Torrent Project documentation

Torrent protocol

Torrent protocol is a file-sharing protocol used to distribute large amounts of data over the internet. Instead of relying on a single source for the data, a torrent file allows users to download and upload small pieces of the file from multiple sources, creating a distributed network of computers that share the load of distributing the data.

Project breakdown

1. Read from and writing to file(chunk - wise)
2. Hashing piece
3. Marshalling the struct to bencode
4. Unmarshalling from torrent file to torrentStruct
5. Dialling to seeder
6. Receive signal from leecher

Read from file and write to file

The file to be transferred is only usable in a byte array form. Thus the file is converted to a byte array in the seeder side and the other way round in the leacher side.

Hashing piece

Each piece is hashed for further cross checking on the leacher’s side and if pieces are mismatched the leacher can ask the seeder for that piece.

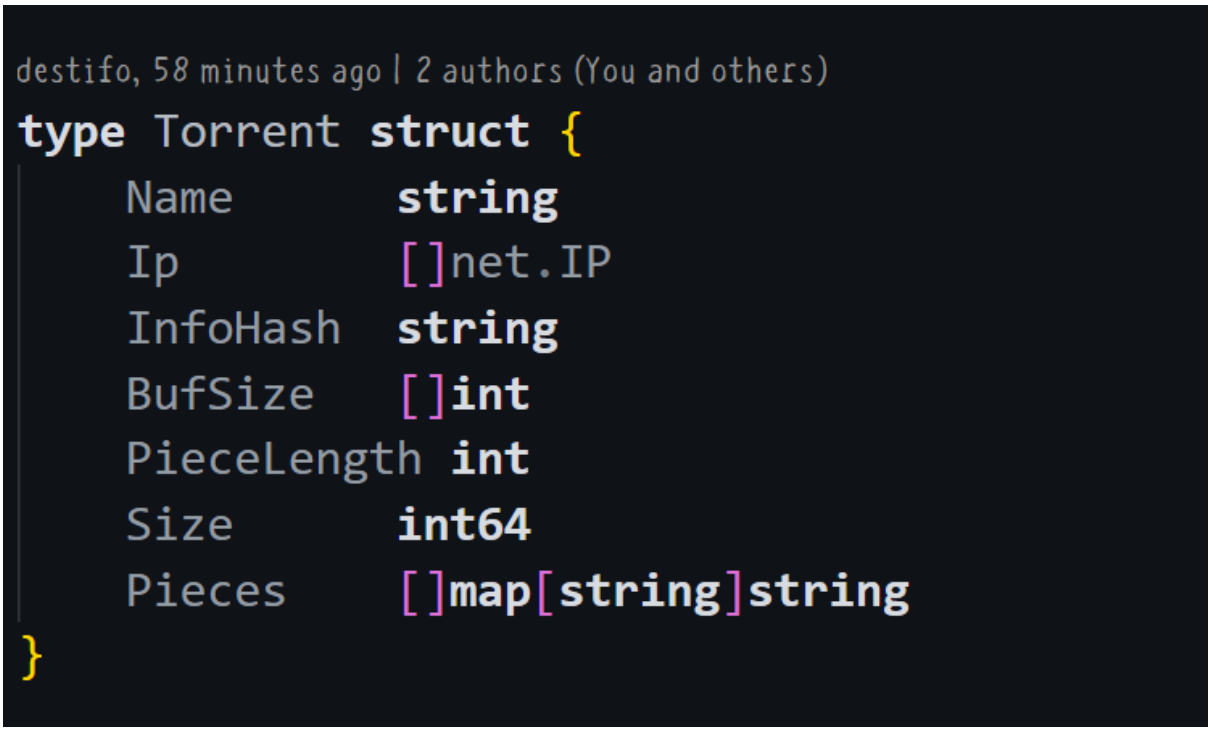
```go
package main      You, yesterday • initial commit …

import (
    "crypto/sha1"
    "encoding/base64"
)

func GeneratePieceHash(piece []byte) (hash string){
    hasher := sha1.New()
    hasher.Write(piece)
    result := base64.URLEncoding.EncodeToString(hasher.Sum(nil))
    return result
}
```

Marshalling the struct to bencode

Marshalling in a Torrent file refers to the process of creating a metadata file that contains information about the data that is being shared via a BitTorrent network. The metadata file is usually a small file with a .torrent extension, which is used by BitTorrent clients to download and upload the shared data.
Marshalling involves gathering all the necessary information about the shared data, such as the file names, file sizes, and the hashes of each file, and packaging it into the metadata file. The metadata file also contains information about the tracker, which is a central server that coordinates the sharing of the data between peers on the network.
Once the metadata file is created, it is shared with other users on the network who can use it to connect to the tracker and start downloading the shared data. BitTorrent clients can read the metadata file to obtain information about the files being shared and to verify the integrity of the downloaded data by comparing the file hashes with those in the metadata file.

Torrent Project documentation

For our project our torrent struct has the following structure



Dialling to seeder

Leecher tries to connect with the seeder igniting the data transfer. The data transfer is done in a chunk size. Retry limit is set inside the code so that if receiver