Low-Latency Handoff for Cellular Data Networks

by Srinivasan Seshan

B.S. (University of California at Berkeley) 1990
M.S. (University of California at Berkeley) 1993

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Professor Randy H. Katz, Chair
Professor Eric Brewer
Professor Robert Harris

1995

Low-Latency Handoff of Cellular Data Networks

by

Srinivasan Seshan

Abstract

Low-Latency Handoff for Cellular Data Networks

by

Srinivasan Seshan

Doctor of Philosophy in Computer Science

University of California at Berkeley

Professor Randy H. Katz, Chair

In this dissertation, we examine the problem of performing handoff quickly in cellular data networks. We define handoff as the process of reconfiguring the mobile host, wireless network and backbone wired network to support communication after a user enters a different cell of the wireless network. In order to support applications and protocols used on wired networks, the handoff processing must not significantly affect the typical end-to-end loss or delay of any communications. This dissertation concentrates on two specific areas of handoff processing: routing updates and state distribution. The techniques we use to solve these problems are:

1. *Multicast* to set up routing in advance of handoff.
2. *Hints*, based on information from the cellular wireless system, to predict handoff.
3. *Intelligent buffering*, enabled by the multicast of data, to prevent data loss without the use of complicated forwarding.
4. *State replication,* enabled by the multicast, to avoid explicit state transfers during the handoff processing.

This dissertation describes the design, implementation and evaluation of these techniques in a variety of networking and computing environments. We have shown that any necessary routing updates and state transfers can be performed in a few milliseconds. For example, our implementation in an IP-based testbed completes typical handoffs in 5—15 msec. In addition, the handoff processing introduces no additional packet delays or data loss. The primary cost of our algorithms to improve handoff latency is the use of excess bandwidth on the wired backbone networks. How-

ever, we have introduced base station layout algorithms that reduce this cost. In current systems, the performance improvement provided by these techniques easily outweigh the resources consumed. Since wired backbone networks will continue to have much greater available bandwidth than their wireless counterparts, this trade-off between handoff performance and network resources will continue to be advantageous in the future.

Randy H. Katz

Acknowledgments

I thank Professor Randy Katz for his support during my years at Berkeley. He provided constant academic guidance and inspired many of the ideas presented here. Randy is a superb teacher, a great communicator and an excellent manager of research projects. I was very fortunate in having the chance to work with him as my research advisor. I look forward to further develop our relationship both as a colleague and a friend.

I thank Professors Eric Brewer, Bob Brodersen, Domenico Ferrari, Robert Harris and David Mowery for being on my qualifying exam and dissertation committees. I would also like to thank Prof. Bob Brodersen, Prof. Jan Rabaey and the many members of the InfoPad team for their involvement in developing the InfoNet architecture. My thanks to the InfoPad project for giving me support over the years and providing me with a testbed for my ideas.

At Berkeley, I have learned as much from the continuous interaction with other students as from my professors. I wish to acknowledge in particular my officemates during the past years, Elan Amir, Hari Balakrishnan, Ann Chervenak, Tzi-cker Chiueh, Mike Dahlin, Ethan Miller, Mario Silva and Mark Stemm. In addition, I would like to give special thanks Kimberly Keeton and Bruce Mah, who participated in the initial brainstorms that lead to the development of the handoff algorithms. I would also like to acknowledge Elan Amir and Hari Balakrishnan for their aid in bringing the Daedalus testbed up to speed.

Theresa Lessard-Smith and Bob Miller performed the administrative work required for this research. They were vital in making my stay at UCB enjoyable.

Finally and most importantly, I would like to thank my parents for their efforts to provide me with the best possible education.

Contents

Chapter 7

List of Figures

List of Tables

Chapter 1

Introduction

Among the most important technological trends of the 1980s and 1990s is the emerging use of cellular wireless communications, portable computing and data networking. Each of these technologies has created their own unique markets. These markets include:

- Cellular Wireless Communications → Cellular Telephony and Paging
- Portable Computing → PDAs and Portable PCs
- Data Networks → On-line services, World Wide Web and Internet services

Only in the past few years have these technologies progressed to the point where they may be combined in useful devices. The combined products of these trends hold forth the promise of providing users with ubiquitous access to information. This ubiquitous access is the culmination of several decades of evolution in providing information access to users (Figure 1.1). In the envisioned environment, users will access information using a number of different devices and technologies. Information access devices will vary in capabilities from systems like today's personal workstations and portable computers to smart cellular telephones. The communications environment will be an internetwork of cellular data net-

