The Final Lecture

EE122 Fall 2012

Scott Shenker

http://inst.eecs.berkeley.edu/~ee122/

Materials with thanks to Jennifer Rexford, Ion Stoica, Vern Paxson and other colleagues at Princeton and UC Berkeley

Why should we care about P2P?

- Largely about stolen content
- Why is that something we should teach?
  - Questionable ethics
  - And you know far more about P2P than I do….

Three Reasons

- P2P represents new economic model for systems
- P2P represents a new design model for systems
- P2P is one last chance to brag

Outline

- History of Telephony
- Transitioning to the Internet
- P2P as a technology
- P2P as a design paradigm
- Where did these ideas come from?

In the beginning…..

- AT&T created the telephone network
- 1st realtime person-to-person communication system
  - Telegraph was station to station
  - Mail was asynchronous
  - …
- Development of phone technology was complicated

Early History of Telephone Technology

- 1844 Innocenzo Manzetti: idea of a “speaking telegraph” (telephone).
- 1854 Charles Bourseul: article on electric transmission of speech
- 1861 Johann Philipp Reis: demonstrated the Reis telephone before the Physical Society of Frankfurt.
- 1865 “It is rumored that English technicians to whom Mr. Manzetti illustrated his method for transmitting spoken words on the telegraph wire intend to apply said invention in England on several private telegraph lines”

Continued History

- 1871 Antonio Meucci files a patent describing communication of voice by wire.
- 1874 Meucci, after having renewed the application for two years does not renew it again, and application lapses.
- 1875 A. G. Bell granted patent for “Transmitters and Receivers for Electric Telegraphs”
- 1876 Elisha Gray files patent for transmitting the human voice through a telegraphic circuit.
- 1876 Bell applies for the patent “Improvements in Telegraphy”

More History

- 1876 Gray withdraws application after notified by Patent Office of conflict with Bell's application.
- 1876 Bell granted patent 174,465 "Improvement in Telegraphy"
- 1876 First successful telephone transmission of speech using liquid transmitter when Bell spoke into his device, "Mr. Watson, come here, I want to see you." and Watson heard each word distinctly.
- 1877 Bell granted patent for an electromagnetic telephone using permanent magnets, iron diaphragms, and a call bell.

A new player in the game….

- 1877 Edison files for a patent on a carbon (graphite) transmitter.
- 1892 Edison granted patent after 15 years of litigation.
- *This is 48 years after first discussion of idea!*

Three Things to Note

- Makes Apple vs Samsung look short and sweet
- This is all about the application technology
  - Not the network!
- But this application drove the creation networking

History of Phone Network

- 1876: Bell licensed phone to local companies
  - Creation of local networks
- 1885: AT&T formed to connect local companies
  - Creation of a national network
- 1913: AT&T becomes regulated monopoly
  - Profits are a fixed percentage of costs
- 1984: AT&T decides to get into computer business
  - Forced to give up Regional Bell Operating Companies

More History

- 1998: AT&T buys up cable companies
  - Tries to enter Internet business
- 1999 AT&T funds ACIRI (a long story….)
- 2001: AT&T starts selling cable companies
  - At a huge loss, and AT&T starts circling the drain….
- 2005: AT&T bought by SBC (one of the RBOCs)

AT&T and Verizon the two dominant carriers in US

Implications of Regulated Monopoly

- Terrible management!
- Glacial innovation in novel functionality
  - Why spend money on features no one knows they want?
- Extreme reliability and polish
  - Spend money improving what people notice (failures)
- What else do you spend money on?

Technology!

- Assured percentage of costs in profits
  - Why not support research on basic technology?

- List of inventions from Bell Labs:
  - Transistors
  - Lasers
  - Cellular telephony
  - Digital data transmission
  - Solar cells
  - Communication satellites
  - Unix, C
  - Digital signal processors

