Mobile and Ubiquitous Computing

Due to device miniaturization and wireless networking.

Communication is with base stations or in the form of *ad hoc* networking.

Subfields:

- handheld computing—blurring of distinction between PDAs, mobile phones and purpose-build handheld devices.
- Ubiquitous (pervasive) computing—computing everywhere.
- Wearable computing—wear computing devices.
- Context-aware computing—aware of context, e.g. a "local" printer.
- Smart spaces—physical space with embedded devices designed to provide service in that environment. Devices appear and disappear.

Device Model:

Computing devices have different properties than traditional devices:

- limited energy—power consumption is an issue
- resource constraints
- sensors—measure physical parameters.

  Example are *motes*, which are intended for autonomous operation in applications such as environmental sensing. Also known as "smart dust". Some run TinyOS operating system.

  Another example is a camera phone, which can be used to read (sense) coded values such as bar codes.
- actuators—software-controllable devices—robotics

Natural *volatility* in mobile environments as devices come and go. Opportunities for fixed and ad hoc connectivity change.

Need discovery services such as Jini.

Focus

From amongst the wide variety of work. Look at four topics:

1. Wireless and the Web
2. Power Consumption and Energy Aware Applications
3. Context-Awareness, Ubiquitous computing
4. Sensor Networks

Wireless and the Web

"Supporting the WWW in Wireless Communications Through Mobile Agents" Hadjiefthymiades, Matthaiou and Merakos, Mobile Networks and Applications, August 2002.

Build on proxy approach. See Figure 1. Base stations cache content for mobile hosts.

Issues:

1. How to handle caches when clients move? Figure 2 shows that base/support station cache contents move.
2. How much of cache contents to move? Full or partial? See Fig 2.
3. Relates to the problem of path prediction algorithm (PPA).
4. Also the issue of merging cache contents.

Implement wireless architecture with IBM Aglets (Java mobile agents) framework.

Show that basic framework can be built with these mobile agents. Details less important.

Issues with Mobile Devices for Web Content

- Connectivity. Getting better with 3G networks.
- Physical limitations of devices - CPU, memory, display, input devices, power supply.
- high percentage of sensitive (personal) information.
- intent — not to access all content, but to answer specific questions.

Solutions

- content delivery
  - delta encoding
  - compression
  - image transcoding (distillation)
- base proxy server does work for the mobile device (DNS lookups, prefetch objects)
- reuse cached object for use with "back" button
- WebViews: macros of web navigations from a wired connection, then apply the macros on a mobile device.
- content summarization—either by original content provider or by proxies using heuristics

Caching not as effective when wireless bandwidth is poor because of discrepancy between wired and wireless rates.

Wireless Application Protocol

WAP-enabled phone usability study in 2000 said 70% of 20 trial users would not buy the technology. Jacob Nielsen: WAP “Wrong Approach to Portability”. But using slower technology at the time.

WAP 1.0 1998. WAP 2.0, January 2002. By November, 2005 WAP organization no longer existed. Consolidated into the Open Mobile Alliance.

- support for IP, TCP, TLS and HTTP. “interoperable optimizations”
- WAP application environment (WAE) or “WAP browser”, XHTML Mobile Profile the language for WAP 2.0

WAE supports style sheets (CSS).

WAP Programming Model

See Fig 1 and 2.

Generally a “pull” model based on user requests.

WAP 2.0 does not require a WAP gateway (as WAP 1.0 did), but can use a WAP proxy for enhanced features—location, privacy and presence based services.

One example is push-based services where proxy pushes content to mobile devices.

WAP Protocol Stack

WAP 1.0, many specific protocols (see Fig 3).

- Wireless Session Protocol (WSP)
- Wireless Transport Protocol (WTP)
- ...

WAP 2.0 for wireless networks supporting IP (see Fig 4):

- Wireless Profiled HTTP (WP-HTTP). Message body compression, establishment of secure tunnels.
- Transport Layer Security (TLS) - wireless profile of TLS.
- Wireless TCP (WP-TCP) - connection-oriented services.

