Systems Issues in Mobile Computing

Brian Marsh
Fred Douglis
Ramón Cáceres

Matsushita Information Technology Laboratory
182 Nassau St, Third Floor
Princeton, NJ 08542

Technical Report MITL-TR-50-93

February 1993

Abstract

The decreasing size of computer components and the increasing availability of wireless communication technology make possible ubiquitous mobile computing: access from anywhere, at any time, to computer networks and the rich set of services attached to them. Mobile computers provide a powerful interface to services that allow a mobile user to access diverse sources of information, exchange electronic messages, interact with other users in real time, and utilize remote computing resources.

Previous advances in distributed systems provide a base for such applications, but realizing the full potential of mobile computing requires solutions to new problems. Achieving mobility requires trading performance for weight and power. The hardware weighs less, but is less powerful. To achieve high-performance, mobile computers must utilize wireless networks to access the resources of more powerful but less mobile computers. In addition, moving a computer implies the need for reconfiguration at different levels of the system. Crossing physical boundaries such as those between buildings requires network reconfiguration to provide uninterrupted network access. Crossing administrative boundaries such as those between two divisions of a corporation requires application reconfiguration to allow applications to access local services and to cope with security concerns. Realizing the maximum user benefit from mobile computers demands that the system deal with all the above issues. This paper discusses each of these issues in detail, specifically with regard to how they will affect user applications and how they can be addressed by systems research.

1 Introduction

Increasing miniaturization of virtually all system components is making mobile computing a reality. General-purpose computer systems can now be deployed in a $3\frac{1}{2}$-pound package costing under $2000. Storage densities for both memory and disk make it possible to configure a laptop computer with capacities similar to that of the previous generation's desktop workstation. Wireless network interfaces make it possible for mobile computers to be connected to the Internet at all times. The world of "ubiquitous computing," a term coined by Weiser [12], is becoming a reality.

Wireless networking promises to do for portable computers what traditional networks, such as Novell's, have done for desktop personal computers. Networks turn stand-alone personal computers into distributed systems that allow users anywhere on the network to access shared resources. With access to a wireless network, a mobile user can download news or documentation, query a remote database, send or receive electronic mail, or even share a visual display with other users in real time.

Our goal in the Mercury project is to allow mobile computers to provide a ubiquitous, high-performance, integrated computing environment. This requires addressing two aspects of mobility. The first is low-performance hardware. Mobile computers utilize microprocessors and storage systems that trade computing and storage capacity for reduced power consumption, weight and size. Accepting such limitations enables mobility at a cost in performance. A mobile computer can be taken anywhere, but its user will find it delivers performance an order of magnitude less than current workstations. Wireless networks must be used to provide access to computers that have better performance but are less mobile. Users should be able to have both mobility and performance instead of having to trade them off against each other.

The second important aspect is dynamic network access. As they move, computers cross boundaries imposed by physical subnets and administrative domains. Unfortunately, moving a computer violates an important assumption on which current systems are built. Network topologies are no longer static, but instead change frequently. Popular internetworking protocols such as IP use this assumption to implicitly link the address of a host with its location. The destination host address on a packet is used at gateways to route packets between different subnets. A mobile host that crosses the physical boundary between subnets without changing its IP address breaks the linkage between name and location and is not be able to communicate. Similarly, many administrative tools depend on the fact that new hosts will not suddenly appear on the local network. Hosts cannot be configured without at least obtaining an IP address from the local system administrator. Services cannot be obtained without having such an authenticated IP address added to the appropriate access lists, such as an export list for NFS. Worse, a mobile host could snoop on local network traffic and could impersonate a local machine to gain access to local resources.

The rest of this paper examines the consequences of hardware disparity and dynamic network access for mobile applications. Section 2 presents our assumptions regarding mobile applications. The next three sections discuss the systems issues that arise in trying to support these applications. Specifically, Section 3 discusses hardware-related issues, such as

performance degradation and limited electrical power. Section 4 considers the problems of crossing physical boundaries, and Section 5 discusses issues relating to crossing administrative boundaries. Finally, Section 6 summarizes our observations and future research directions.

2 Mobile Applications

There will be at least four main classes of mobile computing applications: information browsing, personal communication, multiperson interaction, and data entry.

