A Little about me

- Faculty in WPI Computer Science
- Research interests:
  - graphics, mobile computing/wireless and mobile graphics
- How did I get into mobile and ubiquitous computing
  - 3 years in wireless LAN lab (pre 802.11)
  - Designed, simulated, implemented wireless protocols
  - Group built working wireless LAN prototype (pre 802.11)
- Computer Systems/Electrical/Computer Science background
  - Hardware + software
About this class (Administrivia)

- Class goal: give overview, insight into hot topics, ideas and issues in mobile and ubiquitous computing
- Focus: ideas implemented using smartphone
- Meet for 14 weeks, break on March 5 (term break)
- Seminar style: I will present, YOU will present papers
- See big picture through focussed discussions
- Course website: http://web.cs.wpi.edu/~emmanuel/courses/cs525m/S13/
- Projects: 1 or 2 assigned, 1 big final project
- This area combines lots of other areas: (networking, OS, software, machine learning, programming, etc)
  - Most people don’t have all the background!!
  - Independent learning is crucial
  - Projects: Make sure your team has requisite skills
Administrivia: Papers

- **Week 1**: I will present (today)
- **Weeks 2 – 13**: You will present
  - I will present background material on the week’s topic, other stuff
  - 4 student presentations from Required Papers for the week
  - Discussions
- Student presentations: ~25 mins + ~10 mins discussion
- 15-min break halfway
Formal Requirements

- *What do you have to do to get a grade?*
- Seminar: Come to class + Discuss!! Discuss!! Discuss!!
- Present 1 or 2 papers
- Email me 1-page summaries (in ASCII text) for weekly papers
- Do assigned project(s)
- Do term project: 5-phases
  - Pick partner + decide project area
  - Submit intro + related work
  - Propose project plan
  - Build, evaluate, experiment, analyze results
  - Present results + submit final paper (in week 14)
- Grading policy: Presentation(s) 20%, Class participation 10%, Assigned Projects 20%, Final project: 40%, Summaries: 10%
Written Summaries

- Email to me *before class* in ASCII text. No Word, Latex, etc
- Summarize key points of all 4 papers for week
  - Main contributions
  - Limitations of the work
  - What you like/not like about paper
  - Any project ideas?
- Half a page max per paper
- Summary should quickly refresh memory in even 1 year’s time
  - Include main ideas/algorithms, results, etc.
- See handout for more details
Students: Please Introduce Yourselves!

- Name
- Status: grad/undergrad, year
- Relevant background: e.g. coal miner 😊
- Relevant courses taken:
  - *Systems*: Networks, OS,
  - *Advanced*: machine learning, advanced networks, etc
- What you would like to get out of this class?
  - Understanding a hot field
  - Just a class for masters degree/PhD
  - Looking for research area, masters thesis, PhD thesis
  - Compliments your current research interests/publications
  - My spouse told me to 😊
Next... Overview

- Brief overview of area topics/issues
- Define/motivate area, excite (or discourage) you
- Provoke thinking:
  - More questions, problems than solutions
- Sample of topics to be covered in class
- Topics covered in more detail later
- Students may only understand some topics in today’s overview
Mobile computing

- Mark Weiser, Xerox PARC CTO
- 1991, articulated vision (and issues) for ubiquitous and mobile computing
- Weiser’s Vision:
  
  “Environment saturated with computing and communication capabilities, with humans gracefully integrated”

- Core idea: Invisible hardware/software that assist human
  - **Hardware**: smart phones, sensors, tablets, wearable devices, etc
  - **Software**: Voice recognition, Mobile OS, Networking/communication software, protocols, etc
- Weiser’s vision ahead of its time, available hardware and software
- Example: voice recognition was not available then
- Today, envisioned hardware and software is available
Mobile vs Ubiquitous Computing

● Mobile computing
  • deals mostly with *passive* network components
  • Human computes seamlessly while moving, continuous network connectivity
  • Human initiates all activity, clicks on apps!!
  • Example: Using *foursquare.com* on smart phone

