Challenges for Nomadic Computing: Mobility Management and Wireless Communications

Thomas F. La Porta, Krishan K. Sabnani, Richard D. Gitlin
Bell Laboratories
101 Crawfords Corner Rd., Room 4G-508
Holmdel NJ, 07733
tlp{kks, rich}@research.att.com

ABSTRACT

In this paper, we present several challenges and innovative approaches to support nomadic computing. The nomadic computing environment is characterized by mobile users that may be connected to the network via wired or wireless means, many of whom will maintain only intermittent connectivity with the network. Furthermore, those accessing the network via wireless links will contend with limitations of the wireless media. We consider three general techniques for addressing these challenges: 1) asymmetric design of applications and protocols, 2) the use of network-based proxies which perform complex functions on behalf of mobile users, and 3) the use of pre-fetching and caching of critical data. We examine how these techniques have been applied to several systems, and present results in an attempt to quantify their relative effectiveness.

Challenges for Nomadic Computing: Mobility Management and Wireless Communications

Thomas F. La Porta
Krishan K. Sabnani
Richard D. Gitlin

1 Introduction

Following the explosive growth of cellular telecommunication and paging services, there is an increased interest in anywhere, anytime computing. Often called *nomadic computing*, the goal is to provide users with access to popular desktop applications, applications specially suited for mobile users, and basic communication services in a mobile, sometimes wireless, environment. Nomadic computing is enabled by the advancement of portable computing devices, such as laptop computers and Personal Digital Assistants (PDAs). There are many ways in which nomadic computing networks can be realized. There are also many challenges to providing services with reasonable performance and with acceptable reliability in a wireless mobile environment. The ultimate goal is to provide services in both the local and wide area that provide performance achievable in a non-mobile environment.

One approach to realization is to use existing cellular telecommunication networks [1][2][3][4] to provide wireless access to wired computer networks. The advantage of this approach is the wide coverage area already available with cellular networks. Also, these networks provide support for mobile users. New standards defined for cellular telecommunication networks support messaging applications and provide extensions for supporting data services.

A second possibility is to directly extend wide-area packet networks with wireless access. Many wireless local area networks are already available. Recently, a connectionless packet service called Cellular Digital Packet Data (CDPD) [5] has become available as an overlay service on existing cellular networks. CDPD uses many of the techniques defined in the Internet Engineering Task Force (IETF) draft standard to support mobile users communicating using the Internet Protocol (IP) [6]. As advances are made in wired networks, such as the introduction of Asynchronous Transfer Mode (ATM) technology to support quality of service dependent applications [7], they may be extended to wireless networks as well [8].

The nomadic computing environment differs from a fixed computing environment in many important ways. First, network access for many mobile users will be made over a wireless link as opposed to a wired link. Therefore, these users will generally have access to lower bandwidth and experience higher error rates than wired users. Second, the wireless link may be viewed as unreliable in terms of availability. Due to

mobility, users may often not be within the coverage area of a network, thus making the network unavailable to these users. Therefore, networks, applications, and operating systems must support users that intermittently disconnect from a network, either by design, or because of limited network coverage. Third, nomadic users, whether accessing the network by either wired or wireless means, will no longer be stationary. The mobility of users presents challenges for routing, and opportunities for new location-sensitive applications. Finally, mobile devices will likely have limited processing and power capabilities compared to desktop computers.

One of the key concerns with nomadic computing is network and application performance. Performance parameters of interest include reasonable throughput, response time, and latency in the presence of low-bandwidth, high error rate, wireless links, and fast connection establishment and packet delivery in the presence of mobility. These are the critical performance criteria for information retrieval and interactive communication applications. A second concern with nomadic computing is the relative scarcity of bandwidth on wireless networks. For this reason, applications and nomadic computing systems attempt to limit the aggregate bandwidth used on wireless links.

In this paper, we discuss three general techniques that have been applied in a variety of systems designed to support nomadic computing: asymmetric design of protocols and applications to overcome the limitations of mobile devices; the use of network-based proxies that perform computing and communication functions on behalf of simple mobile devices; and the use of intelligent caching and pre-fetching of data to improve performance and availability.

The remainder of this paper is organized as follows: we present the nomadic computing environment in Section 2; in Section 3 we provide an overview of techniques used to overcome challenges presented by the nomadic computing environment; in Section 4 we present several systems that use these techniques, and quantify some of the gains made through their use. Systems and protocols presented include a link-layer protocol called AIRMAIL [9], a transport layer protocol called SNOOP [10], cellular telecommunication standards [3], mobile IP [6], the Wireless Distributed Call Processing Architecture (WDCPA) [11], ParcTab [12], InfoNet [13], the Wireless World Wide Web (W4) [14], and the CODA file system [15]. In Section 5, we conclude.

2 Nomadic Computing Environment

In this section, we discuss the various environments in which a nomadic user may operate, shown in Figure 1, such as an office, office complex, conference room, home, hotel room, or automobile. Such users have

to contend with variable bandwidths ranging from several megabits/second to a fraction of kilobits/second, different link characteristics ranging from practically loss-less links to lossy links with frequent disconnects, and end-devices with varying displays and processing power. These users may communicate through a wired network connection, or via wireless access.

In either a wired or wireless environment, nomadic users will be mobile, changing their points of attachment to the network over time. Therefore, the address of a nomadic user does not indicate their location. As a result, the underlying infrastructure needs a procedure for transparently tracking and locating users in close to real-time. There are two ways of locating a user: (a) keeping track of the user's location through frequent updates in a database; or (b), searching through some part of the network to locate the user. A practical solution is typically a hybrid. In a wired environment, this may require a mobile device to generate a location update when it detects that it has been connected to a network. For wireless devices, a combination of location updates and paging procedures are used. For example, cellular telecommunication networks use a two-tiered hierarchy of location databases along with paging procedures to track and locate mobile users.

In the following subsection, we discuss wired nomadic computing environments. In the subsequent subsection, we discuss the environment of wireless nomadic computing.

