Lettieri99

**Abstract**

Current wireless terminals are limited to voice terminals such as cellular and PCS phones, and traditional laptop computers and PDAs configured with wireless modems and network interface cards. However, the current wireless networks, which are by and large wireless extensions of the circuit-switched voice networks, are being replaced by emerging wireless networking technologies that are intrinsically designed to support packet data and multimedia services. This will lead to novel networked applications and services, which in turn will require wireless terminals capable of exploiting these services. What shape will these next-generation wireless terminals take? The answer, based on the much talked about notion of "convergence," would appear to be a marriage of the laptop or PC with a wireless phone in the same package, leading to terminals such as the Nokia 9000 [1] or Bell Laboratories' wireless handset [2]. We argue that such a complex one-size-fits-all voice-data integrated wireless terminal will, at best, be a point solution. Rather, with the availability of cheap radio and computing hardware and ubiquitous low-cost indoor and outdoor wireless networking infrastructures, the capability to access a wireless network will soon be embedded into a variety of devices, gadgets, and appliances with specialized functions in our environment. In this article we describe the technological challenges and identify potential solutions in designing these myriad future "wireless terminals" that will handle diverse data types, have limited battery resources, and operate in environments that are unplanned, insecure, and time-varying, and have context-dependent services.

# Advances in Wireless Terminals

**Paul Lettieri and Mani B. Srivastava**
**University of California at Los Angeles**

Wireless networks have historically been designed primarily to extend the domain of circuit-switched telephone service over a wireless last hop (i.e., cellular telephony). However, wireless networks that support data in addition to voice have now begun to appear. Initially, this has taken the form of circuit-switched and packet-switched data services overlaid on top of existing wireless networks, such as Cellular Digital Packet Data (CDPD) [3] and General Packet Radio Service (GPRS) [4] on top of the Advanced Mobile Phone Service (AMPS) and the Global System for Mobile Communications (GSM), respectively. Emerging third- and fourth-generation wireless networking technologies are now being designed to natively support packet data and multimedia services in addition to voice. These data- and multimedia-capable wireless networks, bootstrapping on the success of the Internet and the Web, in turn drive novel networked applications and services for use over the wireless networks. Already, even with the limited data capabilities of the paging network, services such as stock quotes and sports score updates via pager are becoming popular. And this is just the tip of the iceberg.

Current wireless terminals, however, are incapable of fully exploiting these emerging integrated service wireless networks. At the present time the choice of wireless terminals is largely limited to simple wireless phones on the one hand, and to complex and bulky laptops or personal digital assistants (PDAs) with built-in or plugged-in wireless cellular and LAN modems on the other. While these devices serve their purposes, they are neither the most integrated nor the most general of solutions, their functionality is often insufficient, and they are generally not unobtrusive enough (i.e., they focus the user's attention too much on the device itself rather than on the task at hand). The search for new wireless devices goes on, as is evidenced by the numerous commercial and research efforts in wireless devices. The natural question, then, is what will the wireless terminals of the future look like? One answer, based on the trendy notion of the "convergence" of the phone and the personal computer (PC), is to marry a palmtop computer or PDA with a phone in the same package. This amounts to an integration effort, albeit a nontrivial one. Early examples include the Nokia 9000 communicator [1] and Bell Laboratories' wireless handset [2]. We believe that such complex one-size-fits-all voice-data integrated wireless terminals are, at best, a partial answer, and will remain point solutions for a class of users

Rather, the trends in digital and radio frequency (RF) microelectronics point to ever cheaper processing and communication costs. While the cost of wireless infrastructure services decreases, allowing ubiquitous deployment, the radio modem hardware and processors are also becoming increasingly miniaturized and lower-cost, and can therefore be embedded in all sorts of devices, gadgets, and appliances with specialized functions. Furthermore, there are a great number of emerging applications which require devices to be connected wirelessly. Consider, for example, home area networks (HANs), where a multitude of devices in a home, such as cordless phones, smart toys, pen tablets, remote controls, TVs, and VCRs, may be internetworked. Many of these devices are mobile and thus well suited to wireless connectivity, while even the nonmobile devices benefit from wireless by not having to install a wired infrastructure. By having them all on the same network, a home owner may control and monitor status of various devices, appliances, gadgets, and, of course, PCs in the home.

Together, these trends suggest that in the future there will be a wide variety of "wireless terminals" and not just a single generic wireless terminal that is a phone-plus-PDA associated with an individual person who carries it as a cellular phone is carried today. This is similar to the ongoing evolution in the computing world, where the complex all-purpose PC is gradually giving way to simpler specialized information appliances such as set-top boxes, Web TVs, PDAs, screen phones, network computers, and thin client terminals [5]. In fact, the technology needed to realize the vision of ubiquitously networked specialized wireless terminals has already begun to take shape in the form of wireless multimedia terminals [6, 7],

1070-9916/99/$10.00 © 1999 IEEE

wireless sensor nodes [8, 9], wireless cameras [10], and wireless toys [11]. In fact, the term *Smart Spaces* [12] has been coined to refer to environments with arrays of deeply embedded and usually wirelessly connected computing and sensing devices that are not fundamentally associated with an individual person.

While coming in diverse shapes and forms with a variety of specialized functions, the wireless terminals of the future will nevertheless have a common core hardware and software functionality that will allow them to be wirelessly internetworked, as would be useful, for example, in the HAN described above. As a result, these terminals will share a set of common technological challenges. In particular, these myriad future "wireless terminals" will handle diverse data types; have limited battery resources; operate in environments that are unplanned, insecure, and time-varying; and have context-dependent services. Describing these challenges to the design of future wireless terminals and identifying possible solutions is the main purpose of this article.

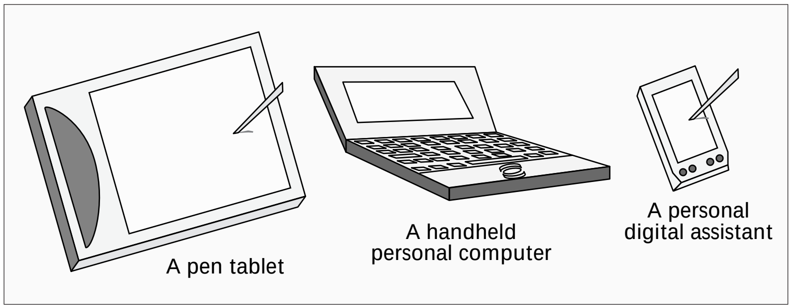


**Figure 1.** *Examples of data-capable terminals.*

## Wireless Terminals Today

As a first step toward understanding the design requirements of wireless terminals, we first take a brief look at the various handheld computing and communication terminals on the market today. Note that many of these terminals have no built-in wireless networking capability, but rather rely on an external plugin wireless modem for wireless connectivity. Also, note that we do not include general-purpose laptops in the following discussion since they are large and bulky, and often overkill in terms of functionality. Broadly speaking, then, current terminals can be classified into the following categories, some of which are depicted in Fig. 1, based on their functions and form factors.

### Pen Tablets

Pen tablets can best be described as laptops without keyboards. Instead of typing on a keyboard for data entry, interaction with the pen tablet is through pen input, as its name implies. In most other respects, however, these devices are effectively low-end laptops. A screen with standard VGA resolution (640 x 480) in color is typical, as are an Intel i80486 or Pentium class processor and a subgigabyte hard disk. As such, Windows9x and in some cases even Windows NT is the operating system (OS) of choice. The terminal itself resembles the LCD half of a laptop, though slightly thicker than that since the entire system resides just below the screen. And while a pen is typically the input method, handwriting recognition is rare. In most cases, the pen replaces the mouse as a pointer device, and any keying must be performed by typing on an external plug-in keyboard or with the pen on an onscreen keyboard. Examples of this type of device are the Orasis from Dauphin, the Stylistic 1000 from Fujitsu, the AMiTY VP from Mitsubishi, and the Panasonic CF-01. Some of these, such as the Fujitsu Stylistic, have an internal radio modem card plugged into the motherboard, whereas others may be made into a wireless terminal by plugging-in radio modem PC cards. But, at heart, these terminals are no different than the average laptop.

### Handheld Personal Computers

Terminals in this second category resemble and are essentially used as miniature laptops. They are characterized by a reduced form factor keyboard and a half-VGA (640 x 240) resolution display, often with color, in a clamshell design. A hard drive is absent, and so they run the Windows CE embedded OS (or, in rare cases, a comparable competing OS) that is designed specifically for systems without mass storage. Handheld PCs (HPCs) generally weigh 1–2 lb, as opposed to the 6- or 7-lb laptop. Since the PC software architecture is implemented on these devices, they usually can run reduced versions of Windows applications, including various word processing, presentation, and scheduling software compatible with their bigger siblings. However, the microprocessor at the heart of a device like this is rarely of the Intel x86 family, but rather is an embedded microprocessor, most often of the MIPS or Hitachi SuperH variety. Finally, as for communication capability, there is usually wireline modem capability, sometimes carried out in software for power savings, along with some combination of a serial port, a docking station port, and an infrared port. There are numerous products of this type on the market. Examples include the Cassiopeia A-20 from Casio, the Palmtop PC 620LX from Hewlett Packard, the Velo-500 from Philips, and the Phenom Ultra from LG Electronics.

