

Measurements on the Spotify Peer-Assisted Music-on-Demand Streaming System

Mikael Goldmann Gunnar Kreitz

KTH – Royal Institute of Technology
Spotify
gkreitz@spotify.com

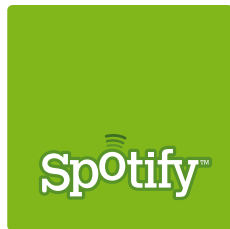
P2P'11, September 1 2011

What is Spotify?

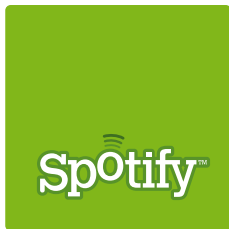
- ▶ On-demand peer-assisted music streaming
- ▶ Large catalog of music (over 15 million tracks)
- ▶ Available in US and 7 European countries, over 10 million users
- ▶ Over 1.6 million subscribers
- ▶ Fast (median playback latency of 265 ms)
- ▶ Legal

Business Idea

- ▶ More convenient than piracy



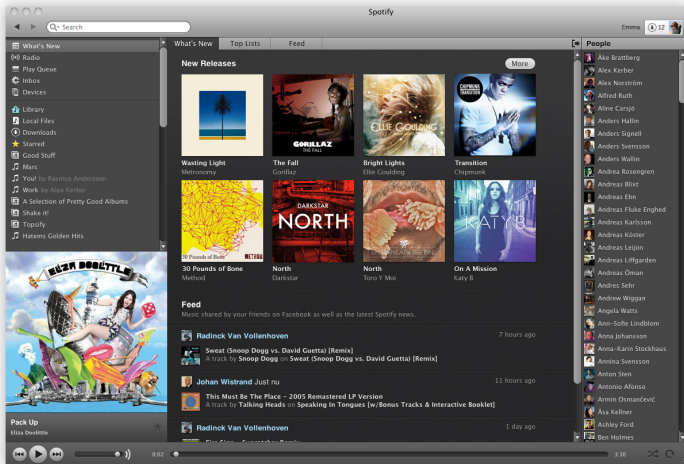
Business Idea



- ▶ More convenient than piracy
- ▶ Spotify Free (ads, 10h/month, invite needed in US)
- ▶ Spotify Unlimited (no ads, on computer)
- ▶ Spotify Premium (no ads, mobile, offline, API)



Desktop Client



Smartphone Client



Hardware Clients



Speed



Photo by moogs <http://www.flickr.com/photos/anjin/23460398/>, CC BY NC ND 2.0

Overview of Spotify Protocol

- ▶ Proprietary protocol
- ▶ Designed for on-demand streaming
- ▶ Only Spotify can add tracks
- ▶ 96–320 kbps audio streams (most are Ogg Vorbis q5, 160 kbps)
- ▶ Relatively simple and straightforward design



Spotify Protocol

- ▶ (Almost) Everything encrypted
- ▶ (Almost) Everything over TCP
- ▶ Multiplex messages over a single TCP connection
- ▶ Persistent TCP connection to server while logged in



Caches

- ▶ Player caches tracks it has played
- ▶ Default policy is to use 10% of free space (capped at 10 GB)
- ▶ Caches are large (56% are over 5 GB)
- ▶ Over 50% of data comes from local cache
- ▶ Cached files are served in P2P overlay



Streaming a Track

- ▶ Request first piece from Spotify servers
- ▶ Meanwhile, search for peers with track
- ▶ Download data in-order
- ▶ When buffers are sufficient, only download from P2P
- ▶ Towards end of a track, start prefetching next one



P2P Structure

- ▶ Unstructured overlay
- ▶ Nodes have fixed maximum degree (60)
- ▶ Neighbor eviction by heuristic evaluation of utility
- ▶ No overlay routing
- ▶ A user only downloads data she needs

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
 - ▶ When doing so, estimate what point P2P will catch up from
- ▶ If buffers are very low, stop uploading

Music vs. Movies

Music

- ▶ Small (5 minutes, 5 MB)
- ▶ Many plays/session
- ▶ Large catalog
- ▶ Active users

Movies

- ▶ Large (2 hours, 1.5 GB)
- ▶ High bit rate

Music vs. Movies

Music

- ▶ Small (5 minutes, 5 MB)
- ▶ Many plays/session
- ▶ Large catalog
- ▶ Active users