2.1 Wired Nomadic Computing Environment

In a wired office environment, normal network connectivity is through a Local Area Network (LAN), such as an Ethernet (~several megabits/second). End-devices are typically powerful workstations. Local communication is inexpensive. Local file servers and compute servers allow a large range of high-performance applications. In this environment, users have access to powerful local processing and high quality displays. Networks provide not only high bandwidth communication, but low error rates and highly available service. The primary concern with nomadic users operating in this environment is detecting their attachment to and detachment from the network.

At home, end devices are usually relatively high-powered Personal Computers (PCs). Network connections are typically made through a telephone line. Telephone connections, especially long distance connections, are expensive. Connection bandwidth is on the order of tens of kilobits/second. Therefore, these users are limited more by communication capabilities than computing and storage capabilities. As a result, programs and data, which are loaded in real-time based on demand from file servers in an office environment, are often stored locally on PCs so that large amounts of information do not have to be transferred over the relatively low bandwidth, expensive telephone connections. An example of a system, called CODA [15], that

automatically pre-loads data onto PCs for local use is presented in detail in Section 4. Because the storage on a PC is limited with respect to distributed file systems which use large file servers, information is selectively pre-loaded.

A user in a hotel room will typically use a laptop computer as end device and a telephone line as communication line. This environment accentuates the limitations of the home environment: local processing power and storage space will be less abundant than in the home or office environment. In this or the home environment, end devices will often be used while disconnected from the network.

2.2 Wireless Nomadic Computing Environment

Nomadic users frequently will be connected to a wired network by a wireless access link. These users may operate in an indoor local environment, or in a wide area outdoor environment. The bandwidth of wireless networks is typically much less than that of the wired networks. When radio signals are sent from a transmitter to a receiver, they can take multiple paths because of reflections. At certain points, these signals can merge to enhance the signal. At other points, these signals interfere with each other and can cause a *multipath fade*. This fade causes the transmitted signal to be spread in time at the receiver. The quantification of this spreading is called the *delay spread*. The delay spread limits the channel rate of the wireless link, assuming that other techniques such as equalization and space diversity are not used. In cellular networks, the delay spread is about 10 microseconds. These channels can support data rates of 64 Kbps with equalization [16]. In networks with small cells, such as indoor LANs, rates can be several megabits/second since the delay spread is only tens of nanoseconds.

Within a building, where a user can walk around with a terminal, the connectivity can be through a wireless LAN. For example, WaveLAN [17] has a shared channel operating at 2 Mbps. In this environment, the end-device and its display will be small. It may be a laptop computer, or a less functional PDA or palm-top device. In these cases, the device may not have a keyboard and instead use a stylus for input. These devices also have limited memory and processing power. In untethered mode, battery power is also a limitation. Therefore, applications and protocols must be designed to operate at a lower bandwidth, remain useful even with limited I/O capabilities of end devices, and distribute processing so that battery power is conserved at the mobile device.

In a wide area outdoor environment, a user can communicate through an outdoor packet service such as CDPD, RAM Mobile Data, or Advanced Radio Data Information Service (ARDIS), or over a circuit-switched cellular telecommunication network. Data rates in such networks are, at most, a few kilobits per

second shared among a large number of users. Service from such networks is typically expensive. Such wireless services may have high error rates and frequent disconnects. In this environment, the limitations posed by local area wireless networks are multiplied. The end-device for such a user is typically small and may require stylus input or voice recognition functions. Applications cannot rely on the wireless network to provide high throughput or fast response times. Therefore, they must be tuned to limit the amount of communication, and perhaps even limit the type of information exchanged with the end devices. The challenge is to maintain a high perceived end-to-end performance without limiting applications to the point where they are no longer useful.

When a wireless mobile unit moves from one cell to another cell[1], an elaborate procedure called a handoff is required. Databases with the location information may have to be updated and new channels or frequencies may have to be allocated. In packet networks, packets in transit and new packets have to be routed to the new cell. During a handoff, some packets may be delayed or dropped. In circuit-switched networks, connections must be re-routed. Again, some information may be lost during this process. Degradation during handoffs affects both applications and protocols used to transfer information. For example, the Transmission Control Protocol (TCP) views such losses as a congestion condition and may trigger recovery procedures which reduce throughput during the transition period [18].

Because of the limitation of wide area wireless connectivity, data users may choose to operate in a disconnected mode much of the time for certain applications, connecting to the network only periodically. Applications must account for such operation and transparently continue to provide useful services with intermittent network connectivity.

A desirable characteristic of a wireless network is the convenient availability of a broadcast channel. A broadcast channel can be used to easily distribute system information or support broadcast applications. An example of such information is traffic conditions on local roads for a location-based service.

In summary, the nomadic computing environment includes both wired and wireless network connectivity. The mobility of users is common to both environments, and affects addressing and routing. In addition, wireless users will require support for real-time mobility, that is changes in points of network attachment during active communication. Bandwidth limitations progressively become more severe as we consider wireless network connectivity. In the worst case, outdoor wireless network connectivity limits users to a few kilobits per second of shared bandwidth. Similarly, the reliability of service and channel error rates worsen when we consider users operating in an outdoor wireless environment. Because the same user, using the same applica-

1. A cell is the area covered by a single wireless transceiver.

tions, will likely operate in a many of these environments, nomadic computing applications and networks must account for the characteristics of these environments, and adapt to users that do not have consistent needs or resources available.

3 General Solution Techniques

In the previous section, we discussed the nomadic computing environment. In this section we discuss three general techniques for addressing the major characteristics of this environment: the use of network-based proxies, judicious acquisition and caching of information, and asymmetric design of protocols and applications. We discuss the general techniques and applicability here. In Section 4, we discuss specific systems which use these techniques and quantify some of the gains achieved through their use.

3.1 Network-based Proxies

Many systems which support wireless mobile users make use of intelligent agents that reside inside the wired network and perform various functions on behalf of the mobile users [11][12][13][14]. Evolving intelligent cellular telecommunication networks [3] use intelligent switches and databases which store user profiles to perform functions on behalf of mobile users. The intelligent agents, sometimes called proxies, can be used to process control information, or to manipulate user information that is being exchanged between the mobile device and a network-based server. A proxy can be executed in a fixed location, or may be mobile, and move as the user that it serves moves. The general benefits of a network-based proxy are:

1. proxies may execute complex functions, relieving processing-limited mobile devices;
2. proxies may be used to reduce the amount of communication required with the mobile device thus reducing the amount of air interface bandwidth consumed;
3. proxies may account for mobile devices that are in a disconnected state;
4. proxies may shield network-based applications from the mobility of their clients; and
5. proxies may shield applications from the heterogeneity of mobile devices.

