Peer-peer Computing & Networking
CS 699/IT 818
Fall 2004

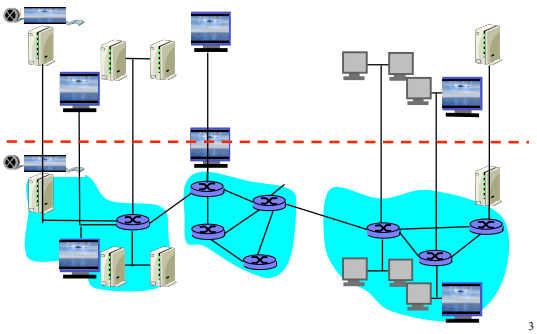
Sanjeev Setia

http://www.cs.gmu.edu/~setia/cs699/

Acknowledgements

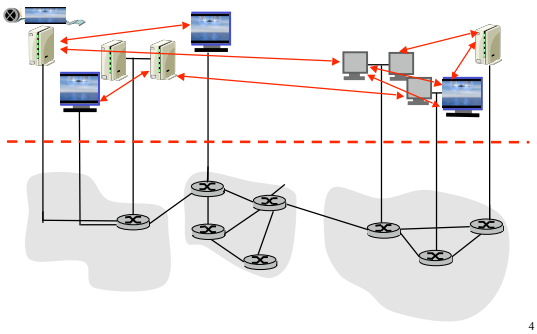
Some of the followings slides are based on the slides made available by the authors of *Computer Networking: A Top Down Approach Featuring the Internet*, 2nd edition.
Jim Kurose, Keith Ross
Addison-Wesley, July 2002.

Peer-peer computing and networking



Peer-peer network

Focus at the application level



Peer-to-Peer: Some Definitions

- A P2P computer network refers to any network that does not have fixed clients and servers, but a number of peer nodes that function as both clients and servers to other nodes on the network.
  Wikipedia.org
- The sharing of computer resources and services by direct exchange between systems
  Intel P2P working group
- The use of devices on the internet periphery in a non-client capacity
  Alex Weytsel, Aberdeen Group
- P2P is a class of applications that takes advantage of resources – storage, cycles, content, human presence – available at the edges of the internet.
  Clay Shirky, openp2p.com

5

Peer-peer applications

- File sharing
  - Napster, Gnutella, KaZaa
  - Second generation projects
    - Oceanstore, PAST, Freehaven
- Distributed Computation
  - SETI@home, Entropia, Parabon, United Devices, Popular Power
- Other Applications
  - Content Distribution (BitTorrent)
  - Instant Messaging (Jabber), Anonymous Email
  - Groupware (Groove)
  - P2P Databases

6

Is Peer-to-peer new?

- P2P concept certainly not new
  - Usenet - News groups first truly decentralized system
  - DNS - Handles huge number of clients
  - Basic IP - Vastly decentralized, many equivalent routers
- What is new?
  - Scale: people are envisioning much larger scale
  - Security: Systems must deal with privacy and integrity
  - Anonymity: Protect identity and prevent censorship
  - (In)Stability: Deal with unstable components at the edges

7

P2P: Related Technologies

- Distributed computing.
  - How is P2P different from distributed computing?
- Grid computing.
  - How is the computational grid different from P2P networks?

  KEY DIFFERENCES: Peers are on the edges of the Internet, are autonomous, have variable connectivity, and temporary network addresses
- Application-level networking.
  - Resilient overlay networks for multicast, video distribution, etc.

8

P2P: Related Technologies

- Wireless ad-hoc networks.
- Sensor networks.
- P2P devices/ubiquitous computing.
  - JINI.
- Web services.
  - .NET framework, SOAP, UDDI.

9

Why the hype???

- File Sharing: Napster (+Gnutella, KaZaa, etc)
  - High coolness factor
  - Served a high-demand niche: online jukebox
- Anonymity/Privacy/Anarchy: FreeNet, Publis, etc
  - Libertarian dream of freedom
  - Extremely valid concern of Censorship/Privacy
  - In search of copyright violators, RIAA challenging rights to privacy
- Computing: The Grid
  - Scavenge the numerous free cycles of the world to do work
  - Seti@Home most visible version of this
- Industry/Management
  - Looking for the next big thing
  - A lot of interest/hype in "autonomic computing"/Computing as a utility

10

Class Logistics: a research seminar

Q: Is a seminar just like a course?

