Wireless systems

- how radio works
- radio spectrum allocation
- examples
  - cell phones
  - RFID: prox, E-ZPass, store tags, passports, ...
  - 802.11 (WiFi)
  - Bluetooth
  - GPS
  - cordless phones
  - ...
- tradeoffs
  - spectrum, power, range, size, weight, mobility
- non-technical issues
  - regulation, competition, ...

Radio

- electromagnetic radiation to carry information
  - without wires => "wireless"
- radiation is a wave of a particular frequency (in Hz)
- "modulate" the wave to impose information on it
  - amplitude (AM): change the power level
  - frequency (FM): change the frequency around nominal value
- received signal strength varies directly with power level
- received signal strength dies off with square of distance
- higher frequencies go shorter distances
Cell phones 101

- all phones are part of the public switched telephone network
- a cell phone is connected by radio instead of wires
- moves long distances, at high speed, appears out of nowhere
- shares a very limited radio frequency spectrum with others
- operates with low power because it uses batteries

- this makes life complicated
Cells (a very idealized picture)

- divide geographical area into cells (notionally hexagonal)
- each cell has an antenna, handles all cell phones in its area
- available radio spectrum is divided into channels
  - two channels for one conversation, one for each direction
  - competing carriers can all operate
  - each has its own independent equipment
- each cell gets 1/7 of the channels
  - adjacent cells can't use the same channels because of interference
  - non-adjacent cells can re-use channels

How it works

- when a phone is turned on, it broadcasts its ID ("registration")
  - nearest base station notices, validates with home system
  - registration uses encryption for fraud prevention
- when phone is called, home system knows where it is
  - contacts base(s) where it is
  - bases broadcast to where last seen ("paging")
- phones talk to base with strongest signal
  - base and phone communicate over 2 agreed-upon channels (up, down)
  - phones continuously adjust power level to signal strength at base
  - uses less battery, creates less interference for other phones
- phones move from base to base and from system to system
  - base initiates handoff when signal gets weak
  - phone picked up by base with strongest signal
  - elaborate protocols at all levels
How it works, continued

• **multiple frequency bands** (different in different parts of the world)
  - divided into channels (frequency multiplexing)
    - digital phones multiplex several calls on one channel (GSM)
    - or spread calls out over the whole spectrum (CDMA)
  - phones usually support multiple bands
• **channels carry both voice and control information** (including data)
  - digital speech is highly compressed (~1 bit/speech sample)
  - elaborate coding & error correction for speech & control information
  - power turned off when nothing is being sent
• **GSM phones store user info on removable flash memory card**
  - SIM (Subscriber Information Module)
  - may be able to replace card to use in a different environment
• **most of the world uses** **GSM**
  - in USA, AT&T & T-Mobile use GSM; Verizon & Sprint use CDMA

Technology meets politics again

• **should texting while driving be illegal (and enforced)?**
  - how about just driving while talking?
• **where determines where cell phone towers are permitted?**
  - property rights versus eminent domain
• **should cell phone jammers be legalized?**
  - in theatres, trains, etc.
• **location tracking and surveillance**
  - FCC mandates that cell phone can be locatable within 125 m radius
  - should real-time location info be available to law enforcement, etc.?
  - how should this evolve as GPS becomes universally available?
  - who can have access to what cell phone records when?
Search engines

- browser uses a FORM to send a query to a server
  - e.g., google.com
- server runs a program to extract query from form
- finds pages that contains word(s) of query
- generates HTML
- returns page to client

- server needs to know what pages contain relevant words
- continuously crawls the web collecting pages
- builds big database that tells what pages contain any given word

- basic problem: scale
  - lots of pages, lots of words, lots of queries

Server processes

- 3 basic processes going on in parallel
  - respond to incoming queries by looking up words in database
  - crawl web looking for new pages
  - extract words from new pages and insert into database
Fetching new pages

