Peer to peer networking (P2P)

- **"direct" connections between peers**
  - peers = participating computers
  - services distributed instead of clients talking to single server
  - all peers provide bandwidth, storage, processing
  - use TCP/IP (same level as HTTP, SSH, SMTP, etc.)
- **an old idea, though with a new name**
  - USENET news service, 1979 (still in use)
- **"peer-to-peer" file-sharing**
  - centralized directories (original Napster)
  - decentralized directories (Gnutella, Kazaa, etc.)
- **once a file is found somewhere**
  - direct connection between supplier and consumer ("peers")
- **other important examples**
  - BitTorrent file distribution system
  - Skype Internet telephony

Peer to peer highlights

- **Napster (1999-2001) [Shawn Fanning]**
  - centralized real-time directory, distributed files
  - mostly MP3 music; ideal for Ethernet bandwidths
  - based in USA; lawsuits put it out of business
- **Gnutella (2000) and friends (Grokster, Kazaa, ...)**
  - decentralized directories: not as fast or reliable but less vulnerable to legal processes
  - most deposit adware and sometimes spyware

    (therefore there is a commercial purpose)
- **BitTorrent (2001)**
  - distributed directories, distributed files
  - distributed peer servers for load-sharing: good for movies

BitTorrent

- **file-sharing for big files in high demand**
- **original file exists on at least one "seed" site**
- **pieces of files distributed among peers of network**
- **"tracker" server knows who has what pieces**
  - coordinates all transfers but does not have any of the file contents
- **clients download blocks of file from multiple sources in parallel**
  - blocks have cryptographic checksum to verify correct content
- **downloaded blocks also then uploaded to others**
  - download rate limited by upload rate: have to contribute
  - tracker knows download and upload statuses

    balances traffic, favors sites that are cooperating
- **blocks reassembled by client**
  - when client has the whole file, it can be a seed for further transfers
- **much faster than single server for right kind of use**
  - less vulnerable to flash crowds
  - but takes time to get started, can't do streaming, etc.

Internet telephony

- **Voice over IP**
  - package speech in IP packets
  - may connect to public telephone network on each end
  - strict requirements on delay (latency), jitter (variable delay), error handling, etc.
- **lots of commercial providers (AT&T, Cablevision, Verizon, Vonage,...)**
  - alternative to conventional telephone service
  - somewhat cheaper, probably less reliable, maybe fewer services

Skype: peer to peer VoIP

- **comes from creators of Kazaa (!),**
  - claims no spyware or adware
- **cost**
  - free within Internet
  - ~2 cents/min to connect to regular phone system
- **security**
  - 256-bit AES to encrypt each call,
  - RSA to establish AES session key
- **proprietary protocol, uses both TCP and UDP**

Copyright issues

- **digital media are intrinsically easy to copy**
  - and hard to protect by technical means
- **peer to peer enables copyright violation on a grand scale**
- **Digital Millennium Copyright Act (DMCA)**
- **test cases**
- **disclaimer**
  - an enormous topic
  - I am not a lawyer (IANAL)

Copyright

- **protects expression, not idea**
- **duration used to be 17 years + one renewal**
- **now life + 70, or 95 for commercial works**
  - (the "Mickey Mouse Protection Act", 1998)
- **"fair use" permits limited copying under some circumstances**
  - criticism, comment, scholarship, research, news reporting, teaching
- **uncertain what fair use really is -- case by case decisions**
- **considerations:**
  - purpose and character of the use
  - nature of the copyrighted work
  - amount and substantiality of the portion used
  - effect of the use on potential market or value of copyrighted work
- **recent copyright laws may prevent some fair uses**
  - can't decrypt to make excerpt for teaching or criticism
  - can't reverse engineer to make copies in different media

DMCA: Digital Millennium Copyright Act (1998)

- **US copyright law: www.copyright.gov/title17, Chapter 12**
- **anticircumvention: illegal to circumvent a technological measure protecting access to or copying of a copyrighted work**
  - limited exceptions for reverse engineering for interoperability, encryption research, security testing
- **illegal to remove or alter copyright notices and management information**
- **"safe harbor": protects ISPs from copyright infringement claims if they follow notice and takedown procedures**

Peer-to-Peer use issues

- **vulnerable to copyright violation lawsuits**
- **decentralized less vulnerable than centralized**
  - no centralized target
    (also decentralized main sites outside USA)
  - not restricted to MP3 files as Napster was
    "substantial non-infringing uses"
  - not invulnerable
    - Grokster sued by RIAA
      RIAA lost appeal in Aug 2004 but won in Jun 2005
    - Grokster now out of business, along with several others
- **Fully distributed (bitTorrent) most general-purpose but still vulnerable**
  - legitimate uses for performance in file sharing
  - can get "takedown" notice even if your computer only holds part of directory and no actual copyrighted content
    may not hold up but still must deal with it

Digital Rights Management (DRM)

- **techniques to control access to and use of digital material**
  - largely unsuccessful
- **CSS (content scramble system) encrypts DVDs to prevent playing except on licensed players (and thus prevent copying)**
  - cracked by "DVD Jon"
- **AACS (advanced access control system) encrypts HD-DVD and Blu-Ray**
  - cracked in 2007
- **Windows Media DRM**
  - cracked in 2006-7
- **iTunes FairPlay**
  - cracked in 2006
- **Sony rootkit on audio CDs (2005)**
  - discovered immediately
- **etc.**