One complex function that may be performed by an end device is the format translation of information sent for display. For example, a wireless web browser may provide a document in PostScript format while the end device can only display ASCII text. To have the end device perform this conversion would require a significant amount of storage and processing. Instead, a proxy may perform this conversion, filter out any graphics, and forward only the ASCII text to the end device for display [14]. This approach has two main benefits. First, the network-based application does not need to be modified, as would be the case if the problem were

attacked at the server application; and second, the amount of processing on the end device is greatly reduced.

Two communication intensive control functions typically performed at end devices are negotiation before end-to-end communication is established, and allocating resources in the end device. For example, signaling standards for Broadband Integrated Services Digital Networks (B-ISDN) [7] define elaborate negotiation procedures between end devices and the network to account for application compatibility and communication capabilities before communication is established. This negotiation is required because advanced multimedia applications are being introduced into an environment of heterogeneous end equipment. In addition to negotiating application and communication capabilities, resource allocation also involves endpoints. For example, a device may be able to process a single audio stream at any particular time, and although it may have spare bandwidth available, it may reject a second audio connection. Negotiation and resource allocation may require several messages to be exchanged between an end device and the network.

A proxy may also be used as a convenient way for a mobile user to request a default connection, such as a voice connection to their home, or a data connection to their office. By introducing proxies to perform these functions [11][12][13], the processing and communication with wireless end devices is reduced.

In addition to performing control functions of negotiation and resource allocation, proxies may be used to filter or modify application information being sent to a wireless end device. For example, a wireless web browser may not have the bandwidth available to receive images embedded in a particular page, but instead would be better served by receiving only text information [14]. By filtering at a proxy, the amount of bandwidth used by the application on the air interface is greatly reduced.

Network-based proxies, if always active, may also be used to account for mobile users that have either been powered off, moved out of range of the wireless network, or simply choose to operate in a disconnected mode. For example, a short messaging service supported by the newly defined cellular mobility management protocol [3] allows messages sent to users that cannot be located to be stored for later retrieval. This service is provided by a combination of intelligence in a network-based messaging center and a location database that is used to track the location and state of the messaging subscribers. The use of proxies in this way overcomes problems associated with limited network coverage to support reliable communication.

Several systems use network-based proxies to hide mobility from network-based applications [12][13]. When used in this fashion, proxies must process all information being sent to a mobile device, both control and application. The advantage of this approach is that the end application written for fixed users may be reused directly for mobile users. The disadvantage is that because a proxy must receive and process all information, performance degradation may occur.

Proxies are also used to hide heterogeneity of end devices. In these cases, instead of performing negotiation functions between end devices and applications, the proxies perform conversion functions as previously discussed. Again, the advantage of this approach is that end applications written for fixed systems can be reused; the price paid is potential performance degradation.

3.2 Judicious Acquisition and Caching of Information

Pre-fetching and caching of data has been used in many applications to improve performance. In the nomadic computing environment, these techniques are used by many systems [3][20][11][6][15][19] to:

1. limit communication caused by mobility; and
2. improve performance and availability of services.

Location information is cached in almost all systems supporting mobile users. This limits the amount of control traffic required to locate a mobile device. This is true in classic telecommunication cellular networks [3], mobile packet networks [6][5], research efforts on mobility management for cellular networks [20], and efforts targeted specifically at supporting personal communication services and nomadic computing [11][12][13]. Examples are given in Section 4.

Besides reducing the amount of control traffic required to locate a mobile device, location information is cached to improve performance. Accurate caching of location information decreases the time taken to locate a mobile device, and hence either establish a connection or deliver a packet. An important consideration for systems that cache location information is the frequency with which this information is acquired. Frequent acquisition leads to fast location times, but eliminates the benefits for reducing control traffic because of the frequent updates. On the other hand, if updates are not frequent enough, and data cached is stale, performance may actually degrade, as is discussed in Section 4.

Location information is not the only information that is beneficial to cache. Applications in a wireless environment may also benefit from techniques of pre-fetching and caching from both a performance and availability standpoint. Two systems that will be discussed in Section 4 make use of types of pre-fetching to improve the availability and performance of a file system application [15][19] and a browsing application [14]. These systems address the problem of limited bandwidth by performing communication functions in the background while mobile users are not using the resources for real-time actions.

3.3 Asymmetric Protocols and Applications

The asymmetric design of protocols and applications helps overcome inherent imbalances of:

1. processing power between the mobile wireless end devices and network-based processors, and

2. uplink and downlink bandwidth due to transmission power available from wireless mobile devices.

As discussed in the earlier subsections, network-based proxies may be used to perform complex functions such as format translation, thus helping alleviate imbalances in processing capabilities. However, for applications and protocols developed specifically for wireless mobile devices, solutions inherent in their design may be more efficient.

For example, lower layer protocols can be designed to place higher processing and memory requirements on fixed servers that are likely to have no serious power or memory constraints. One example discussed in Section 4 is a link-layer protocol called AIRMAIL [9] which off-loads most of the processing associated with error control into processors placed inside network base stations. Similar techniques can also be applied to transport protocols as well.

Two approaches may be taken to address asymmetry at the application level. The first, which has the benefit of not having to change existing applications, uses proxies inside the network to filter and convert information destined for a wireless end device as discussed previously. This makes use of a powerful network-based processor to overcome the asymmetric distribution of processing resources in nomadic computing systems. In the second approach, applications that operate in a mobile wireless environment take this processing imbalance into account. The first approach is necessary to support existing applications. Its disadvantage is largely in performance and efficiency. If an application on an end device can only display ASCII text, it is inefficient for a proxy to retrieve images and audio associated with the text, only to filter it before transmitting the information to the mobile device.