### Personal Digital Assistants

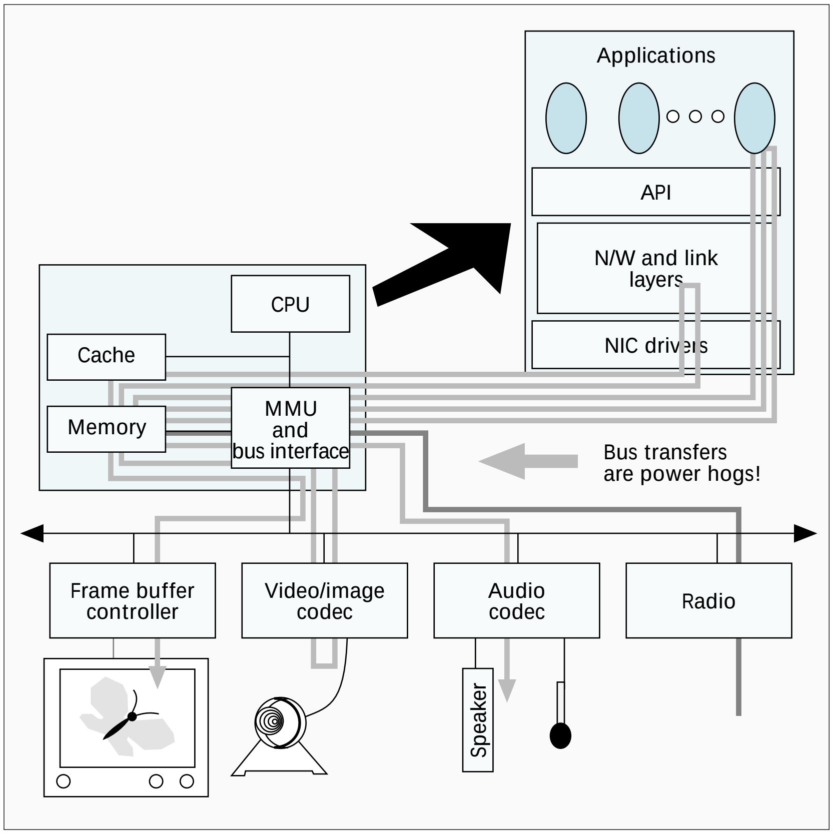
The next category of terminals, the PDA, generally consumes on the order of at most half the resources of an HPC. These devices usually have a quarter-size VGA screen used in portrait mode, half as much memory, and physically occupy half the space of HPCs. While the HPC is usually a clamshell that opens to reveal a keyboard and screen, the PDA is generally a monolithic device without a keyboard (with the possible exception of a handful of control buttons) and fits in the average user's hand at about 3 in x 5 in x 3/4 in. As such, pen input is the norm, and handwriting recognition is common. These devices are generally modeled after the older Palm Pilot and the newer Palm III from 3Com. Other devices which fit into this category include the Everex Freestyle, the Philips Nino, and the Avigo from Texas Instruments. Many of these devices are based on the Windows CE OS for palm-sized devices, and are often called Palm PCs. Communication abilities again involve a docking port or serial port for connecting to and synchronizing data with a desktop computer, as well as possibly a modem. Recently, however, 3Com has announced Palm VII, which is similar to Palm III in form factor but has a built-in two-way radio and a flip-up antenna, and uses Mobitex-based wireless data networking to provide customized Internet service at 8 kb/s.

### Cellular Phones

The cell phone, unlike all of the above terminals, is centered around communication ability. While a PDA or HPC can communicate and synchronize data with a single desktop through a serial or docking station link, or exchange data with other PDAs, printers, and so on through an infrared link, the primary function of these devices is generally to keep appointments on a calendar, lists of contacts, and other personal notes and data. The cellular phone, on the other hand, is first and foremost a communication tool. It may store a phone book of numbers for quick dialing, but beyond that is generally meant for voice transmission. As such, a cellular phone cannot be compared directly to the previous items other than to say that its functionality largely complements that of the HPC or PDA, particularly when it is of the digital variety and thus well suited to data communications. Cellular phones come in all shapes and sizes from companies such as Nokia, Ericsson, Motorola, and Samsung, among others. In general, these phones are smaller but more power-consuming than even the smallest of the information devices, the PDAs. This disparity is related to the cellular phone's RF power requirements.

### Wireless Local Area Networks

Another communication related category encompasses the wireless LAN (WLAN) products. These devices are generally intended for short-range (several hundred meters) indoor use, as opposed to the outdoor several-kilometer range of cellular systems. Typical environments include office buildings, hospitals, and homes. Although there are terminals with built-in WLAN network interfaces, for the most part WLAN products are cards that can be plugged into desktop or laptop computers, or possibly even some of the smaller PC-like devices mentioned earlier. Typically these products are based on the IEEE 802.11 standard, which allows them to operate in both a base-station-oriented mode and a peer-to-peer base-stationless ad hoc network. Data rates are higher than cellular phones, on the order of megabits per second, and size is smaller, although power consumption is comparable, and range is of course decreased significantly. Example devices of this category are the Proxim RangeLAN and Lucent WaveLAN radios.



■ **Figure 2.** *Architecture bottlenecks in a PC-based CPU-centric wireless multimedia terminal.*

### Combination Devices

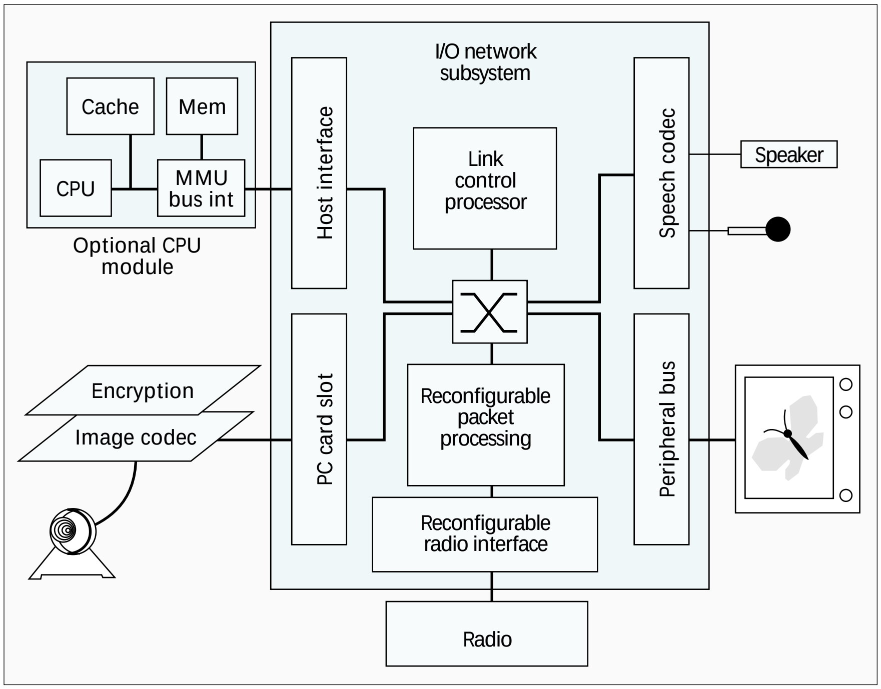
Finally, a number of devices do not fit into any of the above categories, but rather begin to span them. In particular, we note three. At first glance, the Nokia 9000 looks like a standard "brick" cell phone, although slightly larger than average. However, it opens up along the longer edge, revealing a keyboard and LCD inside the clamshell, and suddenly behaves more like a small HPC. Unlike the average HPC, however, this device has wireless communication ability built in and integrated with the HPC applications. Where other products will allow a cell phone to be attached to a data device, in this case the two are physically merged already. Nevertheless, at its core this device is just a PC-like device attached to a cellular phone.

Another device which does not fit the categories presented above is the CruisePAD from Cruise Technologies. Its form factor is similar to the pen tablet computers mentioned earlier, but it has an integrated wireless LAN interface card. Although it contains an i486 CPU, it is nothing more than a dumb wireless terminal peripheral to a server, where the real computation is done. For example, when the user writes on the touch-sensitive display, the terminal simply relays these inputs to a desktop computer acting as a server through a WLAN connection. After performing the requested calculations, the server redraws the screen graphics. Again, the graphic window operations are done at the server, where the screen redraw commands in the graphics driver are trapped and sent over the WLAN link to the terminal. The terminal essentially acts as a wireless graphics frame buffer. Devices like the CruisePAD are known as thin clients, since the client itself does no work. While this terminal architecture has advantages and disadvantages compared to carrying around a standard pen tablet computer which the CruisePAD physically resembles, and does make use of offloading processing and data storage to a server to improve battery life and response time, it is a PC in disguise. The pad itself is nothing more than the wirelessly extended input and output of a nearby desktop machine which acts as the server. The Infopad project at Berkeley also advocated a similar approach to wireless terminals.

Finally, the REX from Franklin Electronics is a bare bones personal organizer. Actually in the form factor of a PC card, this device carries and displays contact and schedule information kept synchronous with desktop application data. The newest form of this device is even capable of light data manipulation tasks through buttons on the card itself. This device exhibits no communication ability beyond that achieved through a standard bus interface, but represents the smallest, lightest, and lowest-power device yet mentioned.

## Problems with Existing Terminals, and the Future of Terminals

From the preceding section it is clear that current wireless terminals are still by and large primarily either data processing terminals *or* communication terminals. The trend in data processing terminals has been to shrink a general-purpose desktop PC into a package that can be conveniently carried. The amount of shrinkage is weighed against reduced capabilities, and the Intel x86 processor may be replaced by a processor better suited to portable applications, but fundamentally these devices resemble general-purpose desktop PCs. Even PDAs have not ventured too far from the general-purpose computer model, neither architecturally nor in terms of the usage model. While PDAs do not follow the "Wintel" PC hardware specification, and in many cases use non-Windows software, their architecture continues to be centered around the model of a single general-purpose CPU orchestrating a few dumb peripherals over a shared bus. Even the PDA applications are a subset of those found on general-purpose computers with the focus being on personal information management and Internet access applications, although their general-purpose roots have resulted in many other PC applications being ported to PDAs as well. In short, current terminals are general-purpose computing devices at heart, with a PC-like use model where the attention of a user is focused on a single device which provides a specific data processing service.



■ **Figure 3.** *An I/O-centric terminal architecture.*

Likewise, communications devices such as cellular phones and WLAN radios are shrinking. Notably, the newest generation of phones, which are digital and support limited data services, are an improvement over the analog alternative. But in general, these devices remain strictly communication devices.

The popular answer in the industry to the extremes of data processing terminals and communication terminals appears to be a convergence of the two types of devices to create wireless terminals that are essentially shrunken PCs or PDAs packaged with cellular phones or wireless modems. This has led to terminals such as the Nokia 9000 communicator, the CruisePAD, and the Fujitsu Stylus. But are these representative of the wireless terminals of the future? While such devices will continue to have a role, there are at least two reasons that we believe the answer is no.

