The Challenges of Mobile Computing

George H. Forman

John Zahorjan

Computer Science & Engineering
University of Washington

March 9, 1994

Abstract

Advances in wireless networking technology have engendered a new paradigm of computing, called *mobile computing*, in which users carrying portable devices have access to a shared infrastructure independent of their physical location. This provides flexible communication between people and continuous access to networked services. Mobile computing is expected to revolutionize the way computers are used.

This paper is a survey of the fundamental software design pressures particular to mobile computing. The issues discussed arise from three essential requirements: the use of wireless networking, the ability to change locations, and the need for unencumbered portability. Promising approaches to address these challenges are identified, along with their shortcomings.

Keywords: mobile computing, hand-held computers, PDAs, surveys, wireless communication, networks, disconnection, low bandwidth, data security, mobility, location dependence, portability, low power, small user interfaces

---

Available as UW CSE Tech Report # 93-11-03 from ftp.cs.washington.edu.
An edited version of this paper has been accepted for publication in IEEE Computer.

Contents

1 Introduction

Recent advances in technology enable portable computers to be equipped with wireless interfaces, allowing networked communication even while mobile. Whereas today's notebook computers and personal digital assistants (PDAs) are self-contained, tomorrow's networked *mobile computers*[1] are part of a greater computing infrastructure. *Mobile computing* constitutes a new paradigm of computing that is expected to revolutionize the way computers are used.

Wireless networking greatly enhances the utility of carrying a computing device. It provides mobile users with versatile communication to other people and expedient notification of important events, yet with much more flexibility than cellular phones or pagers. It also permits continuous access to the services and resources of the land-based network. The combination of networking and mobility will engender new applications and services, such as collaborative software to support impromptu meetings, electronic bulletin boards that adapt their contents according to the people present, self-adjusting lighting and heating, and navigation software to guide users in unfamiliar places and on tours[14].

[1] We use the term *mobile computer* to denote a portable computer that is capable of wireless networking.

The technical challenges to establishing this paradigm of computing are non-trivial, however. In this paper we survey the principal challenges faced by the software design of a mobile computing system, as distinguished from the design of today's stationary networked systems. We discuss those issues pertinent to the software designer, without delving into the lower level details of the hardware realization of the mobile computers themselves. Where appropriate, we identify promising approaches that researchers have applied, as well as their limitations.

The issues described herein divide cleanly into three sections, each stemming from an essential property of mobile computing. Section 2 considers the implications of using *wireless communication*, for example, susceptibility to disconnection, low bandwidth availability, and highly variable network conditions. Section 3 discusses the consequences of *mobility*, including dynamically changing network addresses, location-dependent answers to user queries and system configuration, and communication locality that deteriorates as mobile users move away from their servers. Section 4 investigates the pressures that *portability* places on the design of a mobile system, such as low power, risk of data loss, and small surface area available for the user interface.

In order to expose a greater assortment of issues, the target in mind is large scale, hand-held mobile computing. Of course, special purpose systems may avoid some design pressures by doing without certain desirable properties. For instance, mobile computers installed in the dashboards of cars would be less concerned with the portability pressures than would hand-held mobile computers.

Within the notion of mobile computing, there is considerable latitude regarding the role of the portable device. Is it a terminal or an independent, stand-alone computer? How many purposes shall the device serve? Should it incorporate a telephone (as does the AT&T EO)? Should it provide the work environment of a general purpose workstation, or something more restrictive, such as the Apple Newton MessagePad? These design choices greatly affect the severity of the issues in the following sections. For example, a portable terminal, such as the PARC Tab[14], is more dependent on the network but less prone to loss of storage media than a stand-alone computer. It is important to consider such questions in relation to the issues presented below.

2 Wireless Communication

Mobile computers require wireless network access, although sometimes they may physically attach to the network for a better or cheaper connection when they remain stationary, such as during meetings or while at a desk.

Wireless communication is much more difficult to achieve than wired communication because the surrounding environment interacts with the signal, blocking signal paths and introducing noise and echoes. As a result, wireless connections are of lower quality than wired connections: lower bandwidths, higher error rates, and more frequent spurious disconnections. These factors can in turn increase communication latency due to retransmissions, retransmission timeout delays, error control protocol processing, and short disconnections.

Wireless connections can be lost or degraded also by mobility. Users may outstep the coverage of network transceivers or enter areas of high interference. Unlike typical wired networks, the number of devices in a cell varies dynamically, and large concentrations of mobile users, such as at conventions and public events, may overload network capacity.