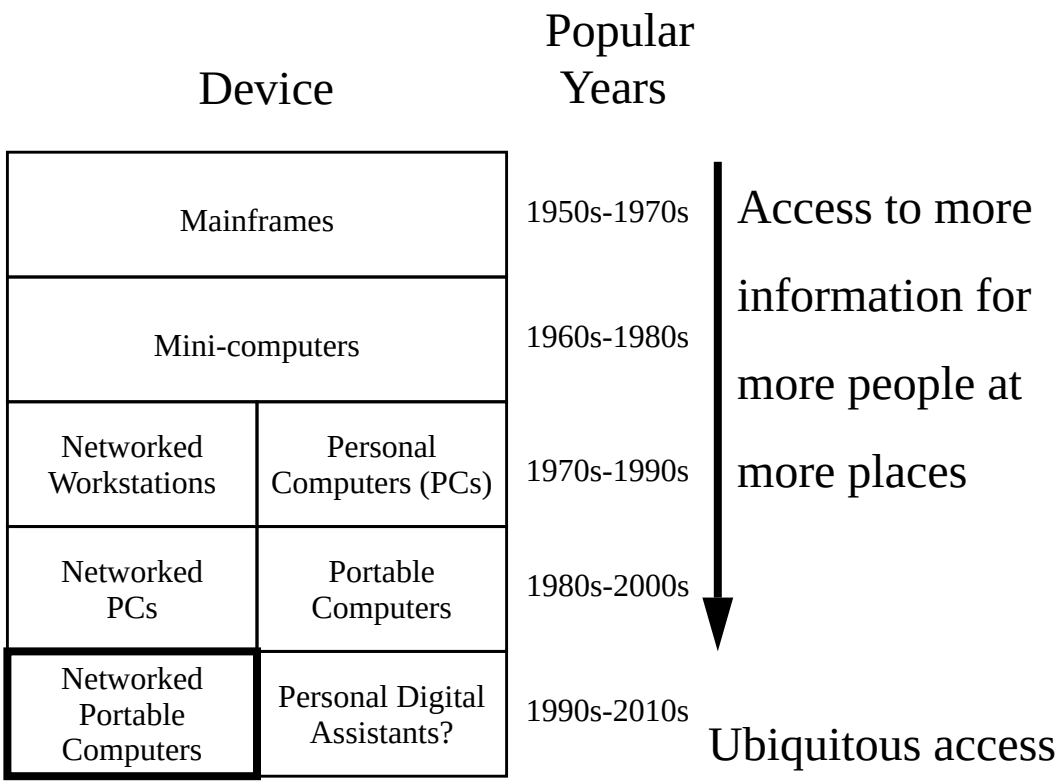


Figure 1.1 Information Access Trend

works and very high bandwidth wired backbone networks. The services available to users will include real-time audio- and video-conference support and access to World Wide Web-like databases containing multimedia information. Figure 1.2 depicts this envisioned environment.

The new computing environment presented by wireless, networked, personal computing systems such as the one described above [Shen92] introduces many challenges, because of the requirements of networked applications and the mobile nature of the hosts. One of these problems is that user mobility forces networks to cope with new dynamics of routing. It is this problem that we address by developing support for low latency handoff in cellular networks. To understand this problem, we must first briefly examine each of the three basic technological areas that make this future environment possible. This examination will provide an understanding of the current and future capabilities of the systems for

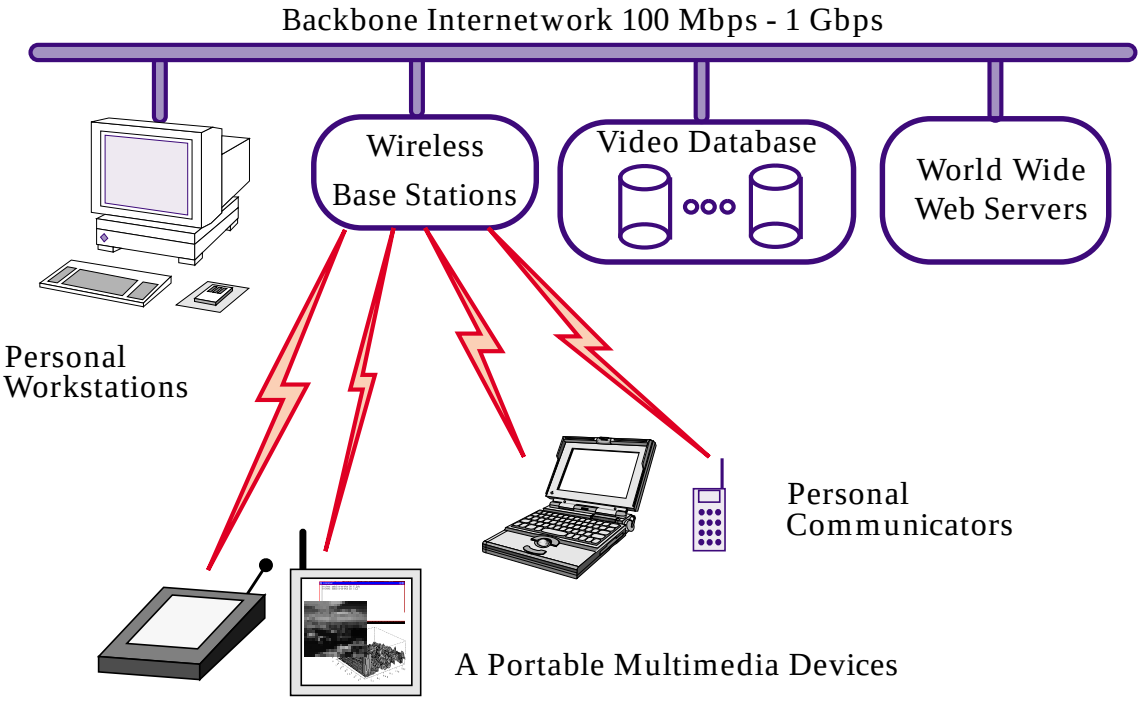


Figure 1.2 Future Environment

which we are developing routing support. Each section also introduces some of the technology related terminology that we will use throughout this dissertation.

1.1 Cellular Wireless Technology

The popularity of cellular telephony clearly shows an existing desire for mobile communication. In this section, we describe the aspects of cellular technology that will likely be used to provide digital communication in a ubiquitous information access system.

Generally, mobile networks are composed of a wired, packet-switched, backbone network and a wireless network. The wireless network is organized into geographically defined cells, with a control point called a *base station* (BS) in each of the cells. The base stations, which are attached to the wired network, provide a gateway for communication between the wireless network and the backbone network. Base stations usually have only one physical connection to the wired network. As a *mobile host* (MH) travels between wireless

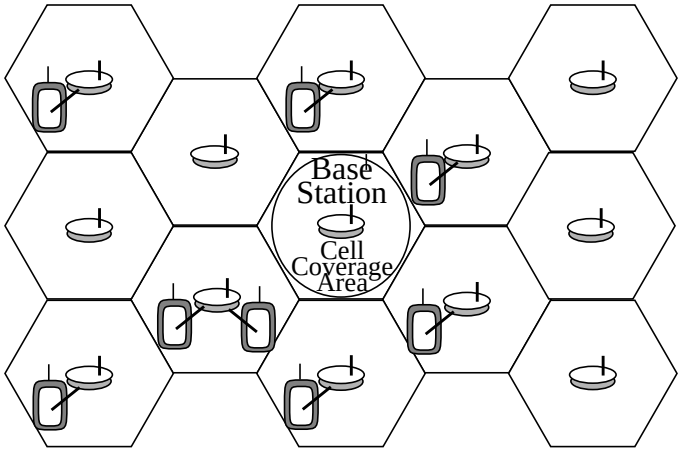


Figure 1.3 Arrangement of a cellular wireless network.

cells, the task of forwarding data between the wired network and the mobile host must be transferred to the new cell's BS. For convenience we refer to the direction towards the MH as the *downlink* and the direction away from the MH as the *uplink*. An example of a cellular network is shown in Figure 1.3.