The second approach will likely become popular as new applications are developed for the mobile environment. These applications will not only account for the asymmetric nature of mobile wireless communication, perhaps by providing users latitude in defining a device type or by negotiating with end devices, but also integrate location information to make the applications location sensitive. Examples can already be seen in ubiquitous computing [12] and web browsing [21].

4 System Examples

In the following subsections we describe how specific systems incorporate the solution techniques presented in Section 3. We examine the application of these techniques to error control, routing, and applications in a nomadic computing environment.

4.1 Error Control

As described in Section 2, wireless links typically experience error rates much higher than their wired

counterparts. Errors may be caused by channel characteristics, such as various fading effects, or by mobility, for example, a mobile user moving out of range of a base station or because of data being lost during handoff procedures. It has been widely reported that errors incurred on wireless links can have a dramatic impact on the end-to-end performance of data applications that require reliable data transport. This is due, in large part, to the reaction of TCP to errors [18]. Because the occurrence of errors has a large impact on the performance experienced by end users of a system, much effort has been applied to overcoming errors in a wireless environment. Below we examine how asymmetric protocol design has been applied at the wireless link layer and how a type of network-based proxy is applied at the transport layer to address this problem.

4.1.1 Asymmetry and AIRMAIL

AIRMAIL [9] is a link layer protocol designed to operate over the wireless link, as shown in Figure 2. AIRMAIL incorporates two key ideas: provide reliable data delivery over the wireless link so that end-to-end error rates as experienced by transport protocols such as TCP are kept low, and support mobility so that reliable delivery is ensured without a large performance degradation during handoffs. The reliable data delivery is provided through a combination of automatic repeat request (ARQ) techniques and forward error correction. The forward error correction technique employed adapts based on the raw error rate being experienced on the channel. Mobility is supported by moving the link state and context information from the old base station to a new base station as a mobile device is handed off. Details can be found in [9]. For the purposes of this discussion we focus on the asymmetric design of the ARQ procedures of AIRMAIL.

As discussed in Section 2, mobile devices will have limited memory, and due to power constraints, limited processing capabilities. For these reasons, AIRMAIL was designed to place the majority of processing complexity at the protocol entity based inside the wired network at the base station. To do this, two guidelines were followed: place no timers, a known bottleneck in protocol processing, on the mobile device, and reduce the processing required due to acknowledgments at the mobile device.

The base station maintains retransmission timers when it transmits packets to the mobile device. The mobile device generates block acknowledgments unless specifically requested by the base station. In this way, no timer is required at the mobile device, and the number of acknowledgments generated is limited. When the mobile device transmits packets to the base station, it maintains a table in which it stores the time at which each packet was transmitted[2]. The base station periodically transmits the status of its receiver to the mobile device. When a status message is received by the mobile device, it checks the received time of the sta-

2. The AIRMAIL transmitter at the mobile device must have access to a local clock, but does not set or clear any timers.

tus message against the transmission times stored in the packet records, and determines if any packet has not received an acknowledgment in more than one round-trip delay. If so, these packets are retransmitted. In this way, the mobile device maintains no timers, and receives acknowledgments periodically instead of after each packet transmission.

The compiled AIRMAIL software at the base station is 150 Kbytes, compared to 100 Kbytes at the mobile device. The processing time taken on the base station to transmit 200 Kbytes of data is 0.7 seconds, compared to 0.23 seconds on the mobile device. Therefore, the asymmetric design of AIRMAIL results in a 2/3 code reduction at the mobile device, and 1/3 reduction in processing time.

4.1.2 Proxies: SNOOP for TCP

Proxies can also be applied to overcome errors on the wireless links. For example, the SNOOP protocol [10] places a proxy in a base station for every TCP connection that passes through it, as shown in Figure 3. The SNOOP protocol acts as a proxy for the user on which the TCP connection terminates. SNOOP monitors every TCP segment sent to and from its mobile host. SNOOP shields TCP in the fixed host from experiencing the effects of lost data on the wireless link. This is important, because as reported in [18], due to the assumption by TCP that errors are the result of network congestion, a severe loss in throughput and increase in response time can be experienced by applications running over TCP in lossy wireless networks.

As TCP segments are sent from the fixed host to the wireless host, they are routed through the base station. SNOOP stores each segment that passes through the base station in a cache. SNOOP also monitors the TCP acknowledgments sent by the mobile device. When an acknowledgment is seen by SNOOP, it removes all acknowledged segments from the cache, and forwards the acknowledgment to the fixed host. If SNOOP detects a duplicate acknowledgment, it realizes that a TCP segment has been lost, and retransmits the segment from its cache. It prevents this duplicate acknowledgment from reaching the fixed host, thus preventing the fixed host from invoking unnecessary procedures to alleviate congestion. In this way, SNOOP hides the errors experienced on the wireless downlink from the fixed host TCP.

SNOOP also monitors TCP segments being sent from the mobile device to the fixed host. If it detects that segments have been lost on the wireless link, it generates negative acknowledgments to the mobile host so that error recovery can begin. In this way, performance is improved when errored conditions are experienced on the wireless uplink. SNOOP also defines procedures for managing handoffs.

Experiments with controlled error conditions show that the SNOOP protocol provides better receiver throughput at the mobile host than traditional TCP under bit error rates of higher than $5x10^{-7}$. For bit error

rates of $2x10^{-6}$, SNOOP provides approximately a 67% increase in throughput. For lower error rates, the throughput of SNOOP is comparable to TCP.

This example illustrates how proxies can be effectively used at a protocol layer other than the application to improve performance on a wireless network. The cost of this approach is that base stations must execute an instance of SNOOP for each TCP connection that they support, and have enough memory to cache all of the outstanding TCP segments.

4.2 Routing

Routing in a mobile environment takes on new complexity because end device addresses which traditionally indicate the identity and location of a device can no longer be interpreted in this way. We examine how caching techniques and network-based proxies are used to assist in routing and improve the performance in networks that support mobility.

4.2.1 Caching and Routing: Telecommunication, PCS, and Packet Networks

Caching of location information is used to varying degrees in telecommunication, PCS and packet networks. We first present examples of telecommunication networks, followed by a research prototype of a PCS network, followed by mobile IP used in mobile packet networks.