Main problem: peer discovery

Movies

- ▶ Large (2 hours, 1.5 GB)
- ▶ High bit rate

Main problem: download strategy

Finding Peers

- ▶ Partial 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 (cherry on top)
- ▶ Client uses all mechanisms for every track

Evaluation

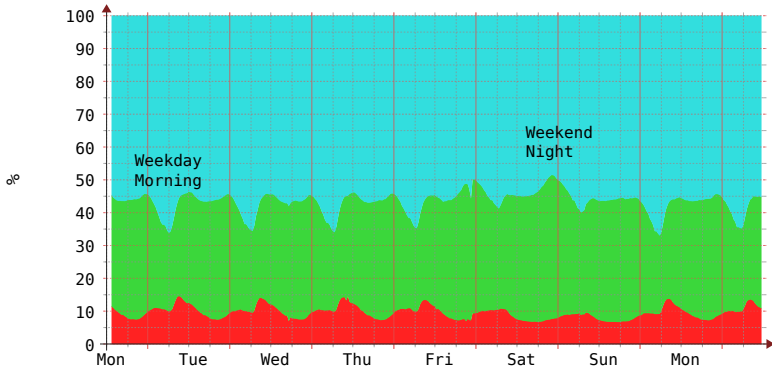
- ▶ So, how well does it work?
- ▶ Data both from 2010 study (P2P'10) and this work



Data Sources (from 2010)

RRDTOOL / TOBI OETIKER

Data source - ratio - by week



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

Data Sources

- ▶ Somewhat sensitive to churn
- ▶ Better P2P performance on weekends
- ▶ 8.8% from servers
- ▶ 35.8% from P2P
- ▶ 55.4% from caches



Finding Peers (from 2010)

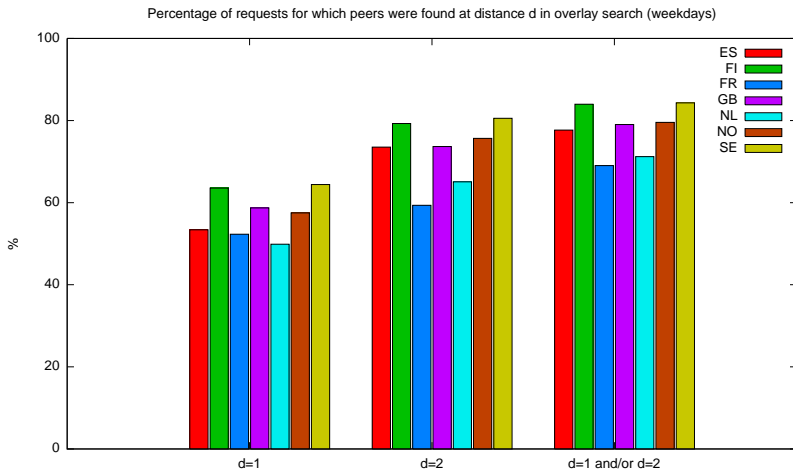


Table: Sources of peers

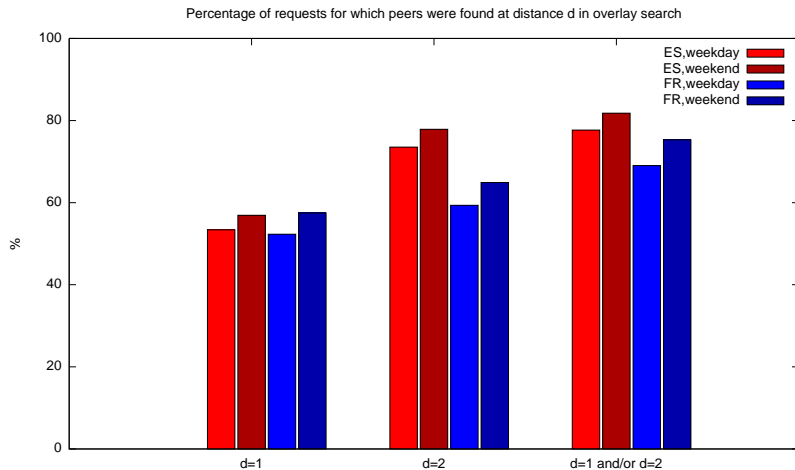
Sources for peers	Fraction of searches
Tracker and overlay	75.1%
Only Tracker	9.0%
Only overlay	7.0%
No Peers Found	8.9%

