distribution**.