## *I/O-Centric Instead of CPU-Centric*

Simply shrinking the data processing terminal and radio modem, attaching them via a bus, and packaging them together does *not* alleviate the architectural bottlenecks in this CPU-centric approach. The goal of wireless terminal design should not be to build the smallest, lightest general-purpose computing device to which the smallest, lightest radio can be attached. Clearly, small, light, low-power devices are useful, but the real design problem is to engineer an integrated terminal where data processing and communications share equal importance and, furthermore, are designed with each other in mind.

Connecting an off-the-shelf PC or PDA with an off-the-shelf communication subsystem as peripheral, even in the same package, is not the solution then. One of the main drawbacks of merely packaging the two is that the power-inefficient general-purpose CPU, with its heavyweight OS and shared bus, becomes not only the center of control but also the center of data flow in the terminal. While the speed bottlenecks of such an architecture may become an issue in future high-speed wireless multimedia terminals, for now typical buses are fast enough to handle wireless terminal data rates. The more serious problem is one of power consumption or energy efficiency. As recent research in [13] observes, a general-purpose CPU time-shared among multiple applications and a single time-shared bus are simply the wrong architectural choices from a power perspective for the signal-processing-type functions expected to dominate wireless terminals. The situation only worsens when the OS overhead of the general-purpose CPU is accounted for. As can be seen in Fig. 2, the conventional CPU-centered shared-bus architecture requires frequent traversal of multimedia streams over the highly capacitive central bus and through the layers of OS software for the simplest operations such as multiplexing/demultiplexing and interstream synchronization. Indeed, measurements with a prototype wireless multimedia terminal at the University of California at Los Angeles (UCLA) [14] with an embedded PC-based architecture show that large amounts of time and power go into memory and I/O transactions across the shared bus.

A better choice, as suggested by [13], is to have dedicated but slower computing resources with lower voltages, and multiple slower interconnects with lower capacitances. The architectural solutions to performance bottlenecks in future high-data-rate wireless terminals, namely parallelism of computing resources to eliminate the single time-shared general-purpose CPU in the data path and parallelism of communication resources to eliminate the shared bus, are also the preferred choice from the power perspective. Therefore, an I/O-centric terminal architecture, where the data streams flow between the radio and devices such as audio and video codecs without needless intervention from a general-purpose CPU, is desirable [7]. As an example, Fig. 3 shows the architecture of WAND [7], a low-power embeddable module built at UCLA for creating multimedia wireless terminals. The general-purpose CPU, called the Link Control Processor, is moved out of the packet flow data path, although it still participates in the control flow for signaling. The data streams flow directly among the radio and the speech and image codecs that are integrated into WAND, and the necessary signal processing functions are done at the data sources and sinks themselves. The CPU is shut down most of the time since, after the initial signaling, operations such as data flow of compressed speech over the air or forwarding of incoming packets in a multihop network can take place without having to wake it up. A full-fledged PC or PDA may be adjunct to WAND, but its presence is optional and, in many wireless terminals, unnecessary.

The WAND architecture uses three features for energy efficiency:

- A switched interconnect instead of a shared bus
- No data processing on audio and video streams at the time-shared general-purpose CPU
- Direct peer-to-peer data transfer without involving the CPU

A solution such as intelligent direct memory access (DMA) applied to the conventional CPU-centric shared-bus architecture will address only part of the problem with conventional architectures.

### Diverse Wireless Terminals

Another assumption behind complex general-purpose one-size-fits-all voice-data integrated wireless terminals resulting from the marriage of the data terminal to the cellular phone is that the wireless terminal is a single device carried by a single user at all times. We believe that this is too limiting a definition, and trends suggest that in the future there will be countless specialized wireless terminals which will have cheap processing and wireless networking capability embedded in them. Already, there are a few examples of the emerging diversity of wireless terminals. Integrating micro-electromechanical sensors (MEMS) and actuators with digital signal processing (DSP) processors and radio modems on the same die, researchers in the Low Power Integrated Wireless Microsensors (LWIM) project at UCLA have created sensor nodes [9] that are internetworked over the wireless network. At the University of California at Berkeley (UC Berkeley) researchers created Infopad [6, 15], which is essentially a wireless multimedia peripheral device with audio, video, and pen capabilities. At the Massachusetts Institute of Technology (MIT), researchers are creating a low-power camera node for a wireless network [10]. Commercially, Micron Communications offers MicroStamp, a single chip that integrates a simple processor with a spread-spectrum radio for use in low-cost ultra-miniature ultra-low-power wireless devices that can be attached to people or objects for applications such as tracking and inventory control. Microsoft has created mechatronic toys called Actimate toys [11] with FM wireless links that allow the toys to receive motion and speech commands from software running on a PC equipped with a radio transmitter, or from specially coded VCR tape recordings and TV programs by looping the video signal through a radio transmitter. None of these and the multitude of other such devices fit the mold.

On one hand, then, a general-purpose PC plus modem is not the answer. On the other, it would still be useful for the core functionality of all these wireless devices to be common. First, it would allow them all to communicate together more easily. Second, and perhaps more important, since the challenges of the wireless channel are shared by each of these devices, it makes sense to solve the problem once and apply it across the board.

## Challenges in Wireless Terminal Design

From the preceding section it is clear that wireless terminals of the future will come in diverse forms with a variety of specialized functions, and not just as hybrid phone-plus-PDA-type devices. Nevertheless, all these diverse wireless terminals should have a common core of hardware and software functionality embedded in them, perhaps in the form of a single chip that would provide the computation, protocol, and communication processing capability needed for wireless internetworking. It is this common core set of functions, and its requirements and design challenges, that are crucial to understanding the wireless terminal design problem. Wireless terminals will share a set of common technological challenges arising from the variety of form factors they will come in, the diverse data types with different quality of service (QoS) requirements they will handle, their limited battery resources, their need to operate in environments that may be unplanned, insecure, and time-varying, and their mobility resulting in a changing set of available services. The following are the key technological challenges that we believe will need to be addressed before the vision outlined above will become real.

### Energy Efficiency

The need for energy efficiency is a problem that stems from the constraints imposed by battery capacity and heat dissipation which are opposed by the desire for miniaturization and portability. Battery technology and technologies for heat removal have traditionally improved at a glacial pace compared to the pace at which the amount of computation expected to be done in wireless terminals is increasing and their size is decreasing. The way out is energy efficiency: doing more work per unit of battery energy consumed and heat dissipated. Traditionally, energy efficiency has been sought via low-level techniques such as improved semiconductor processes, packaging materials with better heat dissipation capabilities, and low voltage circuits. However, with the low-level gains behind, the key to energy efficiency in future wireless terminals will be at the higher levels: low-power protocols, power-cognizant user interfaces, context-dependent and predictive shutdown management, and changed terminal-network functional partitioning to reduce computation done at the terminal.

### Auto-Configuration

Current wireless terminals require well-planned and professionally managed infrastructures to operate in, such as cellular telephone networks and mobile-IP-based wireless LANs. The future wireless terminals will, however, be more like consumer appliances that are bought, turned on, and expected to work in office or home wireless LANs without complex setup and configuration of IP addresses, home agents, DHCP servers, and so on. Clearly, auto-configuration, or the ability of future wireless terminals to automatically discover the needed operation parameters and self-configure to begin operation in a new network, will be critical.

### Adaptivity

A similar issue is that of adaptation to time-varying radio channel conditions. Most current wireless devices and networks operate in structured and regulated cellular environments with well planned spectrum sharing, fixed channel allocations, and so on. However, many future wireless terminals will operate in more difficult, relatively unregulated environments such as home and workplace LANs with time-varying interference levels. Furthermore, as radio bit rates increase, the impact of time-varying multipath fading becomes more pronounced. In short, future wireless terminals are likely to face radio channels with impairments that vary in time over the short and long term. To address this, wireless terminals will need the ability to adapt to changing channel conditions, and will require adaptive radios, protocols, codecs, and so on.

### Reconfigurability

An issue related to adaptivity is that of reconfigurability. While adaptive protocols and radios can vary algorithmic parameters to help a wireless terminal gracefully maintain service quality across changing channel conditions and application requirements, they do not address the much larger variability in operating environment encountered when a wireless terminal moves to a different locale with a different service provider or to a network with a different set of services and perhaps even air interfaces. Similar situations occur when wireless terminals and network equipment of multiple agencies need to interoperate during a joint task, such as in rapidly deployable ad hoc networks established during disaster relief, or in the battlefield with forces from multiple allied countries. It would be desirable for a wireless terminal to have architectural reconfigurability whereby its capabilities may be modified by downloading new functions from network servers. Such reconfigurability would also help in field upgrading of wireless terminals as new communication protocols or standards are deployed, and in implementing bug fixes. Along with the capability to download new capabilities to wireless terminals comes the ability for client terminals to upload new functions into network servers, such as to a proxy server [16] at a base station.

### Location and Context Awareness

One of the newer degrees of freedom, available to wireless devices due to advances in microsensor technology and the shrinking form factor of global positioning system (GPS) receiver hardware, is the ability to have extremely specific location and context awareness. That is, it is now possible to know with great accuracy where a terminal is, what its position is relative to its surroundings, what user or users are proximate, and the condition of the environment. The value of this context information has only barely begun to be explored in some experimental systems, but it is clear that proper exploitation of context information will have a large impact on wireless terminal design. In particular, architectural support will be needed in the form of:

- Terminal hardware that provides context information, such as radios that provide information about channel condition, GPS that provides location information, and sensors that provide proximity, speed, acceleration, temperature, and other environmental information
- Terminal software that implements runtime mechanisms so that applications can efficiently monitor the context and adapt to it

### Functional Partitioning between the Terminal and Network

The design of wireless terminals cannot be done in isolation. With high-speed wireless networks, many different architectural choices become possible, each with different partitioning of functions between the wireless terminal and servers resident in the network. For example, a wireless terminal may provide access to information from the Web by either incorporating full-fledged PC-like capabilities, as in the case of the Nokia 9000 Communicator, or relying on network servers to do the bulk of the work and using simple terminals, as in the case of the AT&T PocketNet telephone [17], CruisePAD [18], Infopad [6], and thin clients such as Palm Pilot and WinCE machines with a proxy server such as Berkeley's Wingman [19] or ProxiNet's ProxiWeb [20]. Partitioning of functions between the wireless terminal and the network is an important architectural decision which dictates where applications can run, where data can be stored, the complexity of the terminal, and the cost of communication services.

