**Minor Project 2015-2016**

**BITTORRENT**

PHASE I EVALUATION



JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY (NOIDA)

(Declared Deemed to be University U/S 3 of UGC Act)

A-10, SECTOR-62, NOIDA, INDIA (12)

##### 

##### 

##### 

##### **SUBMITTED TO- SUBMITTED BY -**

**Aditi Sharma**  **Batch-B7**

##### ***SIDDHARTH UPPAL 13103622***

***AVANTIKA VERMA 13103625***

***KRITI AGGARWAL 13103644***

Introduction

BitTorrent is a communications protocol for the practice of peer-to-peer file sharing that is used to distribute large amounts of data over the Internet. BitTorrent is one of the most common protocols for transferring large files. To send or receive files the user must have a [BitTorrent client](https://en.wikipedia.org/wiki/Comparison_of_BitTorrent_clients); a computer program that implements the BitTorrent protocol.

Objectives

* We will be making a bittorrent client that will be implementing the bittorrent protocol.
* Bittorrent client’s gui will be made using python frameworks like tkinter , PyGTK.
* We will be analyzing the network transmission and will try to secure the transmission from various malwares attacks namely denial of service attack.
* optimizing the peer selection using algorithms.

Work Division

Background Study and Findings

**What is BitTorrent**

The BitTorrent protocol can be used to reduce the server and network impact of distributing large files. Rather than downloading a file from a single source server, the BitTorrent protocol allows users to join a "swarm" of hosts to upload to/download from each other simultaneously. Using the BitTorrent protocol, several basic computers, such as home computers, can replace large servers while efficiently distributing files to many recipients. This lower bandwidth usage also helps prevent large spikes in internet traffic in a given area, keeping internet speeds higher for all users in general, regardless of whether or not they use the BitTorrent protocol

**How bittorrent works?**

A user who wants to upload a file first creates a small torrent descriptor file that they distribute by conventional means (web, email, etc.). They then make the file itself available through a BitTorrent node acting as a seed. Those with the torrent descriptor file can give it to their own BitTorrent nodes, which—acting as peers or leechers—download it by connecting to the seed and/or other peers .

A BitTorrent client is any program that implements the BitTorrent protocol. Each client is capable of preparing, requesting, and transmitting any type of computer file over a network, using the protocol. A peer is any computer running an instance of a client.

To share a file or group of files, a peer first creates a small file called a "torrent" (e.g. MyFile.torrent). This file contains metadata about the files to be shared and about the tracker, the computer that coordinates the file distribution. Peers that want to download the file must first obtain a torrent file for it and connect to the specified tracker, which tells them from which other peers to download the pieces of the file. It is based on segmented file transfer.

**Segmented file transfer** implementation: the file being distributed is divided into segments called *pieces*. As each peer receives a new piece of the file it becomes a source (of that piece) for other peers, relieving the original seed from having to send that piece to every computer or user wishing a copy. With BitTorrent, the task of distributing the file is shared by those who want it; it is entirely possible for the seed to send only a single copy of the file itself and eventually distribute to an unlimited number of peers.

**PROTOCOLS USED:**

**BitTorrent tracker protocol**

The BitTorrent tracker protocol is used by clients to request the IP addresses of other peers associated with a torrent, and to exchange the client's transfer statistics. Clients contact a centralized server, known as a \*tracker\*, which stores their addresses and responds with the addresses of other clients (also known as \*peers\*). The tracker does not know which nodes have which pieces; its job is to tell its clients where to find each other.

Query parameters must be encoded according to the rules for HTML form submissions through HTTP GET: 'reserved character' bytes are encoded in hexadecimal as %HH, and space is encoded as "+"; names and values are joined with "=" and the pairs joined with "&".

The tracker's announce URL is obtained from the announce entry of the root dictionary of the torrent metadata file.

Clients announce themselves by sending a GET request to the tracker's announce URL with "?" and the following parameters appended

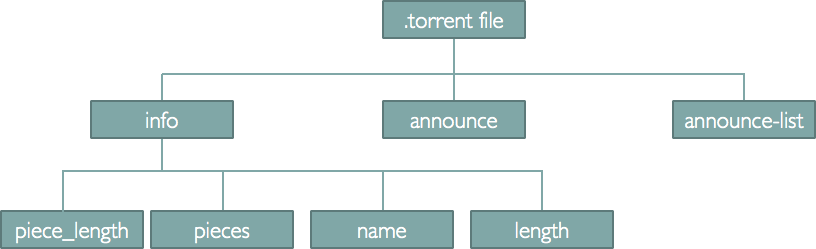
Designing

STEP 1: view the .torrent file in text editor . it contains the info about the file to be downloaded

in form of bencoding.

STEP 2:decode the torrent file and save info like announce url ,info dictionary which includes

piece length,name, path and other things required for tracking.



STEP3: connection to the tracker. **he tracker is an HTTP/HTTPS service which responds to HTTP GET requests. The requests include metrics from clients that help the tracker keep overall statistics about the torrent. The response includes a peer list that helps the client participate in the torrent. The base URL consists of the "announce URL" as defined in the metainfo (.torrent) file. The parameters are then added to this URL, using standard CGI methods (i.e. a '?' after the announce URL, followed by 'param=value' sequences separated by '&').**

STEP4:if the GET request contains correct info then tracKer sends the response containing

bencoded dictionary.

STEP5:connection to tracke:-twisted framework is used to make connection to

multiple clients.**The Twisted Framework involves a Reactor. The reactor is a loop that continously checks for any new event that may have occurred. The new events in our case is the receiving of messages from peers, successful connection with a peer or loss of connection. Once an event occurs, its respective function is called, known as the callback function**

Partial Implementation

Testing

References