The entire process of reconfiguring communication, at the MH, wireless network and backbone wired network, is known as *handoff*. The objective of handoff is to maintain end-to-end connectivity in the dynamically reconfigured network topology. During a handoff, the route of data through the wired network to the MH must be updated to pass through this new BS. In addition any state in the old BS associated with the MH must somehow be transferred to the new BS and the radio communication may also need to be reconfigured. It is support for this reconfiguration that this dissertation addresses.

Several factors affect the characteristics of communication and requirements of handoff in a cellular network. Systems are often designed differently to support vehicular (> 3 meters/sec) versus pedestrian (< 3 meters/sec) movement. Another important design factor is the size of cells. Systems that consist of normal sized cells (> 1mile diameter), micro-cells (1 mile — 100 meter diameter) or nano-cells (< 100 meter diameter) must use different transmission technologies. These two factors combine to determine the frequency and duration of handoffs.

In addition, the wireless technologies used between the MH and BS significantly impact the quality of mobile communication. In the following subsections we describe the characteristics of some of the more popular transmission technologies.

1.1.1 Radio

The characteristics of radio (< 2.5 GHz carrier frequency) propagation and reception define the many of the capabilities and characteristics of cellular radio networks [Linn95, Schi91]. Most building structures do not interfere with these frequencies of radio waves. However, most radio networks have *null regions*, areas in which an MH loses wireless contact with its BS, possibly due to other forms of interference with radio propagation. These propagation characteristics result in many advantages and disadvantages. Since higher frequency radio (>5 GHz) propagation is much more easily stopped by walls, its communication characteristics have more in common with infrared communication than low frequency radio. Although few current systems use these high frequency bands, future systems will likely use these bands to provide higher bandwidth.

Since structures do not prevent propagation, radio signals from a base station are not well contained resulting in poorly defined cell regions. Therefore, there is generally considerable overlap among the cells of a radio network. In these overlap regions, a mobile host may be able to communicate with multiple base stations. The existence of overlap regions relaxes the requirement on how quickly handoff must be initiated or completed in many systems. To avoid a complete disconnection from the network, a system must only initiate a handoff while the MH is in the overlap region and must complete the handoff before the MH exits the region.

Another advantage of radio propagation is that there is a “reasonably consistent” relationship between the signal strength and distance from the transmitter. This allows the simple implementation of a mechanism to identify which BSs or cells are near an MH. Each MH in the system can listen for radio transmissions for any BS “in range” and take signal strength measurements of the transmission. The relative strength of the radio signals from

an MH's current BS and the next strongest radio source gives the MH an indication of when it is close to a new cell. However, the presence of other interfering sources prevents the signal-strength information from being completely reliable or accurate. Therefore, this information cannot always be relied upon to provide accurate location information or advance warning of handoff.

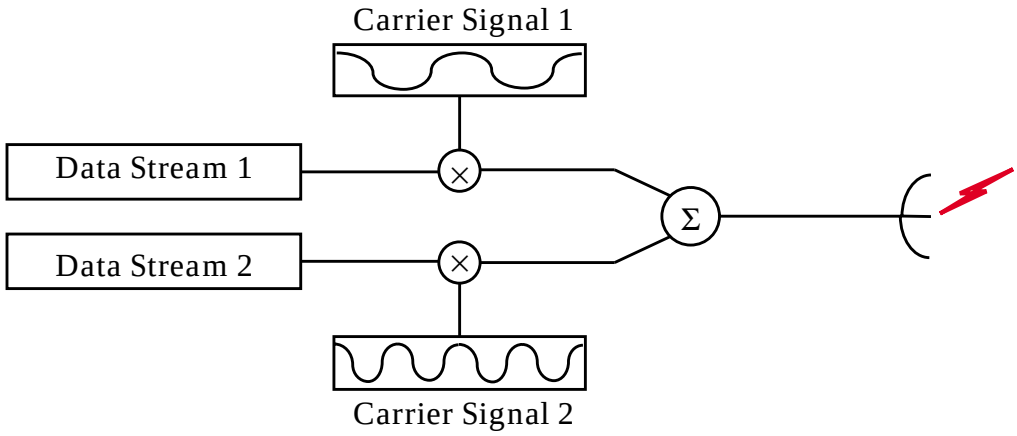
The interference between signals and the wide range of reception characteristics that must be supported result in two significant drawbacks for wireless radio systems: high error rates and low bandwidth. Radio systems typically have 3-8 orders of magnitude higher bit-error rates and 1-2 orders of magnitude lower bandwidth than typical wired links.

A drawback of the signals not being well contained is that a variety of multiple access schemes must be used to prevent interference between overlapping signals. These media access schemes include Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA) and Code Division Multiple Access (CDMA). The access control methods are shown in Figure 1.4. Often these schemes are combined to provide multiple access. The use of these access techniques adds several additional qualities to cellular radio communication. These multiple access schemes and their effects on handoff mechanisms are described in the following sections.

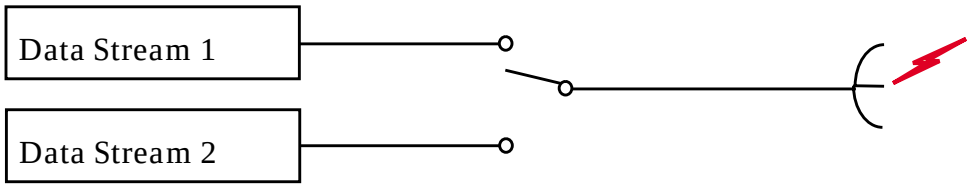
1.1.1.1 FDMA

In an FDMA system, the frequency band used for communications is divided into a number of smaller channels. Each transmitter-receiver pair in a region is uniquely assigned one of the channels to use for communication. Since no other transmitter in range is using the same carrier frequency, communication on this channel experiences little interference. In most cellular FDMA systems, a single cell is only assigned a subset of the frequency channels possible. A color mapping algorithm is then used to ensure that for each cell none of the nearby cells has any of the same channels in its subset. The number of "colors" used in this mapping is referred to as the reuse factor of a cellular FDMA system.