### Security

Security has, of course, always been an important issue in network design, and is only more so when the network is wireless. There are many facets to this problem, including how to keep data safe from loss or corruption, as well as how to keep it private and free from tampering. One must be concerned with security of data at its point of storage, as well as during transit between storage and endpoint. Indeed, trade-offs arise regarding the location of storage, whether it be in the user's endpoint device, in a central location on the network, or distributed between the two. For example, when data is stored in the endpoint, if that endpoint is lost or compromised, the data is also lost or compromised. On the other hand, when the data is stored in a central location, access latencies may be higher, and data may become completely unavailable if the network is down or unreachable. Clearly, then, the issue of security is much more complex than simply worrying about employing encryption and authentication algorithms.

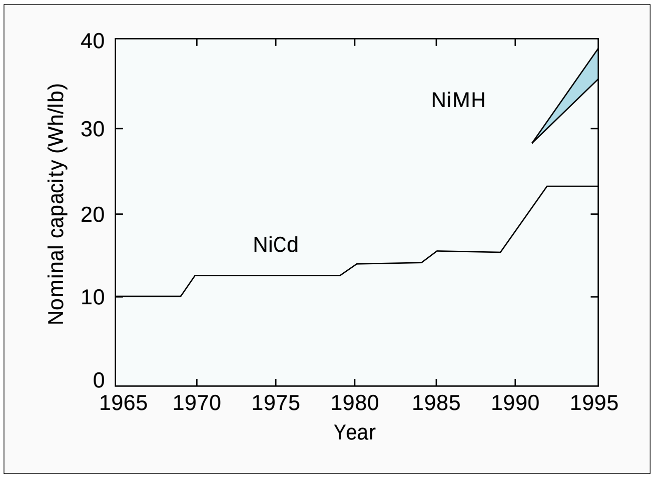
### User Interfaces

The problem of user interfaces encompasses issues such as the need for small size, simplicity, and, most important, lack of direct visibility. While this at first sounds unusual (i.e., user interfaces the user cannot see), for future wireless terminals to be ubiquitously deployed they must blend into the background, unlike today where a hand-carried wireless phone or PDA is the focus of the user's attention. Today's traditional QWERTY keyboard and LCD- or CRT-display-based interfaces are not adequate for the diverse wireless terminals of the future. Certainly, the traditional interfaces will have a place, but are too heavyweight for use everywhere. Instead, intrinsically simpler interfaces based on sound, vision, touch, force, pen, soft buttons, and so forth are more amenable to the small form factors of future wireless terminals and their I/O needs.

### Integrated RF Design

Finally, this discussion could not be complete without at least a brief mention of RF design. With advancements in microelectronics technology in general, and RF complementary metal oxide semiconductor (CMOS) technology in particular, the level of integration and affordability is reaching unprecedented levels. For example, researchers have demonstrated CMOS chips that integrate all components of a radio transceiver [21]. Research is also going on that seeks a single chip combining complete radios with digital hardware/software and MEMS sensors. Clearly, with increasing miniaturization and accompanying low cost, RF communication can be placed inside anything. What possibilities arise then? To start with, every item in the environment can report its status and whereabouts. Anything which is electronically actuated may be controlled remotely. People wearing RF patches can be located within a building, for example, to direct a call to the nearest phone. Other developments in radios, such as the use of RF elements based on MEMS technology [22], ultra-wideband RF front-ends [23], and software radios, are also going to have significant implications on the architecture of wireless terminals.

In the remainder of this article we shall focus on four of the above issues. We will first consider the issue of low-power design, where, after a review of traditional techniques, we will describe newer techniques such as energy-efficient link and medium access control (MAC)-layer protocols. Second, we will look at reconfiguration as a means to continuously adapt the terminal architecture and capabilities to the operational environment. Reconfiguration is particularly important in the face of mobility and changing service availability. Third, we



■ **Figure 4.** *Battery capacity growth over 30 years.*

will consider the issues of context awareness and its role in future wireless terminals. Finally, we will address the problem of the appropriate partitioning of application functions between a wireless terminal and the network infrastructure.

## Energy-Efficient Design

Design for low power is certainly not a new problem, and yet remains one of the most difficult as future wireless terminals attempt to pack more capabilities such as multimedia human–computer interfaces and faster radios into battery-operated portable miniature packages. The primary problem is that in the case of battery technology there is no equivalent of the familiar Moore's Law that forecasts a doubling every 18 months of the complexity of microelectronic chips, and Gilder's Law that theorizes a similar exponential growth in communication bandwidth. In contrast, battery technology has moved at a glacial pace, as indicated by Fig. 4, which plots how the energy density of NiCd and NiMH batteries has increased over a period of years.

With increasing computation and communication functions desired in wireless terminals, the energy density of existing battery technologies are far from what is needed. Table 1 tabulates the characteristics of some of the current battery technologies. Here we see what energy per weight and per volume each of the popular battery chemistries has to offer. It is clear that even the best battery technologies cannot achieve the desired battery life and form factor for current wireless data terminals, let alone for the feature-rich multimedia wireless terminals that are often envisioned for the future. For example, consider that perhaps the most popular form factor for a personal computing and communication device has been the Palm Pilot, which weighs around 6 oz. A wireless terminal of a similar weight made entirely of a zinc air battery (which has the highest gravimetric density), would have an energy capacity of only 52 Whr, which translates to around 300 mW power consumption for merely a week of operation time. In contrast, typical 802.11 WLAN radio modems consume 1–2 W, and low-power general-purpose CPU chips alone consume > 300 mW power.

Clearly, there is a disconnection between the state of battery technology and the visions of future wireless terminals being a convergence of cellular phone and multimedia PC-like functionality. Short of an unforeseen breakthrough in battery technology, the way out is to reduce the energy consumption in the wireless terminal or, in other words, to increase the energy efficiency of computation and communication. Reducing energy consumption has been examined in recent years, particularly from the perspective of semiconductor technology and circuits, and has facilitated the proliferation of current handheld devices. For example, Chandrakasan *et al.* developed numerous low-power techniques for signal processing CMOS very large-scale integration (VLSI), where speed or area or both are traded off for energy savings [13]. Techniques such as parallelism and pipelining used in conjunction with voltage scaling allow a new circuit to achieve the same throughput as the old while using less energy, although at the cost of silicon real estate. Fortunately, with ever shrinking minimum feature sizes, silicon area is getting less expensive.

While these low-level circuit and logic techniques have been well established for improving energy efficiency, they do not hold promise for much additional gain as VLSI technology heads toward interconnect-dominated deep-submicron regimes. The key to energy efficiency in future wireless terminals will be designing higher layers of the wireless terminal, and indeed the entire network, with energy efficiency in mind. In particular, intelligent management of power within the wireless terminal and energy-aware network protocols will be important considerations in future wireless terminals.

### Managing Power Within a Wireless Terminal

Many subsystems of wireless terminals, such as the CPU, wireless modem, and storage system have small usage duty cycles. That is, they are often idle and wait for user or network interaction. Furthermore, they have huge differences in power consumption between their "on" and "standby" or "stopped" states. For example:

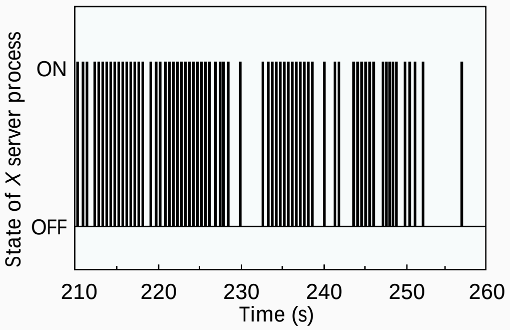
- The ARM-based Ruby II CPU from VLSI Technologies consumes 150 mW when on, 7.5 mW when in standby, and only 0.75 mW when stopped.
- Lucent's 2 Mb/s, 23 dBm, 900 MHz WaveLAN PCMCIA card radio consumes 3 W during transmit mode, 1.48 W during receive mode, and only 0.18 W during sleep mode.
- GEC Plessey's DE6003 20 dBm, 2.4 GHz, 625 kb/s radio transceiver consumes 1.8 W during transmit mode, 0.6 W during receive mode, and 0.05 W during sleep mode.

Many other components in a wireless terminal, such as disks, displays, sensors, and GPS, have similar states with different levels of power consumption.

Clearly, there is a potential for avoiding energy waste by shutting down parts of the wireless terminal that are not in use. Experiments with an X server and typical applications (the workload included contool, olvwm, Framemaker, xclock,

| Battery | Rechargeable ? | Gravimetric density (Wh/lb) | Volumetric density (Wh/l) |
|---|---|---|---|
| Alkaline-MnO$_2$ (typical AA) | No | 65.8 | 347 |
| Silver oxide | No | 60 | 500 |
| Li/MnO$_2$ | No | 105 | 550 |
| Zinc air | No | 140 | 1150 |
| NiCd | Yes | 23 | 125 |
| Li-Polymer | Yes | 65–90 | 300–415 |

■ **Table 1.** *Characteristics of current battery technologies.*



| | Trace | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| Trace length (s) | 5182.48 | 26859.9 | 995.16 |
| $T_{\mathrm{off}}$ (s) | 5047.47 | 26427.4 | 960.82 |
| $T_{\mathrm{on}}$ (s) | 135.01 | 432.5 | 34.34 |
| $T_{\mathrm{off}}$ ($T_{\mathrm{off}}$ + $T_{\mathrm{on}}$) | 0.9739 | 0.9839 | 0.9655 |
| Maximum energy reduction | x 38.4 | x 62.1 | x 29.0 |

**Figure 5.** *Potential CPU power reduction in a wireless X terminal.*

mailtool, cm, and xterms) on a wireless terminal show that, in theory, a large reduction in CPU power can be obtained if it were to be shut down whenever it was idle. Figure 5 summarizes the results from three real-life traces, and shows a portion of one of the traces. Potential energy reduction is from 29 to 62 times.

