International Journal Of Advanced Research In Engineering Technology & Sciences
ISSN: 2394-2819
Email: editor@ijarets.org
March- 2015 Volume 2, Issue-3
www.ijarets.org

Multiple Peers Connections for Mobile Users

Shashank Yadav
Assistant Professor
Department of IT
SRM University NCR Campus
Modinagar

Karan Khanna, Anmol Kakkar, Vijay Veer Bansal
B.Tech Student
Department of IT
SRM University NCR Campus
Modinagar

ABSTRACT
With the rising popularity of internet day by day the data available on the internet is growing with exponential rate. From movies to video, from audios to texts all sorts of files are available on the internet for the users to download them sitting at their home. Download traffic is increasing every day. So we have developed a downloader application that would distribute the task of downloading amongst different clients who want to download the same file from a single server.

Keywords: Trackez, Peers: Initial Seeder, Leechers Tracker URL, Mobile users Torrent File.

Literature Survey

As the internet became bigger and bigger The World came closer. Everyone wanted to share the stuff they got and easily access the stuff others share. Different sites provided the download facility. A typical downloading mechanism involves a server which acts as the source for the file wished to be downloaded and the other a client that downloads the file. Each client that wishes to download even the same file establishes a different connection with the server. Each client connection is independent of each other. This mechanism takes a lot of time for the downloading of even a small file. Moreover in the long course of downloading due to the traffic on the server the connection with any number of the clients may be broken resulting in abortion of downloading task.
With this application the downloading mechanism over the network becomes all the more faster and reliable. Similar style of application is available on the internet as U-torrent downloader.

Aim and Objective

The task here is to develop an application that would provide faster downloading for clients connected over a weak network say a wireless network. Different clients connected to the same network and wishing to download the same file from a system on that network , would help each other to accelerate the task of downloading by serving as a source for some part of that file themselves.
In typical downloading mechanism the file stored at the initial source itself acts as a single source to all the clients who wish to download the same file. This usually takes a lot of time in downloading even a small file. Even there could be a loss of connection between clients and the source.
To make the job of downloading faster and more reliable we can use the following strategy. For this three steps are planned out:

- A tracker that would trace the file requested by a client on other systems over the network and On finding the file it would direct that client to that source.
- Dividing the file into pieces and allow different clients wishing that particular file to download different pieces.
- Now each client can act as a source for the pieces they have already downloaded for other clients.

IJARETS

INTRODUCTION

First, a user playing the role of file-provider makes a file available to the network. This first user's file is called a seed and its availability on the network allows other users, called peers (or leechers), to connect and begin to download the seed file. As new peers connect to the network and request the same file, their computer receives a different piece of the data from the seed. Once multiple peers have multiple pieces of the seed, Bit Torrent allows each to become a source for that portion of the file. The effect of this is to take on a small part of the task and relieve the initial user, distributing the file download task among the seed and many peers.
With Bit Torrent, no one computer needs to supply data in quantities which could jeopardize the task by overwhelming all resources, yet the same final result—each peer eventually receiving the entire file—is still reached.

After the file is successfully and completely downloaded by a given peer, the peer is able to shift roles and become an additional seed, helping the remaining peers to receive the entire file. This eventual shift from lechers to seeders determines the overall availability of the file.

Implementation The modules available in the application are mainly divided into the following two categories:

Tracker

To share a file, a seeder first creates a small file called a "torrent" for that file. 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. The tracker typically has information about the tracker URL, no. of pieces/chunks of the file, SHA1 hash of each piece and the list of peers which are currently seeding/leeching that file.