Information browsing includes querying traditional databases, retrieving information from electronic books, magazines, and newspapers, and navigating through hypertext and other interactive documents. A user equipped with a mobile computer can browse information databases from any location. The latest news no longer requires a trip to the newstand. Stock prices can arrive instantaneously instead of with the morning paper or from a call to a stock broker, and can trigger actions to alert the mobile user. News services can be customized to provide important and relevant information as soon as it is published, instead of waiting for the evening paper or nightly news to be issued. Travelers can query map archives and restaurant reviews whenever they are lost or hungry instead of hunting for a bookstore. Examples of existing applications in this class are WAIS [7] and Gopher [7], although none has been yet deployed on a mobile platform.

Personal communication includes sending and receiving electronic mail and document facsimile (FAX). A mobile computer allows a user to utilize asynchronous, electronic communication from any location. The mobile user should be able to receive important messages immediately. Similarly, it should be possible for a mobile user to transmit replies without a frantic search for a fax machine. Examples of existing applications in this class include the Gold mailer [4] and the Electronic Receptionist.

Multiperson interaction is an important class of applications, allowing users to interact using distributed white boards and interactive games. Mobile computers make such applications possible by providing a high-performance interface that users can carry. Face-to-face discussions can be augmented with electronic data exchange and markup. Multiple computers provide a "virtual whiteboard": electronic scratch paper that appears on each machine. Data can be retrieved from long-term storage and displayed. Graphs can be plotted on the spur of the moment. What was once verbal communication augmented with hand gestures becomes an opportunity for sharing visual information. Moreover, the entire conversation can be recorded, allowing users to retain ideas that might otherwise be lost because of information overload. An example of this application includes the Xerox Liveboard [12] project.

Data entry is an important application for mobile computers because these computers make it possible for data to be recorded and evaluated on-site. Mobile computers put computing power in the field. An example is the recent use of IBM pen computers in an archaeology dig [13]. The dig involved dozens of simultaneous excavations. Each artifact dug up had to be catalogued by its type, and where and when it was found. A pen computer equipped with a digital camera and a global positioning system would have provided an invaluable aid to this cataloguing process.

All of these mobile applications can benefit from distribution by using the mobile computer as an interface to information anywhere at any time. Information browsing requires access to information services such as Dow Jones, the National Weather Service, and the Michelin Guide. Personal communication requires access to a network to send and receive new messages. Without communication, the mobile user must work in isolation. Multiperson interaction requires group communication between mobile computers. Without it, users must look at each other's screens directly, making information exchange much more difficult. Finally, data entry requires network access for transferring data to non-mobile computers for analysis and stable storage. Without communication, the user must do all analysis and archival on the mobile unit, an underpowered and vulnerable platform.

The most important mobile applications will be distributed applications. They will run on under-powered hardware and will access remote services over slow network interfaces that may change dynamically. In the next two sections, we describe the problems intrinsic in this situation.

3 Hardware Disparities

To be mobile, a computer must be untethered and lightweight. Achieving these goals requires trading away performance and availability. Performance is limited by the need to conserve weight and size. A mobile computer must be easy to carry. Anything larger than a notebook will often be left behind. Using a smaller form factor requires tradeoffs in computing power. Processors and networks are slower, and disks are smaller. Availability is limited because power is provided by heavy, on-board batteries. Conventional hardware such as disks and microprocessors consume large amounts of power. Special devices for mobility, such as wireless network interfaces, consume considerable amounts of power, limiting availability even more. Batteries are good and getting better, but they still limit how long a full-functioned mobile computer can run without being plugged into a wall socket. New techniques must be developed that maximize both performance and availability for mobile computers. The closer mobile computers come to providing the speed and uptime of desktop machines, the more useful they will become.

3.1 Performance Limitations

Mobile computers are less powerful than state-of-the-art engineering workstations. The need to limit weight and size constrain the amount of storage and number of batteries that can be carried. Table 1 compares the configurations of state-of-the-art mobile and non-mobile computers. The mobile machine is an EO 880. The non-mobile machine is a DEC AXP workstation. As the table shows, mobile computers are an order of magnitude less powerful than non-mobile computers in many crucial respects: processing power, memory size, disk size, and network bandwidth.