Given the short timescales of idle periods, a manual approach to shutdown and wakeup is infeasible for human reflexes except for turning the wireless terminal on and off over long time durations. Rather, the wireless terminal subsystems should automatically be brought to the right mode (on, off, sleep, standby, etc.) depending on user requirements. This may be done by the terminal OS or some power management monitoring circuit.

Besides the mechanism of shutting down parts of the terminal, there are several policy issues that need to be settled. The first question is *when to shut down*. A typical policy, often employed in laptops, is to shut down after the idle period has exceeded a certain threshold. For example, shutdown may occur after there has been no user input for a few seconds. The problem with this approach is that power continues to be wasted during the threshold interval, whose size is difficult to set rationally. An approach that works better in some cases is predictive [24], where the system uses recent history to predict whether the idle period just entered is going to be long enough to justify the cost of doing a shutdown. A second question is *when to wake up*, where the typical policy is to wake up in response to a certain event such as user interaction or a network packet. The problem with such an on-demand policy is that waking up takes time, and the extra latency is often not tolerable. Again, a predictive approach where the system initiates a wakeup in advance of the predicted end of an idle interval often works better. Such shutdown-based power management policies are applicable to CPUs, disks, displays, radio modems, and so on. In the case of radio modems, the shutdown is closely tied to the MAC protocol that controls the transmit and receive state of the modem.

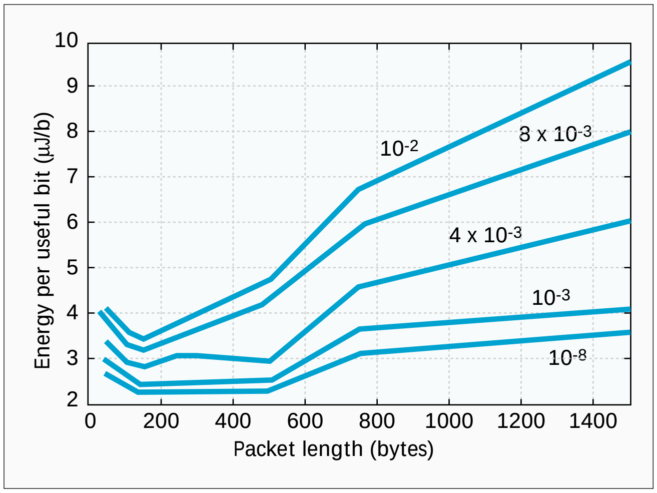
As a further enhancement to the basic shutdown idea, shutdown need not be a binary decision. That is, terminal subsystems may be designed that allow for multiple levels of reduced power consumption at the expense of some other system performance measure (e.g., throughput). For example, the core of the StrongARM microprocessor has the ability to run at multiple voltage levels, with lower voltage levels supporting lower clock speeds. Therefore, if the demands on the processor recently have not been high, the terminal power management circuit may choose to slow the processor and reduce its power supply, both of which have the effect of reducing power consumption. Later, when processing needs again increase, the situation can be reversed and processing power matched to the needs of the application at hand. Note that if the monitor, power, and target circuitry are designed properly, this power cycling can be done on a very short timescale, and thus allow extremely efficient use of resources. In the limit, the circuit uses exactly the energy needed to get the job done with the given silicon technology. Similar ideas can be applied to radio modems, where QoS metrics such as packet loss rates or range may be traded off against transmit power.

Finally, the best policy for deciding when to shut down or wake up a specific part of the terminal would in general be application-dependent since applications are in the best position to know usage and activity patterns for various resources. Power management circuits and terminal OSs that lack application-specific knowledge can only rely on the generic policies discussed above. This suggests that terminal OSs ought to provide application programming interfaces (APIs) so that power-aware applications may directly schedule shutdown and wakeup of terminal resources. In essence, just like CPU, memory, and network bandwidth, there is a need to make battery energy a first class citizen in the wireless terminal resource management scheme.

## *Low-Power Network Protocols and Networked Computation*

In a networked environment, the shutdown-based power management problem is made harder because of the need to interact with other network nodes in addition to the user. However, the networked operation of a wireless terminal opens up additional opportunities for increasing energy efficiency. One opportunity is the possibility of dynamically offloading computation from the local terminal, where battery energy is at a premium, to remote energy-rich nodes such as servers in the wired backbone of the network [25]. In essence, energy spent in communication is traded off for energy spent in computation if it is worthwhile to do so. Another opportunity comes from making various network protocols, such as link, MAC, routing, and transport protocols, power-aware so that they continually strive to provide the most energy-efficient transport of application bits while meeting the desired QoS. In essence, joules per application-level bit ought to be the key performance metric for protocols used with wireless terminals. Recent work has shown that energy-efficient MAC [26] and link-layer [27, 28] protocols can result in substantial reduction in battery energy.

Let us consider these opportunities for energy efficiency in a networked wireless terminal in more detail. In its most



■ **Figure 6.** *Energy efficiency of TCP data transmission with changing packet length under optimal FEC code for a given channel condition.*

abstract form, a networked wireless terminal has two sources of energy drain:

- Communication, due to energy spent by the RF transmitter and receiver
- Computation, due to signal processing and other tasks required during communication

Broadly speaking, minimizing energy consumption is a task that will require minimizing the contributions of computation and communication, making the appropriate trade-offs between the two. For example, reducing the amount of transmitted data and thus the amount of energy spent sending messages may be beneficial. On the other hand, the computation cost (e.g., to compress the data being sent) might be such that it would be better to just send the raw data. Of course, the actual trade-off will depend on the particulars of the system, the nature of the data being sent, and so on. Communication energy is chiefly dictated by the signal-to-noise ratio (SNR) requirements and the separation between the transmitter and the receiver: increasing both the separation and SNR requirement will increase the required transmission energy. Computation energy is a function of the hardware and software used for tasks such as compression and forward error correction (FEC). For long distance wireless links, the transmit communication energy component would dominate. However, for short distance wireless links such as in picocellular environments where transmit power is low, and in difficult channels where much signal processing and protocol computation may be used, the second component can be significant or even dominant. The trend toward smaller cells with low-power transmitters indicates that the ratio of computation to transmit power is likely to increase.

Let us first consider optimizing the energy consumption associated with communication alone. The radio state is controlled by the MAC protocol used. Many MAC protocols for wireless networks are basically adaptations of MAC protocols used in wired networks, and ignore energy issues. For example, random access MAC protocols such as carrier sense multiple access with collision avoidance (CSMA/CA) and 802.11 typically require a wireless terminal radio to be continually in receive mode and monitor the channel for packets. While such asynchronous communication provides a simple mechanism for terminals to exchange data with no prearranged knowledge of each other's state, it also requires that the terminal be constantly "awake" and ready to receive new data. A better strategy would be a MAC protocol that would allow a terminal to wake up at a time according to a schedule that is broadcast by a base station or central coordinator. Similarly, a MAC protocol can reduce power consumption by using slot reservations to minimize or eliminate energy-wasting packet collisions. Such power-aware MAC protocols significantly reduce energy consumption [29]; examples include EC-MAC [26] and LPMAC [22]. Another possibility is for a MAC protocol to "batch" packets by holding non-time-critical packets until there are a certain number of packets to send. Such batching can save power by reducing the total communication overhead.

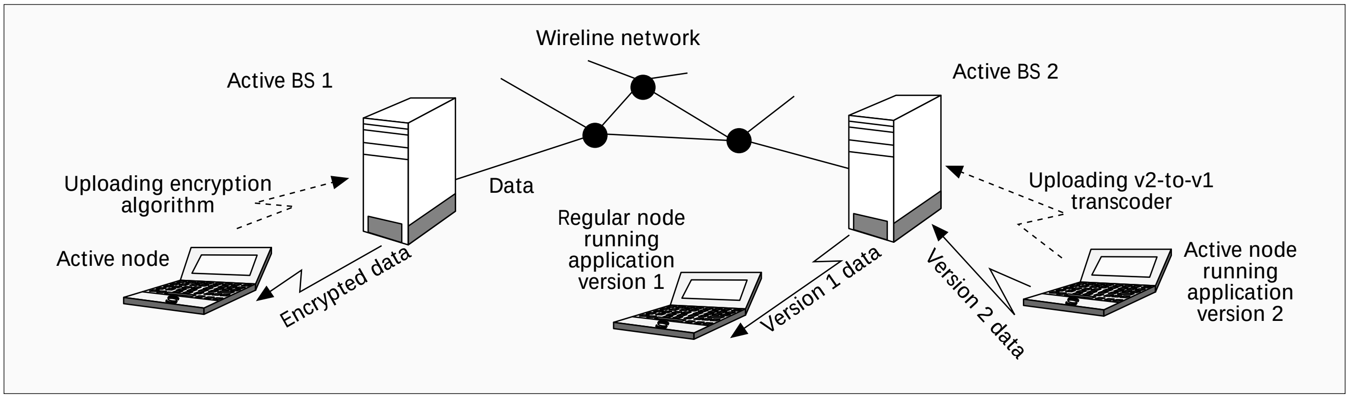
The next step is to reduce the amount of data which must be pushed through the channel. This goal can be realized in a number of ways. One is to reduce the amount of signaling overhead at all layers in the protocol stack, from the application all the way down through the physical layer. This involves developing protocols which are sensitive to the needs of the energy-poor devices. A second way to achieve the goal is to reduce the amount of user data that needs to be transmitted, which is largely an application-layer issue. For example, one might introduce compression, increase the compression rate, or possibly reduce the data resolution. Instead of sending an entire large full-color image, one can send black-and-white half-size images with lossy compression.

