Spotify-Large Scale, Low Latency, P2P Music-on-Demand Streaming

Gunnar Kreitz, Fredrik Niemela

Presenter: Wenzhou Wang

Spotify

- Streaming service
- Over 8 million tracks
- Free vs premium

Spotify

- Low latency: median 265ms
- Peer-to-peer and client-server
- Proprietary protocol

Spotify Overview

- Client read data from sound card buffers
- If buffers underrun, stutter
- Client only upload track when she has whole track
- TCP: (1)simplifies protocol (2)congestion control (3)resending lost packets

Caching

- Repeatedly listening to same track
- Cached data can be served
- Max size: 10% of free disk space
- Eviction rule: Least Recently Used, but not a large effect on cache efficiency

Prefetching &Random Access

- 61% tracks are played in predictable order
- Prefetching data is efficient
- 39% are random access
- Request data from server for 15 secs of music
- Simultaneously search peer-to-peer network

Regular Streaming

- Connection to server: reliable
- Clients seldom download data from server only if low playback quality or high latency
- “Emergency mode”: if buffers become critically low, pause uploading
- Files in p2p network are split into chunks of 16kB

Play-out Delay

- Non delay-preserving: not drop any frames or slow down play-out rate
- Markov chain: 100 simulations of playback of the track while downloading
- If more than one underrun occurs, waits longer before playback

Peer-to-peer Network

- Goal: improve scalability and reduce load of servers
- Unstructured network with trackers
- Two data centers: London and Stockholm

Locating Peers

- Trackers: 20 most recent peers for each track and reply up to 10 peers
  - tracker knows which clients are online
- Query: search a track within distance of two and remember 50 most recent searches

Neighbor Selection

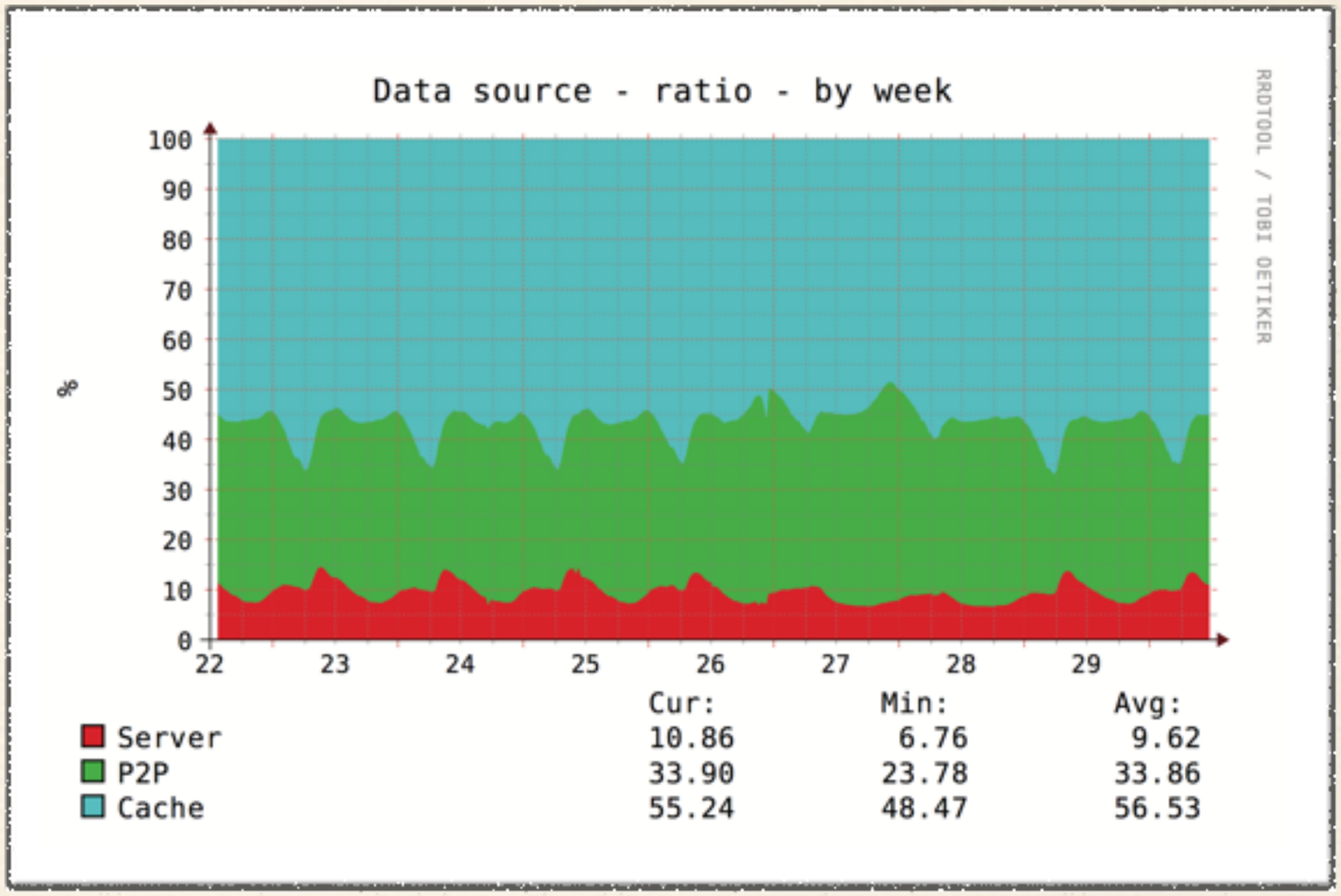
- Keeping TCP connections to peers is expensive
- Maximum number of peers connected: 50(soft), 60(hard)
- Disconnect rule: heuristic evaluation
- Simultaneously upload to at most 4 peers, due to TCP congestion control