Frequency Division Multiple Access (FDMA)



Time Division Multiple Access (TDMA)



Code Division Multiple Access (CDMA)

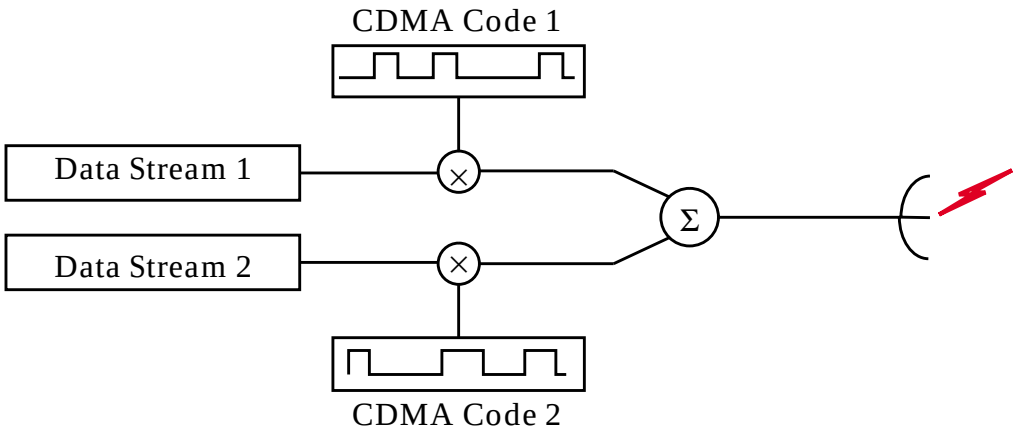


Figure 1.4 Multiple Access Techniques for Cellular Radio Networks

This form of multiple access has several implications on cellular communications. Since the transmitters and receivers are designed to use a single frequency sub-band, they often

do not support listening to multiple transmitters or transmitting to multiple receivers. As a result, special support may be needed for an MH to identify nearby BSs. In addition, an MH must be told to switch frequencies during handoff since adjacent cells do not use the same frequencies. This switch of frequencies may introduce some additional delays into the handoff process. Another drawback of such a system is that the total available bandwidth is divided into a number of smaller channels. This results in a loss in efficiency for data communications, due to the lack of statistical multiplexing. However, the smaller channels make the individual transmitters and receivers simpler and cheaper. Some FDMA systems, such as the GSM-based [Bals93] digital cellular systems, combine FDMA with TDMA to provide multiple access to a single frequency channel. The systems use the combined access methods to implement control channels and to share channel bandwidth among multiple hosts.

1.1.1.2 TDMA

A TDMA system provides multiple access by ensuring that only a single transmitter in an area is active at a time. This is similar to most wired broadcast-based network media access (MAC) protocols, such as CSMA/CD, token passing and time slotting. In TDMA systems each of mobile hosts awaits its turn before transmitting. An MH usually identifies its turn by noticing that the link is idle (CSMA/CD) or by being passed information like a token (token passing) or transmission schedule (time slotting). Unlike wired networks, not all possible transmitters are guaranteed to be in range of each other. Therefore, wireless networks use modified versions of wired network MAC protocols.

A TDMA provides better efficiency for data networking since it allows statistical multiplexing. In addition, the use of a shared channel allows the mobile host to listen to multiple transmitters. This will likely aid the MH in determining its current location. However, the increased complexity of devices usually results in a much lower peak, aggregate bandwidth. Many TDMA systems maintain some form of tight time synchronization among hosts within a cell. In addition, many TDMA systems use other techniques such as FDMA

or CDMA to avoid interference between cells. During handoffs, a mobile host may need to go through some form of clock synchronization and perform the FDMA or CDMA handoff requirements before transmission or reception.

1.1.1.3 CDMA

The most modern of the three techniques, direct-sequence CDMA (DS-CDMA), uses mathematical codes and interference control to provide multiple access. DS-CDMA assigns each host in a cell a carefully chosen unique fixed-length bit sequence, called a CDMA code. For each bit a host wishes to transmit, it actually transmits the xor product of the bit and the host's CDMA code. In addition, each mobile host controls its transmission power such that the base station receives all transmissions at approximately the same power level. The base station may not need to use such power control on transmissions since the base station is the only transmitter for the downlink of data. Each receiver receives the sum of transmissions from several hosts. Since the power associated with each transmitting host should be equal, the portion of the sum associated with each transmitter should also be equal. This fact combined with the carefully chosen codes allow the receiver to retrieve the original bit-sequence from the combination of received bits.

CDMA uses tight power control and synchronization within a cell to provide multiple access. Many CDMA receivers support receiving packets from multiple sources or obtaining signal measurements from multiple sources. A significant drawback of the system is the power control necessary on the uplink. In addition, as in FDMA, the system fragments the available bandwidth. However, unlike FDMA, A CDMA system need not fragment its bandwidth between cells. When an MH moves between cells, it needs to synchronize power and timing for the new cell and it may need to change CDMA codes.

1.1.2 Infrared Links

An alternative to digital radio links is the use of infrared between the base station and mobile host. The propagation of infrared light is very similar to that of the visible spec-

trum. For example, most opaque materials block infrared transmissions and sunlight prevents the use of infrared links. These characteristics make infrared best used for some directional links and for coverage of small interior rooms. Like visible light, infrared has sharp boundaries of coverage (shadows). As a result, entries and exits into cell coverage are very sudden and dramatic. This also implies that no link layer mechanism is available to predict handoff accurately. There is little overlap between infrared cells since there are sharp boundaries between cells. For this and for other technological reasons, infrared systems must use only TDMA-like access algorithms. Since the overlap is small, handoff processing must be fast in order to complete before the mobile user exits the overlap region.

1.1.3 Future Trends

The future trends of wireless networks are as controlled by FCC regulations as it is by technological factors. The FCC is currently making several high-frequency radio bands available for unlicensed use. All unlicensed bands have strict interference and power regulations. As a result, systems in these bands have very small cell sizes and use CDMA technology to reduce and resist interference. The FCC has also auctioned a number of PCS (Personal Communication Services) licenses in the lower frequency radio spectrum (< 2.4GHz) as well as made the 900MHz and 2.4GHz radio bands available for unlicensed use [FCC95]. Therefore, most devices available now and in the near future will provide either micro- or nano-cellular low-frequency radio coverage. The current technological trend in radio devices is towards lower power transmission. This is a result of pressure to provide higher aggregate bandwidth through the spacial reuse of radio spectrum (i.e., more and smaller cells). These technological and FCC trends imply that handoffs will occur more often in the future and must complete more quickly.