This brings us to the second energy component, the computational cost, which may be used to offset the energy costs of communication. It must be remembered that computation is not free and also consumes battery energy. One approach to doing a computation–communication trade-off is to realize that not all nodes on a wireless network are equal: infrastructure nodes, such as base stations, are not battery-operated. One can deliberately exploit this asymmetry, and use protocols that are unbalanced in terms of their resource requirements. A good example is the AIRMAIL link layer protocol [30] which uses a hybrid automatic repeat request (ARQ)/FEC design that strives to put the bulk of the complexity at the base station via an asymmetric ARQ protocol instead of dividing the complexity equally between the mobile and the base station.

In general, it is necessary to optimize all network protocols for low power, as opposed to the traditional metrics of throughput and latency. The right design metric for protocols in wireless terminals is *the amount of battery energy consumed to transmit bits across a wireless link while meeting QoS constraints*. Our research in [27, 28] shows that by appropriately and continually adapting link-layer protocol parameters such as FEC code, ARQ policy, and frame size, one can significantly reduce the energy consumption in highly time-varying fading channels. For example, Fig. 6 shows significant improvements in energy spent per bit in a TCP application by appropriate selection of packet size together with an FEC code rate that is optimally selected according to the bit error rate (BER) encountered in a fading channel. Higher-layer protocols may also be optimized for energy efficiency. For example, route selection in a multihop wireless network may be done to minimize power consumption at the intermediate transit nodes, and the transport protocols may use techniques such as minimizing acknowledgments (ACKs) to reduce energy consumption.

## The Reconfigurability of Terminal Architecture

Wireless terminals face many different types of variability in their environment. Some of these, such as variations in radio channel conditions, are handled by adaptive communication algorithms that vary their parameters according to current


Wireline network
Active BS 1
Active BS 2
Uploading encryption algorithm
Data
Uploading v2-to-v1 transcoder
Regular node running application version 1
Active node
Encrypted data
Version 1 data
Version 2 data
Active node running application version 2

■ **Figure 7.** *Scenarios using uploadable and downloadable functions.*

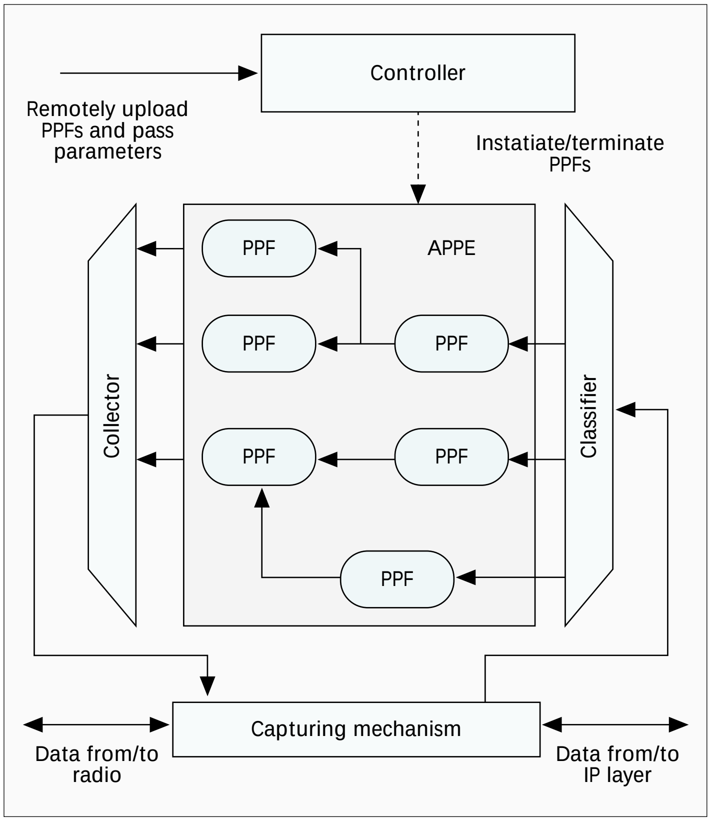
channel conditions. Adaptive equalization and adaptive error control are examples of such techniques. However, wireless terminals may often encounter a much larger degree of environmental variability. Mere algorithmic parameter adaptation does not work under such conditions. Rather, an entirely new set of protocols and algorithms may be required. For example, a wireless terminal may roam to the service area of a different service provider who uses a different air interface, MAC protocol, codec standard, and so forth. A similar situation occurs when wireless terminals and network equipment of multiple agencies and organizations need to interoperate during a joint task or mission, such as in rapidly deployable ad hoc networks established during disaster relief, or in the battlefield with forces from multiple allied countries.

To combat such a high degree of variability in operational environment, one approach would be to have wireless terminals with built-in algorithms corresponding to all possible scenarios. Such multimode terminals become too costly, and the set of scenarios they can handle remains fixed. Another alternative would be wireless terminals whose capabilities can be customized by plugin modules. This, however, requires plugin modules to be carried along with the terminal, detracting from the portability.

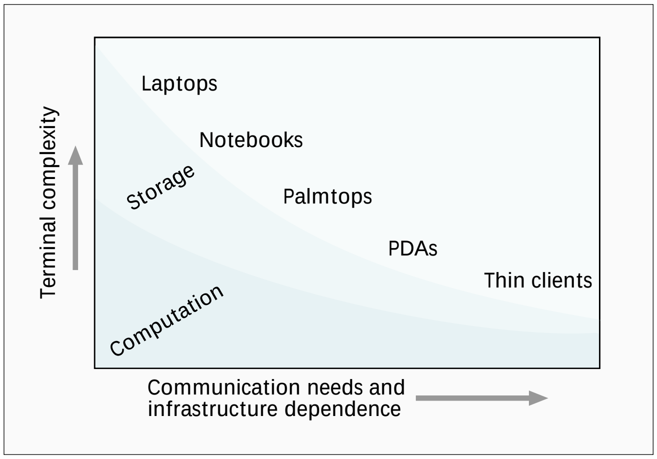
A possible solution we are exploring at UCLA [16] is to have wireless terminals with reconfigurable architectures to which new software and hardware functions can be downloaded from network servers, thereby modifying terminal capabilities. In our wireless terminals, application-specific processing of packets can be done by packet processing functions (PPFs) that are dynamically instantiated at the terminal and interconnected in a pipelined fashion. Thus, these terminals are referred to as *active nodes*. A similar capability exists in our *active base stations*. The PPFs are either software functions (Java) or hardware data paths (in rapidly and incrementally reconfigurable field programmable gate arrays, FPGAs). The PPFs may come from a local library or be remotely downloaded. In a secure setting, the choice of PPF that may be installed at infrastructure nodes such as base stations is likely to be restricted to a library of functions preapproved by the service provider.

Figure 7 illustrates the possible applications of reconfigurable active nodes and base stations via two usage scenarios. In the first, a user with a wireless terminal enters an insecure environment where over-the-air security is not sufficient. Fortunately, base station 1 (BS1) is an active base station, so the user's wireless terminal can upload (or arrange to be loaded from a network server) its own encryption algorithm in the form of a PPF that transforms the incoming packet stream. This particular PPF encrypts all packets destined for the user before they go into the air.

In the second example we see two soldiers in a battlefield who are from different countries or services, and part of a joint force in a United Nations peace mission. Their wireless terminals understand different formats for data, such as compressed speech or image using different versions of a codec. In a more benign civilian setting, this is a common occurrence when different government services collaborate during disaster relief or emergencies. If an active base station were to be present, one of the wireless terminals can upload a transcoder to base station BS2, and thereby exchange data with the other user. The important thing to notice in these examples is that apart from the improvement one can get in link quality using adaptive techniques, with reconfigurable terminals and base stations one is no longer dependent on a central network administrator to deploy or activate new services. The designers of different services may develop new PPFs to provide new services and protocols very easily, thus facilitating the deployment of new technologies. Such reconfigurability would also help in field



■ **Figure 8.** *Reconfigurable packet processing wireless nodes [13].*



■ **Figure 9.** *The spectrum of terminals.*

upgrades of wireless terminals as new communication protocols or standards are deployed, and in implementing bug fixes.

Figure 8 shows one possible architecture to support reconfigurable packet processing. At the core is an active packet processing engine (APPE) in which flow graphs composed of PPFs are instantiated. These PPFs implement low-level communication functions such as error control, encryption, and multimedia signal processing, and are implemented as either *software threads* on processors embedded in the APPE, or *hardware threads* in the form of data paths in reconfigurable hardware in the APPE. PPFs are connected one after the other to form arbitrary flow graphs, and may even be shared between flow graphs. The APPE includes a signaling port via which the network management software on a Controller instantiates/configures/terminates each flow graph as necessary via signaling from a link control processor elsewhere in the system.

As packets pass through the APPE, they are thus processed on the fly via an adaptable flow graph of software and hardware PPFs specific to their class. Toward that end, in front of the APPE's packet input port, a Classifier entity directs the packets onto the different processing paths according to the class to which they belong. The classes correspond to IP flows based on source and destination address and port ranges. The packets then flow through the correct PPF network, with the packet's path having been determined by the class it belongs to. At the other end of the APPE, the packets are collected by a Collector entity and passed to the radio or higher-layer network protocols, depending on whether this was an inbound or outbound packet.