The sections below cover the design challenges resulting from the need for wireless communication: more frequent disconnections, lower bandwidth, greater variation in available bandwidth, greater network heterogeneity, and increased security risks.

2.1 Disconnection

Today's computer systems often depend heavily on the network, and may cease to function during network failures. For example, distributed file systems may block waiting for other servers, and application processes may fail altogether if the network stays down too long.

Network failure is of greater concern to mobile computing designs than traditional designs, because wireless communication is so susceptible to disconnection. One can either spend more resources on the network trying to prevent disconnections, or spend those resources enabling systems to cope with disconnections more gracefully and work around them where possible.

The more autonomous a mobile computer, the better it can tolerate network disconnection. For example, some applications can reduce communication by running entirely locally on the mobile unit, rather than splitting the application and the user interface across the network. In environments with frequent disconnections it is more important for the mobile device to operate as a stand-alone computer, as opposed to a portable terminal.

In some cases both round-trip latency and short disconnections can be hidden by operating asynchronously. The X11 Window system uses this technique to achieve good performance. As opposed to the synchronous remote procedure call paradigm where the client waits for a reply after each request, in asynchronous operation a client sends multiple requests before asking for any acknowledgment. Similarly, prefetching and lazy write-back also decouple the act of communication from the actual time a program consumes or produces data, allowing it to make progress during network disconnections. These techniques, therefore, have the potential to mask some network failures.

The Coda file system provides a good example of handling network disconnections, although it is designed for today's notebook computers where disconnections may be less frequent, more predictable and longer lasting than in mobile computing[6]. Information from the user's profile is used to help keep the best selection of files in an on-board cache. It is important to cache whole files rather than blocks so that entire files can be read during a disconnection. When the network reconnects, the cache is automatically reconciled with the replicated master repository. Coda allows files to be modified even during disconnections. More conservative file systems disallow this to prevent multiple users from making inconsistent versions. Coda's optimism is justified by studies showing that only rarely are files actually shared in a distributed system; less than 1% of all writes are followed by a write by a different user[6]. In those cases where strong consistency guarantees are needed, clients can ask for them explicitly. Hence, providing flexible consistency semantics can allow better autonomy.

Of course, not all network disconnections can be masked. In these cases good user interfaces can help by providing feedback to the user about which operations are unavailable due to network disconnections.