● Ubiquitous computing
  • introduces collection of specialized assistants to assist human in tasks (reminders, personal assistant, staying healthy, school, etc)
  • Networked array of *active* elements, sensors, software agents, artificial intelligence
  • Builds on *mobile computing* and *distributed systems* (more later)
Ubicomp Sensing

- Sense what?
  - *Human*: motion, mood, identity, gesture
  - *Environment*: temperature, sound, humidity, location
  - *Computing Resources*: Hard disk space, memory, bandwidth
  - *Ubicomp example*:
    - *Assistant senses*: Temperature outside is 10F (environment sensing) + Human plans to go work (schedule)
    - *Ubicomp assistant advise*: Dress warm!

- Sensed environment + Human + Computer resources = *Context*
- *Context-Aware* applications adapt their behavior to context
Sensing the Human

- Environmental sensing is relatively straight-forward
  - Use specialized sensors for temperature, humidity, pressure, etc

- Human sensing is a little harder (ranked easy to hard)
  - **When**: time (Easiest)
  - **Where**: location
  - **Who**: Identification
  - **How**: (Mood) happy, sad, bored (gesture recognition)
  - **What**: eating, cooking (meta task)
  - **Why**: reason for actions (extremely hard!)

- Human sensing (gesture, mood, etc) easier with cameras than sensors

- Research in ubiquitous computing integrates location sensing, user identification, emotion sensing, gesture recognition, activity sensing, user intent
Using context

- Accurately determining context = timely feedback
- Inaccurately inferred context = distraction

Example:

- If user is driving and systems thinks they are relaxing on their couch, system may send pop-up messages about doing housework (distracting)
Mobile Devices

- Smart phones (Blackberry, iPhone, Android, etc)
- Tablets (iPad, etc)
- Laptops
SmartPhone Hardware

- Quad core CPUs, Powerful GPUs
- Mobile GPUs support OpenGL ES
- **OpenGL ES** for graphics, OpenCL for GPGPU

<table>
<thead>
<tr>
<th></th>
<th>Nexus 4</th>
<th>Galaxy S III</th>
<th>iPhone 5</th>
<th>Moto Droid</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>APQ8064</td>
<td>MSM8960</td>
<td>Apple A6</td>
<td>OMAP 3430</td>
</tr>
<tr>
<td></td>
<td>1.7 GHz <strong>Quad-core</strong></td>
<td>1.7 GHz <strong>Dual-core</strong></td>
<td>1.3 GHz <strong>Dual-core</strong></td>
<td>600 MHz</td>
</tr>
<tr>
<td>GPU</td>
<td>Adreno 320</td>
<td>Adreno 225</td>
<td>PowerVR SGX543MP3</td>
<td>PowerVR SGX 530</td>
</tr>
<tr>
<td></td>
<td>OpenGL ES 3.0 <strong>OpenCL 1.2</strong> OpenVG 1.1</td>
<td>OpenGL ES 2.0</td>
<td>OpenGL ES 2.0 Shader Model 4.1</td>
<td>OpenGL ES 2.0 Shader Model 4.1</td>
</tr>
<tr>
<td></td>
<td>NA 40-45 GFLOPS</td>
<td>400 MHz 19.2 GFLOPS</td>
<td>266 MHz (<strong>Tri-core</strong> 25.5 GFLOPS)</td>
<td>200 MHz (1.6 GFLOPS)</td>
</tr>
</tbody>
</table>

**GLOPS:** floating-point operations per second

Comparison courtesy of Qian He (Steve)
SmartPhone OS

- Android leader in SmartPhone OS since Q4 2010

**[Diagram showing sales of different smartphone OSes in Q1, Q2, Q3, and Q4 of 2009 and 2010.]**

*Courtesy Margaret Butler*
SmartPhone OS

- Now?
  - Over 80% of all phones sold are smartphones
  - Android share 75% worldwide in Q4 2012
Android System Architecture

- Applications
  - Home
  - Contacts
  - Phone
  - Browser
  - ...