Cellular telecommunication networks provide connection-oriented service. In these networks [3], a mobile end device must be located before a connection may be routed to it. Therefore, mobile location procedures must occur once for each connection (often termed a call) being established. To do this, cellular telecommunication networks maintain a two-tier location database structure as shown in Figure 4a. Pointers are maintained in various locations in the network, as indicated by the dashed arrows in the figure. A database that is associated with the service area in which a mobile device is currently located, tracks the location of the mobile device to the level of the switch which is currently serving the device. This database is called the Visitors' Location Register (VLR) and is assigned to a mobile device based on its location. A second database, called the Home Location Register (HLR), is assigned to each mobile device based on the logical address of the device. The HLR maintains a pointer to the VLR currently serving the mobile device.

When a connection is being established to a mobile device, as shown in Figure 4b, the connection is routed to the home switch of the called mobile device. The home switch is assigned based on the address of the device. The home switch queries the HLR, which in turn queries the VLR, to determine the routing address for the mobile device. The VLR queries the last switch which reported serving the mobile device. This switch pages its base stations to determine the exact location of the device. This address is returned to

the home switch which can then route the connection directly to the mobile device. Once a connection is established, handoffs occur as a mobile device moves between base stations and switch areas.

Registration procedures are defined for assigning new VLRs as a mobile device moves and for updating the HLR associated with the device. At one extreme, these databases may be updated frequently, perhaps each time a base station area is traversed. This approach eliminates the need for paging, and decreases connection establishment time at the cost of increasing the amount of signaling traffic due to registration. A second alternative is for the mobile device to generate registrations infrequently, perhaps periodically. This approach increases the time taken to locate a mobile device during call delivery, but reduces the amount of registration control traffic. This illustrates the trade-off between the frequency and granularity at which the cached location data is acquired, control traffic load, and performance. In current cellular telecommunication networks, these location updates typically occur each time a mobile device moves to an area which is covered by a different switch. As a mobile device moves among base stations served by the same switch, the location information is not updated.

A disadvantage of the current cellular approach is that the signaling load due to registration and mobile location procedures can become prohibitive [22]. One attempt to reduce the signaling load due to mobile location procedures introduces caching of location information into switches from which calls are originated [20], as shown in Figure 5. When a mobile device is called from a switch, the identity of the VLR serving the mobile device is cached in the originating switch. If the same mobile device is called from the same originating switch, the VLR identification is cached locally, and the HLR query can be forgone. In this case, both the signaling load and connection establishment time are *reduced*. If no cache entry exists for the called mobile device, the current cellular location procedures are invoked. If a cache entry is found in the switch, but the data is stale, the VLR query will be sent to the wrong VLR, followed by the normal cellular location procedures being invoked. This results in an extra database query (to the wrong VLR), thus causing both the signaling load and connection establishment time to *increase*.

Under a specific set of assumptions, analysis shows that if the local call to mobility ratio (LCMR), defined as the ratio of calls made from an originating switch to the same mobile device before the mobile device changes switch areas, is greater than 5, both signaling load and call establishment time is lowered. For lower values of the LCMR, call establishment time rises. Several strategies for deciding on whether to cache the VLR identity based on measured user behavior have been proposed. These results illustrate the sensitivity of performance on the accuracy of the cached data.

One effort to significantly modify the current cellular telecommunication structure to support *personal*

communication services, while still using the concept of databases residing in the network, is the Wireless Distributed Call Processing Architecture (WDCPA) [11]. WDCPA has been prototyped at Bell Laboratories. WDCPA distributes call processing functions from switching entities, and for reasons related to this, can reduce the hierarchy of location database to a single level for the majority of mobile devices. Also, WDCPA efficiently supports multi-connection calls with a single location procedure. By reducing the two-tiered database hierarchy present in cellular telecommunication networks to a single level of hierarchy, only a single database update is required when a mobile device registers or moves, and only a single query is required to locate a mobile device. As a result, under a specific set of assumptions related to network size and density of users, WDCPA reduces signaling for mobility management by 25% - 40% over standard cellular telecommunication procedures [23], depending on the call configuration. As more calls contain multiple connections, as is expected with multi-connection nomadic computing applications and multimedia services, the reduction in signaling load experienced by WDCPA increases.

Mobile packet networks, in particular, those based on Mobile IP [6], face a similar, yet slightly different problem. In IP networks, each packet is routed independently. If the database technique used in cellular telecommunication networks is applied directly, a router would have to query a location database each time it routed a packet. Obviously, this would create an unacceptable amount of control traffic and load on the location databases. Instead, location information is cached at routers. In this way, when a router receives a packet destined to a mobile device, it can route the packet based on its locally cached information. The location information may be cached in various locations, depending on the options of Mobile IP enabled.

The structure of a Mobile IP network is shown in Figure 6a. Each mobile device is assigned a home agent. As a mobile host moves, it registers with its home agent via a foreign agent. In the most basic version of Mobile IP, the location of a mobile device is cached in its home agent and a *tunnel* is established between the home agent and the serving foreign agent. The home agent receives all packets destined to the mobile device, and based on the location information cached, tunnels the packets to the foreign agent, as shown in Figure 6a. Tunneling a packet consists of encapsulating the packet received at the home agent inside a second packet that is addressed to the foreign agent, and sending it to the network. The network delivers this packet to the foreign agent, which de-capsulates and recovers the original packet so it can be forwarded to the proper mobile host.

This approach has the drawback of requiring all packets destined to a mobile device to be routed to the home agent before being forwarded to the mobile device, possibly leading to inefficient routing. If the route optimization extension of Mobile IP [24] is used to overcome this problem, packets may be routed directly

from the source, possibly a fixed host, to the mobile device. In this case, the location information is cached in two additional locations as shown in Figure 6b.

When using route optimization, the first packet sent by the fixed host is routed to the home agent of the mobile host as in the basic mobile IP scheme. The home agent forwards the packet, and also send a message to the fixed host informing it of the location of the mobile device. From this point on, the fixed host has the location of the mobile host cached and can route packets directly.

