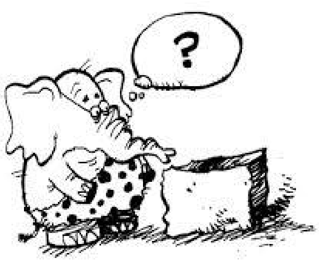
P2P Content Distribution
BitTorrent and Spotify

Amir H. Payberah
amir@sics.se

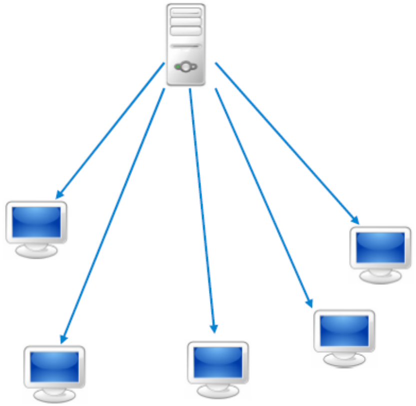
Amirkabir University of Technology
(Tehran Polytechnic)

Possible Solutions for Content Distribution



Client-Server Model

Client-Server Model



P.R.O.B.L.E.M.S.

The Client-Server Model Problems

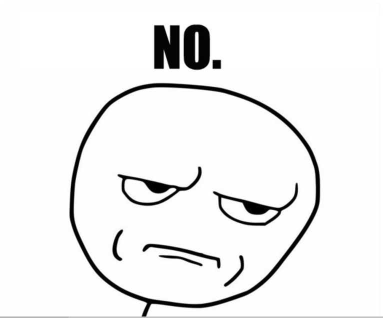
- Scalability?

The Client-Server Model Problems

- Scalability?
- Single Point of failure?

The Client-Server Model Problems

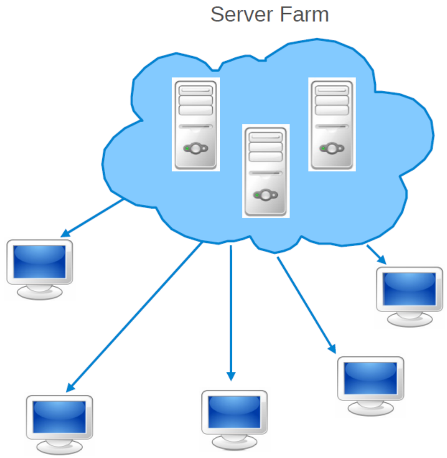
- Scalability?
- Single Point of failure?



Client-Server Systems

The Client-Server Model Problem

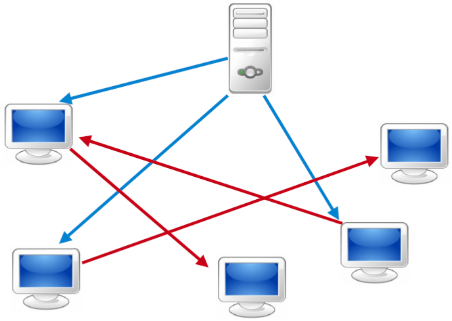
Scalable and Fault-Tolerant Client-Server Model