State Exchange Between Peers

- Priority of requests: currently streaming track, prefetching next track and offline synchronization
- Serving client sort by priority and upload speed and serve top 4 peers

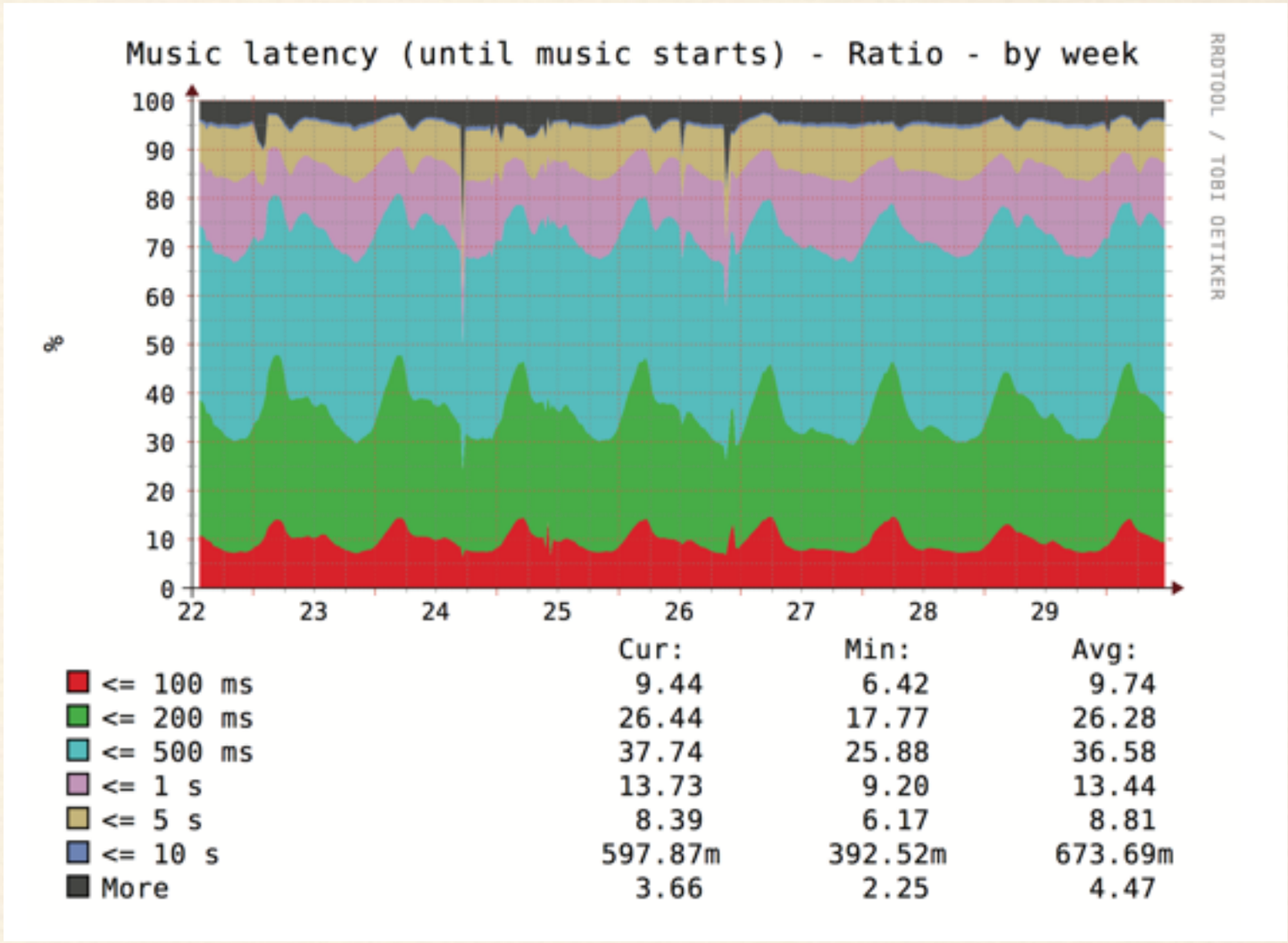
Evaluation: Data Source