1.1.4 Summary of Cellular Data Networks

A summary of some expected cellular data system characteristics is shown in Table 1.1

Cellular Technology	Bandwidth	Multiple Access Technologies	Cell Radius (meters)	Location in Cell Available
900Mhz Unli-censed	100 Kbps — 2 Mbps	CDMA	30 - 1000	often
2.4Ghz	1 Mbps — 2 Mbps	CDMA	30 - 200	often
Infrared	100 Kbps — 10 Mbps	TDMA	room sized, < 10	never
PCS	10 Kbps — 100 Kbps	TDMA, CDMA, FDMA	100 - 5000	usually
HF unlicensed	> 1 Mbps	CDMA	room sized < 10	unlikely

Table 1.1 Characteristics of Cellular Data Networks

Any networking support designed for a ubiquitous information access system must take into account the limitations imposed by cellular data networks and take advantage of the information and structure of such networks. For example, future systems must tolerate higher error rates and lower bandwidths across wireless links than wired links. However, they must also use information provided by many wireless networks, such as location within a cell and logical layout of cells, to aid in solving any new problems.

1.2 Portable Computing

The use of mobile computers is the next logical step in the evolution of the computer industry. Over time, computers have evolved from enormous mainframes to desktop sized personal computers. This trend has also made computing and information more accessible to the individual user. Mobile computers continue this shrinking trend. The rapid growth in sales of equipment such as portable computers clearly demonstrates this demand for portability.

The main limitation of current portable computers is that they must operate on batteries. Unfortunately, battery technology is improving at the relatively slow rate of 30% every 5 years [Eage92] with no breakthrough technologies expected. The typical approach to improving battery life is to re-engineer chips and systems for lower voltage operation. In addition, many of the system capabilities are reduced to decrease power consumption. As a result, currently available portable computers are a factor of two slower and have significantly less storage and memory than their desktop counterparts (Table 1.2). Since battery

Measure	**Typical Portable**	**Typical Desktop**
SpecInt92	30 — 70	100 — 200
Disk Space (MB)	250 — 1000	500 — 4000
Cache Size (KB)	0 — 256	256 — 1024
Video Memory (KB)	512 — 1024	1024 — 4096
Main Memory (MB)	4 — 16	8 — 32

Table 1.2 Typical Portable Computer Capabilities

technology is improving relatively slowly, this gap between desktop and portable performance is not likely to change in the coming years.

Another important trend in portable computer systems is the demand for hands-free or single-hand operation. This operation does not fit well with the traditional keyboard and mouse user interface. As a result, many recent portable devices use pen-based interfaces (some with handwriting recognition) as a replacement. The primary weakness of these implementations is the low accuracy and speed of the handwriting recognition algorithms that portable units support. This is a result of the large computational and memory requirements of good handwriting recognition systems and the weak capabilities of portable computers. This mismatch in requirements has resulted low demand for such implementations

The above trends show that software and algorithms designed for mobile operation must take into account the limited capabilities of portable computers. Not taking these limitations into accounts can create disastrous results like those of pen-based PDAs (Personal Digital Assistants).

1.3 Data Networking

The popularity of Internet-based on-line services such as the World Wide Web (WWW) [Net95, BL95] and the MBone [MBo95, Deer89] indicates the desire for access to global information systems. To provide mobile users access to such services we must understand the requirements of connecting to the networks available today and in the future. In this section, we examine the requirements of these network services, the routing of data in current networks and the future trends in networking technology.

1.3.1 Network Service Requirements

Developers have tuned network based applications and network protocols to the characteristics of current wired networks and stationary users. These characteristics include causes for data loss and variations in end-to-end delays. For example, the TCP [Post81b, Jaco92] transport protocol assumes that all losses in networks are a result of congestion and that any end-to-end delay greater than the average delay plus twice the variance is an indication of loss. When such delays or losses occur, TCP invokes the appropriate mechanism to avoid future problems. Similarly, video conference programs such as `vic` [McCa95] adjust their playback buffering to compensate for the typical end-to-end delays. Any unexpected variation in the delay will cause such programs to drop or improperly display video frames.

1.3.2 Network Routing

The most important difference between the mobile and stationary user is that the routing of data to mobile users must be more dynamic. The route data takes in a network must

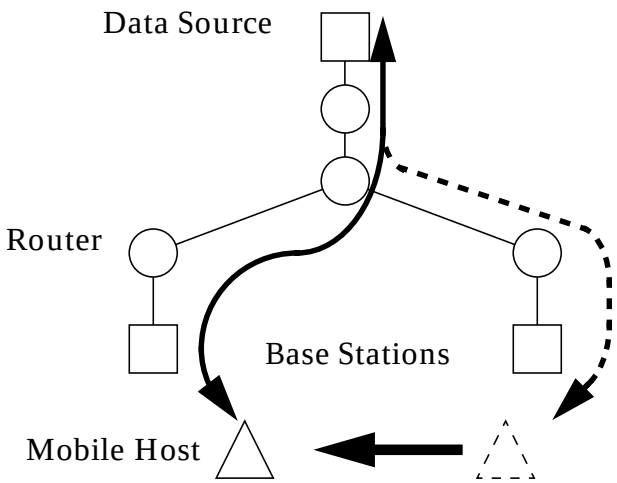


Figure 1.5 A Handoff

The MH has just moved from the right BS to the left BS. The grayed path reflects the previous route of data and the dark path the current route.

change each time a mobile user moves between cells of a wireless network. Figure 1.5 shows the effects of this procedure. The (inter)network layer of standard protocol stacks performs this routing. To understand the implications of adding these routing updates, we examine routing in different types of network layers today.

Datagram routing protocols such as the Internet Protocol [Post81a, Brad89] (IP) are the basis of today's most popular internetwork systems. However, there are also a large number of connection-oriented systems including the telephone system and various Asynchronous Transfer Mode (ATM) networks. We describe the fundamental differences between these two forms of routing in the following sections.