The disparity between mobile and non-mobile platforms has several implications. First, limitations on the processor and storage systems result in slower applications. For stand-alone

System	Processor	MIPS	Memory	Disk	Bandwidth	Network
EO 880	25 MHz Hobbit	13	12 MB	64 MB	14.4 Kb/s	Cellular
DEC AXP	150 MHz AXP(500)	150	1 GB	15 GB	600 Mb/s	ATM

Table 1: Relative hardware configurations of an EO 880 and a DEC AXP workstation.

applications like data entry, the response time for users will not be affected by this disparity. The EO in Table 1 is already the speed of a SparcStation 1, a state-of-the-art machine merely two years ago. However, computationally demanding applications, such as the Gold Index Engine, will experience increased response time. Other applications, such as handwriting recognition, may be forced to offer poorer service to maintain reasonable response times. The alternative to poor performance is to use wireless networks to build distributed applications whose performance approaches that of applications on a non-mobile workstation.

Second, limitations on network bandwidth affect the performance of applications that do bulk data transfer. An example of such an application is the file system. Consider a system such as AFS [5], which retrieves a whole file as soon as any part of the file is accessed. This strategy reduces load on the file server at the expense of the client, resulting in file systems that scale to more clients than systems like NFS. Unfortunately, mobile computers may not have enough physical storage for whole-file caching. Even if they do, any part of the file that is transferred but not accessed represents wasted bandwidth and power. In addition, file access latency is greatly increased if the file system must wait for the entire file to be transmitted over a slow network link. Techniques for reducing the communication requirements of such tasks will result in improved response times, and hence better performance for all applications that rely on them.

3.2 Availability Limitations

Battery-powered operation allows mobility by eliminating the need for a power cord. Unfortunately, today's battery technologies limit the availability of mobile computers. In order to keep the weight of the machine tolerable, its batteries must be limited in size: for instance, most laptop computers today can operate for 3 to 4 hours without recharging on a battery that weighs slightly over a pound. As a result, limited battery capacities may be the single most frustrating aspect of mobile computing today: a user may be in the middle of an important operation and suddenly find that his computer is about to deactivate itself. Many users are therefore forced to carry a spare battery with them, and even the spare may not last if the user is traveling for a long enough time (for instance, a trip between Japan and California). Because it is difficult to predict how quickly a computer will exhaust its batteries, the mobile user may be forced to stop working until reaching another wall outlet.

Battery-powered operation has two implications for applications and system software. First, good performance for mobile computers will be measured not only by elapsed execution time, but by the amount of power that is consumed. Applications and system software that can extend battery life are preferred. Program optimization has traditionally focused on

throughput and latency, but not on power consumption. The result is programs that perform well but lower availability for the mobile system as a whole. Optimization techniques must be developed that recognize this new tradeoff. For instance, researchers at MITL are beginning to look at database query strategies that optimize for power as well as time [1].

The second implication concerns power management features now available in hardware components for mobile computers. Examples of these features are intelligent power supplies that shut down various parts of the system according to programmable timeouts, report on the charge remaining in the battery, and warn of impending battery shutdown. Other features include programmable devices such as magnetic disks and display screens that accept timeout values after which to spin down the disk or dim the screen. Operating systems must adapt to these mechanisms and must export the proper interface to user programs so these programs can customize their behavior according to power considerations. Many mobile computers support a suspend/resume mode that usually operates transparently to the operating system. An untimely command to suspend the system can leave device operations pending and the file system in an inconsistent state, possibly leading to the destruction of file data. Battery state is currently not exported to applications. Timers do not work when the machine is suspended. When batteries finally do fail, operating systems do not respond cleanly by properly shutting down. All of these issues must be addressed by the operating system.

3.3 Research Directions

There are a number of ways in which system software can potentially help to alleviate the disparity between mobile hardware and desktop machines; in many ways the different techniques interrelate. Ultimately these techniques may improve the usability of mobile computers by improving performance and/or extending battery life. They include:

Delegate tasks to stationary computers. The disparity between the hardware characteristics of mobile computers and those of stationary ones suggests that mobile computers should take advantage of the resources of stationary computers whenever possible. One way is to offload processes onto so-called "compute servers." Another example is paging into memory of another computer rather than onto a local disk [8], which might reduce power consumption if network access is less costly than keeping the disk spinning. The benefit to the mobile user is improved response time because the computationally intensive portions of the application execute on non-mobile hardware at a higher speed.

Trade increased data processing for reduced network bandwidth requirements. Wireless networks can be a performance bottleneck by slowing bulk data transfers. One way around this bottleneck is to process data before and after transmission to increase the information density of the data actually transmitted. Techniques for reducing bandwidth requirements include:

- *on-line compression*, which uses processing to squeeze redundant information out of a stream of data;

- *difference-based updates*, which uses the relationship between data on both sides of the network to transmit only the data needed to transform the receiver's data to the sender's data; and
- *filtering*, which allows an application to reduce the amount of data sent over the wireless network by performing operations on the data on a well-connected host.

