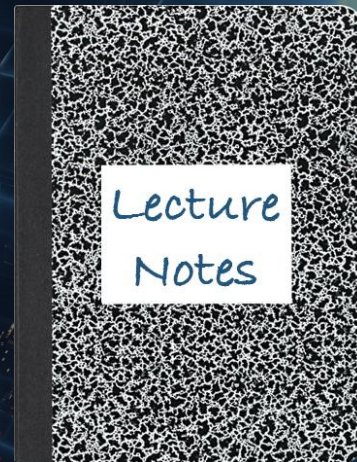


CS 417 – DISTRIBUTED SYSTEMS

Week 11: Content Delivery

Part 3: Peer-to-Peer Content Delivery



Paul Krzyzanowski

© 2022 Paul Krzyzanowski. No part of this content may be reproduced or reposted in whole or in part in any manner without the permission of the copyright owner.

Background

- Introduced in 2002 by Bram Cohen
- Motivation
 - Popular content exhibits temporal locality: *flash crowds*
 - E.g., slashdot effect, new movies, new OS releases, game releases

Big idea: *allow others to download from you while you are downloading*

- Efficient fetching, not searching
- Single publisher, many downloaders

BitTorrent: Publishing & Fetching

To distribute a file

- Create a **.torrent** file, which contains:
 - File name
 - File size, SHA-256 hash of the file
 - Piece size, SHA-256 hash of each piece of the file
 - Address of a tracker server
- Start a **seed node**: a peer that has a copy of the full file
- Start the **tracker** for the file
 - Tracker tracks peers with content and manages uploading & downloading of the content
 - It does not store the contents

BitTorrent: Publishing & Fetching

To get a file

- Get a .torrent file
- Contact the *tracker* named in the file via HTTP (or HTTPS)
 - Download the .torrent file
 - Get the list of seeders and other nodes with pieces of the file
 - Tracker will also announce you to others
- Contact a random node for a list of file piece numbers
- Request a random piece of the file
 - Compare the SHA-256 hash of the piece of the file with the SHA-1 hash in the .torrent
 - During this time, other peers may contact you for pieces of the file
- When all the pieces have been downloaded:
 - Compare the SHA-256 hash of the file with the file's SHA-1 hash in the .torrent
 - Become a seeder

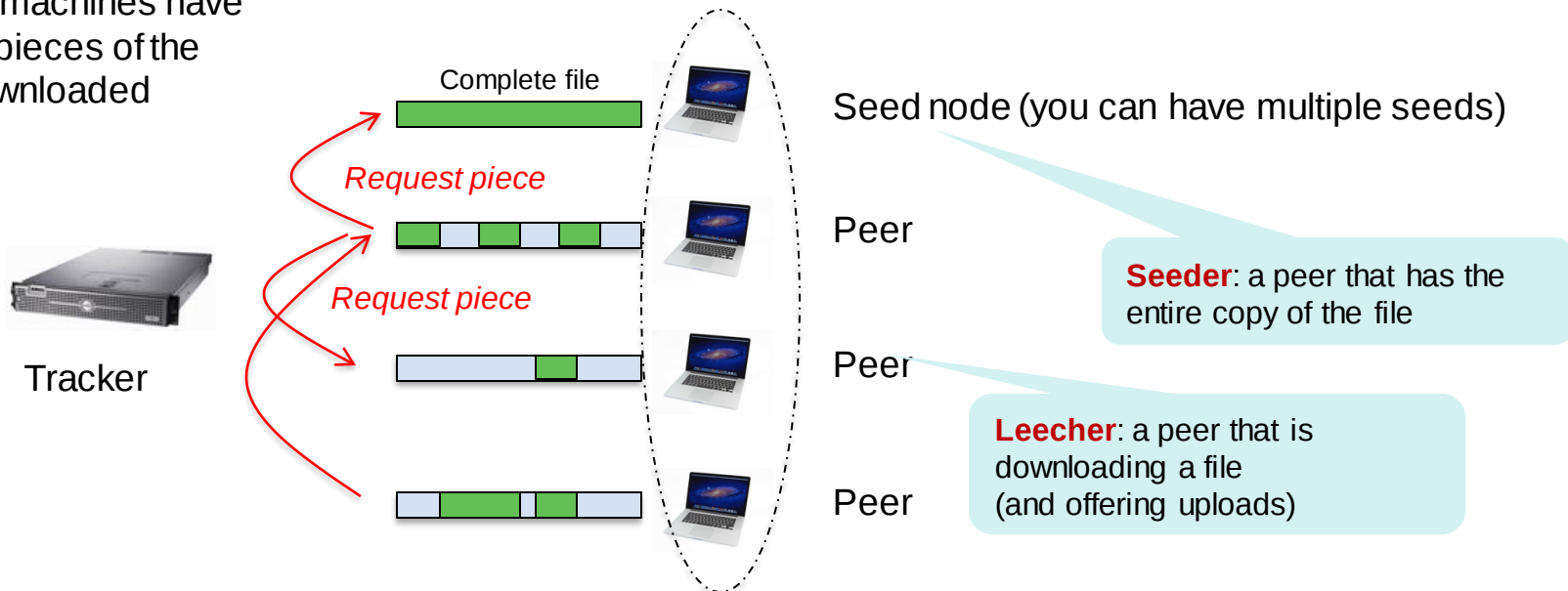
Note: In 2006, BitTorrent added support for using a distributed hash table to support torrents to that don't have a BitTorrent tracker.

Each peer is responsible for a set of .torrent files whose name hashes to a certain range and knows about nearby peers to forward requests for other .torrents.

BitTorrent: Downloading a file in chunks

Tracker identifies:

- (1) System(s) that has 100% of the file – the seed(s)
- (2) Which machines have some pieces of the file downloaded



Swarm: set of peers involved in upload/download for a file

When a peer finished downloading a file, it may become a seed node and remain online without downloading any more content

BitTorrent: Overview

Enable distributed downloads from peers

- **Join**

- No need to join
(seed registers with tracker server; peers register when they download)

- **Publish**

- Create a torrent file; give it to a *tracker server*
- Alternate architecture: use a distributed tracker – DHT among all the peers

- **Search**

- Outside the BitTorrent protocol
- Find the tracker for the file you want, contact it to get a list of peers with files

- **Fetch**

- Download pieces of the file from other peers
- At the same time, other peers may request pieces from you

BitTorrent Summary

- Pros

- Scales well; performs well when many participants
- Gives peers an incentive to share
 - It is sometimes not possible to download without offering to upload

- Cons

- Search is not a part of the protocol; relies on torrent index servers
- Files need to be relatively large for it to work well
- Rare files do not offer distribution & may be slow or difficult to download
- A tracker needs to be running to bootstrap the downloads

The End