1.3.2.1 Datagram Network Routing

Each host in an datagram network has a unique address or host name that the network uses to route packets to the host. Routers in an datagram network forward each packet based on the destination address in the packet header and a table containing the next network hop in the route to all different host addresses. This table may also contain some additional state associated with the destination host. The fact that most datagram-based internetworks

have a hierarchical structure allows routers to easily combine entries in a routing table. This combined with the fact that hosts usually remain in fixed locations results in small, infrequently updated routing tables on most routers. Unfortunately, as mobile hosts (MHs) become more popular, these two characteristics will no longer be true. For datagram networks to use the standard routing protocols for mobile hosts, routers need to store routes to each mobile host. During handoffs, the system must update routes and transfer or invalidate any mobile hosts associated state located on the routers. Clearly, the existing routing solution will not scale to large numbers of mobile hosts.

1.3.2.2 Connection-Oriented Network Routing

A connection-oriented network also assigns a unique address or host name to each host. However, the network uses this address to route connections and not individual packets to the host. Therefore, two machines must establish a connection before communication can take place between them. Connection establishment typically involves a single round-trip pass of control messages from the source to the destination of the connection. These control messages install state for the connection in each router along this path. Each router maintains a table containing the next network hop and possibly some other state (for QOS or other purposes) for each of the connections that passes through it [Mah93].

Routers forward each packet based on the connection ID in the packet header and contents of the connection routing table. Although the system may take a long time to establish a connection, it can perform the routing of data for a connection very quickly and efficiently. As long as connections are relatively long lived, the duration of connection establishment does not adversely affect performance. Unfortunately, when a mobile host (MHs) moves, the network must re-route its connections to reflect the updated network topology. In addition, the network may also need to transfer any state present in the routers for this connection to the new routers along the path. Clearly, an efficient connection re-routing algorithm is necessary to support mobile hosts.

1.3.3 Networking Future Trends

User demands for higher bandwidth and new network services drive the future trends in networking. ATM networks use connection-oriented routing, fixed format headers and fixed sized packets to simplify processing in routers and switches. This allows implementation of extremely fast packet routing in hardware. As a result, ATM networks are capable of providing much higher bandwidths and lower latencies than competing technology. The demand for higher performance has resulted in an increase in popularity of ATM networks. Similarly, users have also been demanding real-time, shared, multimedia applications such as video and audio conferencing. This has driven the development of systems that support quality of service (QOS) [Ferr90b, Ferr92] requests and multicast delivery. A QOS guarantee is a promise made by the network to provide a certain end-to-end communication performance for an application. Since these guarantees are made on a per-connection basis, connection-oriented networks are most often used to provide QOS support. The multicast delivery is a mechanism provided by some networks to deliver a single packet transmitted by a host to multiple recipients. In the past, the source of the packet must perform the replication and transmission necessary to deliver the packets to multiple destinations. To improve the efficiency of this delivery, many networks now perform the necessary maintenance of destination membership and duplication of packets. The MBone [MBo95, Erik94], a virtual network inside the Internet that uses IP Multicast [Deer89, Deer91], implements this efficient form of delivery. The popularity of MBone applications indicates that future networks will support this multicast. To summarize, future inter-networks are more likely to contain some connection-oriented networks, support multicast delivery and provide QOS guarantees.

1.3.4 Data Networking Summary

From examining existing network layers, networks clearly need new routing mechanisms to support the envisioned ubiquitous information access system. In addition, a fast method to move state between routers may also be necessary. To allow current network services to be available in this future system, these mechanisms must not violate any of the assump-

tions made by upper layer protocols, such as TCP and multimedia applications. Table 1.3

Protocol Layer	Requirement
Applications/Transport	loss and delays characteristics as similar to existing networks as possible
Connection-Oriented Network	infrequent connection establishment
Datagram-Based Network	scalable routing tables without frequent updates

Table 1.3 Summary of Data Network Requirements

summarizes the requirements of the different network layers. Any protocols we develop must support both connection-oriented and datagram-based networks. In addition, they may take advantage of any multicast facilities provided.

1.4 Support in a Ubiquitous Access System

As a mobile host (MH) travels between wireless cells, the system must update the backbone data network to continue providing communications for the MH. The entire process of reconfiguring communication, at the MH, wireless network and backbone wired network, is knows as *handoff*. In is support for this reconfiguration that this dissertation addresses.

Each of the three components of a handoff, the MH, wireless network and wired network, places unique requirements on the handoff support. To effectively provide service to mobile computing in the future, a handoff protocol should have the following characteristics:

1. Capability to provide consistent end-to-end data loss and delay performance
2. Scalable to support global sized WANs, long-lived connections, high handoff rates, and high user densities

3. Mobility of computers transparent to the transport layer and above — A host's address must be fixed to allow the transport layer protocol to use the same host identifier regardless of the host's location.
4. Support for mobile hosts with little intelligence — However, the algorithms should exploit any processing available in the mobile host.

The environment we are attempting to support provides some mechanisms for performance improvements. One that we found most useful is the use of handoff prediction to aid the reconfiguration. A number of sources can provide these "hints" about impending handoff. The layout of cells in a network limit what handoffs are likely or possible. The in-cell location information provided by many cellular networks can further limit the choice of handoff targets. Other sources of hints include user movement patterns and topological information about the environment, such as the layout of hallways and roads. However, handoff algorithms cannot rely on the availability of this information about possible handoff targets. Many systems do not provide it and those that do may not guarantee its reliability. Due to these two characteristics of wireless networks, our approaches use information about an impending cell transition as a hint to improve performance, rather than as an integral component of the handoff algorithms.

1.5 Thesis Overview

In this dissertation, we present the design, analysis and implementation of an efficient protocol to support the handoff of a mobile host between cells of a wireless network. This handoff protocol uses a few basic techniques

1. *Multicast* to set up routing in advance of handoff.
2. *Hints*, based on information from the cellular wireless system, to predict handoff.
3. *Intelligent buffering*, enabled by the multicast of data, to prevent data loss without the use of complicated forwarding.

4. *State replication,* enabled by the multicast, to avoid explicit state transfers during the handoff processing.