These techniques improve application performance by reducing the time spent waiting for network transmissions. On pay-per-use networks, they will also reduce transmission costs. As a by-product, they may also improve battery life by minimizing unnecessary data transfers.

Reduce network latency. Maintaining good application response time requires hiding the network latency inherent in accessing slow wireless networks. Applications that send multiple messages to satisfy a single user request will be particularly hard hit. For instance, information browsing applications such as the Neon project's pen-based database query language [2] may issue multiple database queries in the course of computing a join. Two techniques for addressing network latency include:

- *data prefetching*, which allows data to be transferred before needed over an otherwise idle network interface;
- *extensible interprocess communication (IPC)*, which allows an application to batch together multiple IPC requests into a single wireless transfer for execution on a well-connected machine.

These techniques minimize latency by either overlapping network accesses with computation or by eliminating unnecessary accesses altogether. The ultimate benefit to users of mobile applications will be response times comparable to those of applications with access to much faster networks than are in reality available.

Reduce power consumption. Stationary computers have never had to deal with the issue of power consumption since they have always used current from a wall outlet. For mobile computers, power consumption can overwhelm other aspects of the system. Hardware already provides some support for reduced power consumption, such as shutting off a disk when it has not been used for a length of time. However, only the software can arrange not to use the disk for an extended period. There are several possible techniques:

- *Shift processing* from the mobile machine to a workstation. The mobile computer can idle in a low-power mode while the computation proceeds. Normal power consumption can resume when the operation results are retrieved. This may be slower than performing the operation locally, but can extend battery life significantly, making it an attractive tradeoff for users.
- *Aggressively cache and prefetch data* to reduce disk traffic. Keep files in memory as much as possible, while bringing in a significant amount of data at once once

a cache miss occurs. Once the disk spins up it costs almost as much for it to do nothing as for it to transfer data.

- *Exploit asymmetric network power demands,* by transmitting less data while possibly receiving more. One possible application of this would be to perform file differencing by requesting a copy of the original version of a file, and transmitting the difference between the old and the new. The whole file would be received, but much less data might be transmitted.

Collectively, the approaches described above may improve battery life by 25–50% or more. If the expected battery life can be increased enough to allow typical users to operate their mobile computers without a spare battery on a day-to-day basis, mobile computing will become much more desirable.

4 Crossing Physical Boundaries

Physical boundaries are imposed by either the physical realities of the transmission media. The strength of wireless signals received by a mobile computer decreases as the distance from a transceiver grows; shifting to a closer transceiver is necessary to maintain good signal characteristics. Maintaining communication with an old transceiver may violate local administrative constraints on traffic; mobile computers are then obliged to begin communication with a local transceiver. From the perspective of the mobile computer, crossing the boundary occurs when its network interface shifts between physical subnetworks. Such a shift is an inevitable consequence of computer mobility, but it should be hidden from users. Failure to do so requires a user to manually reconfigure his machine every time he moves. For situations in which movement occurs frequently, such as in micro-cellular networks[1], this burden is unacceptable.

Crossing physical boundaries disrupts two key functions provided by current internetworking software: host tracking and packet routing. The host tracking function determines where the mobile computer is physically located. The packet routing function determines how to get packets to the mobile computer as efficiently as possible. Current internetworks assume that a host always resides on the same physical subnetwork; they deduce the location of a host from a fixed address. Mobile hosts violate this assumption. To allow an application to maintain transport-level connections as its host moves, the internetwork must track the moving computer and route traffic to its current network access point. Several efforts are under way to address these problems (*e.g.*, [6, 10, 11]). Each utilizes a slightly different approach for host tracking and packet routing.

[1] Micro-cellular networks are composed of a large number of cells with diameters of only a few meters. These microcells and nanocells offer three advantages over larger cells: higher aggregate throughput, lower power consumption, and enhanced positional accuracy. Aggregate network throughput can be higher despite low bandwidth within a single cell because the bandwidth of many cells can be simultaneously utilized. Power consumption can be lower because transmissions need only cover the small area occupied by a single cell. Finally, positional accuracy is improved because routing information can reveal to applications useful information about their precise location

4.1 Host Tracking

Host tracking is essentially a directory service. A mobile host that crosses a physical boundary must notify the service. A host of any type that wants to contact the mobile host first queries the service to find out where the host is currently located. Clearly a centralized directory service will be a bottleneck. The challenge is to allow updates from multiple hosts simultaneously, but only to notify those hosts that actually care. The task becomes harder when hosts move about since the areas of interest change.

