#ASLI ENGINEERING

Understanding the Torrent File

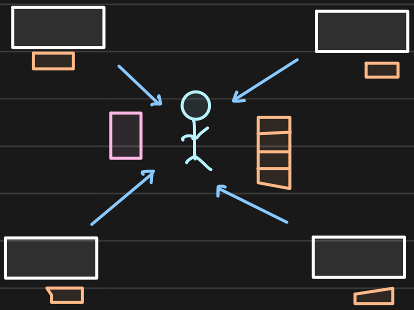
SWIPE

BY

ARPIT BHAYANI

The Torrent File

The Torrent file is the session of transfer of a single content to the set of peers

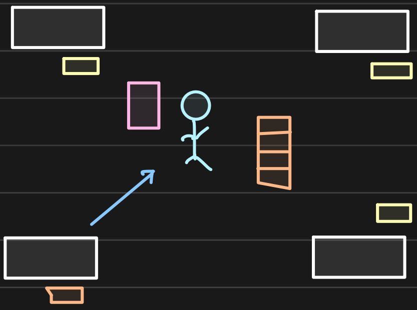


* Each torrent is independent

For a user to download anything from the network, it needs a Torrent file of the content.

Lifecycle of a torrent

Torrent is alive as long as there is at least one seeder



There is no incentive for anyone to join a torrent and become a seeder.

A user downloads the torrent file from wesites via normal HTTP req.

User user the torrent file and a client to download the file and upon completion, it can discard the torrent file.

ARPIT BHAYANI

What torrent file holds?

The .torrent file holds meta information about the content, like

1. announce : the announce URL of tracker
2. created by : name and version of program who created it
3. creation date : creation time of torrent in UNIX epoch
4. encoding : encoding of strings as part of 'info' dictionary
5. comment : some additional comment about author / content
6. info : a dictionary that describes file(s) of the torrent
    1. single file format
        - name : filenome of content
        - length : file size in bytes
        - md5sum : md5 of the file
    2. multi-file format
        - name : name of the directory
        - files : list of dictionaries, one for each file
            - length : length of the file
            - md5sum : MD5sum of the file
            - path : list of string representing the path of the file

              /a/b/c.txt → [a, b, c.txt]

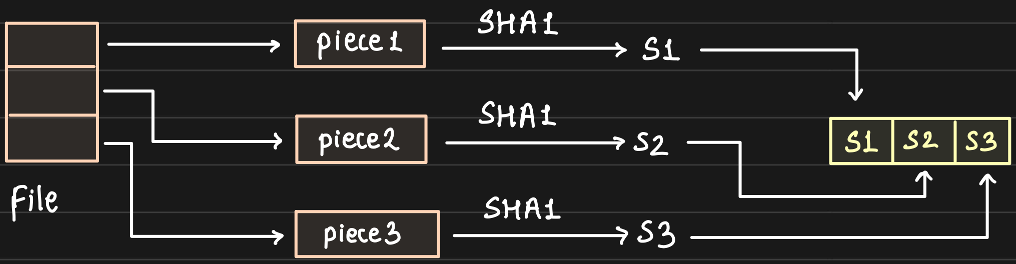
But where is the information about the file data?

It is also stored in the 'info' dictionary

The 'info' dictionary also contains

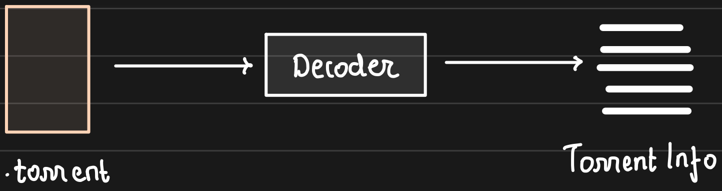
1 piece length : number of bytes in each piece

2. pieces : 20 byte SHA1 hash value concatenated



Torrent Fileformat - Bencoding

Torrent files are 'Bencoded' and to extract the above fields we would need to parse the torrent file (Bencoded)



ARPIT BHAYANI

Bencoding Specification

Bencoding supports: strings, lists, integers, and dictionaries

1. Strings

Format. <length>:<string>

Example: arpit → 5:arpit

2 Integers

Format: i<integer>e

Example: 10 → i10e

3. list

Format: l<bencoded values>e

Example: ["a", "b", 1] → $l\ \underbrace{1:a}_{\text{bencoded String}}\ 1:b\ \underbrace{i1e}_{\text{bencoded integer}}\ e$

4. Dictionary

Format: d<bencoded string><bencoded value>...e

Example: { "a": 1 , "b": 2 } →

$d\underbrace{1:a}_{k}\underbrace{i1e}_{v}\underbrace{1:b}_{k}\underbrace{i2e}_{v}e$

k ↙ bencoded str   v ↘ bencoded value

* it is very fun to write your Bencoding parser