;login:

THE MAGAZINE OF USENIX & SAGE
August 2003 • volume 28 • number 4

inside:

CONFERENCE REPORTS

MobiSys 2003

USENIX & SAGE
The Advanced Computing Systems Association &
The System Administrators Guild

conference reports

This issue's reports focus on MobiSys 2003, HoTOS-IX, and the 2003 European Tcl/Tk User Meeting

Our thanks to the summarizers:

for MobiSys 2003
James Mickens

for HoTOS
Ranjita Bhagwan
David Oppenheimer
Amit Purohit
Matt Welsh

for Tcl/Tk
Clif Flynt

Mobisys 2003

The First International Conference on Mobile Systems, Applications, and Services

San Francisco, California
May 5–8, 2003

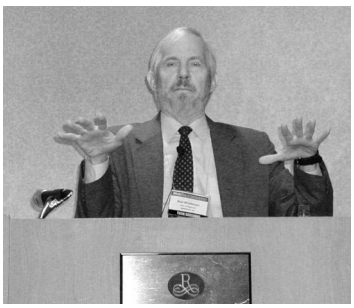
Summarized by James Mickens

KEYNOTE ADDRESS

Design of Wireless Systems-on-a-Chip

Bob Brodersen, Berkeley Wireless Research Center, University of California, Berkeley

A "system-on-a-chip" (SoC) provides integrated components for computation and network communication in a single piece of hardware. SoCs are particularly well suited for small mobile devices which must execute user tasks and transfer data over the Internet. Bob Brodersen argued that the key question for any SoC design is, "What is the cost of flexibility?" There is a fundamental tension between simple designs that do one thing well and larger, more complex designs that provide many features. More flexible designs are often attractive for business reasons (e.g., backward compatibility or the ability to sell one core design for multiple operating environments). However, Brodersen showed that flexible chips are much less efficient than specialized chips, often by several orders of magnitude.



Bob Brodersen

Brodersen examined three types of chips: microprocessors, general-purpose DSPs, and dedicated chips for specialized environments. He noted that supply voltage is relatively constant across all three chip types. However, microprocessors have much higher switched capacitances – in their quest to support high clock rates, they introduce a lot of logic to drive and multiplex the clock signal.

Brodersen then discussed the key relationship between chip area and operations per unit time. Microprocessors have the highest ratio of chip area per operation, whereas dedicated processors have the lowest. Dedicated processors are more energy-efficient because they can extract application-specific parallelism. At any given moment, the vast majority of a dedicated chip's area is actively utilized. In contrast, flexible designs must support many general-purpose operations whose corresponding chip areas are often irregularly utilized. Brodersen drew several conclusions from these results. First, he observed that from an energy perspective, it is better to increase parallelism than to increase the clock rate. He then forcefully noted that, for any given problem, a dedicated hardware solution always has better performance and energy efficiency than a software solution running on a flexible chip. In fact, the popular notion of a "hardware/software" tradeoff is imaginary, because the dedicated hardware solution is always the best! Once again, he stressed that flexible-chip designs are often motivated by business concerns as opposed to efficiency concerns. In general, dedicated parallel processors are the best solution.

Brodersen then described his methodology for rapid prototyping of parallel SoCs. Instead of using a low-level design language like VHDL, Brodersen uses the Simulink and Stateflow programs provided by Mathworks. A chip design is decomposed into timing dataflows and finite-state machines. The design can then be instantiated in an FPGA or ASIC

chip. The FPGA solution is particularly attractive, for several reasons. First, it is very easy to connect multiple FPGAs in parallel. Second, even though the clock rates of FPGAs are not exceptionally fast, FPGAs take full advantage of technological advances in hardware density. Thus, as time progresses, an individual FPGA can contain more and more parallel units.

An audience member asked Brodersen, "How much parallelism can we expect from the real applications that users want to run?" This was an insightful question – if users typically run serial applications, then explicitly parallel chips would not be commonly optimized. Brodersen quickly replied that the current computational paradigm is wrong: Instead of writing serial applications and then trying to run them on parallel chips, we should write explicitly parallel programs for explicitly parallel processors. Brodersen said that the computer community was committing a "great injustice" upon the current generation of programmers by forcing them to learn C. Instead, the community should focus on generating natural methods for describing concurrent phenomenon. Brodersen admitted that his current Mathworks prototyping system is not optimal. However, it is much better adapted to the creation of explicitly parallel applications than many other prototyping systems.