NO!

- I will present papers for the first 7 weeks
- Second half of the semester: student-presentations
  - 2 presentations per class
  - Each presentation based on 1-2 papers
  - student-led discussions
- no well-defined body of knowledge to impart
  - seminar is about searching for answers (and questions)
- great opportunity to find interesting research projects

11

Requirements

Workload:

- Reading assigned papers
- Write short reviews of papers to be discussed in class (20% of grade)
  - Email them to me before class.
  - Template for reviews is on class web site
- Participate in class discussions
- Make a class presentation (25%)
- Assist others in preparing their presentations (5%)
  - Each presenter will be assigned a partner who will go over the presentation with them
- Course project (50%)

Pre-requisites:

- previous class on networking, operating systems,algorithms, distributed systems

12

Presentations

- after first seven weeks:
  - *student-led* presentations and discussions
- *you* will need to:
  - read, think deeply about paper, topic area
  - look for additional outside material
  - prepare ~60 minute class presentation
  - Go over presentation with partner, instructor
  - lead in-class discussion of material
- everyone wants your presentation to be well-prepared, interesting, thoughtful

13

Presentations (more)

Preparing your presentation in advance:

- read documents about preparing a good talk (on web site)
- 1 week in advance: meet with presentation partner
  - Practice your talk!
- meet with instructor about presentation
- post overheads in advance of class

What's in a presentation?

- paper contents
- additional material you have found
- *critical analysis:* questions, strengths, weaknesses, improvements, future work
  - For every paper find:
    - 3 most important points
    - 3 weaknesses (flaws with the assumptions/methodology/etc.)

14

Class Project

- Various Options
  - Explore an open topic
    - Read literature, find open problem, propose a new solution, evaluate it
    - Could be the beginning of a MS/Ph.d. thesis
  - Re-evaluate/Validate the conclusions of a previous study
    - do experiments that have been reported in a paper
    - Measurement, Implementation, Simulation
  - Programming-oriented project
    - Implement a P2P application/service, e.g. use JXTA to develop a P2P app, do measurements, etc.
- Identify project by end of Sept.
  - Discuss with instructor
  - Make a presentation in class (5-10 minutes)

15

Schedule

- First 7 weeks
  1. Intro (Week 1)
  2. Structured P2P systems (DHTs) – Tapestry, Chord, Pastry, CAN (Weeks 2 & 3)
  3. Unstructured P2P systems (Week 4)
  4. P2P File Systems (Week 5)
  5. Performance & P2P Workload Issues (Week 6)
  6. Security/Applications (Week 7)
- Next 6 weeks – student presentations
  - See reading list on class web site (soon!)
- Last week + Finals week
  - Project Presentations

16

P2P Applications Taxonomy

- Content and File Sharing
  - Napster, Gnutella, KaZaa, etc.
  - Most research has focused on this class of apps
- Parallelizable
  - Compute Intensive (Same task on every peer using different parameters)
  - Componentized applications – different components on each peer (not yet widely supported/recognized)
- Collaborative
  - Instant messaging, groupware, games
  - Many startups but not that much academic research

P2P file sharing

Example

- Alice runs P2P client application on her notebook computer
- Intermittently connects to Internet; gets new IP address for each connection
- Asks for "Hey Jude"
- Application displays other peers that have copy of Hey Jude.
- Alice chooses one of the peers, Bob.
- File is copied from Bob's PC to Alice's notebook: HTTP
- While Alice downloads, other users uploading from Alice.
- Alice's peer is both a Web client and a transient Web server.

All peers are servers = highly scalable!

