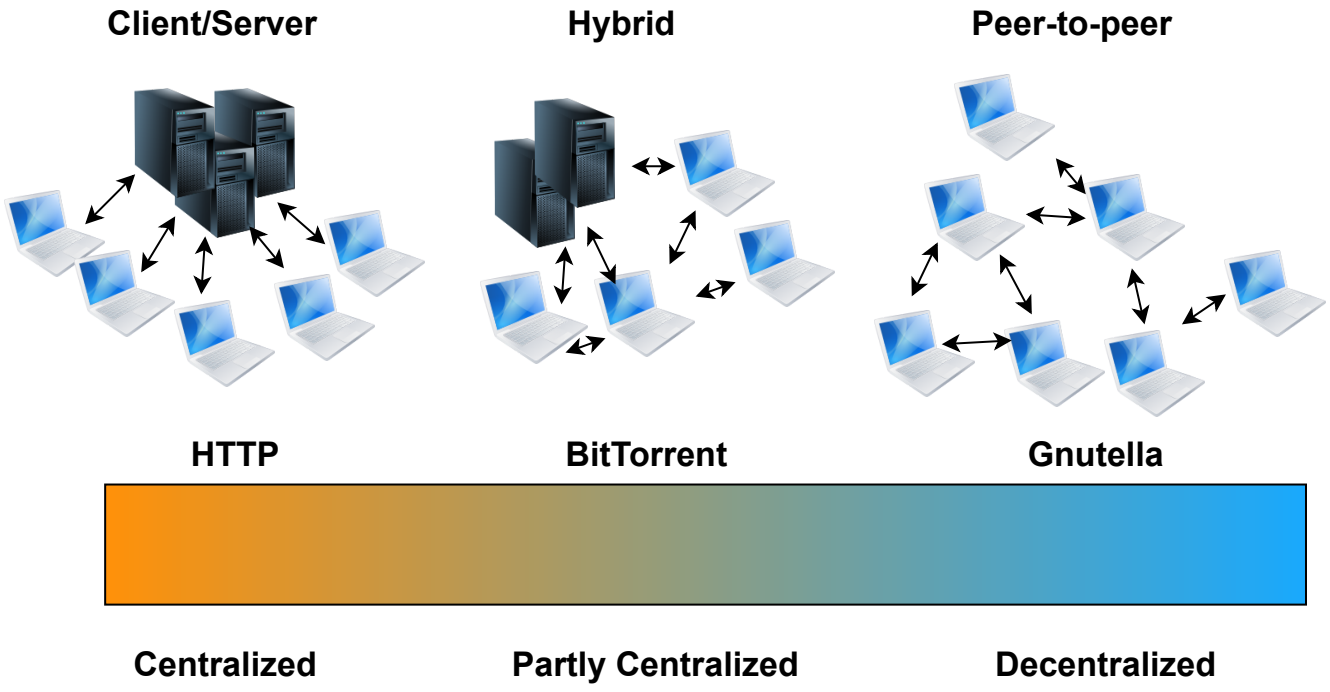
BitTorrent

BitTorrent is a key protocol and has millions of users simultaneously and actively sharing and downloading hundreds of thousands of files of all types: music, movies, books, and so on.

We'll cover the following

We have already had an overview of what the Peer-to-peer and Hybrid architectures are. BitTorrent falls more in the **hybrid** category than pure P2P. Here it is on a spectrum of decentralization:



On a spectrum of decentralization, BitTorrent falls in the middle.

Overview of BitTorrent

Overview of BitTorrent

BitTorrent is a protocol for peer-to-peer file sharing. A **BitTorrent Client** is an application that uses this protocol.

Since BitTorrent is based on a hybrid architecture, it retains some centralized components.

- For example, a **central controller** that maintains a list of participating nodes is involved.
- But the centralized component is not involved in resource-intensive operations. So there will never be too much load on it.
- Data is instead **downloaded or uploaded directly to and by peers**.
- The file is first supplied to a peer in pieces called chunks, and then they also distribute the file to other peers.
- This is sometimes called a **peer-assisted** system.