Another extension to mobile IP route optimization is designed to help reduce packet delay and loss during handoffs. When a mobile host moves, it changes its foreign agent. The new foreign agent updates the old foreign agent with its identity. This information is cached by the old foreign agent so that it may forward any packets it receives for the mobile device to the proper serving foreign agent. In this way, any packets that were enroute to the old foreign agent when the mobile device moves will not be lost.

Much of the detail in the Mobile IP specification deals with the amount of time that location information is considered valid, and how frequently it must be updated. Unlike cellular telecommunication networks, mobile IP does not require that foreign agents be explicitly told when a mobile device has moved out of its area. Likewise, if many hosts are directly routing packets to the mobile host using the route optimization scheme, it may not be feasible to update each host when the mobile device moves. Therefore, procedures have been defined for cache information to be updated both periodically and based on certain events, such as mis-routed packets being detected.

In both the cellular telecommunication and mobile IP networks caching of location information is used to reduce the control signaling load to locate mobile devices, and to improve performance. In the cellular telecommunication networks, this involves lowering connection establishment time. In mobile IP networks, the caching in the route optimization scheme provides more efficient routes and more robust handoff procedures.

4.2.2 Network-based Proxies and Routing: Telecommunication, Packet, and Research Networks

Network-based proxies are also used to assist in routing in mobile networks. The proxies are used to either assist in routing, or to completely hide mobility from applications and servers designed to work with stationary hosts.

In cellular telecommunication networks, the home switches shown in Figure 4 act as a type of proxy; they provide a fixed location to which calls are routed and from which connections are forwarded to the proper destination. The same is true of mobile IP home agents. In both of these solutions, in order to achieve more efficient routes, and hence better performance, some intelligence is required on originating entities. For the

telecommunication networks, if originating switches have the intelligence to start mobile location procedures, they can query HLRs directly and then route connections directly to the mobile device. In mobile IP, using route optimization extensions, the source host must learn and cache the location of the destination mobile host to achieve optimal routing. Neither of these approaches hide mobility from the end system, but rather rely on the end systems to improve performance and network efficiency.

A different approach is taken in two nomadic computing research efforts. In both the ParcTab [12] effort at Xerox Parc and the InfoNet effort at U.C. Berkeley [13], each mobile device is assigned an agent that resides at a fixed location inside the network. These agents are responsible for linking applications to the mobile device. In both of these efforts, the mobile devices are simple units designed primarily for display purposes. All data sent to the mobile device is routed to the agent and then forwarded. Besides allowing the agent to manipulate the data, this hides the mobility of the end devices from the network-based servers and applications. This is shown generically in Figure 7.

This approach is well-suited for a local area nomadic computing environment. Because the ParcTab and InfoPad are not general computing devices, most users will have a desktop computer as well as the mobile device. Therefore, the desktop computer of a user can act as its proxy. Because users are always in the proximity of their office in a local area environment, the fact that all data passes through a fixed point is not a major concern from a routing standpoint. Handoff procedures are also simplified, because a handoff affects only the route between the proxy and the mobile device. However, the fact that all data must enter a proxy and then be forwarded, does cause performance degradation.

Performance measurements have shown that the performance bottlenecks in InfoNet are the read and write operations in the wired-to-wireless gateways and mobile proxy. To solve this performance problem, in the InfoNet project, the concept of *proxy connections* are being examined. A proxy connection extends directly from a source to the mobile device, eliminating the mobile proxy bottleneck. In this case, the proxy of the mobile device will provide location information used to establish the route, much like location databases in cellular networks. Benefits of this approach are that end-to-end delay will be decreased for data transfer, and the system may become more scalable to wide area networks. The disadvantage is that handoffs will become more complicated, and end systems will have to become aware of the mobility of clients.

We have seen that proxies may be used to assist in routing in two ways: they can provide routing information so that optimal routes may be established, or they can be used as central points through which all data is routed, thus hiding mobility from fixed devices.

4.3 Applications

In this section, we describe how the techniques of caching and proxies are applied to nomadic computing applications. We take examples of file systems, multimedia communication, and browsing.

4.3.1 Caching: File Systems and Web Browsing

Caching techniques have traditionally been used to improve the performance of distributed file systems. In the CODA file system [15], caching is used not only to improve performance, but to increase system availability. CODA is designed to support users that operated in a disconnected mode, i.e., without connectivity to a network. The initial CODA work was targeted primarily towards users that had planned disconnection, such as those that would unplug a laptop computer from a network and use it as a stand alone machine for periods of several hours or even days. Users that were unexpectedly disconnected were also supported. Recent extensions to CODA enhance the performance of weakly connected mobile computers. A weakly connected computer is one that has either a low bandwidth connection, intermittent connectivity, or does not desire to use a large amount of bandwidth for other reasons, such as cost.

CODA uses whole file caching to ensure that when a user becomes disconnected, the user may access and operate on any files it has cached. Files are organized into sub-trees called volumes. When connected, CODA uses a callback mechanism to ensure file consistency. If a file is modified on any client, all those with the file in their cache receive a call back break notifying them that their version has become inconsistent. CODA uses an optimistic caching scheme which allows users to modify locally cached files even when disconnected without locking the file on the server. This increases the availability of the files to network users, but may lead to inconsistencies if other users access the file and modify it when the user is disconnected. These inconsistencies are discovered and resolved when a user becomes re-connected.

In a normal state, a CODA user operates in connected mode. In this state, the client periodically performs *hoard walks*. During a hoard walk, certain files are retrieved and cached on the client. Files are chosen for a variety of reasons. Each user may have their own hoard profile in which they assign priorities to certain files. These priorities may be modified based on the activities of a client. For example, if a file is not accessed for a period if time, its priority will decrease. During a hoard walk, the files with the highest priority are cached. If a user is disconnected unexpectedly, these files are available for their use. If a user has prior knowledge of a disconnection, they may request a hoard walk so that specific files are cached.

When a user is in disconnected state, all file operations are logged. Any operations on files that are not cached return an error. When a user becomes connected once again, reintegration takes place. During reinte-

gration, the log of operations is replayed on the file server. If there are no conflicts, for example conflicting write operations by different users on the same file, all updates are made to the files on the server. If conflicts are found, the user is notified and must reconcile the inconsistencies.