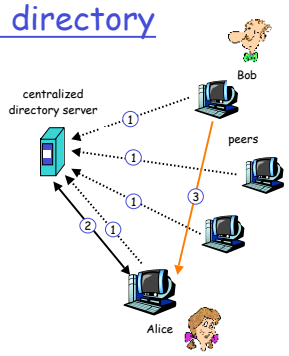
P2P Content Location & Routing

- Three approaches
  - Centralized directory (Napster)
  - Decentralized directory + Flooding-based search (Gnutella)
    - Unstructured P2P systems
  - Distributed Hash Tables (DHT) based document search and publication
    - Structured P2P systems (Chord, CAN, Tapestry, etc)
    - Presented in weeks 2 & 3

P2P: centralized directory

original "Napster" design

1) when peer connects, it informs central server:
   - IP address
   - content
2) Alice queries for "Hey Jude"
3) Alice requests file from Bob



P2P: problems with centralized directory

- Single point of failure
- Performance bottleneck
- Copyright infringement

file transfer is decentralized, but locating content is highly centralized

Napster

- program for sharing files over the Internet
- a killer application?
- history:
  - 5/99: Shawn Fanning (freshman, Northeasten U.) founds Napster Online music service
  - 12/99: first lawsuit
  - 3/00: 25% UWisc traffic Napster
  - 2000: est. 60M users
  - 2/01: US Circuit Court of Appeals: Napster knew users violating copyright laws
  - 7/01: # simultaneous online users:
    Napster 160K, Gnutella: 40K, Morpheus: 300K
  - 2001: Napster shut down; Bertelsmann acquire assets, etc.
- Today
  - Napster 2.0 music download service (Roxio)
  - Also OpenNap (open source napster server)

Napster: how did it work

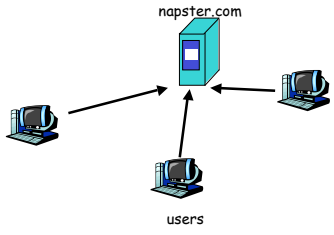
Application-level, client-server protocol over point-to-point TCP

Four steps:

- Connect to Napster server
- Upload your list of files (push) to server.
- Give server keywords to search the full list with.
- Select "best" of correct answers. (pings)

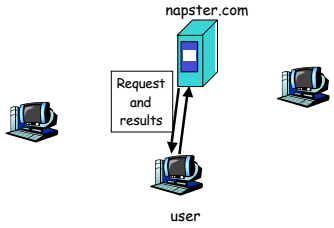
Napster

1. File list is uploaded



Napster

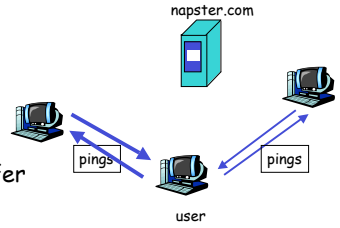
2. User requests search at server.



Napster

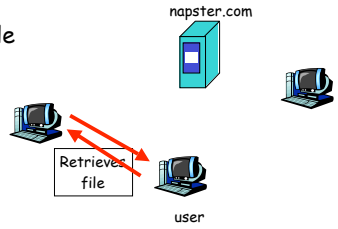
3. User pings hosts that apparently have data.

Looks for best transfer rate.



Napster

4. User retrieves file

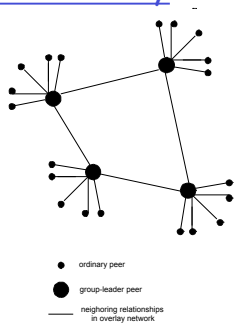


Napster: architecture notes

- centralized server:
  - single logical point of failure
  - can load balance among servers using DNS rotation
  - potential for congestion
- no security:
  - passwords in plain text
  - no authentication
  - no anonymity

P2P: decentralized directory

- Each peer is either a group leader or assigned to a group leader.
- Group leader tracks the content in all its children.
- Peer queries group leader; group leader may query other group leaders.