- ▶ Each mechanism by itself is fairly effective

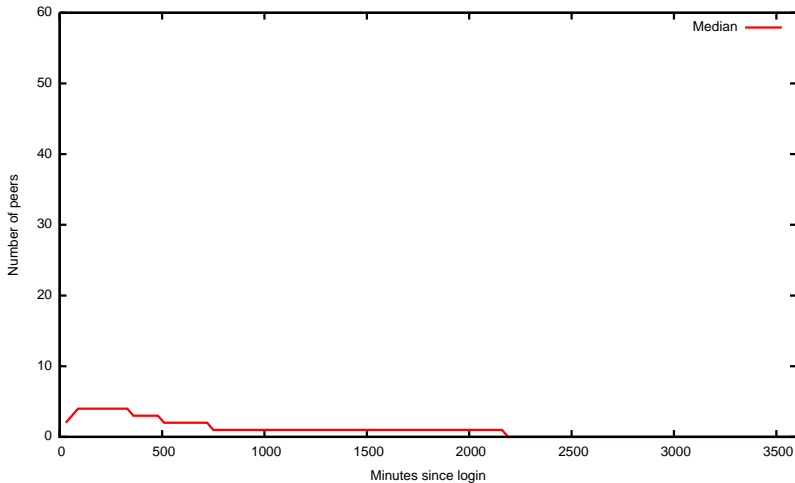
The overlay peer discovery mechanism



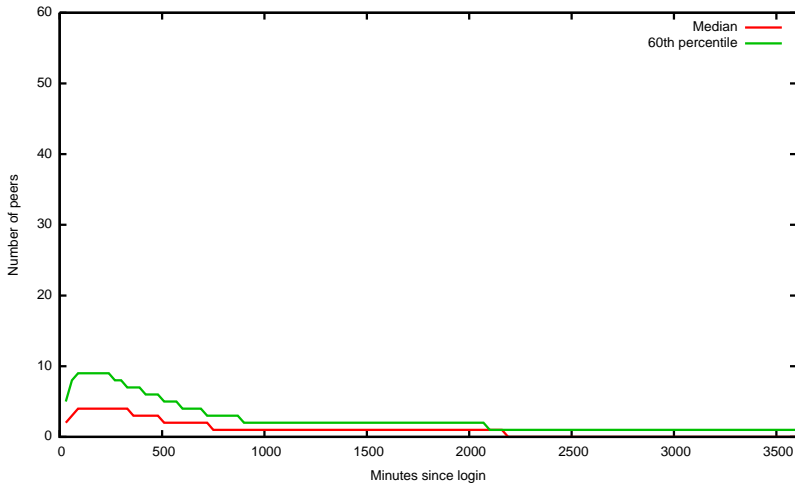
The weekend effect in peer discovery



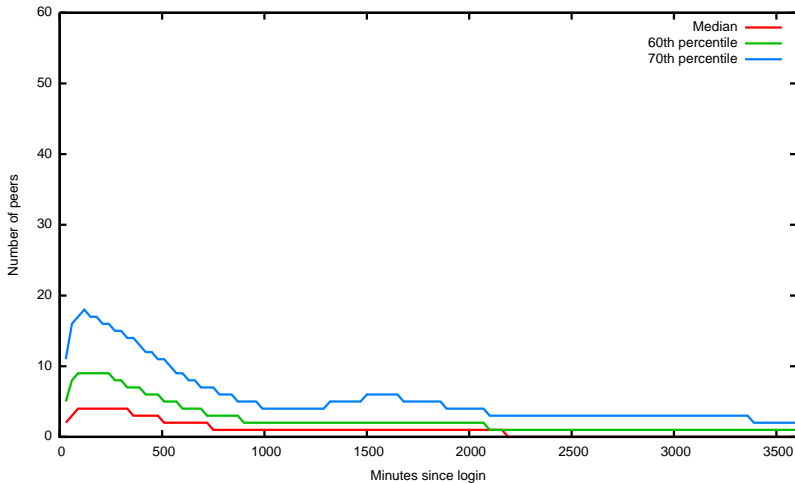
Users become slowly disconnected from overlay