- Application framework
  - Activity manager
  - Window manager
  - Content providers
  - View system
  - Package manager
  - Telephony manager
  - Resource manager
  - Location manager
  - Notification manager

- Libraries
  - Surface manager
  - Media framework
  - SQLite
  - OpenGLES
  - FreeType
  - Webkit
  - SSL
  - libc

- Android runtime
  - Core libraries
  - Dalvic virtual machine

- Linux kernel
  - Display driver
  - Camera driver
  - Flash memory driver
  - Binder (IPC) driver
  - Keypad driver
  - WI-FI driver
  - Audio drivers
  - Power management
Mobile Devices: Droid

- This class: Google Droid as main mobile device
- Google donated Motorola Droid smart phones
- One assigned project and final project based on Droid
  - Connects to Verizon network, WLAN or Bluetooth
  - Google Android OS (updated 4.0.4, ice cream sandwich)
  - 5 MegaPixel camera
  - Streaming video: mpeg, H.264
  - GPS, google maps, etc
  - Sensors: accelerometer, proximity, eCompass, ambient light
Sensor Node

- Sensor? Think of automatic doors
- Automatic door sensor has single purpose: detect human
- New multi-functional sensors, programmable for various tasks (intrusion detection, temperature, humidity, pressure, etc)
- Low cost ($1 per sensor), 1000’s per room, attach to objects
- Capabilities: Sense, process data, communicate with sink node
- Constraints: Small CPU, OS, programmable

(courtesy of MANTIS project, U. of Colorado)  
RFID tags  
Tiny Mote Sensor, UC Berkeley
Wireless Sensors for Environment Monitoring

- Embedded in room/environment
- Many sensors cooperate/communicate to perform task
- Monitors conditions (temperature, humidity, etc)
- User can query sensor (What is temp at sensor location?)
Classic Wireless Sensor Network

- **ZebraNet**: Novel studies of zebra migration and inter-specie interactions
- **Basic idea**: Put sensors on zebras, study them
Ubiquitous Computing: Wearable sensors for Health

remote patient monitoring

Body Worn Activity Trackers
Wellness Smart (Bluetooth) Devices
Worldwide cellular subscriber growth

Global Subscriber Base (in million)

Source: Informa
Explosion of Devices

- *Recent Nokia quote:* More cell phones than tooth brushes
- Many more sensors envisaged
- *Ubiquitous computing:* Many computers per person
Definitions: Portable, mobile & ubiquitous computing

- **Distributed computing**: system is physically distributed. User can access system/network from various points. E.g. Unix, WWW. (huge 70’s revolution)

- **Portable (nomadic) computing**: user intermittently changes point of attachment, disrupts or shuts down network activities

- **Mobile computing**: continuous access, automatic reconnection

- **Ubiquitous (or pervasive) computing**: computing environment including sensors, cameras and integrated active elements that cooperate to help user

- Class concerned mostly with mobile and ubiquitous computing
Distributed Computing

- **Distributed computing example:** You, logging in and web surfing from different terminals on campus. Each web page consists of hypertext, pictures, movies and elements anywhere on the internet.
- Note: network is fixed, Human moves
- Issues:
  - Remote communication (RPC),
  - Fault tolerance,
  - Availability (mirrored servers, etc)
  - Caching (for performance)
  - Distributed file systems (e.g. Network File System (NFS))
  - Security (Password control, authentication, encryption)
Nomadic computing
Nomadic Computing

- **Portable (nomadic) computing example:** I own a laptop. Plugs into my home network, sit on couch, surf web while watching TV. In the morning, wake up, un-plug, shut down, bring laptop to school, plug into WPI network, start up!

- **Note:** Network is fixed, device moves and changes point of attachment.