PANEL

How Should We Evaluate Systems Contributions to Pervasive/Ubiquitous Computing?

Keith Edwards, Palo Alto Research Center; Armando Fox, Stanford University (moderator); Anthony LaMarca, Intel Research; Brian Noble, University of Michigan; Yi-Min Wang, Microsoft Research

Armando Fox opened the panel discussion by observing that ubiquitous-computing (ubicomp) research is a combination of two seemingly disparate fields: HCI work and systems research. Some ubicomp research focuses too heavily on the HCI aspect, leading to interesting proof-of-concept projects that cannot handle real-life workloads. Other ubicomp research emphasizes the lower-level systems aspects; unfortunately, this often results in "solutions" that do not truly address the needs of typical users. How can the ubicomp community find a satisfactory methodology for designing and evaluating ubicomp systems?

Brian Noble argued that ubicomp researchers must create metrics that quantitatively describe users' subjective experiences. Noble said that current metrics are often chosen for their mathematical tractability and are only indirectly related to actual user preferences. Keith Edwards echoed these sentiments. He reminded the audience that ubiquitous computing is inherently user-centric, so the ultimate evaluation metric for any such system must be end-user utility.

Yi-Min Wang and Anthony LaMarca agreed that the subjective end-user experience is important. LaMarca's three evaluation criteria for ubicomp systems were robustness, programmability, and manageability. Yi-Min argued that improved fault modeling is critical to the acceptance of ubiquitous computing. For example, people will not buy a house filled with pervasive computers unless they believe that they can fix most errors without professional help, and that a total system failure would not result in personal danger.

The audience members asked many thought-provoking questions. One attendee wondered why system designers and HCI experts should communicate at all – shouldn't they both stick to their respective areas of expertise? Noble responded that each discipline must have an understanding of the problems in the other. Without synergistic interaction between the systems and HCI communities, it is impossible to create ubicomp systems that excel in both aspects.

Another audience member proposed that ubiquitous computing is not about "killer demos." Instead, it is about a "killer lifestyle." The ubicomp community has many vignettes that illustrate simple ways in which pervasive computing is useful, but it has not effectively demonstrated the full power of the ubicomp model. The panel found no easy solutions to this challenge. LaMarca noted that pervasive computing is a new computational paradigm; it forces researchers to revisit traditional notions of what a computer can and cannot do. As the field matures, researchers will discover new ubicomp applications that can change popular perceptions of the technology.

The audience loved an attendee suggestion that conferences like MobiSys offer tutorials on conducting user studies. Many felt that a gentle introduction to the user evaluation process would help systems researchers better understand the role of the end-user in a ubicomp environment.

DEMO/POSTER SESSION

E-textiles

Traditional sensor-node systems use wireless communication protocols and provide each node with an individual power supply. An e-textile is a piece of fabric that contains embedded processors, sensors, and actuators. Unlike a traditional sensor node, an e-textile node communicates via wires that are threaded along the surrounding fabric. E-textile nodes also draw their energy from embedded power lines. Thus, unlike more common sensor networks, an e-textile grid allows nodes to share energy resources as easily as they share information. An obvious application of e-textiles is in the domain of wearable computers. For example, one could create an e-textile glove that sensed the movements of the user's fingers, providing a virtual keyboard or musical instrument. One could also take advantage of the ease with which e-textile sensors can

be deployed and later recovered. Building inspectors could test for asbestos by rolling an e-textile carpet onto a floor or crawlspace; once the sensors had finished their sampling, the inspectors could simply roll up the carpet and move to the next room.

SmartView: Enhanced Document Viewer for Mobile Devices

Small mobile devices like PDAs have difficulty displaying Web pages that are large and have complex layouts. The PDA is often forced to break these pages into screen-sized chunks. These chunks do not have semantically meaningful boundaries and are usually poorly formatted; the user must engage in extensive scrolling to fully understand the document. SmartView analyzes the source of a Web page and breaks it into properly formatted segments. It then creates thumbnails for each segment, allowing the user to preview a segment before expanding it to its full size. SmartView also provides annotations for pages returned by Google searches. Each segment of a returned page will contain a certain number of the total keyword hits. SmartView graphically depicts the number of hits that each segment has in its thumbnail. Users can then directly jump to segments with the most hits.

