

# Homework 3: bvTorrent

*Due: 11:59 PM, March 11, 2021*

In this exercise you will implement your very own Bit Torrent client that conforms to a Tracker protocol similar to what we designed in class on Tuesday. The tracker software has been provided for you and it can be found in the following location:

```
\\CS-Data\Courses\cmsc431\homework\hw_3\bvTorrent-tracker.py
```

Please read carefully over the code in the tracker file. It very specifically lays out protocols used between clients and the tracker. With this information you are to implement a fully functioning client.

Note: This will be a group assignment working in groups of 2. When submitting the assignment to CS\_Data, please include a file (partners.txt) that identifies who you worked with on this project.

## bvTorrent-client (100 points)

To implement the bvTorrent client will require fully investigating the provided bvTorrent-tracker file. The tracker identifies a handful of protocols for which you must implement the client-side functionality in bvTorrent-client. These protocols include:

1. “New connection protocol” - In which file/chunk information is sent to a newly connected client and the client informs the tracker of: 1) which port it will listen on to receive incoming client connections; and 2) it's current chunk mask (denoting which chunks it has).
2. “Update mask protocol” - In which a client informs the tracker of which chunks it has fully finished downloading.
3. “Client list request protocol” - In which the client requests a newly updated list of all clients participating in the swarm and their chunk masks. With this information, a client will be able to identify who it should download chunks from.

Please note that your bvTorrent client will need to support simultaneous connections. View my code in bvTorrent-tracker to see hints of how to do this (e.g., threading). Ultimately, your client will need to establish and maintain a single connection with the tracker and then it will also start establishing connections with other peers in the swarm (client list). Not only will it establish new connections with other peers, it will be listening for connections from other peers as they reach out and try to download chunks from your client.

## Client-to-Client Communications

It will also be necessary for your client to implement a client-to-client protocol in which it can establish socket connections between clients. Such connections will be established when a client wants to download a chunk from another client. One socket should be established per transferred chunk. When the chunk is fully transferred then the socket should be closed. If the client wishes to download another chunk, it will simply open a new socket connection and start the process again.

The protocol will be as follows:

1. The client requesting a chunk will establish a connection with a client that is known to have the chunk.
2. The requesting client will SEND a single integer in the form of an ASCII string that is terminated by a new line. This integer represents the id (0-based index) of the chunk they wish to receive.
3. The client that possesses the chunk will then send back a byte array containing exactly the number of bytes that are in the chunk (no more – no newline termination).

The receiving client will then produce a digest for the recently downloaded data and confirm that it matches the corresponding digest obtained from the tracker. Note: revisit the bvTorrent-tracker file to see the 1-line of code that can be used to generate this digest. The same code can be used for the purpose of digest verification.

### Implementation suggestions and notes

In order for you to test your client, you will need to connect to a tracker AND connect to other clients from which you can download chunks. For testing purposes, you may ultimately need to produce a seed client which joins the swarm already containing all of the chunks. When it connects to the tracker it can send to the tracker a full chunk mask identifying that it is in possession of all chunks. Then the client you are implementing can fetch chunks from this client that is seeding the file.

You will ultimately need to make a number of design decision including:

- How many simultaneous connections to other clients should I make?
- How many simultaneous connections from other clients are you willing to support?
- How often will you need to fetch an updated client list from the tracker?
- How often will you need to inform the tracker of the chunks your client has successfully downloaded?

There are many design decisions to make and much flexibility along with it. Please be thoughtful in your design decisions and **WRITE COMMENTS IN YOUR CODE** that explain your design decisions as well as your justification for them.

### Submission: Turning in the assignment

After completing all of the exercises in this homework assignment, you should have the following files/folders saved in your CS-Data folder:

```
\\CS-Data\Students\your_user_name\cmssc431\hw_3\bvTorrent-client.py  
\\CS-Data\Students\your_user_name\cmssc431\hw_3\partners.txt
```

This is what I will be looking for to grade. If you do not have these files/folders, something went wrong – come and contact me.