



#ASLI ENGINEERING

Introduction to BitTorrent



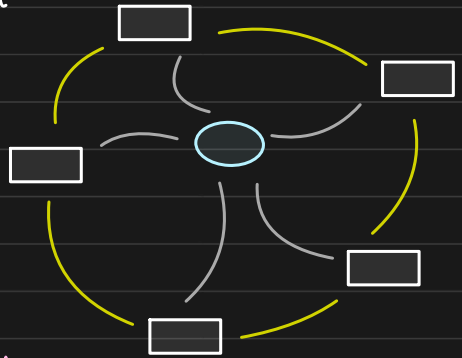
BY

ARPIT BHAYANI

Introduction to BitTorrent

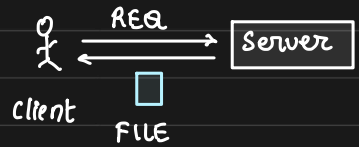
BitTorrent is a **peer-to-peer** protocol that makes distribution of large files

1. easier
2. faster
3. efficient



Classic Download & need of BitTorrent

Client requests for the file from the server, and the server responds. Things become interesting when there are large number of clients or a larger file to download



* Server's bandwidth is limited, so, more clients will slow things down

* Speed of data transfer is limited by the upload capacity

if user B's upload speed is 60mbps, then no matter the download speed of A the overall download speed cannot go beyond 60mbps. Can we do better?



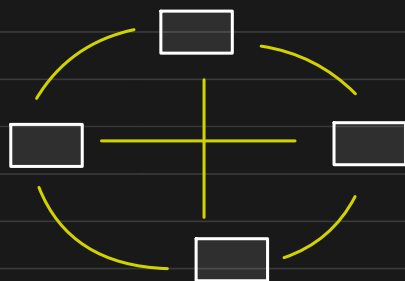
Peer to Peer Networks ^{equal peers}

Each party has the same capabilities, and can initiate conversation with other.

key highlight of P2P: **robustness**

↳ even if you remove one node from the network, there would not be any impact on the service.

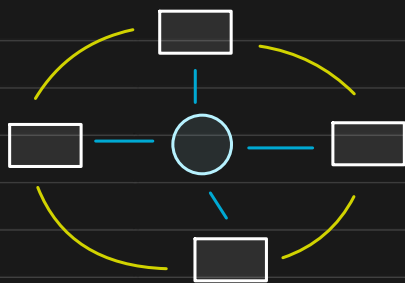
No single point of failure !



There also may be a central entity to provide some functionalities

* The peer nodes are still equal and would still communicate with each other **directly**

But some info can be provided by the central entity.



Note: the network and its services will be affected when the central entity goes down.

Hence, this setup is more vulnerable to failures.

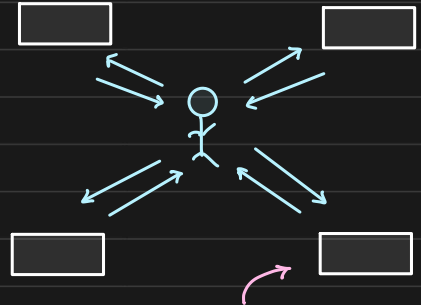
Core idea of BitTorrent

Download the file from multiple machines, concurrently

- ↳ faster downloads
- ↳ upload load is distributed b/w peers
- ↳ better utilization of download capacity

100mbps download

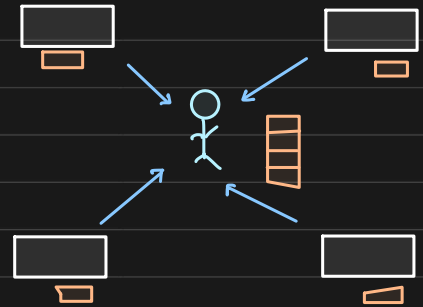
60 Mbps upload



- ↳ large number of downloaders would put only a little extra load
- ↳ breaking file into smaller chunks would boost concurrency

Simplified download flow

When a user wants to download a file, it sniffs around the network to find peers having pieces of it. User then downloads different pieces from different users concurrently → faster download and better utilization of download capacity



Nomenclature and Terminologies

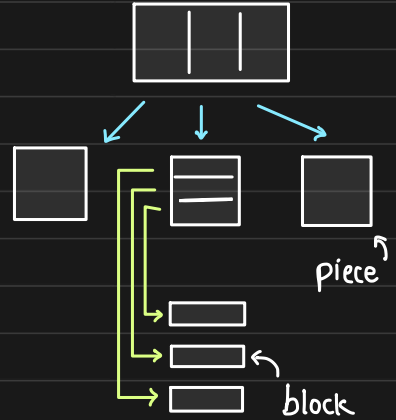
These would come in handy when we deep dive into algorithms

1. Pieces and Blocks

A file that is shared in the BitTorrent network is split into **pieces** and each piece is further split into **blocks**

In one transfer, a **block** is transferred but a **piece** is served by a peer.

* a piece cannot be served if any of the blocks is missing

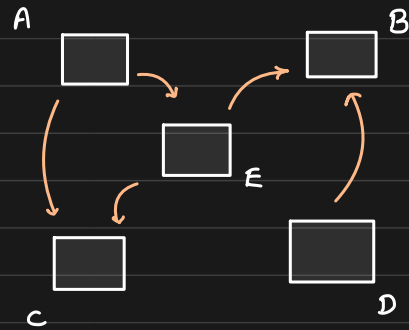


2. Peer Set

Each peer maintains a list of peers that it can send pieces to and this is called its **peer set**

$$\text{peer set}(A) = \{C, E\}$$

$$\text{peer set}(E) = \{A, B, C\}$$

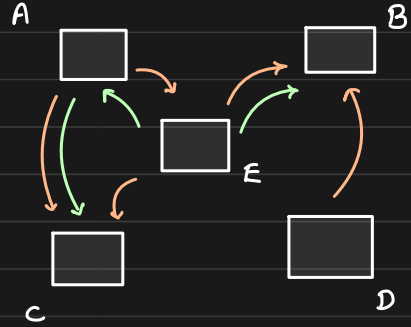


3. Active Peer Set

A peer can only send data to a subset of its peer set and this is called an **active peer set**.

Active peer set (A) = {C}

active peer set (E) = {A, B}

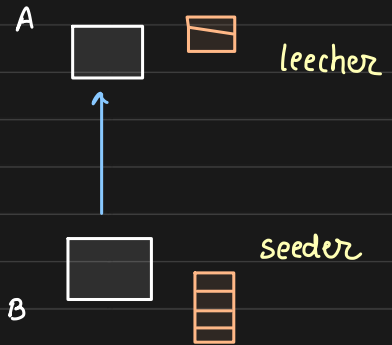


4. Seeders and Leechers

A peer can be a seeder or a leecher

Leecher : when a peer is downloading

Seeder : when a peer has all the pieces of the content



large number of seeders would lead to a faster download speed, as we can pull from multiple seeders quickly.

if leechers \gg seeders, download speed would take a hit.

BitTorrent is popular friendly

The new and popular files will have a lot of seeders, hence it would be downloaded faster. Old or unpopular files will have few seeders, hence a slower download.

Applications of BitTorrent

1. Downloading linux Distributions faster than FTP and HTTP
↳ and large softwares, movies, games, etc
2. Sending patches to users (eg: security patches)
3. Facebook uses this to power their massive deployments
↳ deploying artifacts across servers