MagnetOS: An Operating System for Mobile Ad Hoc Networks

MagnetOS is an operating system for ad hoc networks. It provides a collection of nodes with the abstraction of a single Java Virtual Machine. MagnetOS also automatically splits programs into mobile partitions, and it dynamically migrates partitions in an effort to minimize energy consumption. MagnetOS uses two object-placement strategies: netCenter and netPull. In the netCenter approach, a mobile code object moves directly to the node that generates the largest percentage of the packets that it receives. In the netPull strategy, mobile code is moved in the general direction where most packets are generated. In other words, given an object receiving packets from multiple sources, the object will move to the "center of gravity" of the aggregate packet flow. Experiments show that both techniques result in large energy savings when compared to static or random migration models.

Mobile Web Services

IBM Research Technologies had several interesting demos. One dealt with providing Web services via mobile devices. Suppose that you own an IBM Linux watch with embedded Bluetooth networking. Your watch can export multiple services. For example, IBM demonstrated a watch that exported a payment protocol; if you go to a store that is Bluetooth-enabled, you can purchase your items via your watch. Your watch could export a time service that allows it to synchronize its clock with other network devices. Your watch could also exchange electronic business cards with other watches.

Constraints: An Abstraction to Express Semantics for Reconciliation

Mobile devices often have intermittent network connectivity. This means that if several devices want to modify a shared database, there will be periods when a node's updates cannot be immediately propagated to its peers or the central repository. When devices regain connectivity and submit their updates, the database must reconcile all of the updates to ensure that the final database state is "sensible." Individual applications often have specific reconciliation semantics, but the database would prefer to support multiple applications in a generic fashion. When mobile devices in the IceCube system emerge from disconnected operation, they do not transmit their disconnected updates directly to the database. Instead, they send these transactions to an automatic inference module. This module outputs generic constraints for a device's application-specific updates. For example, one set of operations may need to commit in sequential order; in another update group, perhaps all of the actions must commit or none of them must commit. The automatic inference module delivers a log of actions and a constraint list to the IceCube Generic Reconciler. If the reconciler can generate a feasible reconciliation schedule, it delivers the associated transactions to the central database, which is oblivious to application-specific reconciliation semantics.

LOCATION MANAGEMENT

Single Reflection Spatial Voting: A Novel Method for Discovering Reflective Surfaces Using Indoor Positioning Systems

Robert Harle and Andy Hopper, University of Cambridge; Andy Ward, Ubiquitous Systems Limited

A key goal of many pervasive computing systems is to generate a map of their surroundings. In the Single Reflection Spatial Voting system, people wear tags that emit ultrasonic pulses. As users walk through a room, the reflections from their tags' pulses are detected by sensors in the ceiling. By observing the intersections of these reflections, the sensors can determine the locations of walls and furniture in the room.

The Lighthouse Location System for Smart Dust

Kay Römer, ETH Zurich

"Smart dust" networks consist of millimeter-scale autonomous devices with integrated computing, sensing, and wireless communication capabilities. A base station acts as a data sink for information collected by the nodes. To impose a geographic ordering over this data, the nodes must have a sense of their relative spatial orientations. How can we provide this topological information without consuming an excessive amount of power? Nodes cannot use active radio communication, because the required antennas are too large and require too much power. In the Lighthouse approach, the nodes detect their location in a passive fashion. The base station emits a continually rotating light

stream. During each rotation, a node can measure the amount of time that it is illuminated by the beam. If the node knows the rotation rate of the lighthouse, it can use simple trigonometric formulas to determine its distance from the base station. If we introduce a second and third lighthouse, all with perpendicular beam sweeps, then a node can determine its location in two and three dimensions, respectively.

The primary advantage of the lighthouse protocol is that a node does not expend energy talking to other nodes or to the base station. Furthermore, the code that performs the trigonometric calculations has small CPU and memory requirements.

A member of the audience observed that the sensor nodes are very small and thus can be jostled by the wind or other vibrations. These movements could disturb a node's observation of the lighthouse beam and thus upset its location calculations. The speaker said that this problem can be solved by equipping nodes with accelerometers. The accelerometers would measure any unexpected movement, and the trigonometric calculations could be adjusted by the necessary amount.

Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking

Marco Gruteser and Dirk Grunwald, University of Colorado, Boulder

Fifteen years ago, Tim McCarthy of Motorola's GPS business noticed that many devices suddenly had embedded clocks. McCarthy now predicts that every device will soon have an embedded location sensor. As these sensors become ubiquitous, they will introduce new threats to location privacy. For example, if a malicious party has access to accurate location information about you, he can infer whether you have recently visited a hospital or a political organization. The key idea underlying cloaking is k-anonymity. A subject is k-anonymous if its associated location data is indistinguishable from that of k-1 other subjects. In other words, given a rectangular bounding area and a time interval, this data must describe at least k unique subjects. To achieve this anonymity, mobile nodes indirectly communicate with location services via a trusted anonymity proxy. To provide k-anonymity, this proxy alters the position data in a location service request before forwarding it to the actual service. Communication between a node and its proxy is authenticated and encrypted to prevent eavesdropping. There are two primary areas of future work. Even though the proxy is trusted, it introduces another principal that can be subverted; a better system would eliminate the need for a proxy that is separate from the node itself. More research is also needed to discover appropriate values for k in different application environments.

SUPPORTING APPLICATIONS OVER MOBILE NETWORKS

Reservations for Conflict Avoidance in a Mobile Database System

Nuno Preguiça, J. Legatheaux Martins, Miguel Cunha, Henrique Domingos, Universidade Nova de Lisboa

Mobile devices often have intermittent network connectivity. If multiple clients can autonomously manipulate a shared central database, there must be a method for reconciling updates that occur when the devices are disconnected. To guarantee that client updates can always be successfully reconciled, the Mobisnap database gives clients reservations before they disconnect. For example, a reservation might allow a client to use a record value for a given amount of time, even though that value may be outdated when the client reconnects. A client can determine whether its transactions will commit on the central server by examining its personal reservation set. This introspection only requires local state. Thus, even disconnected clients can be confident that their transactions will commit if they have the appropriate reservations.

Protecting Applications with Transient Authentication

Mark D. Corner and Brian D. Noble, University of Michigan

Current authentication systems typically retain long-term authority to act on their users' behalf after login. Unfortunately, if your laptop is stolen after you login, the authentication system will not prevent a thief from rummaging through your private information. There is a fundamental tension between requiring frequent authentication, which is secure but irritating, and permitting infrequent authentication, which is more usable but less safe. In the transient authentication system, a laptop's hard disk data is always encrypted. The user wears a token that has wireless networking capabilities. The token automatically and securely releases the keys that enable the laptop to decrypt its disk data. When the user leaves (as indicated by the token moving out of communication range), the laptop encrypts its memory. When the user returns, the token transparently authenticates the user and provides the necessary keys for the laptop to decrypt its memory and resume execution. Corner and Noble also provide an API that allows applications to selectively protect sensitive in-memory information.

iFlow: Middleware-assisted Rendezvous-based Information Access for Mobile Ad Hoc Applications

Zongpeng Li, Baochun Li, and Xin Zhou, University of Toronto; Dongyan Xu, Purdue University

iFlow is a middleware framework for disseminating information in mobile ad hoc applications. iFlow leverages node mobility to support "information rendezvous": data suppliers spread popular content on third-party nodes as they travel, and data consumers collect these information deposits as they move around the network. Suppliers use Tor-

nado codes and network coding to efficiently break content into smaller units that are easily distributed and reconstructed. Using these techniques, iFlow uses less communication bandwidth than systems that deliver data directly from supplier to consumer.

SYSTEMS SUPPORT FOR MOBILITY

Full TCP/IP for 8-bit Architectures

Adam Dunkels, Swedish Institute of Computer Science

Conventional wisdom states that the TCP/IP protocol suite is too complex to fully implement in a constrained resource environment. Dunkels presented compact implementations of the TCP/IP stack for 8-bit architectures. These stacks satisfy the necessary properties from RFC 1122 that enable a host to act as an endpoint for generic TCP traffic. 8-bit applications interact with the stacks through an event-driven API.

Several audience members challenged the need for an 8-bit TCP/IP stack. One person suggested that there are no 8-bit applications that need complete TCP/IP support and that more complex chips have enough resources to support the unmodified stack. Another person proposed that the full protocol stack should be run on a proxy machine. This solution would provide embedded devices with the reliability of TCP while keeping embedded code size small.

System Services for Ad Hoc Routing: Architecture, Implementation, and Experiences

Vikas Kawadia, University of Illinois, Urbana-Champaign; Yongguang Zhang, HRL Laboratories, LLC; Binita Gupta, Qualcomm Inc.