## Location/Context Awareness

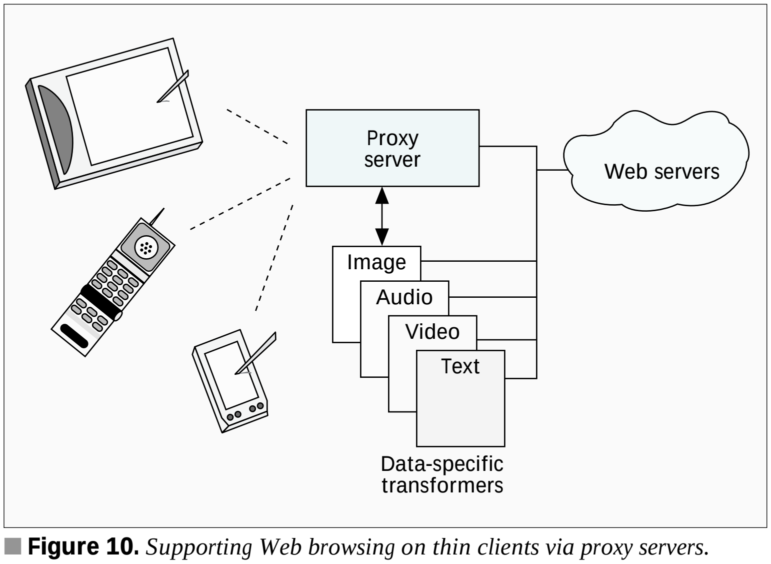
Due to their mobility, wireless terminals can move to different locations and encounter varying operating environments. Potentially, applications can adapt their behavior to such changes in context. For example, an application may combat degradation in its operating environment by using more advanced algorithms, or enhance the service quality by making use of new resources in its operating environment, or tailor its behavior to the presence of other objects and persons nearby.

Such context adaptivity and awareness are relatively new tools for the designers of wireless systems and services, and their impact has yet to be fully exploited. However, proper exploitation will require support from the terminal architecture in the form of terminal hardware that provides context information, and terminal software that implements runtime mechanisms so that applications can efficiently monitor the context and adapt to it. Technological advances, particularly in microsensor and GPS technologies that can be integrated into a wireless terminal, promise to soon make such context awareness possible.

Location is perhaps the most useful context information for a wireless terminal. While the technology to obtain location information has been around for some time, recent reductions in size and power consumption of the needed hardware has started to make such location information possible in the small battery-operated mobile nodes where that information is most useful. One enabling technology in this case is GPS, and the ever shrinking and lower-power GPS receiver hardware. For example, single-chip GPS receivers consuming a few tens of milliwatts of power will soon be possible [31]. With an unobstructed view of a few GPS satellites, a GPS receiver module can determine with great accuracy its own coordinates in space and time (longitude, latitude, altitude, and time). A GPS receiver module can locate itself with a precision on the order of meters in any of the spatial dimensions, and to atomic clock accuracies in time.

Another enabling technology making location and context awareness possible in wireless terminals is microsensor technology. Among other things, very fine positional and velocity measurements can be achieved through the use of inertial sensors and accelerometers that can now fit on a single MEMS chip [9]. Sensors can potentially provide detailed context information including speed, acceleration, orientation, temperature, humidity, sound, light, presence of biological, chemical, and nuclear hazards, proximity to other objects, and so on. Already, research groups at UCLA and Rockwell Science Center [9] and at the Royal Institute of Technology, Stockholm [32], are exploring wireless nodes that incorporate tightly integrated sensors to enable novel context-aware applications and services.

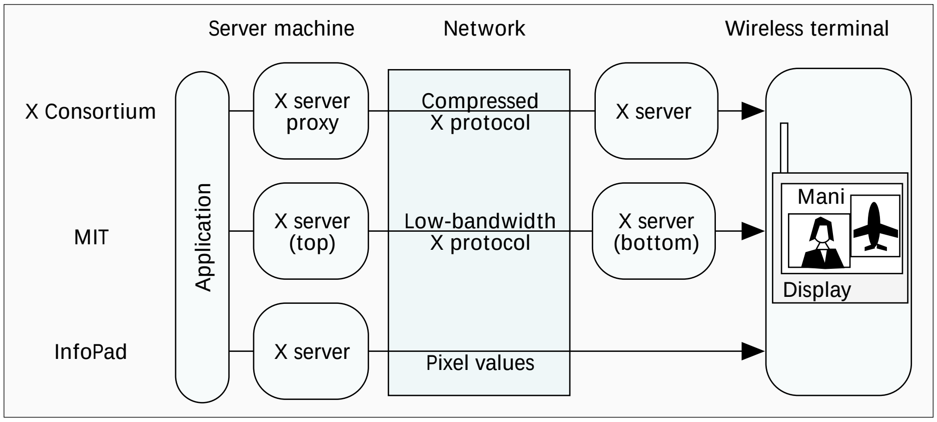
An interesting effort to make use of location awareness in wireless terminals is the notion of Geo-routing from Rutgers University [33]. This network-layer concept allows data messages to be routed according to geographical position, as opposed to the usual IP addresses and multicast groups. All entities within a particular region during an interval of time, regardless of their particular IP address, can be targeted to receive a message. For example, if the weather service were



■ **Figure 10.** *Supporting Web browsing on thin clients via proxy servers.*

to become aware of tornado danger in a particular county, the entire county could be the target of a tornado alert message. Any terminals in the area, including both permanent residents and terminals in cars passing along the highways, would receive the message. This technique clearly makes use of geographic knowledge.

The above technique, however, only scratches the surface of the possibilities with context-aware terminals. Consider this idea on a much smaller scale. Imagine a pen which can keep track of its orientation using a MEMS sensor, and can sense when pressure is being applied. If this pen were to also have a wireless link to the network, one could envision recording one's pen strokes simply by writing on a standard piece of paper. Pen input devices today tend to refer to a pressure-sensitive tablet. However, this is limiting in that the tablet is usually rather large and bulky, relatively speaking. A pen is much easier to carry around. In addition, writing on a tablet does not provide the user with the same feedback as writing with pen on paper, something most people are already quite used to. An average user is likely to find a "real" pen a more satisfying input device.



■ **Figure 11.** *Different ways of partitioning a graphics window server.*

As another scenario, consider the use of smart wireless toys. Imagine a toy which can recognize the child with whom it is playing, and tailor its behavior correspondingly. Through either voice recognition or a personal identification badge sewn into the child's clothing, the educational toy identifies the child playing with it. With this knowledge, an application running somewhere in the infrastructure may search a database to determine the child's proficiency at the particular educational game in which the toy is used, and run the toy at the appropriate level. The child does not need to do anything "unnatural" to evoke the right behavior from the toy. The wireless terminal, in this case the toy, uses the context information obtained from sensor information to modify behavior accordingly.

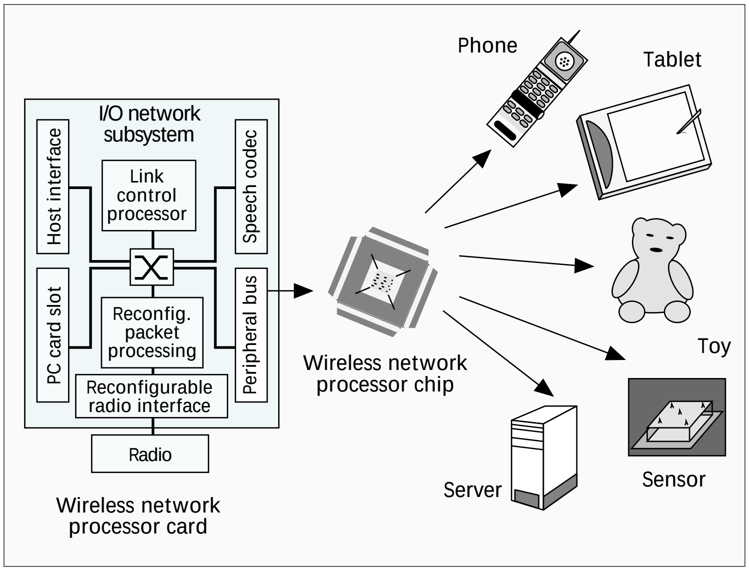
As the above examples show, if wireless terminals have knowledge of their context, many possibilities suddenly open up. Context awareness married to computation and wireless communication take the notion of wireless terminals far beyond the PDAs, PCs, cell phones, and other wireless terminals of today. Single-chip GPS receivers and MEMS sensors will make such wireless terminals possible soon. However, the full extent of what can be done with context information most certainly has not yet been explored by applications and services.

## Functional Partitioning between the Wireless Terminal and Network Infrastructure

A shown in Fig. 9, there is a spectrum of wireless terminals with different trade-offs of terminal complexity (local storage and computation), communication needs, and infrastructure dependence. This computation–communication trade-off affects the terminal cost, service cost, and application structure. Indeed, the design of wireless terminals must be done in a "network-integrated" fashion, and may be dictated by the needs of the service being offered. In particular, with a high-speed wireless data network available, many different architecture choices become possible, each with different partitioning of functions between the wireless terminal and servers resident in the network. Services whose economics would only allow simple terminals with limited CPU and memory resources might offload tasks to network servers. In security-conscious applications where loss of a terminal with its stored data may be undesirable, terminals may not keep data locally and instead rely solely on network storage servers together with encryption and authentication. A terminal whose battery runs low or context changes may dynamically move some of its functions to agents running on servers in the network.

Partitioning of functions between the wireless terminal and the network is an important architecture decision which dictates where applications can run, where data can be stored, the complexity of the terminal, and the cost of communication service. For example, a wireless terminal may provide access to information from the Web by either incorporating full-fledged PC like capabilities, such as in the case of Nokia 9000 Communicator, or relying on network servers to do the bulk of the work and using simple terminals, as in the case of AT&T PocketNet telephone [17], CruisePAD [18], Infopad [6, 15], and thin clients such as Palm Pilot and WinCE machines with a proxy server such as Berkeley's Wingman [19] or ProxiNet's ProxiWeb [20]. In Berkeley's InfoPad wireless terminal, the terminal itself is essentially a set of peripherals: frame buffer with display, camera, pen, speaker, and microphone. No local computation, except for appropriate coding/decoding of the I/O data, is done at the terminal. In Wingman, as shown in Fig. 10, a proxy process running on a network server transforms incoming Web data into a form suitable for the limited display resources of a Palm Pilot PDA. For example, high-resolution color images are converted into lower-resolution black-and-white images suitable for the PDA, while allowing the user to selectively zoom in. The resulting reduction in size reduces transmission time and more than compensates for time spent in processing at the proxy server. By leveraging network-resident compute resources, Wingman gives users on limited thin clients a fully graphical Web browsing experience as opposed to the typical text-only Web browsers that such PDAs natively support. Another example of different possible functional partitioning between terminals and networks are the three different implementations of a wireless terminal with the X window system, as shown in Fig. 11. The three implementations split the X server across the wireless link differently, with corresponding trade-offs of terminal complexity, performance, and communication bandwidth usage.