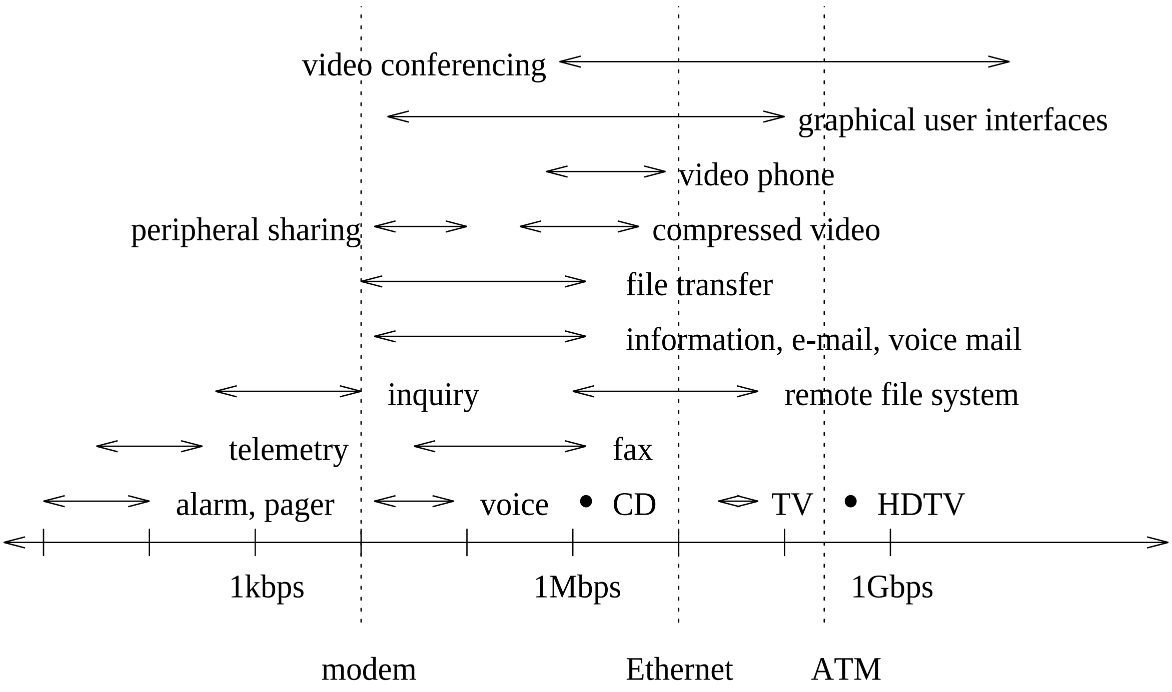

Figure 1:
This figure shows application bandwidth requirements laid out on a horizontal log-scale axis in bits per second (bps). The vertical lines show the bandwidth capability of a few network technologies. This figure clarifies which applications are suitable for a given bandwidth technology. The newest cellular modems are achieving speeds adequate for the everyday informational needs of mobile users, such as electronic mail, and some day may be able to support remote file systems.

2.2 Low Bandwidth

Mobile computing designs need be more concerned about bandwidth consumption and constraints than designs for stationary computing, because wireless networks deliver lower bandwidth than wired networks— cutting-edge products for portable wireless communications achieve only 1 Mbps for infrared communication, 2 Mbps for radio communication, and 9–14 kbps for cellular telephony, while Ethernet provides 10 Mbps, FDDI 100 Mbps, and ATM 155 Mbps. Even non-portable wireless networks, such as the Motorola Altair, barely achieve 5.7 Mbps.

Network bandwidth is divided among the users sharing a cell. The deliverable bandwidth per user, therefore, is a more useful measure of network capacity than raw transmission bandwidth. But because this measure depends on the size and distribution of a user population, Weiser and others promote measuring a wireless network's capacity by its bandwidth per cubic meter[15].

To improve network capacity, one can install more wireless cells to service a user population. There are two ways of doing this: overlap cells on different wavelengths, or reduce transmission ranges so that more cells fit in a given area.

The scalability of the first technique is limited, because the electromagnetic spectrum available for public consumption is scarce. This technique is more flexible, however, because it allows (and requires) software to allocate bandwidth among users.

The second technique is generally preferred. It is arguably simpler, reduces power requirements (see section 4.1), and may decrease corruption of the signal because it may interact with fewer objects in the environment. Also, there is a hardware tradeoff between bandwidth and coverage area— transceivers covering less area can achieve higher bandwidths.

Certain software techniques can also help cope with the low bandwidth of wireless links. Modems typically use compression to increase their effective bandwidth, sometimes almost doubling throughput. Because bulk operations are usually more efficient than many short transfers, logging can improve bandwidth usage by making large requests out of many short ones. Logging in conjunction with compression can further improve throughput because larger blocks compress better.

Certain software techniques for coping with disconnection can also help cope with low bandwidth. Typical network usage occurs in bursts, and disconnections are similar to bursts in that demand temporarily exceeds available bandwidth. For example, lazy write-back and prefetching use the valleys to reduce demand at the peaks. Lazy write-back can even reduce overall communication when the data to be transmitted are further mutated or deleted before they are transmitted. Prefetching involves knowing or guessing which files will be needed soon and downloading them over the network before they are demanded[10]. Bad guesses can waste network bandwidth, however.

System performance can be improved by scheduling communication intelligently. When available bandwidth does not satisfy the demand, priority should be given to those processes for which the user is waiting. Backups should be performed only with "leftover" bandwidth. Mail can be trickle fed onto the mobile computer slowly before the user is notified. Although these techniques do not increase effective bandwidth, they are equally important to improving user satisfaction.

2.3 High Bandwidth Variability

Mobile computing designs also contend with much greater variation in network bandwidth than traditional designs. Bandwidth can shift one to four orders of magnitude between being plugged in versus using wireless access. Fluctuant traffic load seldom causes this much variation in available bandwidth on today's networks.

An application can approach this variability in one of three ways: it can assume high bandwidth connections and operate only while plugged in, it can assume low bandwidth connections and not take advantage of higher bandwidth when it is available, or it can adapt to the currently available resources, providing the user with a variable level of detail or quality. Different choices make sense for different applications.