- **Issues:**
  - File/data pre-fetching
  - Caching (to simulate availability)
  - Update policies
  - Re-integration and consistency models
  - Operation queuing (e.g. emails while disconnected)
  - Resource discovery (closest printer while at home is not closest printer while at WPI)

- **Note:** much of the adaptation in “middleware” layer
Mobile Computing Example

- **Mobile computing:** Sarah owns SPRINT PCS phone with web access, voice, SMS messaging and can run apps like facebook and foursquare. She remains connected while she drives from Worcester, Massachusetts to Compton, California.

- Note: Network topology changes, because sarah and mobile users move. Network deals with changing node location.

- Issues
  - Mobile networking (mobile IP, TCP performance)
  - Mobile information access (bandwidth adaptive)
  - System-level energy savings (variable CPU speed, hard disk spin-down, voltage scaling)
  - Adaptive applications: (transcoding proxies, adaptive resource management)
  - Location sensing
  - Resource discovery (e.g. print to closest printer)
Ubiquitous Computing Example

- **Ubiquitous computing**: John is leaving home to go and meet his friends. While passing the fridge, the fridge sends a message to his shoe that milk is almost finished. When John is passing grocery store, shoe sends message to glasses which displays “BUY milk” message. John buys milk, goes home.

- **Core idea**: ubiquitous computing assistants **actively** help John

- **Issues**:
  - Sensor design (miniaturization, low cost)
  - Smart spaces
  - Invisibility (room million sensors, minimal user distraction)
  - Localized scalability (more distant, less communication)
  - Uneven conditioning
  - Context-awareness (assist user based on her current situation)
  - Cyber-foraging (servers augment mobile device)
  - Self-configuring networks
Smartphone Sensors

- Typical smartphone sensors today
  - accelerometer, compass, GPS, microphone, camera, proximity

Future sensors?
- Heart rate monitor,
- Activity sensor,
- Pollution sensor,
- etc
Mobile CrowdSensing

- **Internet of things:** Sensing data from consumer-centric devices including
  - Smartphones (iPhone, Google Nexus,)
  - Music players (iPods)
  - Sensor embedded gaming systems (Wii, Xbox, kinect)
  - In-vehicle sensors (GPS)
  - Body-worn sensors (e.g. fitbit, Nike+)
- **Mobile crowdsensing:** sense these devices
  - personal, community- and Internet-wide
- Sensing applications at community scale possible
Mobile CrowdSensing

- **Personal sensing**: phenomena pertain to individual
  - E.g: activity detection and logging for health monitoring

- **Group**: friends, co-workers, neighborhood
  - GarbageWatch to improve recycling, neighborhood surveillance

- **Community sensing (mobile crowdsensing)**:
  - Large-scale phenomena monitoring
  - Many people contribute their individual readings
  - **Examples**: Traffic congestion, air pollution, spread of disease, migration pattern of birds, city noise maps
Mobile CrowdSensing Types

- **Participatory sensing**: *active* involvement of individuals (e.g. taking a picture, reporting potholes)
- **Opportunistic sensing**: *passive* user involvement (continuous location sampling without explicit user action)
Mobile Crowdsensing Enablers

- **Cheap phone sensors:** are now available
- **Easily programmable:** Smartphones are easily programmable (Android SDK, PhoneGap, AppInventor)
- **Easy deployment:** App stores make deployment easy
- **Cloud resources:** Compute-intensive or storage-hungry applications can be offloaded
Sense What?