There are several potential problems with disconnected systems that are addressed by CODA. First, a large enough amount of storage must be available on the client for a reasonable amount of information to be cached. Experience with CODA shows that 50-60 Mbytes of storage is sufficient space to allow disconnected operation for the period of about one day. Second, the size of the replay log must be controlled. CODA does this by eliminating obsolete or redundant operations from its log. This technique reduces the log size to a few percent of the cache size. Finally, if a large number of conflicts exist upon reintegration, the system may not be worth using. However, file system traces have shown that write conflicts occur on only about 0.75% of files in the period of one day. Therefore, this is not a serious problem.

Recent extensions to CODA[19] attempt to take advantage of low speed or intermittent network connectivity to improve performance and availability. To improve performance and data consistency, reintegration may be performed in the background while users are weakly connected. Because the bandwidth available is low, an effort is made to keep communication low. Therefore, instead of checking for consistency for each individual file, consistency for each volume of files is checked. If a volume is found to have changed, then each individual file in the volume is checked for consistency. This technique allows the reintegration process taking place at 9.6 Kbps to take only 25% longer than the same process at 10 Mbps. In a study of 26 clients, 97% of the volumes were consistent upon reintegration, saving the checking of an average of 53 files in each volume. Also, by performing the reintegration in the background, there is likely to be fewer inconsistencies when a user resumes its normal network connection and must reintegrate.

Weak connectivity may also be used to service cache misses. Whereas in the original CODA system, a cache miss while disconnected caused a error, in a weakly connected system, if the file retrieval time is below a certain threshold, the cache miss is transparently serviced. This increases the availability of the system.

Other research efforts on file systems for mobile users have also been undertaken [25] [26].

Caching and pre-fetching has also been applied to a wireless web browsing application called W4 [14]. In this system, the mobile device is a simple PDA. The PDA is mated with a proxy executing on a fixed workstation. This aspect of the system is described in more detail in the next section. When the PDA requests a web page, it is sent from the proxy to the PDA where it is cached. In addition, subsequent pages are pre-fetched and cached on the PDA. When accessing cached pages, the time taken to display the page is approximately one second. For pages that are not cached on the PDA, response time is 2-3 seconds over a 4.8 Kbaud

cellular connection.

4.3.2 Network Proxies: Multimedia and Browsing Applications

Network-based proxies are used to alleviate processing constraints of mobile devices, overcome heterogeneity of end devices, and provide location information for many wireless applications. In this section, we examine the use of proxies in the WDCPA, ParcTab, and InfoNet projects for multimedia applications, and W4 for a browsing application.

To allow negotiation in multimedia services, in the WDCPA project, an agent called a *user process* [27] which resides inside the wired network is responsible for negotiating on behalf of the mobile user. With the advent of multimedia applications in a wireless environment, and with many terminal types and applications likely to co-exist on a single network, negotiation procedures may be extensive. In order to reduce the amount of control traffic over the air interface, negotiation protocols terminate on the user process.

The user process is loaded with the user device capabilities and desired services upon power-up. It also tracks the status of the device through updates from the mobile device. With this information, the user process can serve incoming requests for communication, perform negotiation functions, and can reject or accept the requests based on the capabilities and state of its device, all without having to use air resources to contact the device directly.

Because WDCPA is designed to work in a wide area environment, the user process migrates as its mobile device moves. By keeping the user process close to the mobile device, status changes to the mobile device only have to be sent a short distance to reach the user process, thus reducing the amount of long distance signaling. To enable the user process to migrate, it is kept fairly light-weight. Unlike other user agents, the user process in WDCPA performs only control functions and is not involved in the processing of user information. To migrate a user process of a device that is not in an active connection requires the transfer of approximately 50 bytes of information. The migration of a user process with a device in an active connection requires the migration of approximately 100 bytes of information.

In both the ParcTab and InfoPad projects, the agents for the mobile devices allocate resources on the mobile devices. Resource allocation is critical for multimedia applications. Displays must be allocated for video, shared blackboards, etc., and audio resources must be allocated for audio applications. For example, the ParcTab has a limited display which can only be used by a single application at a time. If multiple applications are active, the Tab agent is responsible for multiplexing their output to the Tab screen. The ParcTab agent is also responsible for supplying location information to location-sensitive applications. For example,

when using a group drawing application, all users in the same room will be given a pointer that is active on the screen in the room in which they are located.

Both the ParcTab and InfoPad agents execute in fixed locations. Therefore, they may be extended to perform more complex operations without concern for their complexity because, unlike the WDCPA user process, they do not move.

The W4 [14] project applies proxies to *web browsing* applications. As previously described, each PDA is mated with a workstation. When the user requests a web page from its PDA, the request is sent to the workstation. The workstation retrieves the page, parses it, formats it to be displayed on the PDA, and then forwards it to the PDA. All images are filtered. This architecture hides the limited capabilities of the PDA from the web application. It also places complex processing functions of parsing, format translation, and filtering inside the network. Finally, this structure limits the traffic on the air interface by filtering out images that cannot be displayed on the PDA. The drawback of this approach is inefficiency. While it facilities the rapid introduction of wireless browsing applications, it is inefficient to retrieve entire web pages onto the workstation, only to have forms and images filtered. As wireless web browsing becomes more popular, applications or web pages will be modified to account for simple wireless end devices, leading to more efficient end-to-end solutions.

5 Conclusions

In this paper, we described the characteristics of the nomadic computing environment, and how three general techniques, asymmetric design of protocols and applications, the use of network-based proxies, and data caching, can be applied to accommodate this environment.

The characteristics of the nomadic computing environment include lower bandwidth and higher error rates on wireless access links than in current wired networks. User profiles have changed from fixed users with powerful desktop machines and reliable network connectivity, to mobile users with processing and power limited end devices. Network connectivity will be unreliable, because as users move, they often become disconnected from the network. Many applications and services currently available on desktops in a wired environment, such as distributed file systems and web browsing, must be supported in a wireless, mobile environment. In addition, new location dependent services are also being developed. A challenge exists to make access to these applications comparable with their fixed counterparts in terms of performance and reliability.

We described how the asymmetric design of protocols and applications can help overcome the processing

and power limitations of mobile devices. We presented AIRMAIL, a link layer protocol that employs asymmetric ARQ procedures for error recovery, as an example.