2.4 Heterogeneous Networks

In contrast to most stationary computers, which stay connected to a single network, mobile computers encounter more heterogeneous network connections. As they leave the range of one network transceiver they switch to another. In different places they may experience different network qualities, for example, a meeting room may have better wireless equipment installed than a hallway. There may be places where they can access multiple transceivers on different frequencies. Even when plugged in, they may concurrently use wireless access.

Also, they may need to switch interfaces when moving from indoors to outdoors. For example, infrared interfaces cannot be used outside because sunlight drowns out the signal. Even if only radio frequency transmission is used, the interface may still need to change access protocols for different networks, for example when switching from cellular coverage in a city to satellite coverage in the country. This heterogeneity makes mobile networking more complex than traditional networking.

2.5 Security Risks

Precisely because it is so easy to connect to a wireless link, the security of wireless communication can be compromised much more easily than wired communication, especially if the transmission range encompasses a large area. This increases pressure on mobile computing designs to include security measures.

Security is further complicated if users are to be allowed to cross security domains, for example, allowing the untrusted mobile computers of hospital patients to use nearby printers while disallowing access to distant printers and resources designated for personnel only.

Secure communication over insecure channels is accomplished by encryption, which can be done in software, or more quickly by specialized hardware, such as the recently proposed CLIPPER chip. The security depends upon a secret encryption key being known only to the authorized parties. Managing these keys securely is difficult, but can be automated by software such as MIT's Kerberos[9].

Kerberos provides secure authentication services, provided the Kerberos server itself is trusted. It authenticates users without exposing their passwords on the network and generates secret encryption keys that can be selectively shared between mutually suspicious parties. It also allows roaming mobile units to authenticate themselves in foreign domains where they are unknown, thus enhancing the scale of mobility. Methods have also been devised to use Kerberos for authorization control and accounting. Its security is limited, however.For example, the current version is susceptible to off-line password guessing attacks and to replay attacks for a limited time window.

3 Mobility

The ability to change locations while connected to the network increases the volatility of some information. Certain data considered static for stationary computing becomes dynamic for mobile computing. For example, although a stationary computer can be configured statically to prefer the nearest server, a mobile computer needs a mechanism to determine which server to use.

As volatility increases, cost-benefit tradeoff points shift, calling for appropriate modifications in the design. For example, greater volatility of a data object reduces its ratio of uses per modification. For lower ratios, it makes less sense to cache the data, or even to store it at all if it can be recomputed from scratch easily enough. As another example, where management of static information is often done by hand, automated methods are required to handle higher rates of change. Even where automated methods exist, many are ill-suited for the dynamicism of mobile computing.

The following three sections discuss the main problems introduced by mobility: the network address of a mobile computer changes dynamically; its current location affects configuration parameters as well as answers to user queries; and as it wanders away from a nearby server, the communication path between the two grows.

3.1 Address Migration

As people move, their mobile computers will use different network access points, or 'addresses.' Today's networking is not designed for dynamically changing addresses. Active network connections usually cannot be moved to a new address. Once an address for a host name is known to a system, it is typically cached with a long expiration time and with no way to invalidate out-of-date entries. In the Internet Protocol (IP), for example, a host IP name is inextricably bound with its network address— moving to a new location means acquiring a new IP name. Human intervention is often required to coordinate the use of addresses.

In order to communicate with a mobile computer, messages must be sent to its most recent address. There are four basic mechanisms for determining the current address of a mobile computer: broadcast[5, 4], central services[8], home bases[12], and forwarding pointers[5]. These are the building blocks of the current proposals for 'mobile-IP' schemes.

Selective Broadcast: With the broadcast method, a message is sent to all network cells, asking the mobile computer sought to reply with its current address. This becomes too expensive for frequent use in a large network, but if the mobile computer is known to be in some small set of cells, selectively broadcasting in those cells alone is workable. Hence, the methods described below can employ selective broadcast to obtain the current address when only approximate location information is known. For example, a slightly out-of-date cell address may suffice if adjacent cells are known.

Central Services: With the central service method, the current address for each mobile computer is maintained in a logically centralized database. Each time a mobile computer changes its address, it sends a message to update the database. Although this database is logically centralized, the common techniques of distribution, replication, and caching can be employed to improve availability and response time.

Home Bases: The home base method is essentially the limiting case of distributing a central service— only a single server knows the current location of a mobile computer. This brings with it the availability problems of aggressive distribution without replication. For example, if a home base is down or inaccessible, the mobile computers it tracks cannot be contacted. Note that if users sometimes change home bases, another instance of the address migration problem arises, albeit with much lower volatility.