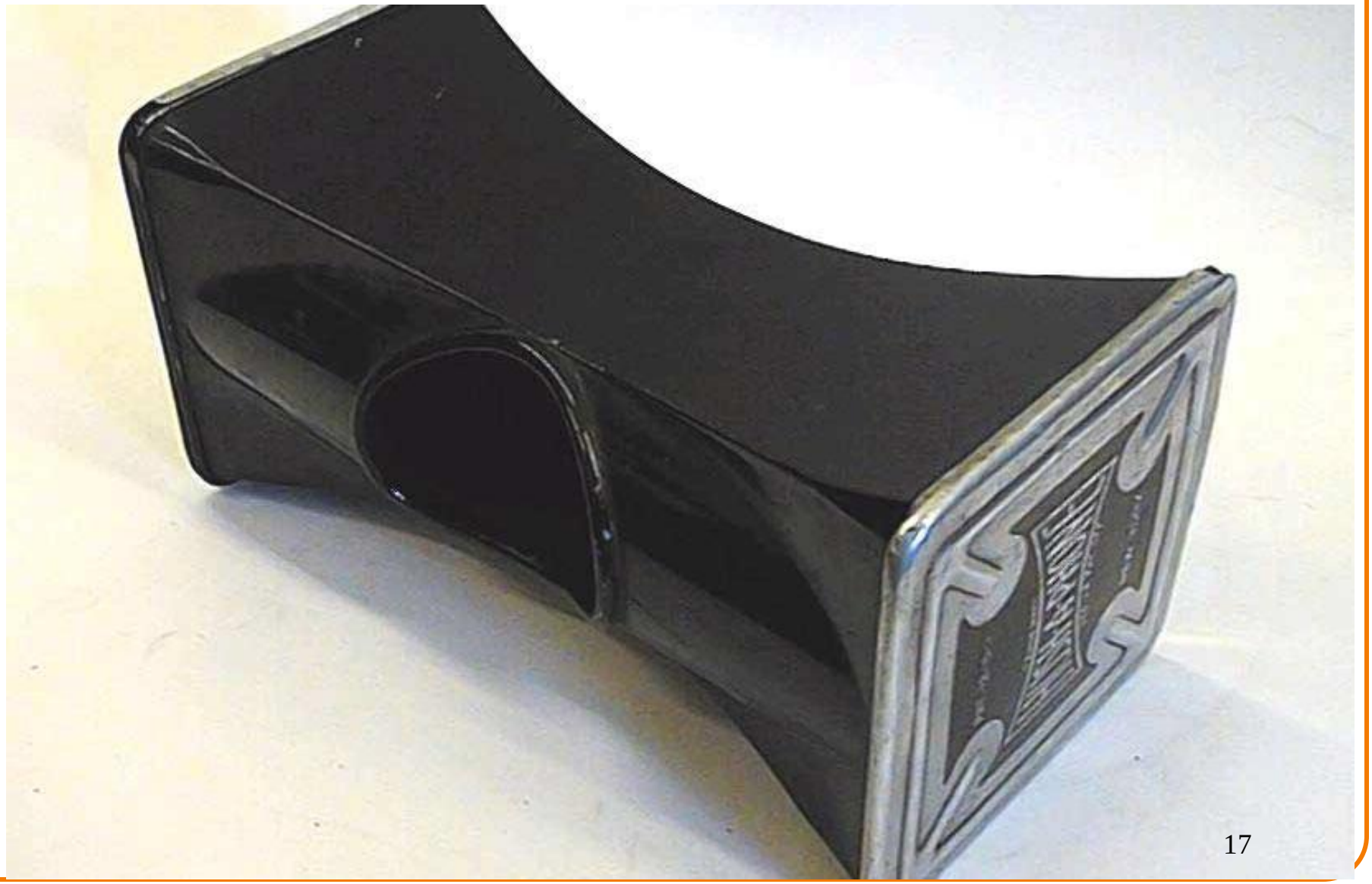
Monopolies and Research

- Greatest research centers:
  - Bell Labs (physics, UNIX)
  - Xerox PARC (invented modern computing)
  - IBM (databases, etc.)
- Why not in academia?
- Why not in regular companies?

Original Telephone Economic Model

- Functionality controlled by network operator
  - They sink the money into the infrastructure
  - They get to decide what that infrastructure does
- End-user only has “dumb terminal”
  - Legally restricted in its use of that terminal
- AT&T had stranglehold over network until 1956
  - Courts gave end users limited freedom
  - What machine was this battle fought over?

The Hush-a-Phone



More involuntary openness followed

- Later court cases opened way for FAX machines, modems, and the like.

- But the endpoints were still pretty dumb, until the PC was developed….

Which then gave us the Internet…

- End points had complete freedom, and substantial computing power
  - Infrastructure just carried bits
- Completely different economic model
  - Small guys can innovate
  - Big guys run dumb infrastructure (like utilities)
- Result:
  - Rapid innovation in applications (e.g., email, web)
  - Diversity of content (on web)
  - Low barrier to entry
- And finally, even the big companies noticed….