Recent work has considered how to construct efficient tracking algorithms. Awerbuch and Peleg describe a hierarchical directory service whose communication overhead is polylogarithmic in the size and diameter of the network [3]. Unfortunately, the authors do not consider the issue of fault tolerance. As described, their scheme is prone to single-point failures, a serious possibility in a mobile environment. In addition, while their algorithm is intended to optimize for locality of motion, it does not consider the impact of existent communication patterns. Such temporal locality could significantly reduce the overhead for their scheme. Finally, this work was primarily theoretical. The practical utility of the scheme has yet to be tested in a real application such as one of the mobile IP schemes described above.

4.2 Packet Routing

Packet routing for mobile hosts requires that routes change to reflect the mobile host's current location. Many schemes currently use a packet forwarding server located at a well-known address. These schemes suffer from non-optimal routes and unnecessary network traffic. Routes are non-optimal because they must always go through the server. Besides being a bottleneck, the traffic sent to the server for redirection may actually travel over several of the same network links twice, generating superfluous network traffic.

Many current mobile packet routing schemes also limit the number of hosts they can support per physical subnet. These schemes require that a mobile host have a local class-C Internet address. The advantage to this approach is that it eliminates the need to modify existing Internet routers. Unfortunately, it also bounds the number of mobile hosts per subnetwork by the number of address bits in the local network address space, instead of by the bandwidth available. The functional limit on the number of mobile hosts per subnetwork is imposed by the hardware; the limited bandwidth of wireless networks constrains the number of hosts that may be transmitting simultaneously. This limit should be a lower bound: a mobile host should always be able to communicate as long as there is available bandwidth. Beyond this limit, users will accept degraded performance as long as they eventually get some bandwidth.

Another problem is the susceptibility of current schemes to single-point failures in the area in which the mobile computer is operating. Schemes such as Columbia's [6] and Matsushita's [11] require that packets be tunneled through a server on the mobile computer's "home" subnetwork. As the mobile computer moves further from home, the odds of short- to medium-term disconnection from the home subnetwork increase. Such disconnection may be due to hardware failures, congestion on the intervening network, or even security concerns (as

discussed in the next section). In any case, packet routing services should be fully operational even if only the local network is accessible.

4.3 Research Directions

Systems research must address the problems outlined above. Scalable, fault tolerant algorithms for host tracking and packet routing must be developed for networks where a large percentage of hosts move frequently across physical boundaries. Users can expect increased availability and performance as a result. Data replication, a well-studied topic, can be used as a starting point for addressing these issues. This work can be extended along the following lines:

Exploit locality. Host directory hierarchies should exploit communication locality to limit the number of packets transmitted when a host moves. Location updates should be propagated only to cells in a small neighborhood around a boundary crossing. More distant cells need be aware only of which neighborhood a host is currently in, relying on cells within that neighborhood to do the final stages of routing messages to the host. The user benefit will be networks that scale more effectively because of reduced message traffic and are more fault tolerant because of reduced dependency on global information.

Account for velocity. Data updates should adapt for host velocity. Hosts will move with varying speeds, perhaps crossing multiple nano-cells in a short period of time. The network should not route packets to cells that the host has already left and it should anticipate the next cell a mobile host will enter. The user benefit will be better performance while a host is actually moving.

Exploit weakened consistency semantics. Update traffic can be minimized by allowing certain data to become stale. For infrequently used routes, routing updates that sometimes provide sub-optimal delivery for initial transmission may be acceptable. As long as a packet arrives and the route improves to account for computer motion, stale routes may be acceptable. Likewise, update traffic can be reduced if host location directories can provide slightly inaccurate information. The user benefit in both cases is higher performance and increased availability due to lower update traffic and reduced dependence on global data.

5 Crossing Administrative Boundaries

An administrative domain is a set of resources managed by a particular authority (i.e., system administrator). At the lowest level, these resources include computing platforms and physical networks, but also include IP addresses and host names. At the highest level an administrative domain can include access to any service provider. As cell sizes shrink and platforms become mobile, machines are likely to cross into new administrative domains with increasing frequency. Administrative tasks currently assume a relatively static configuration

of hosts and networks. Mobile hosts change that assumption, requiring a reevaluation of administrative mechanisms to adequately support mobile computing.

