Peer to Peer Applications: BitTorrent

Kameswari Chebrolu

Break

Fetch

- As part of Join, a leecher gets a list of peers who are downloading same file
- Which piece (sub-piece) to request?
  - Piece Selection Algorithm
- From whom to request? And whose request to accept or deny
  - Choking Algorithm

Piece Selection

- Goal: Enable fast download of the entire file
- Challenge: Peers come and go; initial seeder may be taken down
  - Do not want a situation where none of the peers have the missing pieces → Need to ensure small overlap of pieces across peers
- Solution:
  - General Rule: Rarest First
  - At beginning: Random First Piece
  - At end: Endgame Mode

- Rarest first: request piece that is owned by least number of peers
    - Initial Seed can get more information out
    - Replicates rarest pieces as quickly as possible
- Random first piece: Do it at beginning
    - Helps assemble first piece fast so that upload can begin
    - Switch to rarest first after this

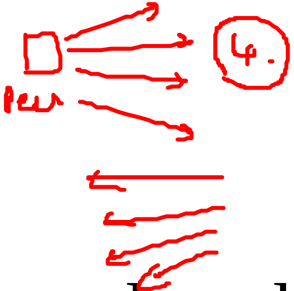
- End-game mode: request sub-pieces from all peers
  - Sometimes download stalls due to slow download of a piece from a peer with low transfer rate
  - Send requests of all sub-pieces to all peers.
  - Can cancel requests later for downloaded sub-pieces
  - Can waste bandwidth but end-game mode is short

Choking Algorithm

- Goal: Utilize all available resources
- Challenge: Freeloaders (peers who download but not upload)
- Solution: Tit-for-tat
  - Download from whoever than can but upload to via tit-for-tat strategy (i.e upload to peers which upload to ~~them~~ you; choke others)
  - Results in connections actively transferring in both directions
  - Probe new peers for better transfer rates

Choking

- Temporary refusal to upload to a peer
- Peers always unchoke (i.e. upload to) a fixed number of peers (default is 4)



- Who to unchoke?
  - Upload to 3 peers that provide the best download rate; Revised periodically (say every 10 sec)
  - Optimistic unchoke: probe a random peer for better choices (rotate peer every 30 sec)

Anti-snubbing

- A peer is snubbed if choked by all peers it was downloading from

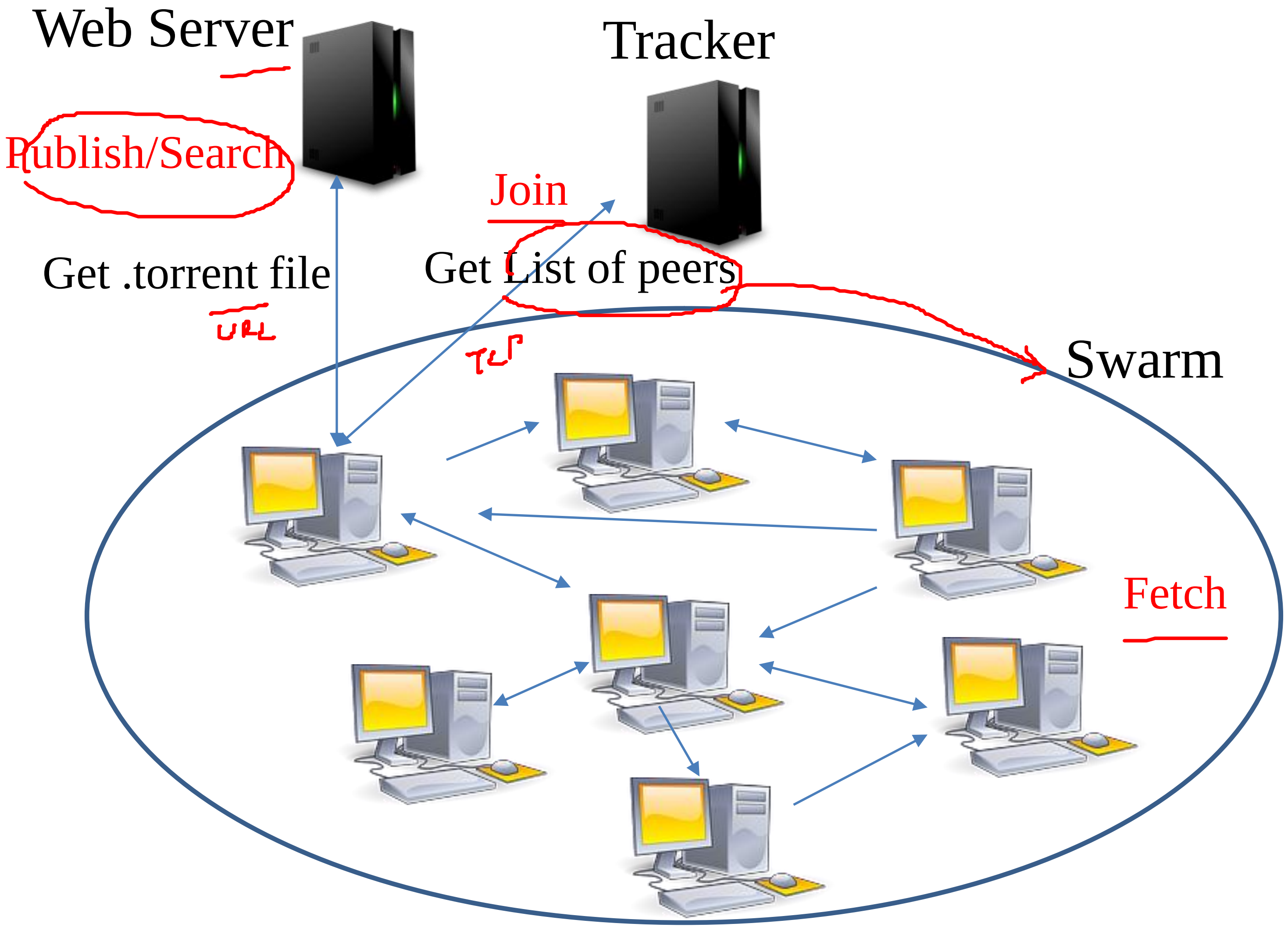
- Solution:
  - Snubbed peer stops uploading to its peers

  - Optimistic unchoking done more often
    - May discover a new peer that will upload to it

Upload Only

List

- After complete download, who to upload to?
- Upload to those with the best upload rate
  - Ensures pieces get replicated faster → more seeders in future



Web Server

Publish/Search

Get .torrent file

URL

Tracker

Join

Get List of peers

TCP

Swarm

Fetch

Discussion

- Pros
  - Encourages peers to share resources, discourages freeloaders
  - Can resume partially downloaded files → chunks
- Cons
  - Works well for “hot” content; not so much for obscure content
    - Performance deteriorates if swarm cools off
    - Search may be difficult
  - Single point of failure (tracker)

Summary

- BitTorrent proposes a novel way to perform file-sharing; suitable when handling flash crowds
- Focus on efficient fetch as opposed to search
- Tackles freeloaders efficiently via a tit-for-tat strategy
- Many mechanisms built in to enable faster downloads (piece selection, anti-snubbing, upload only etc)