How It Works

Trackers and torrent files

How do clients find peers to connect to? Well, clients connect to a special tracker node first. The tracker responds with the IP and the port of a few other peers who are downloading the same file.

> **Note**: Modern BitTorrent clients are trackerless and use a Distributed Hash Table instead, but that's beyond the scope of this course.

So clients can find peers through trackers. But how do clients find the tracker in the first place? Clients begin by downloading a 'torrent file' from a web server which has the URL of the tracker. The torrent file also contains a SHA1 hash of each file chunk. Can you guess why?

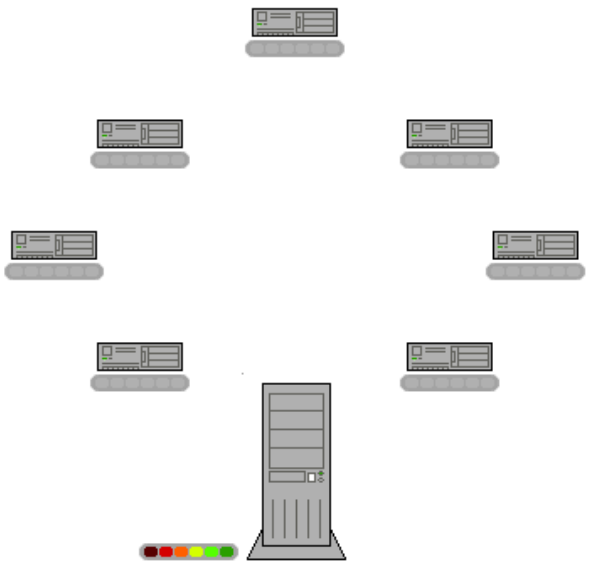
A Simplified BitTorrent Session

1. Download the 'torrent file' from a web server.

2. Connect to the tracker and get a list of peers.
3. Connect to the peers - initially as a ‘leecher.’
4. While the file is not yet fully downloaded:
    - Advertise to peers which blocks are available locally.
    - Request blocks from peers.
    - Compare hash of downloaded blocks to the hash in the torrent file. Can you guess why?
5. Turn into a ‘seeder,’ i.e., continue uploading to peers without downloading.

Swarming

Study the following animation to get an idea of how a file is distributed in chunks. Note that it starts with one peer that has the file in its entirety and then the peers start distributing the file to each other.



Once a peer has the entire file it can choose to leave (selfishly) or choose to stay as a seeder.

The distinct chunks of the file are represented by different colors in the diagram above, as are the bits being transferred. The initial blank gray boxes represent that none of the machines has any bit of the file. Eventually, all the machines have the entire file.

So what this protocol needs really is to give peers the incentive to upload. That is where the **tit-for-tat** scheme comes in.

Tit-for-tat

- Every ten seconds, a peer in the network will calculate which four peers are supplying data at the **highest rate** to it. It will then supply data to them in return. These 4 peers are said to be **unchoked** in the sense that they are now receiving data in return.

> **Note**: This list of top four peers may change every 10 seconds.

- A peer in the network will randomly pick another peer every thirty seconds and supply data to them. The best-case scenario would be that the peer becomes one of the randomly picked peer's top 4 suppliers. Naturally, that random peer would start supplying data in return. Then if the randomly picked peer is sending data at a fast enough rate, it may also become part of the peer's top four suppliers. In other words, two peers partnered randomly will continue working with each other if they are satisfied with the trading. This randomly picked partner is said to be **optimistically unchoked**.

The result of this scheme is that everyone has an incentive to upload. The scheme is an instance of an old successful idea that stems from Axelrod's tournament.

Quiz on BitTorrent

1 What category does BitTorrent fall in?

- A) Cleint/Server
- B) Hybrid

C) Peer-to-peer

COMPLETED 0%

1 of 2

That concludes our study of application layer protocols! Let’s move on to the next layer at last.