Network-based proxies are used to overcome processing imbalances, perform error control, and assist in routing. W4, WDCPA, ParcTab, and InfoNet were all presented as examples of how network-based proxies can assist in overcoming the asymmetric nature of the nomadic computing environment, and limit bandwidth usage. W4 uses a network proxy for a web browsing application. By filtering and formatting information, it reduces the use of air interface bandwidth and the processing on its mobile device. WDCPA employs a user process to perform negotiation functions. ParcTab and InfoPad use network-based agents for resource allocation. The SNOOP protocol was presented as an example of the use of a proxy to perform error recovery. SNOOP uses a network proxy to perform error control and to shield TCP from high error rates experienced on the wireless link. Cellular telecommunication and packet networks were presented as examples of how proxies can be used to assist in routing. These systems use home switches and routers as proxies to assist in routing.

Caching and pre-fetching of data are used to improve performance and availability of wireless applications. Telecommunication and packet networks were presented as examples. Telecommunication networks use a hierarchy of databases to store location information. Research efforts illustrate how the location of the data stored, how frequently it is updated, and how much hierarchy in the location structure is present, have serious impacts on system performance. Routers in mobile packet networks cache location information to eliminate the need to constantly locate a mobile device to deliver data. Applications which are sensitive to response time also use caching and pre-fetching of data. We presented the CODA file system and the W4 web browsing application as examples. In addition to improving performance, CODA uses caching to increase system availability to users that are disconnected or weakly connected.

These techniques will be extended and applied as new nomadic computing applications are developed. As wireless networks become more widely deployed and reliable, and end devices achieve a comfortable compromise between capabilities and portability, many more location-sensitive applications will be used. These applications will require more intelligence than their current counterparts, and will likely rely on the techniques presented here to provide good performance, reliably, under the constraints of the nomadic computing environment.

References

[1] EIA/TIA IS-54 (Revision B), "Cellular System Dual-Mode Mobile Station - Base Station Compatibility Standard," April, 1992.

[2] EIA/TIA IS-95, "Mobile Station - Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular System," July, 1993.

[3] EIA/TIA IS-41 (Revision C), "Cellular Radio-TelecommunicationsIntersystem Operations," 1995.

[4] ETSI, "European Telecommunications Standards Institute GSM Recommendations."

[5] CDPD Forum, "CDPD System Specification, Rel. 1.1," 1995.

[6] IETF, "IP Mobility Support," C. Perkins, Ed., Internet Draft, 1995.

[7] CCITT Recommendation I.311, "B-ISDN General Network Aspects," 1991.

[8] K. Y. Eng, at al, "BAHAMA: A Broadband Ad-Hoc Wireless ATM Local Area Network," *ACM/Baltzer Wireless Networks*, 1995.

[9] E. Ayanoglu, S. Paul, T. F. La Porta, K. K. Sabnani, R. D. Gitlin, "AIRMAIL: A Link-Layer Protocol for Wireless Networks," *ACM/Baltzer Wireless Networks*, Vol. 1, No. 1, 1995.

[10] H. Balakrishnan, S. Seshan, E. Amir, R. H. Katz, "Improving TCP/IP Performance over Wireless Networks," *ACM MobiCom*, 1995.

[11] T. F. La Porta, M. Veeraraghavan, P. A. Treventi, R. Ramjee, "Distributed Call Processing for Personal Communication Services," *IEEE Communications Magazine*, June, 1995.

[12] R. Want, et al, "An Overview of the ParcTab Ubiquitous Computing Environment," *IEEE Personal Communications Magazine*, December, 1995.

[13] M. T. Le, F. Brughardt, S. Seshan, J. Rabaey, "InfoNet: The Networking Infrastructure of InfoPad," *Proceedings of Compcon*, March, 1995.

[14] J. F. Bartlett, "W4 - the Wireless World Wide Web," *Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, Dec. 1994.

[15] J. Kistler, M. Satyanarayanan, "Disconnected Operation in the CODA File System," *ACM Transactions on Computer Systems*, Vol. 10, No. 1, Feb., 1992.

[16] R. Steele, *Mobile Radio Communication*, Pentech Press, 1992.

[17] B. Tuch, "Development of WaveLAN, an ISM Band Wireless LAN," *AT&T Technical Journal*, vol. 72, No. 4, July/August, 1993.

[18] R. Caceres, L. Iftode, "Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments," *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 5, June, 1995.

[19] L. B. Mummert, M. R. Ebling, M. Satyanarayanan, "Exploiting Weak Connectivity for Mobile File Access," *Proc. of 15th Symposium on Operating Systems Principles*, Dec., 1995.

[20] R. Jain, Y.-B. Lin, C. Lo, S. Mohan, "A Caching Strategy to Reduce Network Impacts of PCS," *IEEE Journal on Selected Areas in Communications*, Vol. 12, No. 8, Oct., 1994.

[21] G. M. Voelker, B. N. Bershad, "Mobisaic: An Information System for a Mobile Wireless Computing Environment," *Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, Dec. 1994.

[22] K. S. Meier-Hellstern, E. Alonso, "The Use of SS7 and GSM to Support High Density Personal Communications," *Proc. of IEEE ICC*, 1992.

[23] T. F. La Porta, M. Veeraraghavan, R. Buskens, "Comparison of Signaling Loads for PCS Systems," submitted to *IEEE/ACM Transactions on Networking*, 1995.

[24] "Route Optimization in Mobile IP," D. B. Johnson, C. Perkins, Ed., Internet Draft, 1995.

[25] C. D. Tait, D. Duchamp, “Detection and Exploitation of File Working Sets,” *Proc. 11th International Conference on Distributed Computing Systems*, 1991.

[26] P. Honeyman, L. Huston, J. Rees, D. Bachman, “The Little Work Project,” *Proc. 3rd IEEE Workshop on Workstation Operating Systems*, 1992.

[27] R. Ramjee, T. F. La Porta, M. Veeraraghavan, “The Use of Network-Based Migrating User Agents for Personal Communication Services,” *IEEE Personal Communications Magazine*, December, 1995.