Forwarding Pointers: With the forwarding pointer method, each time a mobile computer changes its address, a copy of the new address is deposited at the old location. Each message is forwarded along the chain of pointers until it reaches the mobile computer. To avoid the inefficient routing that can result from long chains, pointers at message sources can be updated to reflect more recent addresses.

Although the forwarding pointer method is among the fastest, it is prone to failures anywhere along the trail of pointers, and in its simplest form, does not allow forwarding pointers to be forgotten. Hence, forwarding pointers are often employed only to speed the common case and another method is used to fall back on for failures and to allow reclamation of old pointers.

Note that the forwarding pointer method requires an active entity at the old address to receive and forward messages. This does not fit standard networking models, where a network address either is a passive entity, such as an Ethernet cable, or is specific to the mobile computer, which cannot remain to forward its own messages. This mismatch introduces subtle difficulties in implementing forwarding efficiently (such as with intra-cell traffic, or when multiple gateways are attached to a network address).

3.2 Location Dependent Information

Because traditional computers do not move, information that depends on location is configured statically, such as the local name server, available printers, and the time zone. A challenge for mobile computing is to factor out this information intelligently and provide mechanisms to obtain configuration data appropriate to the present location.

Besides this dynamic configuration problem, mobile computers need access to more location sensitive information than stationary computers if they are to serve as guides in places unfamiliar to their users, for example, to answer queries like "where is the fiction section (in this library)?" or "where is the nearest open gas station heading north?"

Whereas such queries require static location information about the world, Badrinath and Imielinski are studying a related class of queries that depends on the dynamic locations of other mobile objects, for example, determining where the nearest taxi is[4].

Privacy: Answering these queries requires knowing the location of another mobile user. In some cases this may be sensitive information, more so if given at a fine resolution. Even where it is not particularly sensitive, such information should be protected against misuse, for example, to prevent a burglar from determining when the inhabitants of a house are far away.

Privacy can be ensured by denying users the ability to know another's location. The challenge for mobile computing is to allow more flexible access to this information without violating privacy, for there are many legitimate uses of location information, including contacting colleagues, routing telephone calls, logging meetings in personal diaries, and tailoring the content of electronic announcement displays to the viewers[15].

3.3 Migrating Locality

Mobile computing engenders a new kind of locality that migrates as users move. Even if a mobile computer spends the effort to find the server that is nearest for a given service, over time it may cease to be the nearest due to migration. Because the physical distance between two points does not necessarily reflect the network distance, the communication path can grow disproportionately to actual movement. For example, a small movement can result in a much longer path when crossing network administrative boundaries. A longer network path means communication traverses more intermediaries, resulting in longer latency and greater risk of disconnection. This also consumes more network capacity, even though the bandwidth between the mobile unit and the server may not degrade.

To avoid these disadvantages, service connections may be dynamically transferred to servers that are closer[3]. When many mobile units converge, such as during meetings, load balancing concerns may outweigh the importance of communication locality.

Table 1: Characteristics of Personal Digital Assistant products and the AT&T EO tablet computer. Each has a pen interface and a black & white reflective LCD screen. The portable PC is included for comparison. (These data were gathered from advertisements, company representatives, and product reviews, such as those in PC Magazine, October 1993.)

Product	RAM	MHz	CPU	Batteries (hours,# & type)	Weight (lbs.)	Display (pixels, sq.inches)
Amstrad Pen Pad PDA600	128 KB	20	Z80	40, 3 AAs	0.9	240×320, 10.4
Apple Newton MessagePad	640 KB	20	ARM	6–8, 4 AAAs	0.9	240×336, 11.2
Apple N. MessagePad 110	1 MB	20	ARM	50, 4 AAs	1.25	240×320, 11.8
Casio Z-7000 PDA	1 MB	7.4	8086	100, 3 AAs	1.0	320×256, 12.4
Sharp ExpertPad	640 KB	20	ARM	20, 4 AAAs	0.9	240×336, 11.2
Tandy Z-550 Zoomer PDA	1 MB	8	8086	100, 3 AAs	1.0	320×256, 12.4
AT&T EO 440 Personal Communicator	4–12 MB	20	Hobbit	1–6, NiCad	2.2	640×480, 25.7
Portable PC	4–16 MB	33–66	486	1–6, NiCad	5–10	640×480, 84 (or 1024×768)

4 Portability

Today's desktop computers are not intended to be carried, so their design is liberal in their use of space, power, cabling, and heat dissipation. In contrast, the design of a hand-held mobile computer should strive for the properties of a wristwatch: small, light weight, durable, water-resistant and long battery life. Concessions can be made in each of these areas to enhance functionality, but ultimately the value provided to the user must exceed the trouble of carrying the device. Similarly, any specialized hardware to offload from the CPU tasks such as data compression or encryption should justify its consumption of power and space.