The advantages of each of these techniques have been successfully demonstrated in the mobility support implemented for the Daedalus [Dae95] and InfoPad [Shen92, Sesh94] testbeds. The Daedalus testbed uses off-the-shelf portable computer and wireless networking hardware to create the desired ubiquitous information access system. The InfoPad system presents a much different environment by using a custom portable terminal with limited computation capabilities. We developed these concepts in a three stage process. We first examined the problem of mobile routing by performing an analysis of different possible routing update algorithms. This analysis identified several promising techniques that we investigated in the second stage of this work, implementation of the routing update scheme in the Daedalus and InfoPad systems. The third stage of the work was the evaluation of the performance of the implementations to see if the algorithms worked as expected. In addition, during the implementation of routing updates in the Daedalus, we identified state distribution as another obstacle, in addition to routing updates, in supporting low latency handoff. To provide a more complete solution to supporting user mobility, we also developed a low-overhead method to distribute state before handoff. The rest of this dissertation is organized as follows:

In Chapter 2, we examine the previous research work and commercial products that have addressed the problem of supporting mobile networked users. Much of this work has concentrated on solving the problem of performing routing updates. However, there is some preliminary work in moving state as a result of mobility.

Chapter 3 presents our initial analysis of various techniques to update routing in networks. This analysis examines the use of these techniques in a connection-oriented network. This analysis shows that the use of hints to set up routing in advance is the most important factor for routing updates. It also shows that the use of multicast and intelligent buffering are the most effective method of exploiting hints. Although the analysis uses several assump-

tions about network technologies, the estimated performance of the different algorithms are applicable to much more general networks. The performance of implementations on very different networking systems verifies this belief.

In Chapter 4, we describe the implementation and performance of the routing update protocol in an IP-based testbed called Daedalus. The implementation applies the use of multicast, hints and intelligent buffering to achieve low latency handoff in an IP-based system. The performance measurements match our expectations from the analysis and effectively show the advantages of hints, multicast and buffering.

Chapter 5 describes our solution to transferring state during a handoff. We describe how to use the multicast delivery of packets provided by the routing protocol to replicate state at the target of a handoff. This allows the handoff to occur without an explicit transfer of state. As a case study, we describe the application of this state replication technique to a TCP packet retransmission protocol called snoop. We present the performance of the modified snoop protocol to verify that the technique works as expected.

In Chapter 6, we examine how to modify some of the techniques to support unintelligent mobile hosts. The basic techniques assume sophisticated processing at the mobile host to produce the movement prediction hints. Moving this prediction elsewhere has a significant impact on the processing of handoffs. We describe the implementation of this modified handoff processing on the InfoPad testbed. Due to the state of the testbed, we only present performance estimates and not actual measurements.

Chapter 7 summarizes the contributions of this work and identifies some of the important lessons of this thesis. We also examine the areas for future work in handoffs and related areas in mobile computing.

Chapter 2

Related Work

In this chapter, we review some of the previous efforts at supporting communications to mobile hosts. We examine existing systems such as cellular telephony and proposed solutions including several Mobile IP proposals. The purpose of this chapter is to provide an understanding of the various alternatives in providing continuous connectivity to the mobile host and identify the weaknesses of these alternatives.

2.1 Introduction

There have been many past efforts to support wireless mobile data communication, including ALOHA [Abra70], DARPA Packet Radio [Lein87, Jubi87], commercial paging systems, Mobile-IP and cellular telephony. In this section, we only examine previous work that has concentrated on supporting host mobility in cellular style systems. This previous work falls into the following three categories:

1. Support for re-routing connections in connection-oriented networks — cellular telephony, group-based routing and VC trees.

2. Support for routing updates in datagram-based networks — several proposals for mobile routing in IP-based networks.

3. Support for state transfer during handoff — state transfer in the I-TCP system.

The following sections summarize these systems and identify the advantages and shortcomings of their mobile network support. We examine how well each system matches the handoff support goals presented in Section 1.4. In each description, we explain and use the unique terminology used by the system designers.

2.2 Connection-Oriented Network Routing

In this section, we examine three different efforts to perform routing updates for mobile hosts in a connection-oriented network. The first system we examine is the cellular telephone system, called AMPS. The AMPS system must support a single connection to a relatively simple mobile host. We also examine the VC Tree and Groups systems that provide support for more sophisticated mobile hosts in ATM environments.

2.2.1 Advanced Mobile Phone System [Anon93, Bals93, Ioan93a]

The North American cellular telephone system, Advanced Mobile Phone System (AMPS), is the largest mobile communication systems currently in operation. The AMPS system uses centralized control to provide connection-oriented network service to mobile users. The system consists of the following three types of devices:

1. *Cellular Telephones* — The mobile hosts in AMPS are called cellular or portable phones. They contain an antenna, RF transceiver, a unique ID and some simple control electronics.

2. *Cell Sites* — The base stations in AMPS are called cell sites. They consist of an antenna, RF electronics, some control electronics and an interface to a Mobile Telephone Switching Office (MTSO).

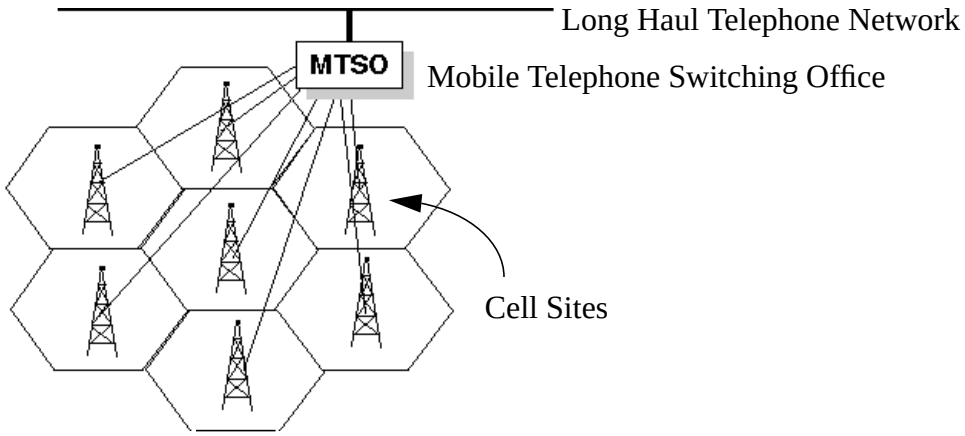


Figure 2.1 Traditional Cellular Communications System

3. *MTSO* — The MTSO interfaces a region of the cellular telephone network to the land line telephone system. It performs the billing, call processing, handoff control and all other control functions for its cellular region.

