P2P vs. Client-Server

Before we move on with the details of BitTorrent, it's useful to do a quantitative comparison of the hybrid architecture with the client-server architecture.

We'll cover the following

Quantitative Comparison of P2P with Client-Server

Let’s calculate how long it will take to transmit a file from one server to a number of clients based on both the P2P and server-client architectures. The calculations will be performed based on the following givens.

- A **server** that can upload at a rate of up_s where up_s is the upload speed in bits/second.
- There are N **clients** all wanting to download the same file from the server. Client i can upload at a rate of up_i bits/second and download at a rate of dwn_i bits/second.
- The size of the file that all the peers want is S.

Client-Server

Let’s start with the **client-server** architecture. The following can be observed.

- Since N clients each want a file of size S, the server will have to upload NS bits. The upload rate of the server is up_s so the server will take at least $\frac{NS}{up_s}$ time to transmit the file to all N clients.
- The client with the lowest download rate ($dwn_{min} = min(dwn_i)$) will take at least $\frac{S}{dwn_{min}}$ time to download the full file.

dwn_{min}

So, in total the time taken to transmit the file will be the maximum of both of the times above, i.e.:

$$\max\left\{\frac{NS}{up_s}, \frac{S}{dwn_{min}}\right\}$$

P2P

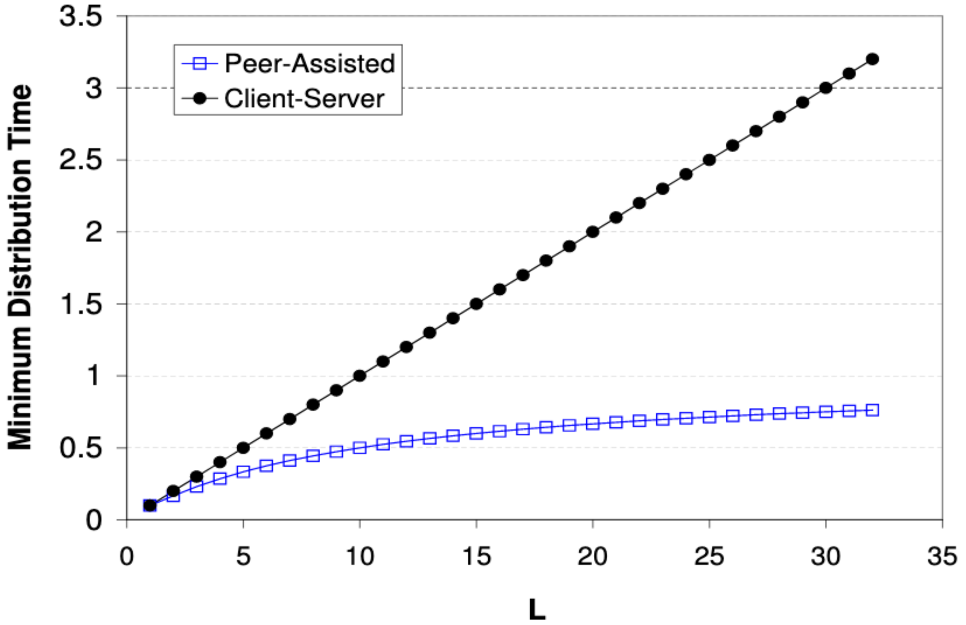
We can make the following observations:

- Initially, only the server has the file. It has to throw the file out into the network and to do that, it will take at least $\frac{S}{up_s}$ time. While the file is being sent out into the network of peers, they start to distribute it amongst themselves.
- The peer with the lowest download rate (dwn_{min}) will take at least $\frac{S}{dwn_{min}}$ time to download the full file.
- The file cannot be transmitted faster than the total upload speed of the entire network: ($up_{sum} = \{up_1 + up_2 + up_3 + ... + up_N\}$). Since the file has to be distributed to all N peers, NS bits have to be transmitted, that will take $\frac{NS}{up_{sum}}$ time.

Therefore, the time taken in total to distribute a file of size S to N peers is:

$$\max\left\{\frac{S}{up_s}, \frac{S}{dwn_{min}}, \frac{NS}{up_{sum}}\right\}$$

Note that as the number of clients/peers, N, grows, the time taken by the client-server architecture also grows. Here is a graph of how the distribution time grows for each architecture as the number of clients/peers grow:



Graph of How p2p Scales vs Client-Server attributed to:
https://pdfs.semanticscholar.org/3de3/1a9b45a3d071c638574117af8e046b578004.pdf

P2P networks are extremely mathematically scalable. The resources of a P2P system grows with the number of peers in the system. Thus, applications with P2P architecture are self-scaling.

Quick Quiz!

1

In a client-server model, the rate at which a client can download a file is limited by the ______.

- A) server upload rate
- B) client download rate
- C) min(a , b)

Let’s now get into how processes communicate!