UNIT
LIBERTAD
свабода
LIBERTY OR DEATH
LIBERTY OR DEATH
UNIT

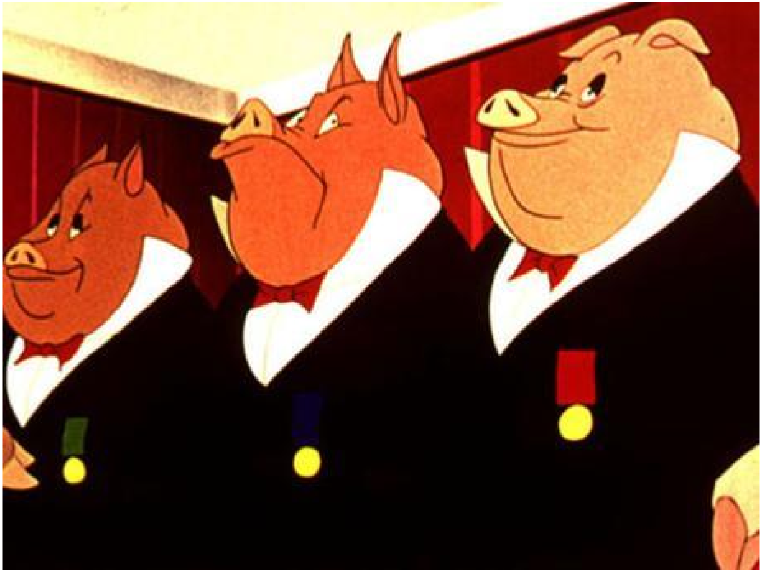
Peer-to-Peer Model

Peer-to-Peer (P2P) Model



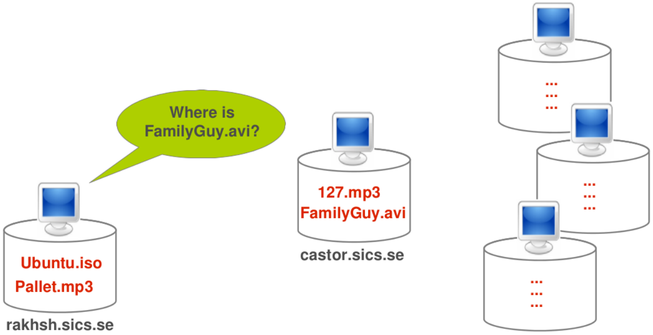
P2P Challenges

- Churn in the system
- Free-riding problem
- Bottleneck in the overlay network
- Connectivity problem, e.g., NAT



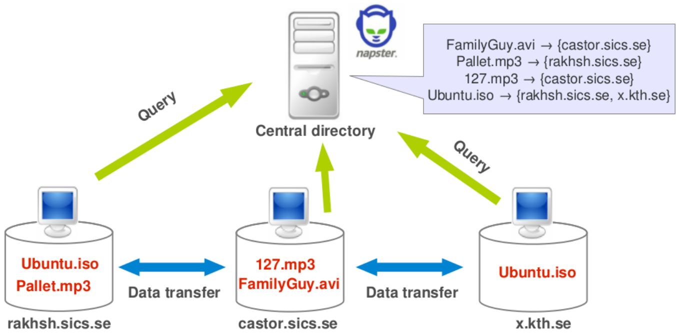
P2P

How To Discover Data?



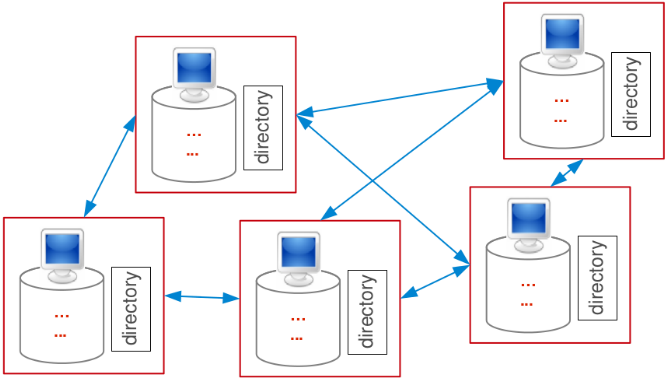
Possible Solutions - First Generation

- Central directory



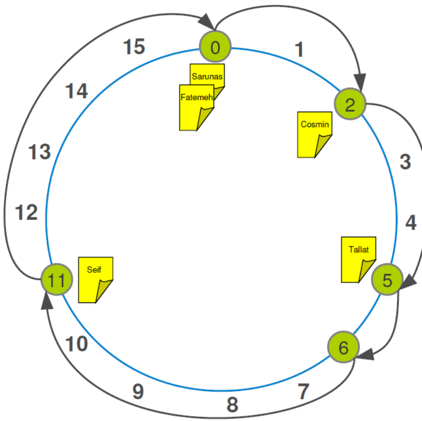
Possible Solutions - Second Generation

- Flooding



Possible Solutions - Third Generation

- Distributed Hash Table (DHT)



