Peer-to-Peer: Part I

Theophilus Benson
CS 0138 Spring ‘18
Judge Tells Movie Company That it Can’t Sue Alleged BitTorrent Pirate

BY ANDY  ON JANUARY 18, 2018

A federal judge user that it will
Judge Michael Fathers & Dat under the Uni

Despite a consensus streaming piracy are still finding the world. Alle
number one so tracked. Howe

Top 10 Most Pirated Movies of The Week on BitTorrent – 01/29/18

BY ERNESTO  ON JANUARY 29, 2018

The top 10 most downloaded movies on BitTorrent are in again. 'Thor Ragnarok' tops the chart this week, followed by 'The Shape of Water'. 'Jumanji: Welcome to the Jungle' completes the top three.

This week we have three newcomers in our chart.

Thor Ragnarok is the most downloaded movie.

The data for our weekly download chart is

Screenshots of https://torrentfreak.com/
INDY/TECH

HACKERS INFECT FACEBOOK MESSENGER USERS WITH MALWARE THAT SECRETLY MINES BITCOIN ALTERNATIVE MONERO

screen shot of
CRYPTO MINING IN BROWSERS

- CryptoJacking (Running crypto mining software in the browsers of unsuspecting clients)
  - Legality of cryptojacking is under debate

- Multiple in-browser miner available (coinhive, crypto-loot etc.)

- Not just limited to BitCoin – (Monero, Electroneum etc)

- Can have serious implications on device and web performance

https://gwillem.gitlab.io/2017/11/07/cryptojacking-found-on-2496-stores/
https://steemit.com/cryptocurrency/@fortified/java-script-crypto-browser-mining-or-illegal-or-revolutionary
What is a Peer-to-Peer network?

- Peer-to-Peer —> P2P
- leverage resources at “edge” or clients
  - Resources —> storage, CPU
Centralized Administered Distributed Systems

- Netflix → storage
- Amazon → Compute

CPU/storage is always at a known location, i.e., always at Netflix or Amazon

Client must know how to get to Amazon/Netflix
What is a Peer-to-Peer network?

- Peer-to-Peer —> P2P
- leverage resources at “edge” or clients
- Resources —> storage, CPU

Client must know how to locate and find resources

CPU/Storage could be at any location —> hard to search/find resources
P2P Abstractions

- API Interfaces
  - Join (leave) Nodes
  - Put: Add file (or compute)
  - Get: Find file (or compute results)

P2P systems are often implemented using a distributed hash table (DHT)
- DHT aligns with the Put/Get functionality required by P2P systems
P2P Abstractions

- API Interfaces

  - **Join** (leave) Nodes
  
  - **Add** file (or compute)
  
  - **Find** file (or compute results)

- allocate all work to one node (miner)

- spread work across all nodes (miners)
P2P Challenges

- How do you consistently name content?
- How do you locate (and use) resources?
- How do you provide high performance? (balance load)
- How do you support very large networks?
Consistent Hashing

Put (Theo’s Video)

Get (Theo’s Video)

nope!
Consistent Hashing
Consistent hashing

- How do you coordinate in a stateless manner
  - Name or ID = hash(content of file)
  - use same hash —> always get the same name
  - Put content on “closest server” —> go clockwise to find nearest

Put \((\text{Hash(Theo’s Video)})\)

Get \((\text{Hash(Theo’s Video)})\)
Consistent hash challenges

- Popular content $\rightarrow$ creates imbalance
  - A few servers will be used more frequently than others
- Fault tolerance
  - If you duplicate content $\rightarrow$ same hash
  - Stored on the same node $\rightarrow$ no benefits
P2P Challenges

- How do you consistently name content?
- How do you locate (and use) resources?
- How do you provide high performance? (balance load)
- How do you support very large networks?
Location and routing

• Approach 1: Each node stores a list of all nodes
  • Lookups: very fast $O(1)$
  • Overhead: requires a lot of stores at each node
Location and routing

- Approach 2: Each node stores a list of $L$ nodes. $l/2$ before in chain and $l/2$ after in chain
  - Lookups: a bit slower (N/L)
  - Overhead: requires only $n$ entries.
Can you come up with something better?
P2P in the Wild

- Napster: music sharing program
- Gnutella/BitTorrent: file sharing program
- CryptoCurrency: compute sharing program
Napster: Publishing Content

Hello! I have the following files.
Napster: Publishing Content

Napster Central Server

I have a new file

Publish
Napster: Searching for Content

Where can I find X?

Napster Central Server

look up content

location

fetch

Get (X)

transfer content
Napster subtleties

• P2P WORKS!!!!!!
  • Napster was very simple
  • Easily controllable! (Control central server control everything)
• Limitations: Central server —> single point of failure
  • Server does all the processing
• Lessons:
  • Locality is key: get content from actual closest users
  • Uniqueness
  • No build in replication: People are willing to wait for music
Enter Gnutella —> 
Distribution to the extreme
BitTorrent Overview

- Centralized tracker — maintains list of peers
  - **BUT** there are many different trackers
- Getting a file:
  - First find the right tracker
  - Get list of peer from tracker
  - Download chunks from peers
  - Publishing content == creating a new tracker

- How do you find a tracker?
  - Use google or a different search engine

- Other differences from Napster:
  - Chunk based downloading instead of whole file
  - Napster for music, so naturally smaller files
  - BitTorrent has video and large linux ISOs —> large files
  - Anti-freeloading mechanisms: ‘tit for tat’
Next Class

- P2P Continued
  - Addressing distributed hash table (DHT) limitations
  - Alternative DHT designs
    - Tapestry
    - Chord