29

More about decentralized directory

overlay network

- peers are nodes
- edges between peers and their group leaders
- edges between some pairs of group leaders
- virtual neighbors

bootstrap node

- connecting peer is either assigned to a group leader or designated as leader

advantages of approach

- no centralized directory server
  - location service distributed over peers
  - more difficult to shut down

disadvantages of approach

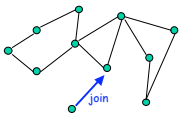
- bootstrap node needed
- group leaders can get overloaded

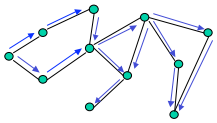
30

P2P: Query flooding

- Gnutella
- no hierarchy
- use bootstrap node to learn about others
- join message

- Send query to neighbors
- Neighbors forward query
- If queried peer has object, it sends message back to querying peer





31

P2P: more on query flooding

Pros

- peers have similar responsibilities: no group leaders
- highly decentralized
- no peer maintains directory info

Cons

- excessive query traffic
- query radius: may not have content when present
- bootstrap node
- maintenance of overlay network

32

Gnutella

- peer-to-peer networking: applications connect to peer applications
- focus: decentralized method of searching for files
- each application instance serves to:
  - store selected files
  - route queries (file searches) from and to its neighboring peers
  - respond to queries (serve file) if file stored locally
- Gnutella history:
  - 3/14/00: release by AOL, almost immediately withdrawn
  - too late
  - many iterations to fix poor initial design (poor design turned many people off)
- What we care about:
  - How much traffic does one query generate?
  - how many hosts can it support at once?
  - What is the latency associated with querying?
  - Is there a bottleneck?

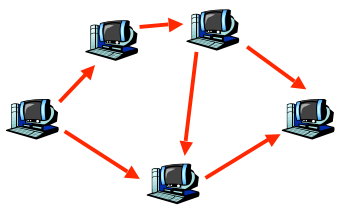
Gnutella: how it works

Searching by flooding:

- If you don't have the file you want, query 7 of your partners.
- If they don't have it, they contact 7 of their partners, for a maximum hop count of 10.
- Requests are flooded, but there is no tree structure.
- No looping but packets may be received twice.
- Reverse path forwarding

Note: Play gnutella animation at:
http://www.limewire.com/index.jsp/p2p

Flooding in Gnutella: loop prevention



Seen already list: "A"

Distributed Computing

- Current supercomputers are too expensive
  - ASCI White (#1 in TOP500) costs more than $110 million and needed a new building
  - Few institutions or research groups can afford this level of investment
- There are more than 500 million PCs around the world
  - some as powerful as early 90s supercomputers
  - they are idle most of the time (60% to 90%), even when being used (spreadsheet, typing, printing,...)
  - corporations and institutions have hundreds or thousands of PCs on their networks

Try to harness idle PCs on a network and use them on computationally intensive problems

How it works

- Embarrassingly parallel applications
  - Large computation to communication ratio
  - Master/worker model
  - Applications can use local disk for checkpointing
- Provider farms out work to idle PCs across the internet
  - PC owners volunteer idle cycles (for money or altruistic purposes)

37

Entropia network

- Born in 1997 to apply idle computers worldwide to problems of scientific interest
- In 2 years grew to more than 30,000 computers with aggregate speed of over 1 Tflop/second
- Several scientific achievements, e.g. Identification of largest known prime number
- Gone commercial: www.entropia.com and used for applications from:
  - Life sciences
  - Financial services
  - Product design, etc.
- Today: appears to not have succeeded as a business
  - Business model for distributed computing not yet successful

38

SETI @ home project

setiathome.ssl.berkeley.edu

- SETI = Search for Extraterrestrial Intelligence
- Started in 1996 to enlist PCs to work on analyzing data from the Arecibo radio telescope
- Good mix of popular appeal and good technology

⇨
- Now running on more than _ million PCs
- delivering ~ 1,200 CPU years per day
- ~ 35 Tflops/sec
- fastest (but special-purpose) computer in the world

39

Folding @ home project

www.stanford.edu/group/pandegroup/Cosm

- Enlists PCs to work on the protein folding problem
- most important problem in modern molecular biology
- From genome to structure:
  - Genome sequence of DNA specifies amino acids that make up proteins, but says little about their functions: what is needed is how a protein fold (3D structure)
  - Protein folding is very fast (microseconds) and complex
  - Simulation timescale is of the order of nanoseconds
    → 10^3 gap → distributed computing
- Currently around 20,000 users

40

Readings

- P2P Survey Article on Class web page
- For next week:
  - Tapestry
  - Pastry

41