Peers

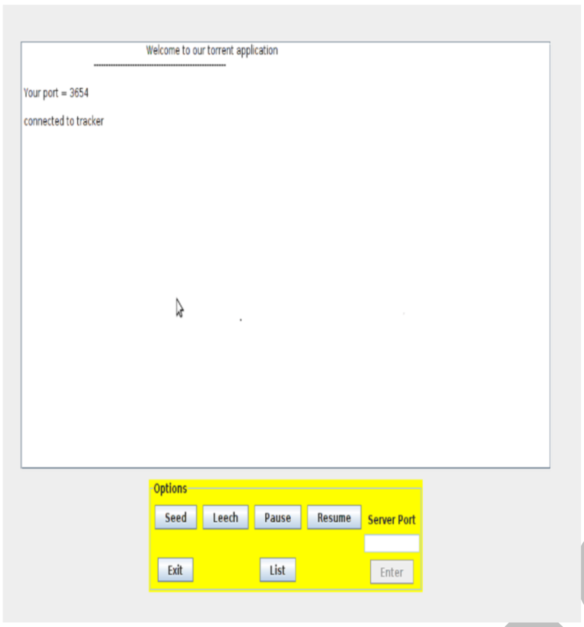
a. **Initial Seeder**
Initial seeder is the torrent client which originally decided to share the file by uploading its .torrent file on the tracker.

b. **Leechers**
These are the clients which download a particular file using the torrent application. The process involves downloading the corresponding .torrent file from the tracker, then connecting to the peers, via a TCP connection, which are currently seeding or leeching the file. Then the leecher enquires about the available chunks of that file from every peer. Based on a certain policy like random/rarest first, it requests chunks from each peer and downloads the entire file by arranging these chunks locally.

The beauty of torrent application lies in the fact that one can download parts of the file from multiple locations thereby reducing the load on each peer. The leecher can simultaneously seed the chunks which it has already downloaded.

IJARETS

Result

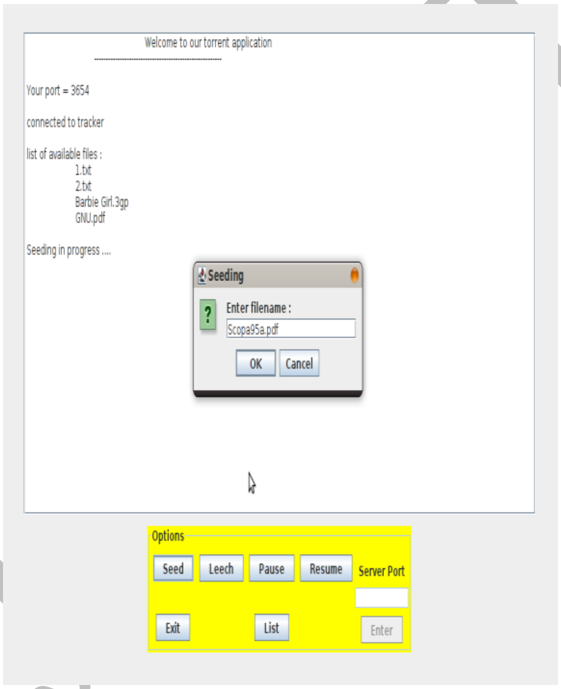


Welcome to our torrent application

Your port = 3654

connected to tracker

Tracker



Welcome to our torrent application

Your port = 3654

connected to tracker

list of available files :
1.txt
2.txt
Barbie Girl.3gp
GNU.pdf

Seeding in progress ....

Seeding

Enter filename :

Scopa95a.pdf

Seeder

IJARETS

CONCLUSION

In our project, we have implemented a basic torrent application comprising of a tracker and multiple peers. We have stressed on factors which are of significance to a mobile client viz. loss of connection/ no duplicate chunks/ efficient use of resources et cetera. The code has been written in java and an effort has been made to follow the bit torrent protocol specifications. This is evident in our use of Ben coding for torrent file, SHA1 hash check for chunks, keeping a tracker for each file and not flood the medium with chunk queries.
This project is lightweight and can be suitably modified to work on any cell phone with java technology. We have tested our application for audio/video and image files and got the desired results. We have also done elementary error handling, thus protecting our system from crashing down in case of loss of connection to some of the peers/ unrecognized response from a peer or loss of connection to the tracker. The aim was to come up with a robust system which can handle the mistakes made by a novice user and also address connection issues which are typical for mobile clients.

REFERENCES:

1. K.-J. Peng and Z. Tsai, "Distortion and Cost Controlled Video Streaming in a Heterogeneous Wireless Network Environment," Proc. IEEE Int'l Symp. Personal, Indoor and Mobile Radio Comm., Sept. 2006.
2. Herbert Schmidt, "Java 2: The Complete Reference, Fifth Edition
3. www.bitcommet.com

IJARETS