The authors argued that current operating system architectures are inappropriate for supporting ad hoc routing. Most OSes separate the notions of packet forwarding and packet routing. However, in ad hoc protocols, a data packet can also have a routing function (e.g., route discovery). The authors defined a generic API to add ad hoc routing support to an OS's existing networking framework. They implemented this API for the Linux 2.4 kernel using a user-level library and a small loadable kernel module. The authors used their new API to implement AODV and part of DSR. Referring to the difficulties they encountered in correctly implementing these protocols, the authors argued that separating routing and forwarding mechanisms in ad hoc protocols is "profoundly important" for ensuring protocol efficiency, extensibility, and ease of implementation. This proposition was vigorously criticized by Dave Johnson, who said that ad hoc protocols combine forwarding and routing for valid reasons. Johnson dismissed the notion that implementation difficulty always justifies changes to valid design decisions; he cited the example of TCP, which is complex but effective.

Predictive Resource Management for Wearable Computing

Dushyanth Narayanan, Carnegie Mellon University; Mahadev Satyanarayanan, Carnegie Mellon University and Intel Research Pittsburgh

Applications for wearable computers (e.g., speech recognition or translation software) always desire more resources than are available in such a constrained environment. To ensure low response times for these resource-intensive applications, a system can support multi-fidelity computations. A multi-fidelity computation is one which accepts a computation request and a description of available resources, and generates the highest fidelity result that it can achieve with those resources. The authors describe a concrete system which, given the current resource supplies, can automatically predict the latencies associated with generating outputs of varying fidelity. The system executes the highest quality operations that still have tolerable latencies; it observes the actual correlations between output fidelity and resource usage to dynamically calibrate its predictions. Experiments show that this approach can reduce mean latency by 60% and latency variability by 30%.

SENSOR NETWORKS

Design and Implementation of a Framework for Efficient and Programmable Sensor Networks

Athanassios Boulis, Chih-Chieh Han, and Mani B. Srivastava, University of California, Los Angeles

SensorWare is a framework for creating distributed applications in wireless ad hoc sensor networks. All programs are represented as event-driven state machines. After a user injects a program into the network, the program autonomously migrates and/or produces multiple copies of itself in response to changing environmental conditions. Users are not burdened with the chore of assigning tasks to nodes. The SensorWare runtime environment provides abstractions for radios, sensors, batteries, etc., so writing portable code is easy. SensorWare also provides support for threading and message queues.

An Entity Maintenance and Connection Service for Sensor Networks

Brian Blum, Prashant Nagaraddi, Anthony Wood, Tarek Abdelzaher, Sang Son, and Jack Stankovic, University of Virginia

The primary goal of sensor networks is to monitor environmental events. Blum et al. described an API for associating addresses with these events, making it easy for applications to communicate with nodes in the vicinity of the event. The API also allows the network to associate state with each event; this state migrates with the event as it moves through the network. One audience member questioned whether this model pushed too much work onto the lightweight sensor nodes. Blum noted that the amount of data sent to the base station is reduced, since information about each event is only conveyed to the base station by a single "leader node." However, he admitted that the complexity of

pattern recognition for different event types is not explicitly addressed by his system.

ENERGY MANAGEMENT

Operating System Modifications for Task-Based Speed and Voltage Scheduling

Jacob R. Lorch, Microsoft Research; Alan Jay Smith, University of California, Berkeley

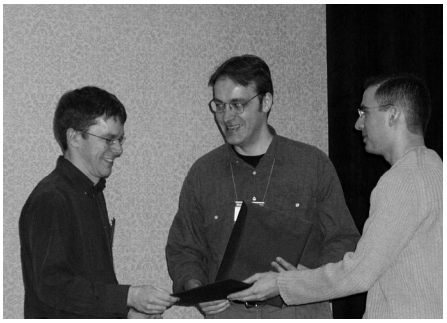
RightSpeed is a dynamic voltage scheduler for Windows 2000. Given a set of task deadlines, RightSpeed minimizes the voltage (and thus the processor speed) needed to meet these deadlines. Applications can explicitly provide deadlines, or RightSpeed can infer them by observing user-interface events and thread activity. A voltage/speed setting is "worthwhile" if using it saves more power than an emulated version that uses a combination of faster and slower settings. RightSpeed needs at least three worthwhile settings to improve performance. Interestingly, the authors discovered that several popular voltage-scaling processors do not have enough worthwhile settings for RightSpeed to save energy! However, simulations show that future processors with more worthwhile settings will reap large energy savings.

