

# Bittorrentpy

## 1.Protocol

There are 4 types of messages in the **Tracker protocol**. Each of them is implemented in Tracker class. You can check the class for details.

...

```
class MessageTypes(Enum):  
    GET_FILE_LIST = 1  
    GET_CHUNK_LOCATIONS = 2  
    FILE_REGISTER = 3  
    CHUNK_REGISTER = 4  
...
```

An example communication:

1. Client send GET\_CHUNK\_LOCATIONS message containing the payload "filename"
2. Server gets the metadata for that file and sends the size of the next outgoing message
3. Then the server sends the metadata, a list of chunk hashes and peers that have that chunk.

Others message types follow a similar pattern and the flow can be found in network.py Tracker.handle\_message.

---

There is 1 type of message in the **Peer protocol**.

1. Requester sends filename and chunk\_id.
2. Provider sends the data.

## 2.Structure

The system has 3 modules.

1. Start.py: Gets user input and has the UI logic.
2. Network.py: Handles communication to Peers and the Tracker. Chunk\_size is a constant.
3. Transfer.py: Downloads chunks concurrently. Concurrency\_count is a constant.

Files are downloaded into uploaded from the /files folder.

## 3.What Works

Everything works

## 4. Sample output

Example of integrity check:

```
Welcome Onboard to Bittorrentpy
[?] What would you like to do?: Join as tracker
  Assign a tracker
  > Join as tracker

[?] What would you like to do?: Download a file
  > Download a file
  Upload a file
[?] What would you like to do?: Upload a file
  Download a file
  > Upload a file
  See current status
  Exit the program

[?] Which file would you like to upload?: newfile
  > newfile
  Go back

[?] What would you like to do?: Exit the program
  Download a file
  Upload a file
  See current status
  > Exit the program

Requesting: {'filename': 'newfile', 'chunk_id': 1} from 130.203
.16.32
Requesting: {'filename': 'newfile', 'chunk_id': 5} from 130.203
.16.32
Requesting: {'filename': 'newfile', 'chunk_id': 6} from 130.203
.16.32
Requesting: {'filename': 'newfile', 'chunk_id': 7} from 130.203
.16.32
Requesting: {'filename': 'newfile', 'chunk_id': 8} from 130.203
.16.30
Requesting: {'filename': 'newfile', 'chunk_id': 9} from 130.203
.16.30
Incorrect chunk received, chunk: 9 downloading it from another
peer, calc hash: f49ba523d07e73a2e6640b1ce62527db, original has
h: 682b3d081a6623dfd048a5a5649ac48d
Requesting: {'filename': 'newfile', 'chunk_id': 9} from 130.203
.16.32
Incorrect chunk received, chunk: 8 downloading it from another
[?] What would you like to do?: Exit the program
  Download a file
  Upload a file
  See current status
  > Exit the program

.16.32
Requesting: {'filename': 'newfile', 'chunk_id': 95} from 130.203
.16.32
Requesting: {'filename': 'newfile', 'chunk_id': 96} from 130.203
.16.32
Requesting: {'filename': 'newfile', 'chunk_id': 97} from 130.203
.16.32
Requesting: {'filename': 'newfile', 'chunk_id': 98} from 130.203
.16.32
Requesting: {'filename': 'newfile', 'chunk_id': 99} from 130.203
.16.32
Requesting: {'filename': 'newfile', 'chunk_id': 100} from 130.20
3.16.32
Requesting: {'filename': 'newfile', 'chunk_id': 101} from 130.20
3.16.32
Requesting: {'filename': 'newfile', 'chunk_id': 102} from 130.20
3.16.32
[?] What would you like to do?: Exit the program
  Download a file
  Upload a file
  See current status
  > Exit the program

Incorrect chunk received, chunk: 17 downloading it from another
peer, calc hash: 95e6149c11482113b877586f3e37bb64, original has
h: 0c8f81c3a98ca234b60e7d5fb69cc54a
Incorrect chunk received, chunk: 15 downloading it from another
peer, calc hash: ab8175dfc76a17e739fdf52677c16c5d, original has
h: 547b6172ef65fa5334283e4fe5e27ce9
Requesting: {'filename': 'newfile', 'chunk_id': 17} from 130.20
3.16.30
Requesting: {'filename': 'newfile', 'chunk_id': 15} from 130.20
3.16.32
Incorrect chunk received, chunk: 17 downloading it from another
peer, calc hash: 95e6149c11482113b877586f3e37bb64, original has
h: 0c8f81c3a98ca234b60e7d5fb69cc54a
Requesting: {'filename': 'newfile', 'chunk_id': 17} from 130.20
3.16.30
Incorrect chunk received, chunk: 17 downloading it from another
[?] What would you like to do?: Exit the program
  Download a file
  Upload a file
  See current status
  > Exit the program
```

## 5. Source code

Is provided inside the zip.

## 6. Run

The program is implemented with python3. The requirements are provided inside requirements.txt. To run the program use

```
$ python3 src/start.py
```

To run it with verbose on

```
$ python3 src/start.py v
```