- start with a list of likely URLs
- fetch data from next URL from the list
  - obey robot exclusion standard
- extract parts to be indexed, deliver to index builder
- extract URLs
- delete duplicate URLs (ones seen recently)
- delete irrelevant ones (advertisements, ...)
- add remaining URLs to end of list
- go back to the top

questions:
  - how to start
  - how to detect duplicates quickly
  - what to preserve (text, .html files, .txt files, PDF, gif/jpg, ...)
  - how to avoid overloading big/popular sites

Building and searching an index

- for a new page that has just been fetched:
  - isolate words (discard HTML tags, etc.)
  - handle upper and lower case, accents, punctuation,
    other languages and character sets, ...
  - for each word
    add URL to list for that word
    add word position within the page to the list for the URL

- to look up a single word query:
  - go to the list for the word
  - collect all URLs
  - sort them into order by weighting function
    importance, frequency, ...

- queries with multiple words:
  - collect URL lists, combine them, weight them
Hashing: an algorithm to look things up quickly

- problem: how to look up one word in 1 billion words, really fast
  - binary search would be 30 probes if names were sorted
  - sorting takes too long if it has to be updated

- hashing: scramble the word into an integer
  - between 0 and N
  - so that hash values of potential words are spread out uniformly

- store all words with the same hash value together
- searching for a word then requires only
  - compute the hash value
  - look at the list of previously-stored words with that hash value

- example hashing algorithm: add up the numeric values of all the characters in the word

Ranking search results

- how to get the most likely results on the first page (at the top)
  - most people look only at the first few results
  - need for very high precision (relevant documents in the top 10 or so)

- Google uses proprietary "page rank" algorithm based on link structure of web
  - pages that are cited often move higher
  - pages that are cited by higher ranked sites move higher
  - anchor (<a href=...>) text gives more information
  - proximity of search terms within page
  - ...
- other search engines have analogous techniques
- have to defend against attempts to inflate rankings
Privacy and copyright issues

- what privacy standards apply to search engines?
  - how can private / incorrect information be purged?

- search engines versus government
  - should Yahoo have released information about Chinese dissidents to the Chinese government?
  - does Google’s acquisition of DoubleClick concentrate too much information about individuals in one place?
  - can query logs be subpoenaed?
    - AOL’s release of "sanitized" information permitted identification of individuals from their queries

- copyright
  - Viacom v YouTube: vicarious liability or DMCA safe harbor?
  - should newspaper stories be indexed without permission?

- trademarks
  - can someone buy someone else’s trademark as an advertising keyword?
    - e.g., Microsoft buys "iPod"

- ...
Hardware

• logical/functional/architectural structure
  - bus connects CPU, RAM, disks, other devices
  - caching
  - 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
    linear: searching, counting, ...
    quadratic: simple sorting
    logarithmic: binary search (logarithm = number of bits needed to store)
    n log n: quicksort
    exponential: towers of Hanoi, traveling salesman problem, ...

• 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
  - 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
  - caching
- security & privacy: viruses, cookies, spyware, ...
  - active content: Javascript, ActiveX
- cryptography
  - secret key; public key; digital signatures
- compression; error detection & correction
- case studies and the real world
  - prox cards, peer to peer, cell phones, search engines, ...

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  (watch the web page!!!)

• **Thursday January 21 1:30pm,  Friend Center 101**
  - Q/A session January 17; watch the web page for schedule
  - come to office hours or send mail or drop in; watch the web page

• similar to midterm but twice as long

• open notes, problem sets, labs, ...

• bring a calculator if you can — it might make something easier

• **hints**
  - I’m usually looking for something *brief* that shows that you understand or can reason
  - if you’re writing or calculating a lot, you’re likely on the wrong track
  - questions are meant to test understanding of basic ideas and critical distinctions
    meant to be simple and straightforward, not complicated, *if you understand*
    not meant to be tricky or rely on obscure facts
  - think about plausibility and where I’m likely coming from
  - if it still seems ambiguous, say "I’m assuming this..." and carry on