The above discussion illustrates the diversity of functional



■ **Figure 12.** *The wireless system-on-chip as the technology enabler for the variety of future wireless terminals.*

partitioning across the terminal and network, and the dependence of partitioning on changing context and usage. In addition to indicating that it is often meaningless to talk about a wireless terminal in isolation from the network services supporting it, the above discussion also suggests the need in many cases for an adaptive terminal–network boundary across which application functions may be migrated in a context-dependent fashion. The key implication for the future wireless terminal architecture is that the runtime hardware and software environment on the terminal and in the network should be able to support such adaptivity, and provide application developers with appropriate interfaces to control it. Software technologies such as proxies and mobile agents/applets, and hardware technologies such as adaptive and reconfigurable computing hardware are likely to be the key enablers.

## Conclusions

In this article we have identified energy-efficient design, autoconfiguration, adaptivity, architectural reconfigurability, location and context awareness, terminal–network functional partitioning, security, user interfaces, and integrated RF design as the key issues in future wireless terminal design. However, we want to emphasize that more important than these specific technical challenges and opportunities are the coming shift in paradigm away from the notion of wireless terminals as a portable computer-cum-phone toward a myriad of wireless terminals in the form of gadgets, appliances, and devices with embedded wireless networking capability. A key enabler will be progress in microelectronics that will enable an entire *wireless systems on a chip*, with radio, internetworking protocol, signal processing, and computation functions to be integrated on a single die. As proof of concept, our research at UCLA is pursuing this vision of wireless terminals, where all the necessary hardware and software functions are being integrated into a single embeddable module. The module, called the Wireless Adaptive Network Device (WAND), embodies the architectural ideas articulated in this article to integrate radio, packet processor, link controller, protocols, and codecs in a single package that is currently a small board and which we strive to integrate on a single-chip wireless system. As shown in Fig. 12, we expect technologies such as this to be embedded in all types of wireless devices in the environment, not just hand-carried terminals.

## Acknowledgments

This work was partially supported under DARPA/ITO contract DABT63-95-C-0100, NSF CAREER award MIP-9733331, GTE Foundation, and the UCLA School of Engineering and Applied Science. We would also like to thank Athanassios Boulis for his help and contributions.

## References

[1] http://www.nokia.com/com9000/n9000.html

[2] S. Narayanaswamy, J. Hu, and R. S. Kashi, "Using Data on Digital Cellular and PCS Voice Networks," *Bell Labs Tech. J.*, vol. 3, no. 2, Apr.-June 1998, pp. 58–75.

[3] T. Melanchuk, P. Dupont, and S. Backer, "CDPD and emerging digital cellular systems," *Dig. Papers COMPCON '96*, Santa Clara, CA, Feb 25–28, 1996, pp. 2–8.

[4] S. Akesson, "GPRS, General Packet Radio Service," *Proc. ICUPC '95,* Tokyo, Japan, Nov. 6–10 1995, pp. 640–43.

[5] "The Future of Computing: After the PC," *The Economist*, Sept. 12–18, 1998, pp. 79–81.

[6] R. Brodersen, "The InfoPad Project: Review and Lessons Learned," *Proc. 11th Int'l. Conf. VLSI Design*, Chennai, India, Jan. 4–7, 1998, pp. 2–3.

[7] P. Lettieri, A. Boulis, and M. B. Srivastava, "Design of Adaptive Wireless Terminals," *1998 Int'l. Symp. Sys., Signals, and Elect.*, Pisa, Italy, Sept. 29–Oct. 2, 1998.

[8] G. Asada *et al.*, "Low power wireless communication and signal processing circuits for distributed microsensors," *Proc. 1997 IEEE Int'l. Symp. Circuits and Sys.*, Hong Kong, June 9–12, 1997, pp. 2817–20.

[9] http://www.janet.ucla.edu/lpe.lwim

[10] J. Goodman *et al.*, "Signal Processing for an Ultra-Low-Power Wireless Video Camera," *Proc. 3rd Int'l. Wksp. Mobile Multimedia Commun.*, Princeton, NJ, Sept. 25–27, 1996, pp. 267–74.

[11] http://www.microsoft.com/actimates

[12] W. Mangione-Smith, "Mobile computing and smart spaces," *IEEE Concurrency*, vol. 6, no. 3, Oct.–Dec. 1998, pp. 5–7.

[13] A. P. Chandrakasan and R.W. Brodersen, "Minimizing Power Consumption in Digital CMOS Circuits," *Proc. IEEE*, vol. 83, no. 4, Apr. 1995, pp. 498–523.

[14] C. Chien *et al.*, "Design Experience with an Integrated Testbed for Wireless Multimedia Computing," *Proc. 3rd Int'l. Wksp. Mobile Multimedia Commun.*, Princeton, NJ, Sept. 25–27, 1996, pp. 231–38.

[15] S. Narayanaswamy *et al.*, "A low-power, lightweight unit to provide ubiquitous information access application and network support for InfoPad," *IEEE Pers. Commun.*, Apr. 1996, vol. 3, no. 2, pp. 4–17.

[16] A. Boulis, P. Lettieri, and M. B. Srivastava, "Active Basestations and Nodes for a Mobile Environment," *Wksp. Wireless Networking*, Dallas, TX, Oct. 30, 1998, pp. 31–37.

[17] http://www.attws.com/nohost/data/pocketnet/equipment.html

[18] http://www.cruisetech.com

[19] http://www.isaac.cs.berkeley.edu/pilot/wingman

[20] http://www.proxinet.com/proxiweb

[21] Y. Manoli and W. Mokwa, "Silicon Microsystems — Merging Sensors, Circuits and Systems," *Proc. 1998 IEEE Int'l. Conf. Comp. Design: VLSI in Comps. and Processors*, Austin, TX, Oct. 5–7, 1998, pp. 634–41.

[22] T.-H. Lin *et al.*, "Micropower CMOS RF components for distributed wireless sensors," *1998 IEEE Radio Frequency Integrated Circuits Symp. Dig. of Papers*, Baltimore, MD, June 7–9, 1998, pp. 157–60.

[23] C. Chien *et al.*, "Processing-Gain Adaptive Multiband Radio Technology for Wireless Multimedia Communications," *IEEE Int'l. Wksp. Mobile Multimedia Commun.*, Oct. 2, 1997.

[24] M. B. Srivastava, A. P. Chandrakasan, and R. W. Brodersen, "Predictive System Shutdown and Other Architectural Techniques for Energy Efficient Programmable Computation," *IEEE Trans. VLSI Sys.*, vol. 4, no. 1, Mar. 1996, pp. 42-45.

[25] A. Rudenko *et al.*, "Saving portable computer battery power through remote process execution," *Mobile Comp. and Commun. Rev.*, Jan. 1998, vol. 2, no. 1, pp. 19–26.

[26] K. M. Sivalingam et al., "Low-Power Access Protocols Based on Scheduling for Wireless and Mobile ATM Networks," *Proc. ICUPC '97*, San Diego, CA, Oct. 12–16, 1997, pp. 429–33.

[27] P. Lettieri, C. Fragouli, and M. B. Srivastava, "Low Power Error Control for Wireless Links," *Proc. Mobicom '97*, Budapest, Hungary, Sept. 26–30, 1997, pp. 139–50.

[28] P. Lettieri and M. B. Srivastava, "Adaptive Frame Length Control for Improving Wireless Link Throughput, Range, and Energy Efficiency," *Proc. IEEE INFOCOM '98*, San Francisco, CA, Mar. 29–Apr. 2, 1998, pp. 564–71.

[29] J.-C. Chen *et al.*, "A Comparison of MAC Protocols for Wireless Local Networks Based on Battery Power Consumption," *Proc. IEEE INFOCOM '98*, San Francisco, CA, Mar. 29–Apr. 2, 1998, pp. 150–57.

[30] E. Ayanoglu *et al.*, "AIRMAIL: a link-layer protocol for wireless networks," *Wireless Networks*, vol. 1, no. 1, 1995, pp. 47–60.

[31] http://havefun.Stanford.EDU/mobilenet/research/gps

[32] G. Q. Maguire Jr., M. T. Smith, and H. W. P. Beadle, "SmartBadges: A wearable computer and communication system," *Codes/CASHE '98*, Seattle, WA, Mar. 16, 1998.

[33] J. C. Navas and T. Imielinski, "Geographic Addressing and Routing," *Proc. MobiCom '97*, Budapest, Hungary, Sept. 26–30, 1997, pp. 66–76.

## Additional Reading

[1] W. Mangione-Smith *et al.*, "A Low Power Architecture for Wireless Multimedia Systems: Lessons Learned from Building a Power Hog," Proc. 1996 Int'l. Symp. Low Power Elect. and Design, Monterey, CA, Aug. 12–14, 1996, pp. 23–28.

## Biographies

PAUL LETTIERI (lettieri@janet.ucla.edu) received a B.S.E.E. magna cum laude from Rensselaer Polytechnic Institute, Troy, New York, in 1995. While pursuing this degree, he did an internship with Pitney Bowes in Shelton, Connecticut, designing and evaluating embedded systems. Funded in part by a MICRO fellowship, in 1997 he obtained an M.S.E.E. at the University of California, Los Angeles, where he now continues his studies working toward a Ph.D. His current research focuses on wireless systems with an emphasis on link-layer protocols. Additional interests include embedded systems and reconfigurable computing.

MANI SRIVASTAVA (mbs@janet.ucla.edu) is an associate professor of electrical engineering at UCLA. He received his B.Tech. in electrical engineering from the Indian Institute of Technology, Kanpur, and his M.S. and Ph.D. from Berkeley. From 1992 through 1996 he was a member of technical staff at Bell Laboratories in networked computing research. His current research interests are in mobile and wireless networked computing systems, low-power systems, and hardware and software synthesis of networked embedded systems. He received the Okawa Foundation Grant award in 1998, the NSF CAREER award in 1997, and the President of India Gold Medal in 1985.