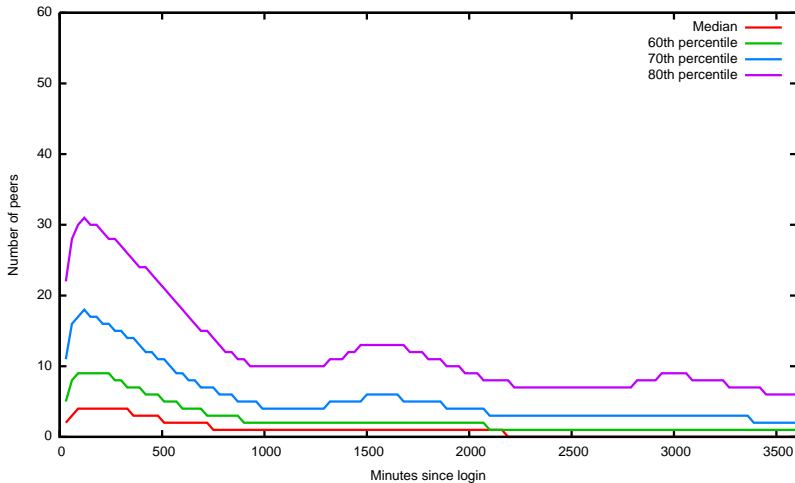
Users become slowly disconnected from overlay



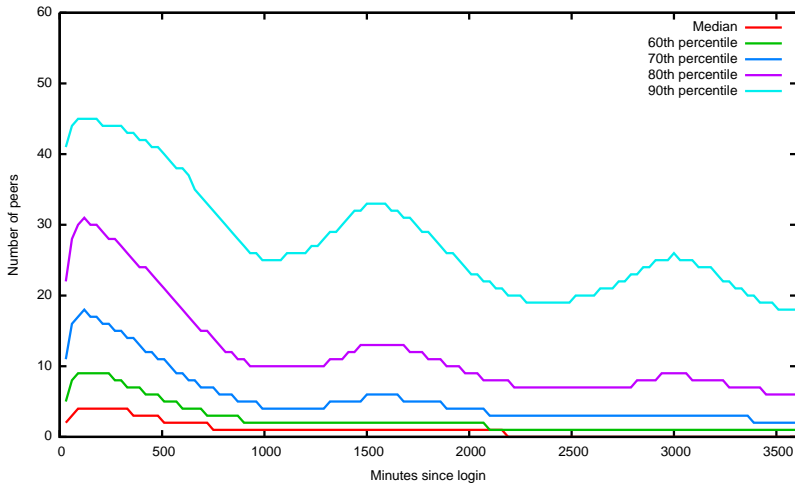
Users become slowly disconnected from overlay



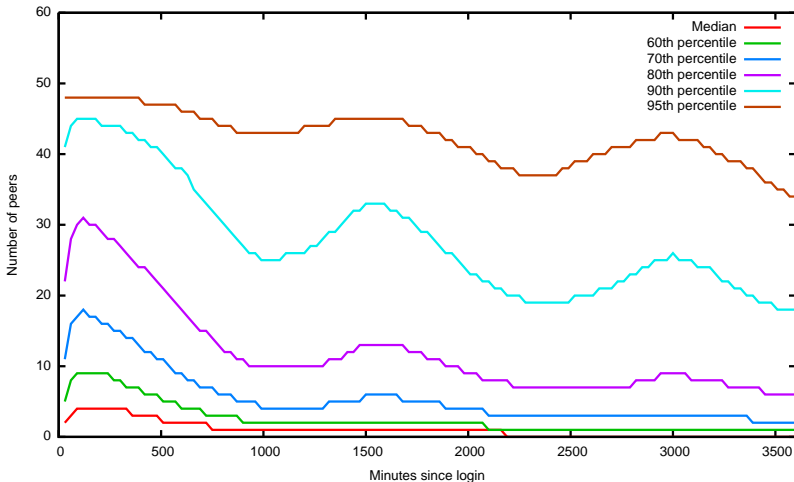
Users become slowly disconnected from overlay