- Servers: 8.8%
- P2P: 35.8%
- Cache: 55.4%



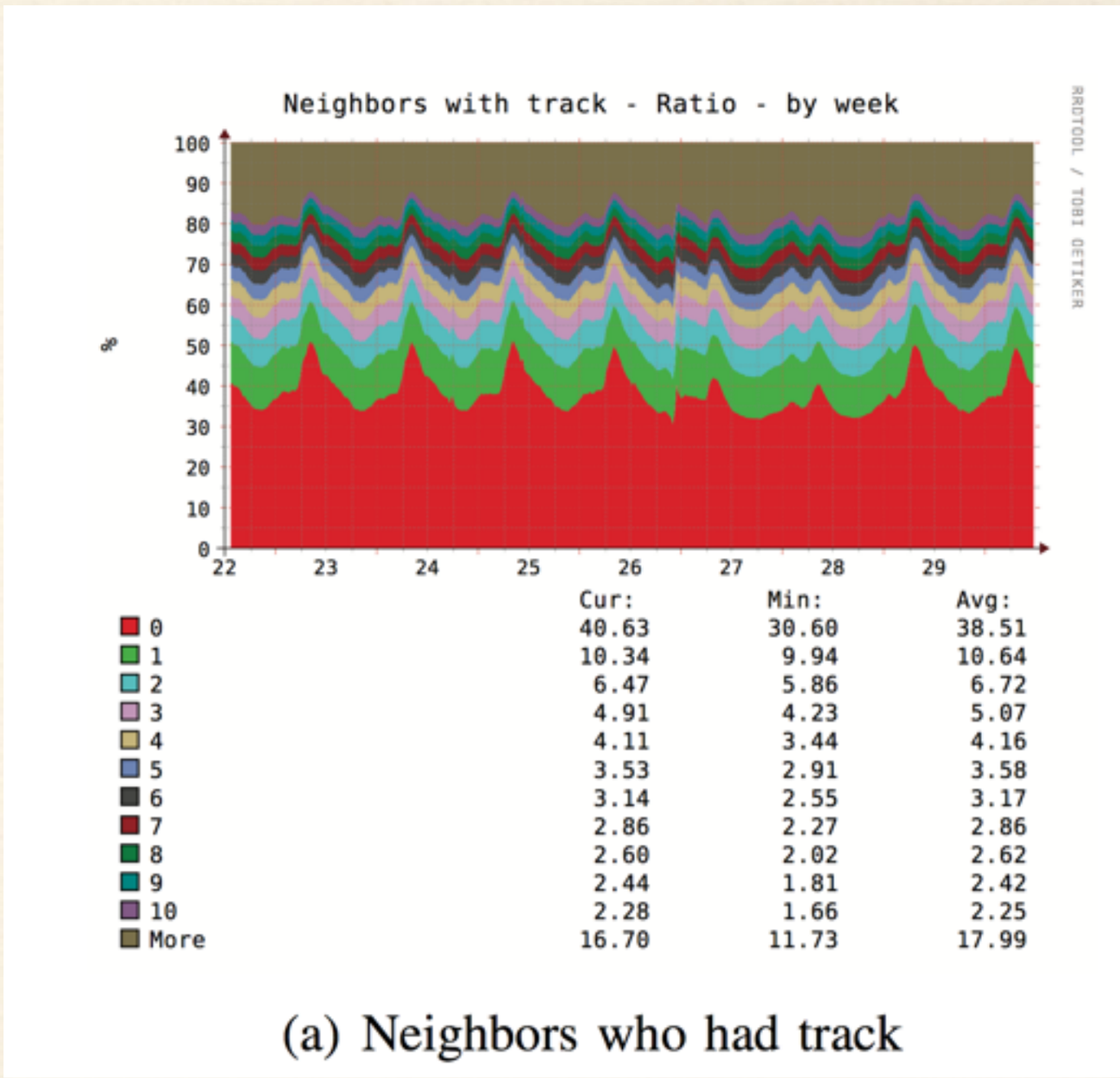
Evaluation: Latency

- At least 1 RTT
- Lower at night



Evaluation: Locating Peers

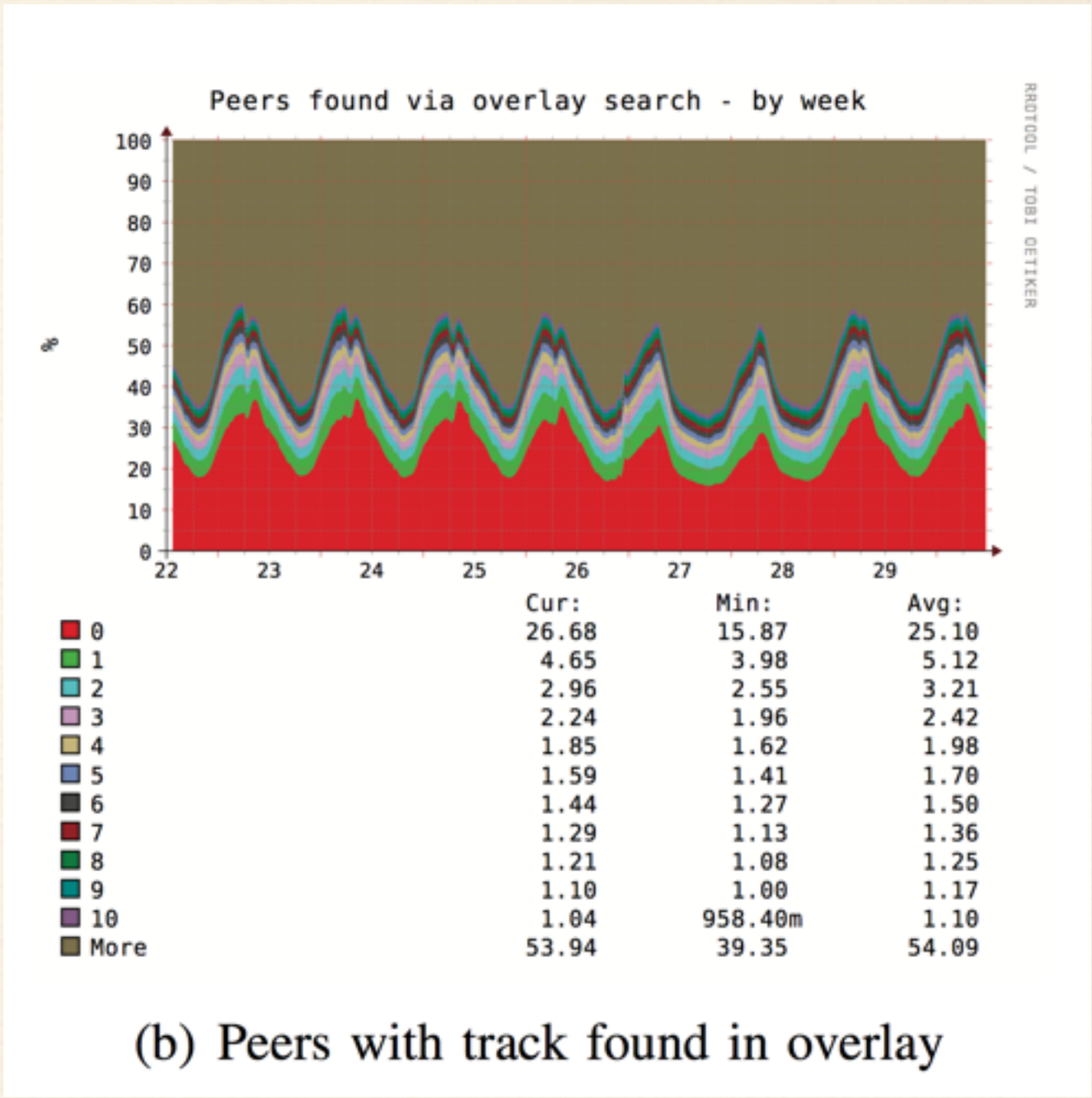
- With distance of 1, much peers do not have track



(a) Neighbors who had track

Evaluation: Locating Peers

- With distance of 2, peer receive much responses



(b) Peers with track found in overlay

Summary

- Combination of server-based and peer-to-peer streaming is good
- Trackers and queries are efficient for locating peers