In the sections below we describe the design pressures caused by portability constraints: low power, heightened risk of data loss, small user-interfaces, and limited on-board storage. These pressures are evident in the designs of the recent PDA products listed in Table 1, as will be related below.

Table 2: Power consumption of the components of a portable computer and accessories. (The data for the computer components were derived from the Sharp PC 6785 manual. The data for the accessories were obtained from manufacturers; starred figures are estimates for PCMCIA products that are soon to be released.)

Device	Power (Watts)
base system (2MB, 25 MHz CPU)	3.650
base system (2MB, 10 MHz CPU)	3.150
base system (2MB, 5 MHz CPU)	2.800
screen backlight	1.425
hard drive motor	1.100
math co-processor	.650
floppy drive	.500
external keyboard	.490
LCD screen	.315
hard drive active (head seeks)	.125
IC card slot	.100
additional memory (per MB)	.050
parallel port	.035
serial port	.030
Accessories:	
1.8" PCMCIA hard drive	0.7–3.0
cellular telephone active	5.400
cellular telephone standby	.300
infrared network– 1 Mbps*	.250
PCMCIA modem– 14400 bps	1.365
PCMCIA modem– 9600 bps	.625
PCMCIA modem– 2400 bps	.565
global positioning receiver*	.670

4.1 Low Power

Batteries are the largest single source of weight in a portable computer. While reducing battery weight is important, too small a battery can undermine portability— users may have to recharge frequently, carry spare batteries, or use their mobile computer less. Minimizing power consumption can improve portability by reducing battery weight and lengthening the life of a charge.

Power consumption of dynamic components is proportional to CV^2F, where C is the capacitance of the wires, V is the voltage swing, and F is the clock frequency. This function suggests three ways to save power. (1) Capacitance can be reduced by greater levels of VLSI integration and multichip module technology. (2) Voltage can be reduced by redesigning chips to operate at lower voltages. Historically, chips operate at five volts, but to save power, the Apple MessagePad operates at three volts. Manufacturers are rapidly developing a line of low-power chip sets for 2.5 and 3.3 volt operation. (3) Clock frequency can be reduced, trading off computational speed for power savings. PDA products have adopted this concession, as shown in Table 1. In some notebook computers, the clock frequency can be changed dynamically, providing a flexible tradeoff; for example, the Sharp PC 6785 can save power by dynamically shifting its clock from 25 MHz down to 10 MHz or even 5 MHz, as seen in Table 2. In order to retain more computational power at lower frequencies, processors are being designed that perform more work on each clock cycle[1].

Power can be conserved not only by the design but also by efficient operation. Power management software can power down individual components when they are idle, for example, spinning down the internal disk or turning off screen lighting. Li et al. determined recently that for today's notebook computing it is worthwhile to spin down the internal disk drive after it has been idle for just a few seconds[7]. Applications can conserve power by reducing their appetite for computation, communication and memory, and by performing their periodic operations infrequently to amortize the startup overhead. Since cellular telephone transmission typically requires about ten times as much power as reception, trading talking for more listening can also save power. The potential savings of these techniques can be evaluated using Tables 2 and 3, which show example power budgets for notebook computers. Although screen lighting consumes a large amount of power, it has been found to greatly improve readability, for example, on EO models it enhances contrast from 6:1 to 13:1. Nevertheless, PDA products have elected to omit screen lighting in favor of longer battery life.

4.2 Risks to Data

Making computers portable heightens their risk of physical damage, unauthorized access, loss, and theft. This can lead to breaches of privacy or total loss of data. These risks can be reduced by minimizing the essential data that is kept on board— notably, a mobile device that serves only as a portable terminal is less prone to data loss than a stand-alone computer. This is the approach taken for PARC's Tabs and for Bershad's BNU system[13].