Users become slowly disconnected from overlay



Users become slowly disconnected from overlay

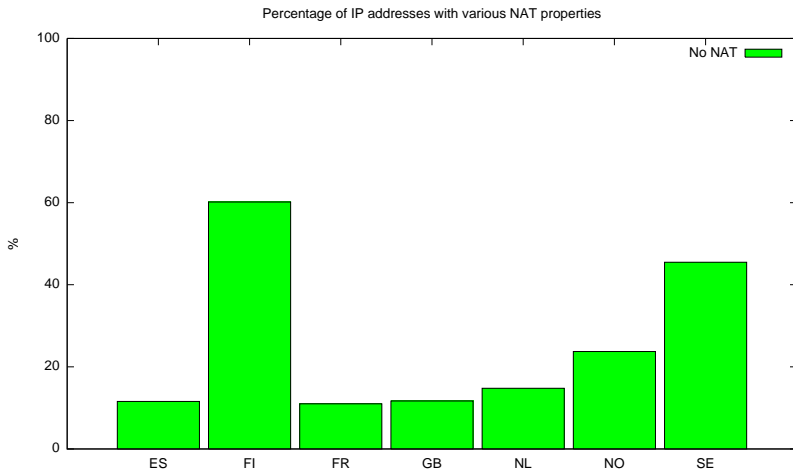


NAT types in the wild

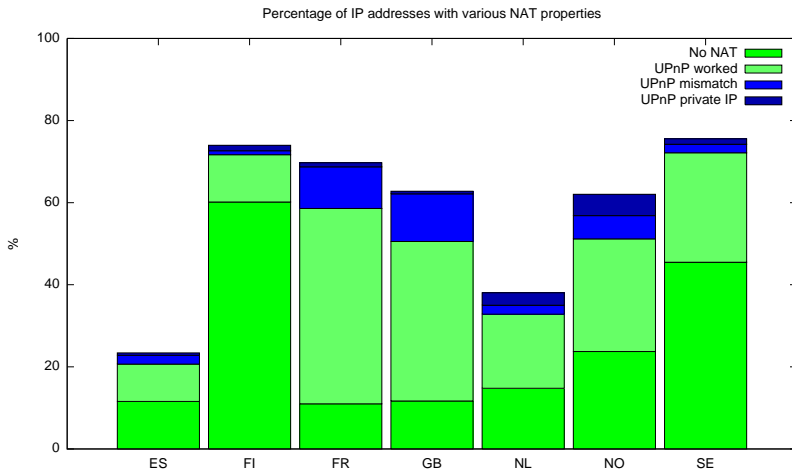


Photo by roychung1993 <http://www.flickr.com/photos/roychung1993/2856744333/>, CC BY NC 2.0

NAT types in the wild



NAT types in the wild



How many IPs do users have

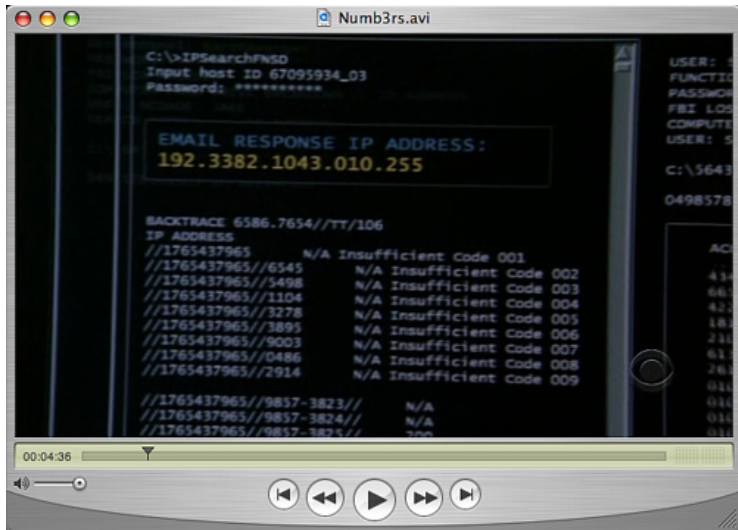
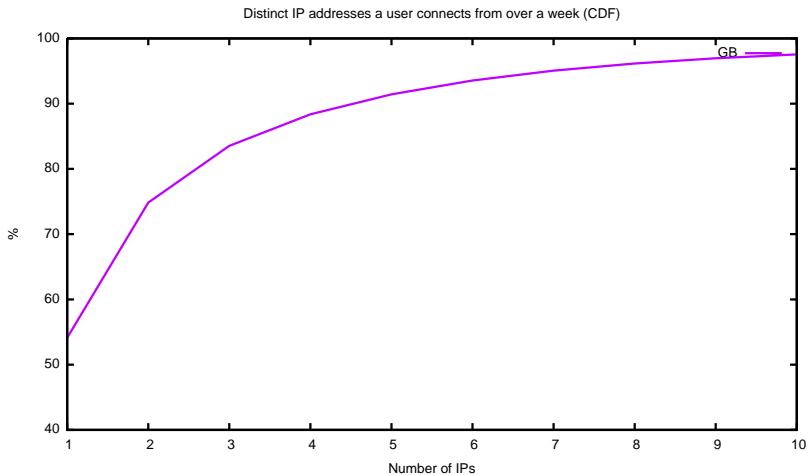
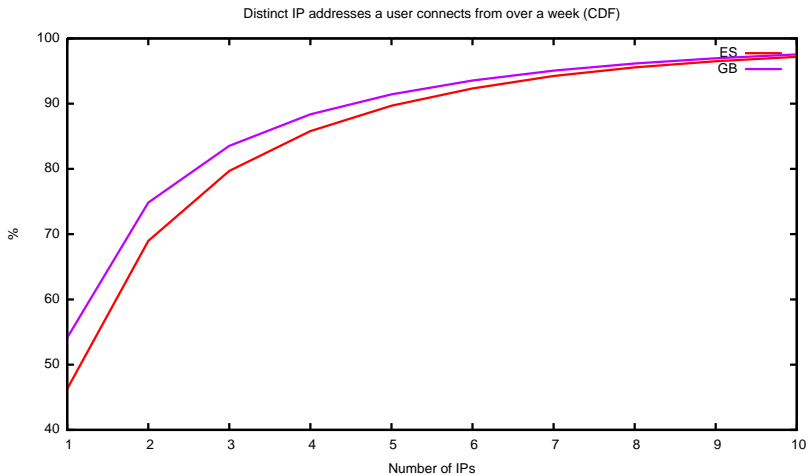