P2P Content Distribution Applications - File Sharing

Lime
wire

BitTorrent™

P2P Content Distribution Applications - Media Streaming

Spotify

PPTV

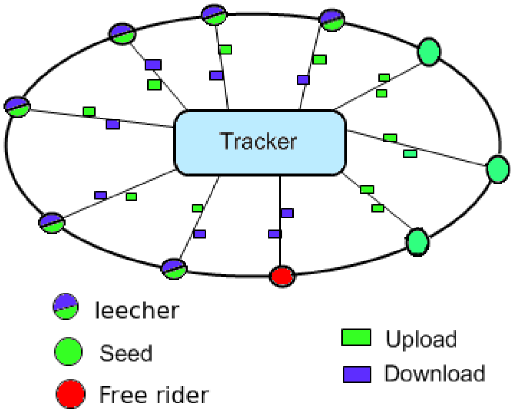
joost

BitTorrent™

BitTorrent

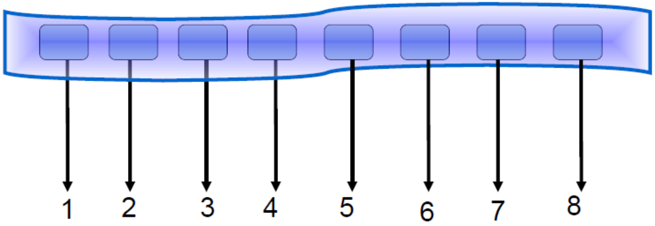
- BitTorrent is a system for efficient and scalable replication of large amounts of static data.

BitTorrent Players



Files

- Files are broken into pieces of size between 64KB and 1MB.



.torrent Files

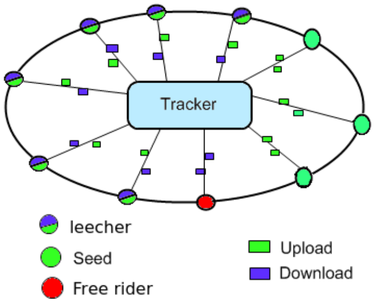
- Metadata
- Contains:
  - URL of tracker
  - Information about the file, e.g., filename, length, ...

The Pirate Bay
Search Torrents | Browse Torrents | Recent Torrents | TV shows | Music | Top 100
Preferences
Languages
All
Audio
Video
Applications
Games
Other
(search titles only)
Pirate Search
I'm Feeling Lucky
How do I download?
Login | Register | Language / Select language | About | Legal threats | Blog
Contact us | Usage policy | Calendar | Doodles | Search Cloud | Tag Cloud | Forum | TPB T-shirts
SlopsBox | BayWords | BayImg | IPREDator

3.499.339 registered users. Last updated 12:50:04.
IPv4 21.370.368 peers (10.310.574 seeders + 11.059.794 leechers) in 1.660.172 torrents on tracker.
IPv6 32.210 peers (15.477 seeders + 16.733 leechers) in 31.800 torrents on tracker.

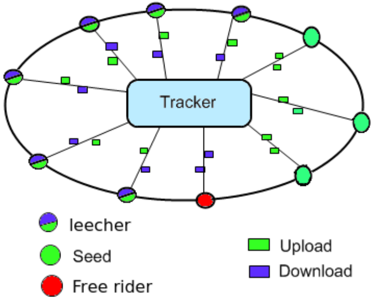
The Core Idea

- A peer obtains .torrent file.



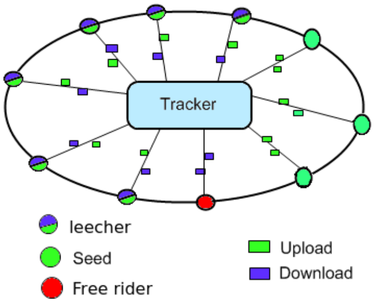
The Core Idea

- A peer obtains .torrent file.
- It, then, connects to the tracker.



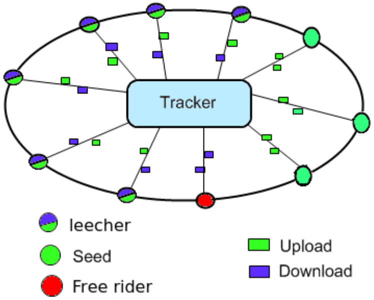
The Core Idea

- A peer obtains .torrent file.
- It, then, connects to the tracker.
- The tracker tells the peers from which other peers to download the pieces of the file.