To help prevent unauthorized disclosure of information, data stored on disks and removable memory cards can be encrypted. For this to be effective, users must not

Table 3: Power consumption breakdown by subsystems of a portable computer. These data were obtained from the Compaq LTE 386/s20 manual.

System	% Power
display edge-light	35%
CPU/Memory	31%
hard disk	10%
floppy disk	8%
display	5%
keyboard	1%

leave authenticated sessions (logins) unattended.

To safeguard against data loss, one can keep a copy that does not reside on the portable unit. One solution is to have the user make backup copies, but users often neglect this chore, and data modified between backups is not protected. With the addition of wireless networks to portable computers, newly produced data can be copied immediately to secure, remote media. This can be accomplished with replicated file systems such as Coda and Echo[6].

4.3 Small User Interface

The size constraints on a portable computer require a small user interface. Desktop windowing environments may be sufficient for today's notebook computers, but for smaller, more portable devices current windowing technology is inadequate. On small displays it is impractical to have several windows open at the same time regardless of screen resolution, and it can be difficult to locate windows or icons when stacked atop one another deeply. Also, window title bars and borders either consume significant portions of screen space or become difficult to operate with the pointing device.

Duchamp and Feiner have investigated the use of head-mounted virtual reality displays for portable computers[3]. As the user turns his head, the image displayed in the eye shifts to give the sensation that there is a screen all around. This effectively increases the screen area available for windowing systems. Disadvantages of this approach include the hassle of the head gear, low-resolution (one tenth that of conventional displays), eye fatigue, and the requirement for dim lighting conditions.

Buttons vs. Recognition: The shortage of surface area on a small computer can cause us to trade buttons in favor of recognizing the user's intention from analog input devices: handwriting recognition, gesture recognition, and voice recognition. Although handwriting is about three times slower than typing on average, handwriting recognition allows the keyboard to be eliminated, which reduces size and improves durability. This approach has been adopted by all the PDA products in Table 1.

Handwriting recognition rates for high-end systems are typically 96–98% accurate when trained to a specific user. (Tappert et al. give a thorough survey[11].) With context information, recognition rates can be enhanced effectively to 100%, but context constraints do not help for all kinds of input, such as when entering words that are not in the dictionary. Popular reports indicate that the Apple Newton's handwriting recognition, while among the best of the PDAs, is nevertheless a source of frustration. Finally, recognition of the user's intention in a general setting is inherently hard because the interpretation of pen strokes is ambiguous. For example, by drawing a circle a user may intend to select an object or an area, write a zero, degree sign, or the letter 'o'.

Speech production and recognition seem an ideal user interface for a mobile computer in that they require no surface area and allow hands-free and even eye-free operation. The voice commanded VCR programmer by Voice Powered Technology demonstrates the feasibility of this interface on a hand-held device for a narrow domain. Speaker-independent recognition rates of nearly 96% have been reported for the Sphinx research project; 98% for speaker-trained recognition. However, general purpose speech input and output places substantial storage and processing demands on a mobile device. Also, speech is inappropriate in common situations: it disturbs others in quiet environments, it cannot be recognized clearly in noisy environments, and it can compromise privacy. Finally, speech is ill-suited for skimming data because of its sequential nature.

Pointing Devices: The mouse is the standard pointing device for desktop computers, but does not suit mobile computers. Pens have become the standard input device for PDAs because of their ease of use while mobile, their versatility and their ability to supplant the keyboard.

Switching to pens requires changing both the user interface and the software interface because mice and pens are really quite different[3]. Users with pens can jump to absolute screen positions and enter path information more easily than with mice; it is nearly impossible to write with a mouse. Pen positioning resolution on current tablet computers is several times that of screen resolution, for example, on the EO pen resolution is 0.1mm while screen resolution is 0.23–0.3mm. Parallax between the pen tip and the screen image can mislead pointing; with mice, there is no parallax because the mouse cursor provides feedback in the image plane. Finally, the mouse cursor obscures much less of the screen than is obscured by one's hand when writing with a pen.

4.4 Small Storage Capacity

Storage space on a portable computer is limited by physical size and power requirements. Traditionally, disks provide large amounts of non-volatile storage. To a mobile computer, however, disks are a liability. They consume more power than memory chips, except when off-line, and may not be non-volatile when subject to the indelicate treatment a portable device endures. Hence, none of the PDA products have disks.