- **Environmental**: pollution, water levels in a creek
- **Transportation**: traffic conditions, road conditions, available parking
- **City infrastructure**: malfunctioning hydrants and traffic signs
- **Social**: photoblogging, share bike route quality, petrol price watch
- **Health and well-being**:
  - Share exercise data (amount, frequency, schedule),
  - share eating habits and pictures of food
Sensing with Smartphones vs Motes

- **More resources:** Smartphones have much more processing and communication power
- **Easy deployment:** Millions of smartphones already owned by people
  - Instead of installing sensors in road, we detect traffic congestion using smartphones carried by drivers
- **Time-varying data:** population of mobile devices, type of sensor data, accuracy changes often due to user mobility and differences between smartphones
Sensing with Smartphones vs Motes

- **Reuse of few general-purpose sensors:** While sensor networks use dedicated sensors, smartphones reuse relatively few sensors for wide-range of applications
  - E.g. Accelerometers used in transportation mode identification, pothole detection, human activity pattern recognition, etc

- **Human involvement:** humans who carry smartphones can be involved in data collection (e.g. taking pictures)
  - Human in the loop can collect complex data
  - Incentives must be given to humans
Mobile Phone Sensing Architecture

- **Sense:** Phones collect sensor data

- **Learn:** Information is extracted from sensor data by applying machine learning and data mining techniques

- **Inform, share and persuasion:** inform user of results, share with group/community or persuade them to change their behavior
Sensor Processing

- **Machine learning** commonly used to process sensor data
  - Action to be inferred is hand-labelled to generate training data
  - Actual data is mined for combinations of sensor readings corresponding to action

![Diagram showing raw data, extracted features, and classification inferences](image)
LOCALIZED ANALYTICS

- Sensor data processed before sending to server
- Saves energy, time, and processing at backend
- **Examples:** filtering outliers, context inference (user state)
Other Issues

- **Resources (Bandwidth, computational):**
  - heterogeneous and time-varying

- **Privacy, security and data integrity:**
  - Contributing user’s information (e.g. location) becomes known (e.g. criminals learn user’s path to work)
  - Criminals can contribute bad data to repository
Location-aware mobile computing apps

- Focus mostly on mobile and ubiquitous computing apps that use Smart Phone and Internet connectivity.

- Example: Location-aware mobile computing apps. Issues:
  - **Entropy**: Inferring how close two Facebook friends are based on locations mutually visited
  - **Anonymity**: May not want all Facebook friends to know where I am
  - Automatically anonymize location information hierarchically
    - Fact: User is at Starbucks, 180 Main St, Worcester, MA
    - Status update to friend A: Emmanuel is at “coffee shop”
    - Status update friend B: Emmanuel is at “Starbucks, 180 Main St, Worcester”
    - Algorithms to automatically generate status update (based on closeness)
Internet as a data source for Location-aware apps

- [Identifying the Activities Supported by Locations with Community-Authored Content, Dearman and Truong, Univ. of Toronto]

- User at location X would like to make location-based queries
  - What activities can I do here?
  - What’s a good close place to do X activity (e.g. soccer)

- **Solution:** Yelp is a community-authored reviewer website for restaurants, activities, etc

- Yelp has: activities + location + goodness of venues

- Scrape + mine yelp: augment with location as searchable tag
Location-Aware Apps

- Easier location check-in
  - Ubicomp 2010 video p395
Context-Aware Search

- [Hapori: Context-based Local Search for Mobile Phones using Community Behavioral Modeling and Similarity, Nicholas D. Lane, Dartmouth College]

- Goal: Improves Internet search results using context, such as weather, age, profile of user, time, location and profile of other users to improve search.

- Example: a teenager gets a completely different set of recommendations from and elder.
Mobile Social Networking

- Partipatory sensing: Many people cooperating on a task
- *Classic example:* Comparative shopping
- At CVS, ready to buy toothpaste. Is CVS price the best locally?
- Phone has software to query other members of my network
- People at other local stores (Walmart, Walgreens, etc) respond with prices
UCLA Partipatory Sensing Video

- Demo from UCLA
Mobile Social Networking

- Smart phones have many sensors, cameras, etc
- Imagine ability to access other people’s phones: *Phone Sensing*
- Like a telescopic lens into different locations: *Microblogging*
Sensing Human Behavior