The Core Idea

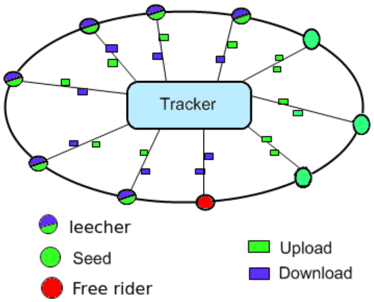
- A peer obtains .torrent file.
- It, then, connects to the tracker.
- The tracker tells the peers from which other peers to download the pieces of the file.



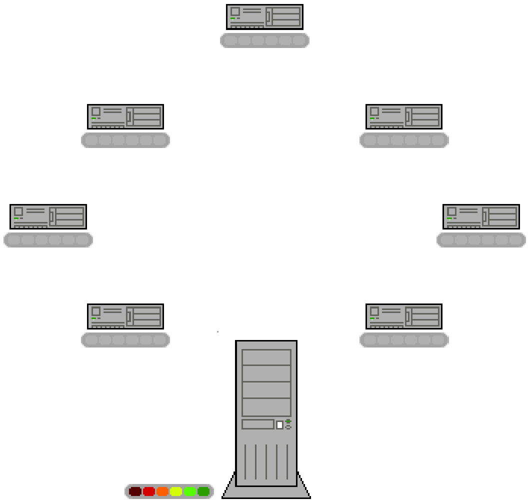
- Peers use this information to communicate with each other.

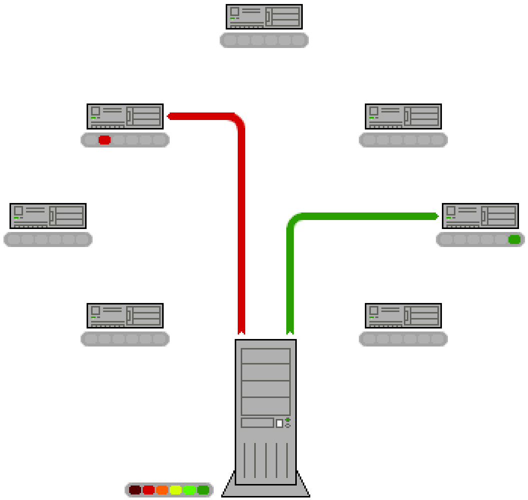
The Core Idea

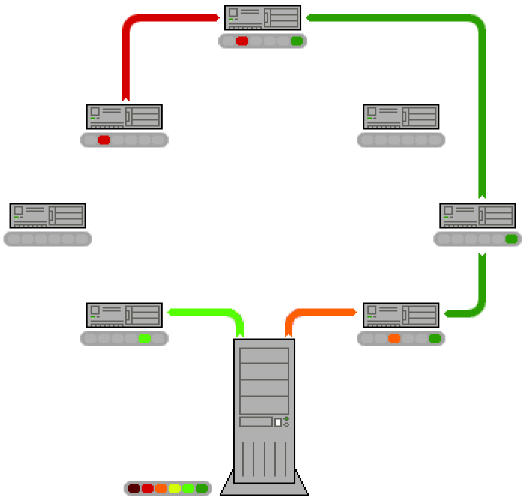
- A peer obtains .torrent file.
- It, then, connects to the tracker.
- The tracker tells the peers from which other peers to download the pieces of the file.