Coping with limited storage is not a new problem. Solutions include compressing file systems, accessing remote storage over the network, sharing code libraries, and compressing virtual memory pages[2]. Although today's networked computers have had great success with distributed file systems and remote paging, relying on the network is less appropriate for mobile computers that regularly encounter network disconnections.

A novel approach to reducing program code size is to interpret script languages, instead of executing compiled object codes, which are typically many times the size of the source code. This approach is embodied by General Magic's Telescript and Apple Technology Group's Dylan and NewtonScript. An equally important goal of such languages is to enhance portability by supporting a common programming model across different machines.

5 Conclusion

In this paper we have examined the repercussions of three principal features of mobile computing: wireless communication, mobility, and portability. Wireless communication brings challenging network conditions, making access to remote resources often slow or sometimes temporarily unavailable. Mobility causes greater dynamicism of information. Portability entails limited resources available on board to handle the changeable mobile computing environment. The challenge for mobile computing designers is how to adapt the system designs that have worked well for traditional computing. To date, few prototype mobile computing systems have been built that include the wireless network[3, 15].

6 Acknowledgments

Support for this work was provided in part by the National Science Foundation (Grants CCR-9123308 and CCR-9200832), Tektronix Inc. (a graduate fellowship), the Washington Technology Center, and Digital Equipment Corporation (Systems Research Center and External Research Program). We thank Robert Bedichek, Brian Bershad, Blake Hannaford, Marc Fiuczynski, Brian Pinkerton, and Stefan Savage for helpful pointers and clarifying discussions that significantly improved this paper.

References

[1] Anantha Chandrakasan, Samuel Sheng, and R.W. Brodersen. Design Considerations for a Future Portable Multimedia Terminal. In *Third Generation Wireless Information Networks,* Kluwer Academic Publishers, pages 75–97, 1992.

[2] Fred Douglis. On the Role of Compression in Distributed Systems. In *5th SIGOPS Workshop on Models and Paradigms for Distributed Systems Structuring,* 6 pages, Sept 1992.

[3] Dan Duchamp, Steven K. Feiner, and Gerald Q. Maguire, Jr. Software Technology for Wireless Mobile Computing. *IEEE Network Magazine,* pages 12–18, Nov 1991.

[4] T. Imielinski and B. R. Badrinath. Data Management for Mobile Computing. *SIGMOD Record,* 22(1):34–39, March 1993.

[5] John Ioannisdis, Dan Duchamp, and Gerald Q. Maguire Jr. IP-based Protocols for Mobile Internetworking. In *Proceedings of SIGCOMM '91 Symposium,* pages 235–245, Sept 1991.

[6] James J. Kistler and M. Satyanarayanan. Disconnected Operation in the Coda File System. *ACM Transactions on Computer Systems,* 10(1):3–25, Feb 1992.

[7] Kester Li, Roger Kumpf, Paul Horton, and Thomas Anderson. A Quantitative Analysis of Disk Drive Power Management in Portable Computers. Technical Report, Computer Science Division, University of California at Berkeley, 1993.

[8] Chaoying Ma. On Building Very Large Naming Systems. In *5th SIGOPS Workshop on Models and Paradigms for Distributed Systems Structuring,* 5 pages, Sept 1992.

[9] B. Clifford Neuman. Protection and Security Issues for Future Systems. In *Workshop on Operating Systems of the 90s and Beyond,* Springer-Verlag Lecture Notes in Computer Science #563, pages 184–201, July 1991.

[10] Carl D. Tait and Dan Duchamp. Detection and Exploitation of File Working Sets. In *11th International Conference on Distributed Computing Systems,* pages 2–9, May 1991.

[11] C.C. Tappert, C.Y. Suen, and T. Wakahara. On-line Handwriting Recognition—A Survey. In *9th International Conference on Pattern Recognition,* 2:1123–1132, 1988.

[12] Fumio Teraoka and M. Tokoro. Host migration transparency in IP networks: the VIP approach. *Computer Communication Review,* 23(1):45–65, Jan 1993.

[13] T. Watson and B. N. Bershad. Local Area Mobile Computing on Stock Hardware and Mostly Stock Software. In *USENIX Proceedings of the Mobile and Location-Indepent Computing Symposium,* pages 109–116, Aug 1993.

[14] Mark Weiser. The Computer for the 21st Century. *Scientific American,* 265(3):94–104, Sept 1991.

[15] Mark Weiser. Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM,* 36(7):75–84, July 1993.