

BitTorrent and P2P File-Sharing

Maxwell Zen and Andrew Zhang

COMPSCI 2620: Introduction to Distributed Computing

Motivation

- Digital content constantly growing in size and demand
 - Need for efficient file-sharing
- Traditional networks feature a single server handling many clients
 - Creates bottlenecks and high loads on a single server
 - Single point of failure
- Enter peer-to-peer networks: scalable ways to make large networks
 - Leverage bandwidth of each individual user
 - Highly fault-tolerant

What is P2P?

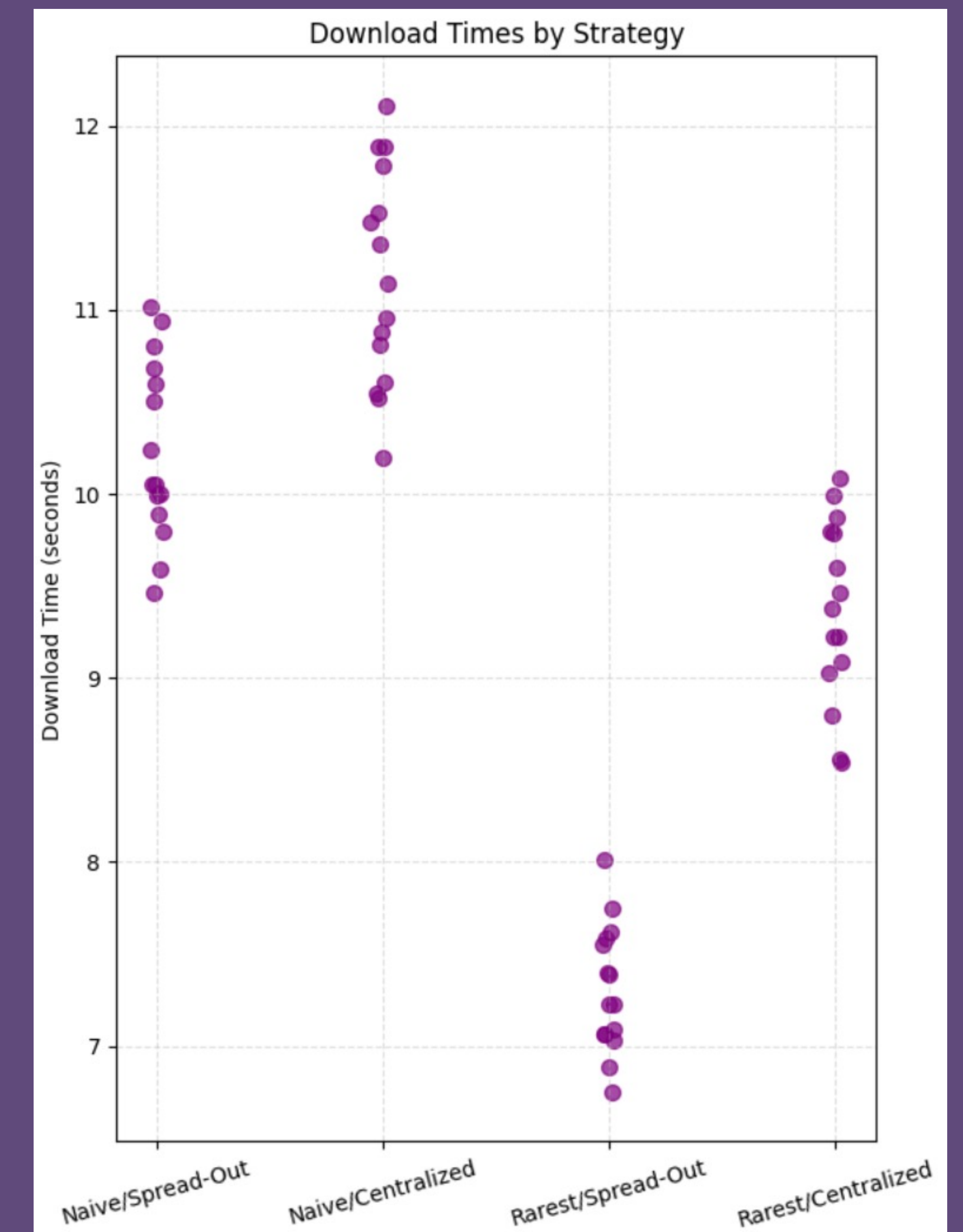
- Peer-to-peer (P2P) networks rely not on a central server but instead on other peers for functionality
 - Peers act as both client and server, downloading parts of file from and uploading other parts of file to other peers in the network
- Key advantage is that network can stay online without need for a central server
 - Much more fault-tolerant
 - No download bottle-necks (at least not due to a central server)

What is BitTorrent?

- One of the most popular and widespread use-cases of P2P networks used for filesharing
 - *Leechers* download files from other peers
 - *Seeders* upload files to other peers
 - *Tracker* serves as centralized way for leechers to find seeders that have the file they want
- Split initial file(s) into many pieces and distribute those pieces to some initial seeders
 - Initial seeders propagate these pieces into wider network
- Some issues:
- Tracker creates single point of failure
 - Distributed hash table (DHT) protocol
- Major free-rider problem
 - Users may choose to leech and leave without seeding to other peers
 - Choking and optimistic unblocking

Piece Selection

- How do leechers determine which pieces to go after first, and which peers to download from?
- Naïve implementation:
 - Have peers simply look for pieces in index order from peers in index order
- Rarest-first algorithm
 - Find rarest piece, if ties choose random
 - Pick random peer that has this piece to download from
- Spread-out means we distributed the pieces among peers first
- Centralized means all pieces started with a single peer which seeded other peers, mimicking an initial server that peers download from before seeding to each other



Future Work

- This poster was made in the wee hours of April 26th, the day this file was to be sent to Lamont for printing—one might call this a file-sharing problem
- By the SEAS Design Fair, we hope to have made some more progress in our implementation of BitTorrent and parts of the more advanced architecture
- Regardless, here are some potential extensions to our work that may or may not be integrated into our project by April 30th:
 - Choking and optimistic unblocking
 - Testing download times of different choking strategies
 - Distributed hash table (DHT)—decentralization of tracker
 - GUI for demo purposes and ease of access
 - More wide-scale testing of download times across a greater number of peers