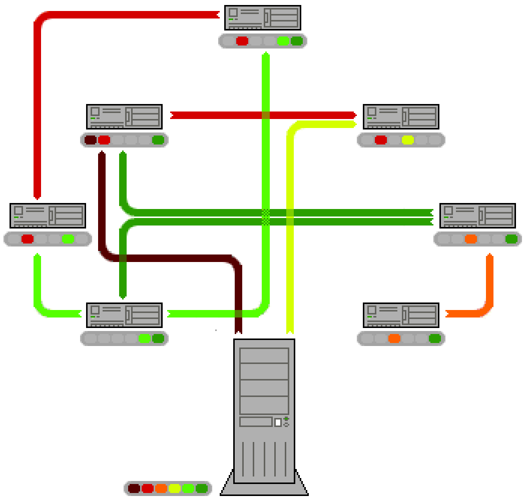


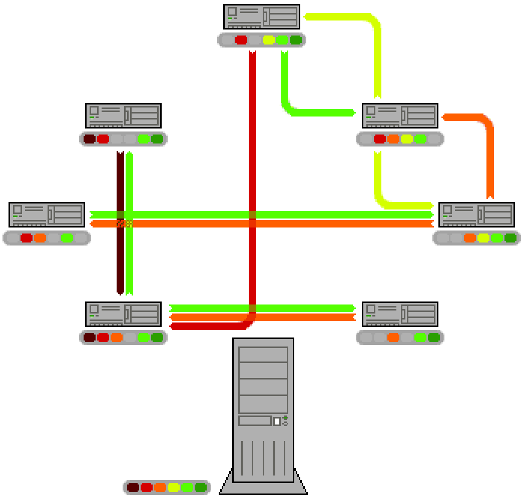
- Peers use this information to communicate with each other.
- The peers send information about the file and themselves to tracker.