Other Features

- WAP Push - a proxy can push content to devices. Client polling is expensive. Could use timed invalidations so wireless devices know when to wake up.
- User Agent Profile - based on Composite Capabilities/Preference Profiles (CC/PC) work of W3C.
- Data Synchronization
- Pictogram - consistent use of images (WML)

XHTML Mobile Profile (MP)

Content language used with WAP2.0.

Characteristics of Mobile Web Content

See paper from IEEE Workshop on Hot Topics in Web Systems and Technologies.
"Characteristics of Mobile Web Content"
`http://www.cs.wpi.edu/~cew/talks/hotweb06.ppt`

Power Consumption

"Managing Battery Lifetime with Energy-Aware Adaptation" by Flinn and Satyanarayanan, TOCS04.

Energy-aware adaptation—dynamic balancing of energy conservation and application quality.

Use energy-profiling tool called PowerScope. Can use it to determine what fraction of the total energy consumed during a certain time period is due to specific processes in the system.

Can also drill down to determine the energy consumption of specific procedures within a process.

Uses two computers: profiling computer where applications execute and a data collection computer—see Fig. 1.

System Monitor consists of a device driver, which collects sample data and a user-level daemon rpocess that reads samples and writes them to a file. Device driver is a Linux-loadable module. No kernel mods!

Energy Monitor runs on data collection computer. Collection varies, but for most laptop computers, simplest method is to sample the current drawn through the laptop's external power source. Use current rather than voltage.

Battery removed during measurements to avoid extraneous power drain.

Synchronized clocks between the two machines. Energy monitor toggles value of input pin on the profiling computer *immediately after* a power sample. This causes an interrupt and system activity is recorded. Done this way so application power activity is recorded, not the System Monitor. Output pin then toggled on the profiling machine to start next power sample.

Ran benchmarks to validate approach.

Energy-Aware Adaptation

Compared application *fidelity* (metric of quality) vs. energy usage.

Applications:

- video player—Xanim
- speech recognizer—Janus
- map viewer—Anvil
- Web browser—Netscape Navigator

Measured different scenarios:

- baseline: hardware mgmt disabled, highest fidelity
- hardware mgmt enabled, highest fidelity
- hardware mgmt enable, lowered fidelity levels.

Lots of idle time, but allows processor to reduce frequency and save energy

Goal-Oriented Adaptation

Have a specific goal and adapt fidelity to meet a criteria—how long the battery needs to last.

Have a slider to control the goal for battery life.

Context-Awareness

Use physical location to influence the response of a system. Early work at Olivetti Research Labs and Xerox PARC.

Use the word *context* to describe the physical circumstances, including presence of other users.

Work on Active Badge allows phone calls to be directed to a user's current location within an environment.

A nearby printer is automatically found and used (might need instructions on how to find the printer!).

Must take into account error behavior. How close, how often?

Work on more generalized toolkits, such as Context Toolkit. Look at Figs 16.5 and 16.6 for examples of how to use. Doesn't have specific approach on how a widget is implemented.

Location Sensing

Another problem is location sensing. Lots of potential approaches as shown in Fig 16.8.

GPS provides absolute positioning.

Beacons provide proximity, but not exact location.

Security and privacy are a related issue.

Privacy in Location Tracking

"Inference Attacks on Location Tracks" Pervasive 2007. John Krumm, Microsoft Research.

What can be inferred from GPS logs? How anonymous is one's location data?

Inferring home and identity. Used log data from test users. For heuristic algorithms for determining home:

1. Last Destination. Before 3am. Rationale that user ends day at home. Used median of collected values (*why not mode???*)
2. Weighted Median. Weight locations according to time spent there (*end of nowhere??*
3. Largest Cluster. Centroid of largest clustering after clustering technique used.
4. Best Time. Maximize liklihood of being home.

First three best, but only 5% or so accurate when doing lookup with lat/long in Windows LiveTM Search.

Then map name address to name. Also mixed on accuracy.

Sensor Networks and TinyOS

Look at:

"The Emergence of Networking Abstractions and Techniques in TinyOS" by Levis, et al, UCB, NSDI'04. Slides: `http://www.cs.wpi.edu/~cs535/s08/levis:nsdi04.pdf`