A mobile computer crosses a domain boundary when it crosses a physical boundary, each side of which is managed by a different authority. The expectation that hosts will appear suddenly and may not be owned by the local authority changes the nature of administrative tasks such as granting access to a service, maintaining security, advertising available services, and providing accounting. For instance, when a mobile host physically attaches to a new network, it requires a new address that may be used to identify the host within the network. In most environments today, adding a machine to a network typically requires manual action by a human, the system administrator. Such overhead is permissible when network configurations are fairly static, changing only when new machines are purchased or replaced. Mobile hosts will appear and disappear far more frequently than desktop machines have changed in the past. Changing network configurations to handle mobile hosts should be completely automated, both to reduce latency and to free the system administrator for more important tasks. Users benefit from such automation because their computers can be easily moved into new administrative domains.

5.1 Naming

When a mobile unit crosses a domain boundary, it must obtain a name that will allow packets to be routed to its new domain. It must also arrange for packets to be forwarded from its old domain(s). Important entities that are named include machines and users. By machine names we mean the symbolic Domain Name System name and the IP address to which that name maps. By user names we mean the user login and user id to which it resolves.

Name allocation is a key issue. When a mobile computer arrives in a new administrative domain, it must be given a name that allows it to contract for and receive local services. This name must allow local services to be able to communicate with it, authenticate it, and bill its owner for services rendered. Likewise, the user of the mobile computer may receive a local name for use by user-specific services, such as `talk`.[2]

In most environments today, obtaining a local name for a new machine requires manual action by a human, the system administrator. Such overhead is permissible when network configurations are fairly static, changing only when new machines are purchased or replaced. In the future, mobile hosts are likely to appear and disappear far more frequently than desktop machines have changed in the past. Changing network configurations to handle mobile hosts can and should be completely automated, both to reduce latency and to free the system administrator for more important tasks.

Naming users is another issue. One possibility would be to create a temporary identity for a mobile user within the context of the foreign domain. For instance, *marsh@mitl.com* might also be *marsh@cs.princeton.edu* for a period of time. This would allow other users to identify the local instance of user *marsh* without resorting to contacting host *mitl.com*.

[2] `Talk` is a Unix application that provides on-line communication for users. Two people can communicate by typing messages which are simultaneously displayed to both.

However, generating a local name has problems associated with it. The name might conflict with another name within the foreign domain. The mobile host might have other names (or numeric user identifiers) associated with files on its local disk, and these names might conflict with the foreign domain as well. Lastly, when would the user be identified by the temporary name in the foreign domain and when would he be identified by the permanent name in the home domain? Outgoing electronic mail, for instance, would have to be identified by the permanent name, or a reply to it might not reach the user after a short while.

Using the user's permanent name within the foreign domain is a preferable alternative, as long as it can be done efficiently. Other hosts within the foreign domain should not need to contact a server at the user's home domain each time they need to interact with or authenticate the user. Instead, a server within the foreign domain should cache information about the "guest" and respond to queries relating to him (or the mobile host). Additionally, protocols that assume that an entity is locally known must be augmented to allow for a globally unique identifier to be presented. An example of such a protocol is Sun's NFS, which assumes that all machines that access a file system have a common set of user identifiers (essentially, a shared password file).

5.2 Security

Security is an issue because a mobile host that enters a new domain has the ability to intercept and generate packets. Both mobile hosts and the networks they visit must be alert for several problems, including impersonation, denial of service, and tapping.

Impersonation means that an attacker can pretend to be someone he's not. For instance, a mobile unit might attempt to impersonate a machine from the administrative domain being visited. If successful, the mobile unit might be able to access files that would otherwise be exported only to machines actually belonging to the current domain. It could also accrue service charges under the assumed identity that would be billed to an unsuspecting user's account. Likewise, a stationary unit might attempt to impersonate a visiting mobile host for the purpose of stealing resources from other domains that trust that mobile host.

Denial of service means that a visiting host might be able to generate so much load on the local infrastructure that service is denied to local machines. Excessive amounts of network traffic can overload local routers. Large mail messages can cause messages originating from non-mobile local hosts to be rejected. Both problems can be caused by non-mobile hosts as well. However, mobile hosts represent a greater threat because they are not under the control of the local system administrator. Such hosts used to be located on the far side of a network gateway, making access to local resources much easier to control.