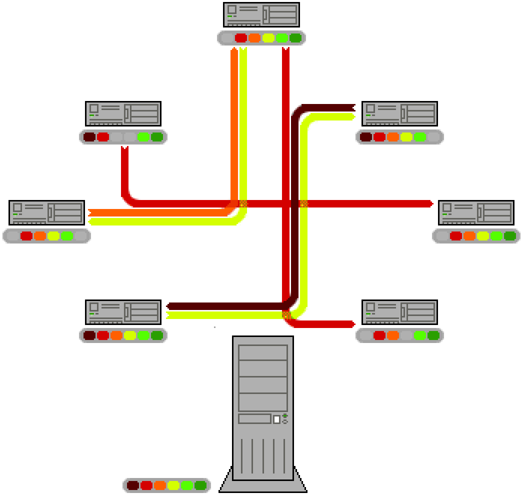


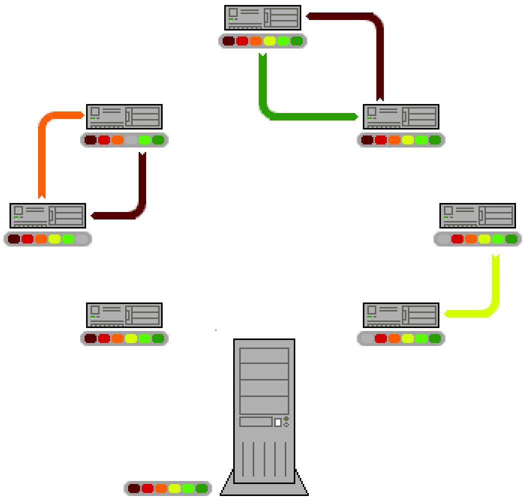


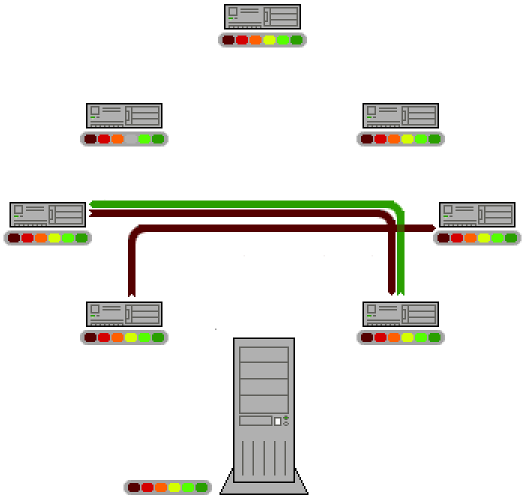


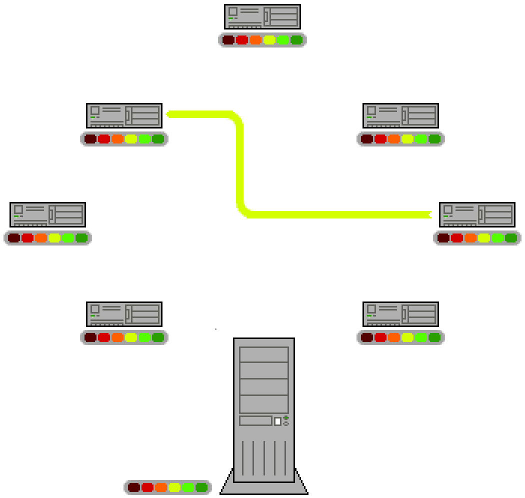


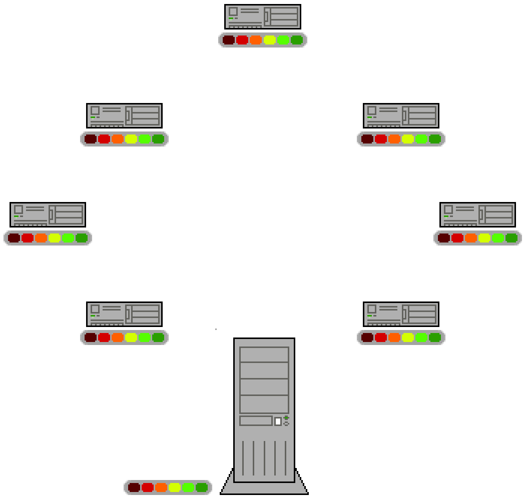












What About Free Riders?

Tit-For-Tat

Question

- From which peers download the pieces?

Peer Selection

- Use choking algorithm to choose peer to download pieces.
- Decision to choke/unchoke based on tit-for-tat.

Discover More Cooperating Peers

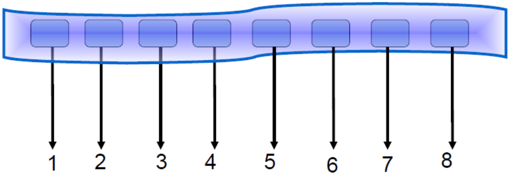
- Optimistic unchoking
- Allocate an upload slot to a randomly chosen uncooperative peer

Snubbed Peers

- If all its peers choke it.
- Increase the number of optimistic unchokes.

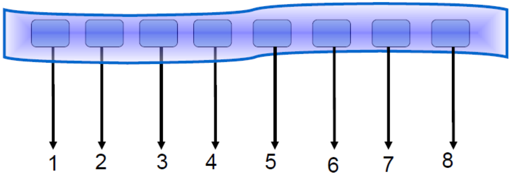
Question

- Which piece?



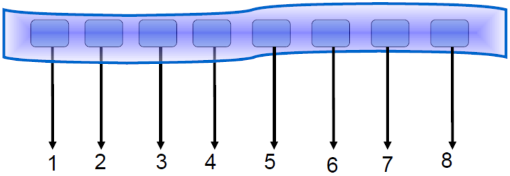
Piece Selection

- **Rarest first**: common parts left for later



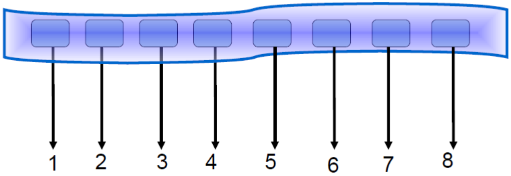
Piece Selection

- Rarest first: common parts left for later
- Random first piece: start-up need to get a complete piece



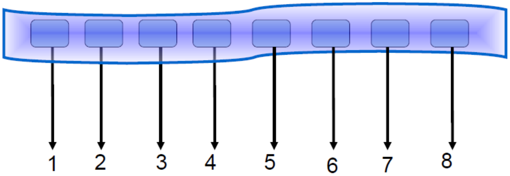
Piece Selection

- Rarest first: common parts left for later
- Random first piece: start-up need to get a complete piece
- Endgame mode: broadcast for all remaining blocks



BitTorrent Extension

- Distributed tracker
- Peer-exchange



Spotify

- Active users: over 50 million
- Number of songs: over 20 million
- Number of songs added per day: over 20000
- Number of playlists: over 1.5 billion created so far
- Available in 58 countries
- Legal

The Core Idea

- Request first piece from Spotify servers.

