【CN】Day19

Created	@June 15, 2022 2:16 PM
Class	
Type	
Materials	P2P File Distribution
Reviewed	□

【Ch2】Application Layer

2.5 Peer-to-Peer File Distribution

In this section we consider a P2P application that distributes a large file from a single server to a large number of hosts(called peers).

Scalability of P2P Architectures

Let's assume that in a client-server architecture, the server is sending an F-sized file to N clients.

The upload rate of the server is u_s and the download rate of the slowest client is d_{min}.

Thus, the time it takes to send a file to all clients is $D_{cs} = max\{NF/u_s, F/d_{min}\}$

As the number of clients grow, the distribution time increases linearly.

Let's look at the P2P architecture, where each peer can assist the server in distributing the file.

- At the beginning, only the server has the file. The server must send each bit of the file at least once into its access link. Thus, the minimum distribution time is at least F/u_s.
- The peer with the lowest download rate cannot obtain all F bits in less than F/d_{min} seconds.

- Finally, the total upload capacity is equal to the upload rate of the server plus the upload rates of each of the individual peers. That is, $u_{total} = u_s + u_1 + ... + u_N$.

  The system needs to upload F bits to each of the N peers, thus delivering a total of NF bits.

  Thus, the minimum distribution time is also at least $NF/(u_s + u_1 + ... + u_N)$

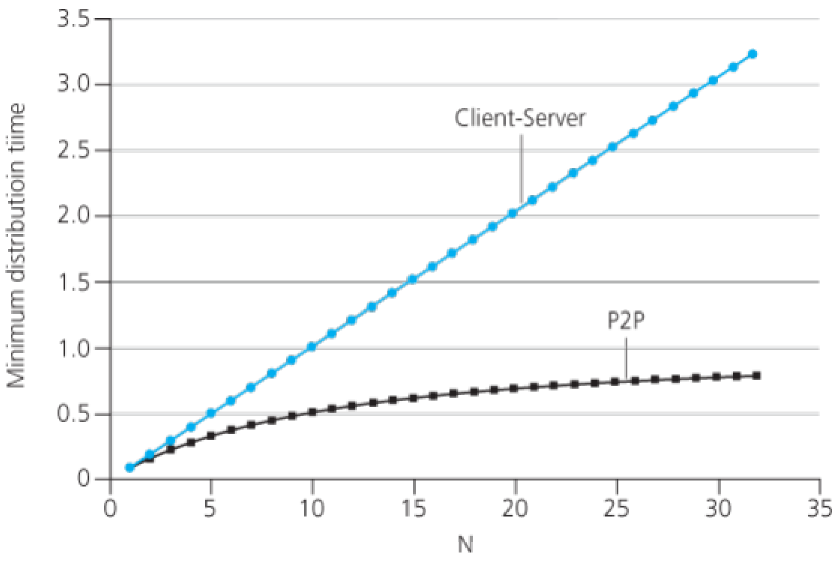

Figure 2.23 Distribution time for P2P and client-server architectures

BitTorrent

BitTorrent is a popular P2P protocol for file distribution.

In BitTorrent lingo, the collection of all peers participating in the distribution of a particular file is called a torrent. Peers in a torrent download equal-size chunks of the file from one another, typically a chunk size of 256 KB.

When a peer first joins a torret, it has no chunks. Over time it accumulates more and more chunks. While it downloads chunks it also uploads chunks to other peers.

When a peer has acquired the entire file, it may leave the torrent, or remain in the torrent and continue to upload chunks to other peers.

Suppose that a peer, Alice, joins the torrent. Alice has two important decisions to make.

First, which chunks should she request from her neighbors? Second, to which of her neighbors should she send requested chunks?

When decising which chunks to request, Alice uses a technique called rarest first. That is, she requests the chunks that are rarest among her nightbors.

To determine which requests she responds to, BitTorret uses a clever trading algorithm.

The basic idea is that Alice gives priority to the neighbors that are currently supplying her data at the highest rate.