Finally, tapping can be done by either local or visiting hosts, simply be accepting any packet broadcast. Physically partitioning mobile hosts from the rest of a local network is unlikely to be practical. It must be possible to make guarantees to mobile and non-mobile users about the security of any data they transmit.

5.3 Accounting

Accounting is an issue because we expect mobile hosts to pay for services, particularly when they move into a new domain. Charges are useful not only for obtaining compensation for services rendered but also to discourage "guests" from using resources needlessly. Pay-per-use services include:

- *Network access* – Typically there would be a charge per packet. Charging users would keep their usage down, leaving the capacity of the wireless networks available for local users, or it would at least compensate for lost capacity. The charge might vary based on expected demand, just as the phone company's rates change based on time and day, or it might vary in response to actual demand.
- *Information retrieval* – Local or remote servers may respond to requests, and in some cases will be compensated. A mobile host may choose to change service providers because a local service is less expensive than a remote service, or vice-versa.
- *Hardware usage* – The administrative domain may choose to provide additional resources, such as printers, processing, disk space, memory, and so on. The system needs to provide a mechanism for these resources to be used on a pay-as-you-go basis.

5.4 Resource Discovery

Resource discovery is an issue because mobile hosts must be able to discover which services are available in new domains (*e.g.*, "Where is the printer and how do I get permission to use it?"). Services in the new domain may provide information not previously available, may be more economical, and may offer higher performance.

New information may become available when a mobile host moves into an administrative domain that does not normally pass packets through to wherever the mobile host was previously. Non-forwarding of packets is used to provide security by denying access to potential intruders. However, such packet filters may make local services inaccessible to any host outside their administrative domain. Mobile computers must be able to discover these services as they become available. Such discovery is especially important because packet filters may make previously used services inaccessible, leaving the mobile computer no alternative but to discover and utilize local service providers.

Resources in the new domain may be more economical than the services previously in use. Mobile users should be able to find the best service they are willing to pay for. Local services may simply have a better per-unit price for the services currently in-use or known to the mobile unit. Local services may also be closer, eliminating any additional charges for long-distance network transmissions. In any case, resource discovery must enable informed consumption of networked resources.

Using a local service may also provide better performance. The local service may utilize a higher performance processor. It will also communicate with the mobile client using a local-area network, which will almost certainly provide lower latency, and perhaps higher

throughput, than the internet route to the original service. Better performance and lower long-distance charges may help to offset a higher per-unit price.

5.5 Research Directions

Mobile hosts should retain network connections when they cross administrative boundaries, regardless of security concerns. Mobile hosts should also be able to find servers within new administrative boundaries while allowing the domain to tightly control the freedom of such access. Services within the domain need to be able to charge for their services, and use of other resources within the domain (such as wireless networks) must be controlled. Finally, it is essential that both mobile hosts and the environments they visit are protected from each other. There are a number of ways that system software can address these problems.

Allow shopping. A mobile host may be willing to pay for better performance and enhanced service, provided those services can be found, and once found, have an acceptable cost. If they are found, existing clients on the mobile unit may choose to switch their service providers. Support for electronic shopping requires:

- *A brokerage service.* Clients must be able discover which services are available and what they cost. They must also be able to select a particular service. A brokerage service would provide this kind of mediation.
- *An accounting service.* Clients must be able to contract for services. Services should be able to ask for prepayment from authenticated accounts. An accounting service would provide an impartial and trusted third party for mediating cost accounting between mobile clients and servers.
- *Rebindable service interfaces.* Once a rendezvous has been agreed upon, the service interface used by existing clients must be switched from the old service provider to the new one.

Use controlled isolation. Total isolation between mobile hosts and stationary hosts prevents access to local services. It may also prevent data sharing between mobile hosts and stationary hosts. Lack of isolation exposes both mobile hosts and stationary hosts to unacceptable security risks. A balance needs to be struck. We see two techniques for achieving this balance:

- *Authorization mechanisms must become more dynamic to cope with mobile hosts.* Authorization mechanisms must be extended beyond simple, relatively static notions such as Unix group and user ids for the purposes of providing service. Capabilities and access control lists provide two extremes to begin with. They must include a globally-unique identifier, consisting of a domain and a user or group within that domain.
- *Authentication and encryption must be ubiquitous.* Computer security is an increasingly well-understood area. Systems such as Kerberos [9] provide authentication

and the means to exchange private keys for encryption. They are not yet widely used, but will have to be in common use for mobile access not to present serious security problems. Furthermore, though encryption is currently left to individual applications, it must be made available at a level appropriate for use by all applications.