Digital (Rights or Restrictions?) Management

- **a disguised form of vendor lock-in?**
- **conflicts with fair use**
  - prevents legitimate operations like time/space shifting, media conversion, backup, ...
- **obsolescent technology may cause things to be lost**
- **incompatible systems make users unhappy**
  - may cause more trouble that it's worth
- **pragmatically, DRM doesn't work and probably can't**
  - long history of failed / cracked systems

Technology meets law/policy/economics/politics

- **should there be laws controlling peer to peer technology?**
- **should content providers like RIAA be permitted to install search (& destroy) software on home computers?**
- **should universities be required to enforce file-sharing laws?**
- **should VoIP be regulated by the FCC?**
  - should VoIP suppliers have to provide services like 911?
  - should VoIP suppliers pay taxes and fees, and for connectivity to public telephone network?
  - should VoIP calls be subject to wire-tapping laws like regular phones?
- **should common carriers like Verizon be permitted to discriminate against traffic from other VoIP suppliers?**
  - should there be different prices and policies for different kinds of traffic?

Course Summary

(not guaranteed exhaustive
use Schedule & Assigments page and slides)

Hardware

- **logical/functional/architectural structure**
  - bus connects CPU, RAM, disks, other devices
  - CPU cycle: fetch-decode-execute; kinds of instructions
    toy machine as an example
    different processor families are incompatible at the instruction level
  - von Neumann: architecture; Turing: equivalence of all machines
- **physical implementation; sizes and capacities**
  - chips; Moore's law, exponential growth
- **analog vs digital**
- **representation of information**
  - bits, bytes, numbers, characters, instructions
  - powers of 2; binary and hexadecimal numbers
  - interpretation determined by context
- **it's all bits at the bottom**

Software

- **algorithms: sequence of defined steps that eventually stops**
  - complexity: how number of steps is related to amount of data
    examples of linear, quadratic, logarithmic, n log n, exponential
    (logarithm = number of bits needed to store value)
- **programs and programming languages:**
  - evolution, language levels: machine, assembly, higher-level
  - translation/compilation; interpretation
  - a program can simulate a machine or another program
- **basic programming, enough to figure out what some code is doing**
  - variables, constants, expressions, statements, loops & branches (if-else, while), functions, libraries, components
- **operating systems:**
  - run programs, manage file system & devices
  - virtual memory and caching
  - file systems: logical: directories and files; physical: disk blocks
- **application programs, interfaces to operating system**

Communications

- **local area networks, Ethernet, wireless, broadcast media**
- **Internet: IP addresses, names & DNS, routing; packets**
  - bandwidth
- **protocols: IP, TCP, higher-level; layering**
  - synthesis of reliable services out of unreliable ones
- **Web: URLs, HTTP, HTML, browser**
  - Enabled services:
    - search engines
    - cloud computing
- **security & privacy: viruses, cookies, spyware, …**
  - active content: Javascript, ActiveX
- **cryptography**
  - secret key; public key; digital signatures
- **peer to peer**
  - (very basic idea)

Real world issues

- **legal**
  - intellectual property: patents, copyrights, contracts, licenses
  - jurisdiction, especially international
- **social**
  - privacy, security
- **economic**
  - open source vs proprietary
  - who owns what
- **political**
  - policy issues
  - balancing individual, commercial and societal rights and concerns

Things to take away

- **some skills, some specific technical knowledge**
  - how computers and communications work today
  - what's ephemeral, what's likely to still be true in the future
- **improved numeracy / quantitative reasoning**
  - what makes sense, what can't possibly make sense, and why
    plausible estimates, engineering judgement, enlightened skepticism
- **another way of thinking**
  - how do things work?
  - how *might* something work?
  - you can often figure it out
- **some appreciation of tradeoffs & alternatives**
  - you never get something for nothing
- **some historical perspective**
  - everything derives from what came before
- **informed opinions about the role of technology**

Final exam information

Exam:

Wed. Jan 19, 7:30 p.m, 104 Computer Science Building

- **similar to midterm but twice as long**
- **open notes, problem sets, labs, …**
  - you are allowed to bring copies of articles posted under "Reading", but overkill. May wish to bring BRIEF notes on important ideas
- **"Laptop computers as well as hand held electronic communications devices (e.g. cell phones, iPods, BlackBerrys, iPhones, etc.) are forbidden in final examination rooms." Rules of University**
- **bring a calculator if you can — it might make something easier**

Preparing for final exam

most important:

lecture content: slides + your notes

problem sets: understand correct answers and where you went wrong

labs: present some important concepts

readings:

some to assist you with lecture content

some to expose you to other ideas or history

should just have main idea of these

there will be a few readings posted to support topics treated Monday and today

watch "Announcements" web page!!!

Q/A session: check Announcments for schedule

Also check Announcements for

Our office hours

Old final exam

Solutions

Other information on exam

EVALUATIONS- PLEASE GIVE FEEDBACK!

Written comments help most – how improve course?

Course must change to keep up - need your thoughts on:

- more topics or fewer?
- broader or deeper?
- different topics?

  like what?