CS 456 Hao Pan
Salahuddin, Mohammad
Spring 2018

Chapter 2.5 P2P Applications

07/06/2018 [Th]

2.5.1 Pure P2P Architecture

- There is no always-on server.
- End systems directly communicate with each other.
- Peers are *intermittently* connected and changes IP addresses often.
- **Examples**:
  - File distribution (BitTorrent)
  - Streaming (KanKan)
  - VoIP (Skype)

2.5.2 File Distribution Time

- How does P2P compare with client-server in terms of file distribution of a size F file from one server to N peers:
- Note that peer upload and download capacity is a *limited resource*.
- Note: $u_s =$ server upload capacity.
- In the client-server architecture:
  - Server transmission must sequentially send N file copies.
  - The time to send one copy is F/u_s, so the time to send N copies is NF/u_s.
  - Client must download the file copies.
  - If d_{min} is the minimum client download rate, then the minimum client download time is F/d_{min}.
  - Thus, the time to distribute file F to N clients using the **client-server approach** is: $D_{c-s} \geq max\{NF/u_s, F/d_{min}\}$
- In the P2P architecture:
  - Server transmission must upload at least one copy, which has time F/u_s.
  - Clients must each download the file copy, which has min download time F/d_{min}.
  - Clients download a combined NF bits, meaning the max upload rate... (?)
- See example graph of client-server vs. P2P on slide 2-80.

2.5.3 BitTorrent

An example P2P file distribution service.

- A file is divided into 256Kb chunks.
- Peers in the torrent send and receive file chunks.
- **Swarm** is a group of peers exchanging chunks of a file.
- **Trackers** track peers who are participating in the swarm.
- The process is usually as follows:
  - Peers join the swarm. They have no chunks, but will receive them from other peers over time.
  - New peers register with the tracker to get a *list of peers* and connects to a subset of them (their neighbours).
  - While downloading, the peer will upload chunks to other peers.
  - Peers have the option of *changing the peers* it exchange chunks with.
  - **Churn**: peers will often come and go.
  - Once a peer has the entire file, it could leave, or they could remain in the torrent.

2.5.3.1 Requesting File Chunks

- At any given time, different peers have *different subsets* of file chunks.
- Peer will sometimes ask other peers for list of chunks that they have and request the missing chunks (*rarest first*).

2.5.3.2 Sending Chunks (Tit-for-Tat)

- Send chunks to (4) peers that are currently sending them chunks at the *highest rate*.
- Other peers are *choked* and do not receive chunks from this peer.
- The top (4) peers are re-evaluated after a given amount of time (10s).
- After a period of time (30s) another peer is *randomly selected* to start sending chunks to, thus *unchoking* this peer.
- This newly selected peer joins the top (4).
- See example of unchoking on slide 2-84.