

Usage:

Our programming language is Java in which we can use the udp/tcp package and the sha1 hash function easily.

Simply run make in the command line, and a jar executable file named bittorrent-client.jar file will be generated in the directory. The directory has many useful resources you can use to test our implantation.

Below is the main structure of our project. And some statements and explanations will be attach.

| - src

| -|--main->his is where we coding

| -|--bencoding->third-party library we use to encode and decode the .torrent file information

| -test.torrent-> the torrent file we use in our experiment.

| -01 MiGente.mp3->the resources we downloaded using the test.torrent and our implementation.

Command:

Download using torrent file:

java -jar bittorrent-client.jar test.torrent

Start a server using 2333 port:

java -jar bittorrent-client.jar test.torrent server 2333 "01 Mi Gente.mp3"

Start a client and establish tcp connection with server:

java -jar bittorrent-client.jar test.torrent client 2333

You can also just type **java -jar bittorrent-client.jar**, the you can see some hint about the usage.

1) Supported features

All the core features and **the support for udp-tracker protocol** in extra credit.

2) Design and implementation choices

Our torrent file choosing.

Before we start, we want to get better result when we use the protocol, so we made some choices below:

- a) We choose some hottest torrent file to do the experiment because we can get a bunch of peers. After carefully choosing, we eventually selected two torrent files. Because many site doesn't offer torrent files instead just a magnet url, we then copy the magnet url and make torrent files use the function of the bittorrent application on our local machine.
- b) We close the bittorrent DHT function.
- c) After we did a) and b) we get download rate at 200kb/s and always get 4~5 peers per download.

After we started coding, we found there are so many problem use the torrent file which

we downloaded on the Internet. The worst problem is the connection to the tcp. We find the truth when we do our experiment: all tcp connection encountered timeout but the same tcp connection work well in Bittorrent application.

```

=====>>>>End Connect To Tracker=====>>>>
=====>>>>Start TCP To Peers=====>>>>
TCP=>/45.41.132.133:14891=>start establish tcp
TCP=>/77.229.99.66:18359=>start establish tcp
TCP=>/139.180.216.203:2334=>start establish tcp
TCP=>/41.244.243.216:22595=>start establish tcp
TCP=>/186.220.89.38:32194=>start establish tcp
TCP=>/141.85.0.123:21689=>start establish tcp
TCP=>/92.58.61.122:7703=>start establish tcp
TCP=>/121.7.41.47:29240=>start establish tcp
TCP=>/139.180.216.203:16314=>start establish tcp
TCP=>/37.116.201.166:28353=>start establish tcp
TCP=>/149.28.149.75:16372=>start establish tcp
TCP=>/184.173.25.79:1337=>start establish tcp
TCP=>/101.177.23.162:4109=>start establish tcp
TCP=>/81.84.86.70:28848=>start establish tcp
TCP=>/139.180.216.203:16314=>ERROR: can't get the socket use the ip and port=>Connection refused: connect
TCP=>/139.180.216.203:2334=>ERROR: can't get the socket use the ip and port=>Connection refused: connect
TCP=>/81.84.86.70:28848=>ERROR: can't get the socket use the ip and port=>Connection timed out: connect
TCP=>/149.28.149.75:16372=>ERROR: can't get the socket use the ip and port=>Connection timed out: connect
TCP=>/141.85.0.123:21689=>ERROR: can't get the socket use the ip and port=>Connection timed out: connect
TCP=>/184.173.25.79:1337=>ERROR: can't get the socket use the ip and port=>Connection timed out: connect
TCP=>/92.58.61.122:7703=>ERROR: can't get the socket use the ip and port=>Connection timed out: connect
TCP=>/186.220.89.38:32194=>ERROR: can't get the socket use the ip and port=>Connection timed out: connect
TCP=>/45.41.132.133:14891=>ERROR: can't get the socket use the ip and port=>Connection timed out: connect
TCP=>/37.116.201.166:28353=>ERROR: can't get the socket use the ip and port=>Connection timed out: connect
TCP=>/41.244.243.216:22595=>ERROR: can't get the socket use the ip and port=>Connection timed out: connect
TCP=>/77.229.99.66:18359=>ERROR: can't get the socket use the ip and port=>Connection timed out: connect
TCP=>/101.177.23.162:4109=>ERROR: can't get the socket use the ip and port=>Connection timed out: connect
TCP=>/121.7.41.47:29240=>ERROR: can't get the socket use the ip and port=>Connection timed out: connect
=====>>>>After TCP and first handshake below peer are working fine
=====>>>>End TCP To Peers=====>>>>

```

And for the Bittorrent application.

IP	Client	Flags	%	Down Spe...	Up Speed	Reqs	Uploaded	Download...
149.28.149.75	BitTorrent 7.10.4	D	100.0	112.7 kB/s	0.2 kB/s	28 0		736 kB
81.84.86.70	Vuze 5.7.6.0	D	100.0	29.5 kB/s	0.1 kB/s	9 0		80.0 kB
101.177.23.162	BitTorrent 7.10.4	D	100.0	15.3 kB/s	0.1 kB/s	9 0		80.0 kB
37.116.201.166	µTorrent 3.5.4	D	100.0	12.1 kB/s	0.0 kB/s	3 0		64.0 kB
92.58.61.122	Vuze 5.7.6.0	D	100.0	3.4 kB/s	0.1 kB/s	3 0		16.0 kB

Name	Status	Update In	Seeds	Peers	D
[Peer Exchange]	working		0	3	
udp://tracker.opentracker.org:1337/announce	working	27m 23s	11	3	
[DHT]	disabled		0	0	
[Local Peer Discovery]	disabled		0	0	
udp://tracker.openbittorrent.com:80/announce		updating...	0	0	

We finally solve this problem by deploying another server and make seed on it. So it's better to use our torrent for this experiment test.

Why we use udp mainly to fetch peers' information.

In the beginning, we just implement http protocol to fetch tracker peers. But after many tries, we realize that http nearly response data with no peers. We googled that the http protocol used in p2p cause too much overload on the Internet and the suggest solution is connect server using udp protocol. As said above, we need get as much peers as possible so that the program can get fast down speed. So It's reasonable that we also implement udp protocol.

3) Problems

The most strangest problem we met is the magic number for the `udp_protocol_id(0x41727101980L)`. Even a slight change will cause fail to connect the tracker. Wiki's resources don't figure out its' importance but we finally find this magic number shouldn't be changed when we googled the solution.

For our two torrent file, only one torrent file contains the http tracker. But none of them has the peers data. That is why we add udp protocol implementation.

Our implementation method is very simple, for now just support for a file download and download from a peer. For amelioration, we want to first implement multiple peer download using multi-threading. Now we are working on that but there are still many bugs.

4) Bugs

For now the program aren't support a mutifile torrent, we suggested use our "test.torrent" file in the project root directory to test.