Photo by feria <http://www.flickr.com/photos/feria/141613794/>, CC BY NC SA 2.0

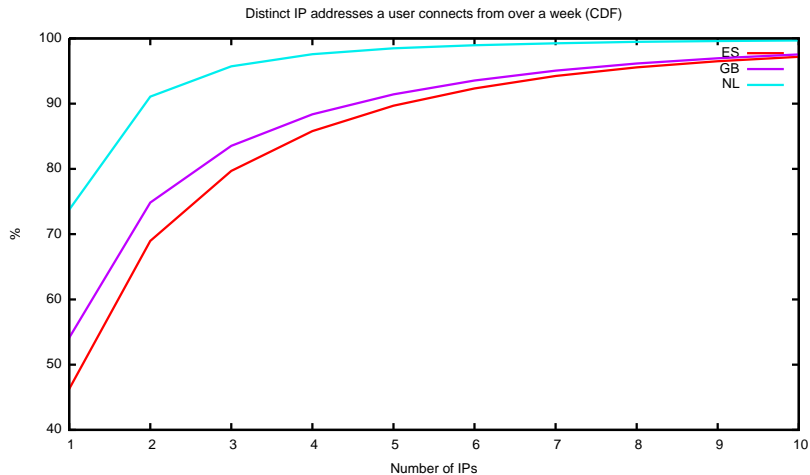
How many IPs do users have



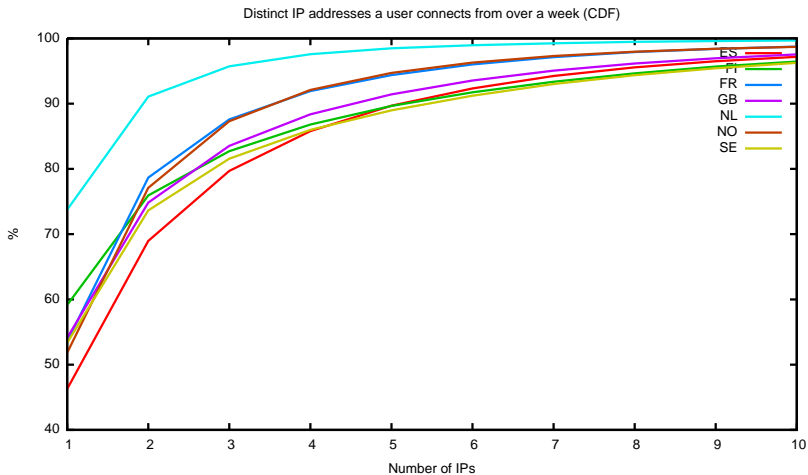
How many IPs do users have



How many IPs do users have



How many IPs do users have



Summary

- ▶ Measurements of a large, deployed system
- ▶ Future work
 - ▶ Scaling to more users
 - ▶ Improvements of P2P protocol
 - ▶ More measurements (what are you interested in?)