The Empire Strikes Back

- Zipf's law restores order to the universe
  - Popularity ~ 1/rank
  - Lots of weight at top (people like the same things)
  - Lots of weight in tail (but lots of idiosyncratic tastes)
- A Tale of Two Markets
  - Lots of action in the tail (anyone can provide content)
  - But only few can create/provide the popular content
- High barrier to entry at top: (CDNs)
  - Bandwidth
  - Servers
  - Management

Revenge of the Nerds

- Peer-to-Peer restores the balance
  - Takes “contributed” nodes from participants
  - Together they provide enough aggregate bandwidth
- The key is in coordinated these peer nodes
  - First: Napster (Shawn Fanning)

P2P Systems Do Three Main Things

- Help user **determine** which content they want
  - Some form of search
  - *P2P form of Google*
- Then **locate** that content
  - Locate where that content is on the Internet
  - *P2P form of DNS (map name to location)*
- Then **download** that content
  - *P2P form of Akamai*

We need P2P forms of

- Search (keyword)
- Directory
- CDN

- What kinds of coordination mechanisms do we need for these tasks?
  - Have 100,000s of nodes at our disposal
  - How to coordinate them for search, directory, CDN?

Design Requirements

- Scalable
- Fault-tolerant
- Able to use commodity parts

Design P2P Search

- How would you do it?

P2P Search

- Basic approach:
  - Since keyword search can be complicated,
  - …just do it on each machine independently,
  - …and keep going for as long as you need

- Examples:
  - Broadcast request to all nodes
  - Broadcast among “superpeers”
  - Random walk (theory, papers)

Random Walk Search (Aside)

- Ask random nodes, each has fixed size cache
- Query rate q_i, where this represents popularity
- How to replicate content "optimally"?
  - What is replication factor r_i
- Goal: find result in shortest number of queries
- What's the right choice for r_i?
  - Uniform? Proportional to query rate? What?
- Uniform and proportional have same performance
- Optimal is proportional to sqrt(q_i)

Design P2P Directory

- How would you do it?

P2P Directory

- In most cases, a few centralized servers will do
- If you need to scale further, then use DHT
  - Distributed Hash Table
  - Put/Get interface
- DHT: simple version is consistent hashing
  - Set up “linked list” of servers, each with ID in [0,1]
  - Each data object has key in [0,1]
  - Map key to server using the successor rule
    - Pick smallest server with an ID “larger” than key
  - Scalable, local repair of failures, etc.

Design P2P Download

- How would you do it?

P2P Download

- The first key is *self-scaling*
  - If every person who downloads something also has to upload it to someone else, the system works
- The second key is *asymmetric bandwidth*
  - Download bwdth much greater than upload bwdth
  - That's where chunks come in
  - Downloading many chunks overcomes asymmetry
    - Because each chunk is uploaded from a different source

Modern P2P Systems Use a Mixture

- Search to find name (wildcard search)
  - Flood among superpeers
- Directory lookup to find host given exact name
  - DHT-like structure
- Chunked download
  - Self-scaling
  - Asymmetric bandwidth

Peer-to-Peer as Design Paradigm

- Once you can coordinate many disparate peers…
- …you can certainly coordinate co-located peers
- Now the dominant design style in datacenters
- DHT-like data structures are everywhere

What is the P2P Design Paradigm?

- *Scale out, not up*:
  - To get more capacity, add more low-cost units
  - Don't increase capacity of individual units
  - (This is why Sun/SGI/etc. are no longer with us)
- This poses problem of coordination
- The key to scalable coordination is how you deal with failures….
  - Because as system grows, failures are inevitable

Dealing with failures

- Design as if failure is common
- Recover from failures at highest possible layer
  - Generalized version of E2E principle
- E.g. If routing fails, use another server
  - Don’t wait for routing to repair itself

This is Google's Design Philosophy

- Permeates culture
  - Much like Jobs' attitudes at Apple
- Provided consistent and winning paradigm
- Where did this come from?

This is the Internet Design Philosophy

- What I've been pontificating about all semester
- The Internet design philosophy has taught us how to build systems at scale
- The Internet is the first scalable P2P system
- The Internet changed the world
- **Its design philosophy changed computer science**

What you should remember….

- I don’t care if you remember anything about TCP
  - Or DNS, or HTTP, or….
- But if you ever build system that doesn’t scale well,
  - **I will hunt you down….**

Mission Accomplished