- [Social Sensing for Epidemiological Behavior Change, Anmol Madan et al, MIT Media Lab]

- **Goal:** infer how falling sick affects the [mobile/network] behaviors of human beings.
- **Examples:** Changes in call rates or visiting low entropy places more could mean person is sick
- Statistics of number of calls, co-location, proximity, WLAN and bluetooth entropy found to be good predictors of illness.
- Findings could be used as an early warning tool.
- If strong inference, then nurse could call the person
Energy Efficiency

- Most resources increasing exponentially except battery energy (ref. Starner, IEEE Pervasive Computing, Dec 2003)

- Strategies:
  - Energy harvesting: Energy from vibrations, moving humans
  - Scale content: Reduce image, video resolutions to save energy
  - Better user interface: Estimate and inform user how long each potential task will take
    - E.g: At current battery level, you can either type your paper for 45 mins, watch video for 20 mins, etc
Popular Wireless Networks Types

- **Cellular Network:** Wide area wireless network operated by Sprint, Verizon, AT&T, 3G/4G

- **WLANs:**
  - **Infrastructure networks:** wired backbone (Internet), wireless last hop. E.g WPI wireless LAN, New: mesh networks
  - **Ad hoc networks:** all wireless, no backbone, no order known in advance. Few deployed examples.. .futuristic

- **Bluetooth:** Short range communications, printers, headsets, etc

- **Sensor networks:** self-organizing network of large numbers of cooperating sensors deployed inside phenomenon. E.g. even more futuristic. Many research projects
Mobile Measurement Studies

- Previous versions of class covered wireless protocols, standards
- This version: focus on measurement studies
  - How existing apps, mobile web, wireless networks are being used
Wireless Security

- Insecure wireless network
  - Wireless signals leak beyond building confines (less secure)
  - Security standards like Wireless Encryption Protocol (WEP) have significant demonstrated flaws
  - Mobility: tracking perpetrators is hard

- Mobile devices/OS almost now as complex as PCs
  - Subject to many of the same vulnerabilities as PCs?

- Mobile devices designed to be carried around => more prone to theft or misplacement

- Mobile devices easily stolen, tampered with (drunk employees)

- Anderson: over 90% of security breaches caused by lapses in physical security:

  - Example: drunk employee at bar with laptop
WLAN Vulnerabilities

- Protocol (e.g. 802.11) vulnerabilities:
  - **Rogue APs:** Attacker inserts access point, hijacks mobile nodes
  - **Jamming:** ISM bands prone to that, microwaves, etc
  - **Induce congestions, collisions:** Induce collisions, congestion, disobey protocol. Delay bad for multimedia
  - **Exhaustion:** Keep sending packets to wireless node, prevent sleep modes, drain battery, DoS
  - **Packet header manipulation:** e.g. sequence/ACK Nos.
Wi-Fi Privacy Ticker

- [Sunny Consolvo et al, Intel Labs Seattle, University of Washington]

- Many wireless security/privacy breeches occur
- Many open problems. Some too hard to solve for now
- Examples:
  - website A may send your information to website B without your knowledge
  - New google search sends typed characters BEFORE you hit enter
- Solution: Alert to user when info is being transmitted unsecurely
- Ticker streams violations of user's pre-defined breeches
- “Breeches“ identified and importance customizable
- Wi-Fi Ticker increased user awareness about security
- Even highly techno-savvy learned about breeches
Final Words

- This is a special topics graduate class
- **Special Topics:** I have picked selected topics that are hot.
- Coverage is not complete
- Graduate class so graduate level work/effort is expected
- **Seminar style classes:** You get out what you put into them
Homework

- **Today:** Sign up for papers to present
  - **Procedure:** Sign up sheet passed around, simply sign
- Summaries of week 2 papers (Healthcare and Personal assistants): due before next class
- **Two weeks:** decide project area and partners (if any)
  - Project? Never too early to start thinking about project, talking to me.