

A Glance at Spotify's Peer-to-peer Streaming

Gunnar Kreitz

Spotify
gkreitz@spotify.com

IETF 75, Jul 30 2009



What is Spotify?

- ▶ Lightweight on-demand streaming
- ▶ Large catalogue of music
- ▶ Fast
- ▶ Legal



Business Idea

- ▶ More convenient than piracy



Business Idea

- ▶ More convenient than piracy
- ▶ Free version with ads
 - ▶ Daypass gets rid of ads for 24 hours
- ▶ Premium version
 - ▶ Extra features
 - ▶ No ads



Technical Overview

- ▶ Client software on Mac and Windows
 - ▶ Works well under Wine
- ▶ Proprietary protocol
- ▶ 160-320kbps audio streams
- ▶ Streaming from
 - ▶ Spotify servers
 - ▶ Peers
 - ▶ Content Delivery Network ([http](http://))

Network Protocol

- ▶ Designed for random-access streaming rather than live streaming
- ▶ (Almost) Everything over TCP
- ▶ (Almost) Everything encrypted
- ▶ Multiplex data over a single TCP connection
- ▶ Persistent TCP connection to server while logged in



Latency Kills!

- ▶ Latency kills (not just in First Person Shooters)



Latency Kills!

- ▶ Latency kills (not just in First Person Shooters)
- ▶ Increasing latency of Google searches by 100 to 400ms decreased usage by 0.2% to 0.6% [Brutlag09]
- ▶ The decreased usage persists

Latency Kills!

- ▶ Latency kills (not just in First Person Shooters)
- ▶ Increasing latency of Google searches by 100 to 400ms decreased usage by 0.2% to 0.6% [Brutlag09]
- ▶ The decreased usage persists
- ▶ Most Spotify playbacks start within a few hundred milliseconds



Peer to Peer (P2P) Goals

- ▶ Minimize bandwidth bills
- ▶ Minimize hardware costs
- ▶ Minimize latency
- ▶ Avoid stutter



Playing a Track

- ▶ Request first piece from Spotify servers
- ▶ Meanwhile, search Peer-to-peer (P2P) for remainder
- ▶ Switch back and forth between Spotify servers and peers as needed
- ▶ Towards end of a track, start prefetching next one



P2P Structure

- ▶ Mesh structure, nodes have fixed maximum degree
- ▶ Neighbor eviction by heuristic evaluation of utility
- ▶ No overlay routing
- ▶ A user only downloads data she needs
- ▶ Weakly clustered by interest



Finding Peers

- ▶ Partial central index (Napster-style)
- ▶ Peers have local knowledge (Gnutella-style)
- ▶ Limited broadcast for local peer discovery

P2P NAT Traversal

- ▶ Asks to open ports via UPnP
- ▶ Attempt connections in both directions
- ▶ Rather high connection failure rate
- ▶ Room for improvement



When to Start Playing?

- ▶ Minimize latency while avoiding stutter
- ▶ TCP throughput varies
 - ▶ Sensitive to packet loss
 - ▶ Bandwidth over wireless mediums vary



When to Start Playing?

- ▶ Minimize latency while avoiding stutter
- ▶ TCP throughput varies
 - ▶ Sensitive to packet loss
 - ▶ Bandwidth over wireless mediums vary
- ▶ Model throughput as a Markov chain and simulate
- ▶ Heuristics



Touching Points With Ongoing IETF Work

- ▶ P2P Architectures Taxonomy [draft-iab-p2p-archs-02]
- ▶ P2Pi Workshop '08 [draft-p2pi-cooper-workshop-report-01]
- ▶ PPSP [draft-zhang-ppsp-problem-statement-04]







Challenges Ahead

- ▶ Continued growth
- ▶ Mobile clients
- ▶ Locality-aware peer selection



References

-  **Brutlag**
Speed Matters for Google Web Search
-  **Camarillo, Ed.**
Peer-to-peer (P2P) Architectures
-  **Peterson, Cooper**
Report from the IETF workshop on P2P Infrastructure
-  **Zhang *et al.***
Problem Statement of P2P Streaming Protocol (PPSP)