6 Conclusions

As mobile computing becomes increasingly commonplace, operating systems for both mobile and stationary computers will have to meet a number of challenges. The first challenge is to provide mobile users with performance and availability that approaches what they experience using desktop computers, even in the face of slow wireless networks, minimal hardware configurations, and battery-powered operation. **Disparity management** enables applications to combine the advantages of mobility with the performance and availability of stationary computers. The intent is to exploit distribution between mobile and non-mobile computers to insure that computation takes place at a site that balances performance and availability concerns. With disparity management facilities in place, the distinction between being "home" and "on the road" will blur, enabling people to make effective use of their computing resources wherever they are.

The second challenge is to provide efficient communication with users and machines as they move, even if they change subnetworks with great frequency or if the connection with their normal service providers should be interrupted. **Physical mobility** requires support for scalable mobile internetworking and the establishment of ubiquitous fault-tolerant services. Until the infrastructure to support large-scale mobile internetworking is in place, mobile computing will be limited to small numbers of computers within a short distance of their "home" environments.

The third challenge is to allow mobile users to take advantage of stationary resources even when those resources belong to another administrative domain. **Consumer computing** is the set of mechanisms that allows mobile users to locate resources as they move, and stationary hosts to bill mobile users for activity. For consumer computing to be acceptable, it must be widely available, while it must minimize the intrusion of a mobile host within the stationary environment. If users can enter a building and gain immediate access to hardware resources such as processing capacity, memory, and disk storage, as well as other functionality such as database retrieval, mobile computing will become more desirable and widespread than in today's standalone environments.

7 Acknowledgements

We'd like to thank Hank Korth, Rosemary Walsh, and Kai Li for their comments on this paper. Their efforts have improved its quality.

References

[1] R. Alonso and S. Ganguly. Energy Efficient Query Optimization. Technical Report MITL-33-92, Matsushita Information Technology Laboratory, November 1992.

[2] R. Alonso, E. Haber, and H. Korth. A Database Interface for Mobile Computers. In *Proceedings of the Globecomm Workshop on Networking of Personal Communications Applications*, Orlando, Florida, Dec 1992.

[3] B. Awerbuch and D. Peleg. Concurrent online tracking of mobile users. In *Proceedings of SIGCOMM*, 1991.

[4] Daniel Barbará, Chris Clifton, Fred Douglis, Hector Garcia-Molina, Stephen Johnson, Ben Kao, Sharad Mehrotra, Jens Tellefsen, and Rosemary Walsh. The Gold mailer. In *9th International Conference on Data Engineering*, Vienna, April 1993. To appear.

[5] John Howard, Michael Kazar, Sherri Menees, David Nichols, Mahadev Satyanarayanan, Robert Sidebotham, and Michael West. Scale and Performance in a Distributed File System. *ACM Transactions of Computer Systems*, 6(1):51–81, February 1988.

[6] J. Ioannidis, D. Duchamp, and G. Maguire, Jr. IP-based Protocols for Mobile Internetworking. In *Proceedings of SIGCOMM '91*, pages 235–245, September 1991. Describes IPIP and IMCP.

[7] E. Kroll. *The Whole Internet: User's Guide & Catalog.* O'Reilly & Associates, 1992.

[8] Bill N. Schilit and Dan Duchamp. Adaptive remote paging for mobile computers. Technical Report CUCS-004-91, Columbia University, New York, NY, February 1991.

[9] J. G. Steiner, B. Clifford Neuman, and J. I. Schiller. Kerberos: An Authentication Service for Open Network Systems. In *Winter 1988 USENIX Conference*, Dallas, TX, 1988. USENIX Association.

[10] R. Teraoka, Y. Yokote, and M. Tokoro. A Network Architecture Providing Host Migration Transparency. In *Proceedings of SIGCOMM '91*, September 1991. Also Sony CSL TR #SCSL-TR-91-004. Describes Sony's approach to mobile internetworking.

[11] H. Wada, T. Yozawa, T. Ohnishi, and Y. Tanaka. Packet Forwarding for Mobile Hosts. In *Proceedings of the Winter UseNIX*, 1993.

[12] Mark Weiser. The computer for the 21st century. *Scientific American*, pages 94–104, September 1991.

[13] tape #4 Worldwide PenPoint Developers Organization. Archaeology and PenPoint. Videotape courtesy of GO Corporation.