The Core Idea

- Request first piece from Spotify servers.
- Meanwhile, search P2P network for remainder.

The Core Idea

- Request first piece from Spotify servers.
- Meanwhile, search P2P network for remainder.
- Switch back and forth between Spotify servers and peers as needed.

The Core Idea

- Request first piece from Spotify servers.
- Meanwhile, search P2P network for remainder.
- Switch back and forth between Spotify servers and peers as needed.
- Towards end of a track, start prefetching the next one.

Main Problem in Using Spotify P2P Network

Peer Discovery

Peer Discovery

- Sever-side tracker (BitTorrent style)
  - Only remembers 20 peers per track.
  - Returns 10 (online) peers to client on query.

Peer Discovery

- Sever-side tracker (BitTorrent style)
  - Only remembers 20 peers per track.
  - Returns 10 (online) peers to client on query.

- Broadcast query in small (2 hops) neighborhood in overlay (Gnutella style)

Peer Discovery

- Sever-side tracker (BitTorrent style)
    - Only remembers 20 peers per track.
    - Returns 10 (online) peers to client on query.

- Broadcast query in small (2 hops) neighborhood in overlay (Gnutella style)

- LAN peer discovery

Downloading in P2P

- Ask for most urgent pieces first.

Downloading in P2P

- Ask for most urgent pieces first.
- If a peer is slow, re-request from new peers.

Downloading in P2P

- Ask for most urgent pieces first.
- If a peer is slow, re-request from new peers.
- When buffers are low, download from central server as well.

Downloading in P2P

- Ask for most urgent pieces first.
- If a peer is slow, re-request from new peers.
- When buffers are low, download from central server as well.
- If buffers are very low, stop uploading.

Spotify vs. BitTorrent

- One (well, three) P2P overlay for all tracks (not per-torrent).

Spotify vs. BitTorrent

- One (well, three) P2P overlay for all tracks (not per-torrent).
- Does not inform peers about downloaded blocks.

Spotify vs. BitTorrent

- One (well, three) P2P overlay for all tracks (not per-torrent).
- Does not inform peers about downloaded blocks.
- Downloads blocks in order.

Spotify vs. BitTorrent

- One (well, three) P2P overlay for all tracks (not per-torrent).
- Does not inform peers about downloaded blocks.
- Downloads blocks in order.
- Does not enforce fairness (such as tit-for-tat).

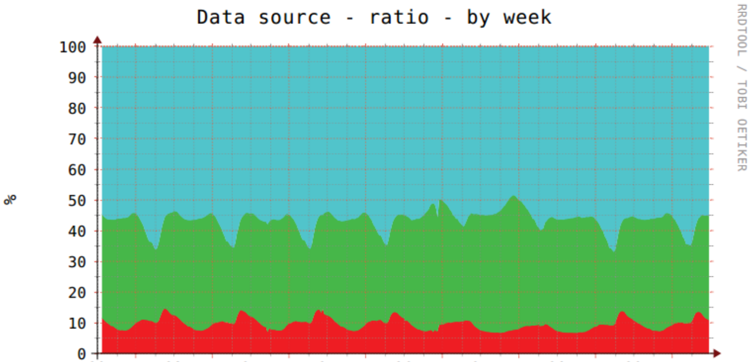
Spotify vs. BitTorrent

- One (well, three) P2P overlay for all tracks (not per-torrent).
- Does not inform peers about downloaded blocks.
- Downloads blocks in order.
- Does not enforce fairness (such as tit-for-tat).
- Informs peers about urgency of request.

Caching

- Player caches tracks it has played.
- Use 10% of free space (capped at 10GB)
- Least Recently Used policy for cache eviction.
- Over 50% of data comes from local cache.

Spotify Data Usage



	Cur:	Min:	Avg:
🟥 Server	10.86	6.76	9.62
🟩 P2P	33.90	23.78	33.86
🟦 Cache	55.24	48.47	56.53

Spotify Says Goodbye to P2P

Summary

Summary

Questions?