Energy-Aware Lossless Data Compression

Awarded Best Paper

Kenneth Barr and Krste Asanović, MIT

An add instruction consumes less than a nanojoule of energy, but sending a single bit over a wireless network can consume 1000 nanojoules. Barr and Asanović explored the energy savings that can be achieved via data compression and decompression over wireless links. They found that receiving and decoding compressed data is usually more efficient than receiving uncompressed data. The authors also observed that compression programs have poor cache behavior. Handling a cache miss is very expensive, so compressing data before transmission can lead to more energy consumption than regular uncompressed transmission! The authors demonstrated how careful selection of data structures can improve cache behavior and provide the desired energy savings. They described



Program Chair Robert T. Morris with Award winners Barr and Asanović

how idle power consumption affects the choice of compression algorithms. Finally, they showed how to optimize the energy savings for communication involving devices with different hardware profiles.

Energy-Adaptive Display System Designs for Future Mobile Environments

Subu Iyer, Robert Mayo, and Parthasarathy Ranganathan, Hewlett Packard Labs; Lu Luo, Carnegie Mellon University

Iyer et al. noted that displays consume over half of the power in mobile devices. They also presented a new user study showing that people using desktop and laptop computers typically focus on only 60% of the total display area. In their new "dark windows" system, the user's focused window retains its normal brightness and color, but the remaining screen areas are dimmed or displayed in a different energy-saving hue. By combining dark windows with new OLED screen technology, displays can consume 30% less power. An audience member questioned the authors' experimental methodology. He noted that he used 100% of his small laptop screen but a much lower percentage of his big desktop display. Therefore, it may be inappropriate to derive behavioral generalizations from a user set having heterogeneous display sizes, and thus potentially divergent usage patterns.

MOVING PARTS OF APPLICATIONS

Tactics-Based Remote Execution for Mobile Computing

Rajesh Krishna Balan, SoYoung Park, and Tadashi Okoshi, Carnegie Mellon University; Mahadev Satyanarayanan, Carnegie Mellon University and Intel Research Pittsburgh

How can users run resource-intensive applications such as speech recognition software on a resource-constrained mobile platform? In the basic remote execution approach, mobile clients offload work to nearby servers. These servers use their more powerful computational resources to calculate the required results, which are then shipped back to the mobile client. Balan et al. introduce the new idea of tactics. In their Chroma system, developers split programs into functional units called modules. Each application also exports a tactics list which enumerates the useful module-level partitions. These tactics constrain the search space for module distribution, reducing the time needed to determine the best allocation. Given current resource availability and expected resource demands, the best tactic is the one that provides the smallest latency and the best fidelity.

An audience member raised the insightful question of whether Chroma is applicable to applications people currently use. Chroma may work well for speech recognition and language translation, but would it improve the performance of popular email clients, Web browsers, or text editors?

Collaboration and Multimedia Authoring in Mobile Devices

Eyal De Lara, University of Toronto; Rajnish Kumar and Dan S. Wallach, Rice University; Willy Zwaenepoel, École Polytechnique Fédérale de Lausanne

To support multimedia collaboration between weakly connected mobile devices, the authors introduce two new concepts. Adaptation-aware editing distinguishes between user updates and fidelity modifications introduced by the adaptive system; users can edit low-fidelity data and later merge their changes with the shared high-fidelity version. Progressive update propagation reduces the upload time of updates by shipping partial or reduced-fidelity versions of these updates. The authors also decompose top-level documents into multiple component documents, e.g., sound, video, text. By reducing communication overhead and sharing granularity, mobile users can issue more frequent updates with fewer conflicts.

UNDERSTANDING AND BUILDING BETTER MOBILE NETWORKS

Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network

Magdalena Balazinska, MIT; Paul Castro, IBM T.J. Watson Research Center

The authors provided detailed traces of a corporate WLAN environment spread over three buildings. They characterized user behavior along two primary dimensions: persistence (session duration) and prevalence (patterns of access point usage). The period of mobility for most users was typically more than one day, and 50%–80% of all users were occasionally or somewhat mobile. Most people spent a majority of their time at a single "home" access point. However, a user's average bandwidth usage was the same at arbitrary access points, and users had more short sessions than long sessions across all access points. The authors also observed that the aggregate bandwidth usage of an access point is somewhat correlated with its number of users, but it is strongly correlated with the identity of these users. An audience member commented that corporate infrastructures usually have fast-wired networks that people use for the majority of their work. He argued that the authors' traces do not truly represent "mobile" users. The session chair suggested that the study examine "portable" users as opposed to "mobile" users.