These three device types interact to perform the connection setup and maintenance necessary to support mobile connection-oriented communication. The interconnection of these devices is shown in Figure 2.1.

2.2.1.1 Connection Establishment

When a call is made to a mobile host, the MTSO is responsible for searching for the mobile and performing the call setup. The MTSO searches for the mobile host by requesting its cell sites to transmit the cellular telephone's host ID across the AMPS paging channel. The MTSO may limit this broadcast if it has knowledge of the telephone's current location. Similarly, if the mobile host is not in the area, the MTSO may expand the broadcast search to other nearby regions that the telephone has coverage. Once the telephone receives the broadcast, it responds across a setup channel through a cell site to the MTSO. At this point, the MTSO allocates a pair of channels for use and performs the call setup processing.

2.2.1.2 Connection Maintenance

The cell site monitors the received signal strength of the channel from the cellular telephone. Should the signal strength drop significantly, the cell site requests the MTSO to handoff the conversation. The MTSO uses measurements from other cell sites to choose a new pair of frequencies and cell site. It then changes its routing and notifies the cellular telephone, new and old cell sites of the handoff decision. The cell sites and telephone reconfigure their reception to complete the handoff. The MTSO is also responsible for maintaining the connection for the duration of the telephone conversation. To improve connectivity quality, it may direct the mobiles and cell sites in its area to adjust transmission power. The MTSO sends these directions via in-band, analog signalling. In addition, any users moving into an already full cell will lose their connections since there is neither admission control nor dynamic allocation of resources.

2.2.1.3 Analysis

The centralized control of all connections in a region by the MTSO is inherently non-scalable. However, the system performs handoffs quickly for the limited number of connections it can support. In addition, handoffs between MTSO regions may create long chains of connection forwarding links. This results in inefficient routing and degraded performance for the end user. The handoff system also assumes a single connection per mobile unit. This system works for two important reasons:

1. Cellular telephony charges a large amount per connect time. As a result, most conversations are short lived and do not experience a large number of handoffs. This also prevents a single conversation from creating long forwarding chains as a result of inter-MTSO handoffs.

2. The AMPS system has a very limited capacity. Each carrier in a region is giver 25 MHz of spectrum to use for cellular telephony. An AMPS telephone uses a channel of 30 KHz for a voice conversation. This limits the total capacity of the system to 416

simultaneous channels in a cell. In fact, given the cell reuse patterns used by most AMPS systems 416 channels are never available in a single cell. A centralized control system can easily handle the routing load imposed by this number of channels.

2.2.2 VC Trees [Acam94]

The VC Tree solution aims to support a PCS environment in which the wired and wireless network use Asynchronous Transfer Mode (ATM) technology. The objective of the VC Tree approach is to reduce the number of router updates caused by handoffs. It does not attempt to solve many of the other problems created by mobility. The approach splits the wireless network into areas called *neighboring mobile access regions*. In each region, a single fixed switching node acts as the root of a tree. This root node is responsible for keeping track of the location of all mobile hosts within its region and routing ATM cells to them.

2.2.2.1 Connection Establishment / Data Routing

All connections to a mobile host pass through the root node of its current region. The root node assigns a set of virtual circuit numbers (VCN) when a mobile connection is created. Each VCN defines a path between the root node of the tree and one of the base stations in the region. The entire set of VCNs creates a tree, called the VC Tree, from the root node to all base station in the area.

When an ATM cell for a mobile host arrives at the root node, it uses the information about the mobile host’s location to forward the cell across the appropriate branch of the VC Tree. This delivers the cells to the base station currently serving the mobile host.

2.2.2.2 Connection Maintenance

When an MH wishes to route packets through another base station in the VC Tree, it simply transmits its ATM cells using the virtual connection ID that has been pre-assigned for communication between it and the new base station. When these cells reach the root node

of the VC Tree, it identifies that the cells have arrived on a different VCN and recognizes that a handoff has occurred. The root node updates its location information to reflect this handoff.

When an MH moves to another region, the system must build an entirely new VC tree. The regions are large enough such that this virtual connection tree handoff are infrequent.

2.2.2.3 Analysis

Most handoffs in the VC Tree approach require a single update at the root node of a region. As a result, the approach accomplishes its goal of reducing router updates. However, this approach does not solve many other problems and introduces new ones. These weaknesses include:

- The virtual connection tree handoff is a long and expensive procedure. The overhead of this form of handoff depends on the size of the defined regions. If the regions are large, the overhead is high but the handoffs are rare. Conversely, if the regions are small the overhead is lower but such handoffs occur more often.
- The approach does not attempt to eliminate data loss. While the root node is being notified, the data is incorrectly routed and never forwarded.
- The approach has possible scaling problems. There is a single root node per region. However to reduce inter-region handoffs, regions must be made as large as possible. As a result, the root node must process a large number of routing updates as well as route a large amount data.

2.2.3 Groups-Based Routing [Ghai94]

This approach attempts to solve the problem of connection-oriented MH to MH communication in a pico-cellular environment. The approach divides the wireless network into regions controlled by a supervisor host. In addition, the design puts the cells within a region into different groupings of likely handoff targets. A single cell may be part of sev-

eral of these groups. The approach assigns each of theses groups a unique IP multicast ID. This design introduces a three level hierarchy. This hierarchy consists of:

1. Mobile Hosts (MH) — The system requires each mobile host to maintain a large amount of state and receive numerous updates. Each MH keeps track of its previous mobile support station (MSS), current MSS and current supervisor host (SH). It also maintains the following for each active connection:

    - Sequence of last packet received
    - Virtual connection ID
    - Status — (1) connection active with other end of connection in the same region, (2) connection active with other end in different region, and (3) connection still in establishment phase

    For MHs in the same region, the source of the connection is responsible for routing its packets to the destination MH. As a result, it must receive updates whenever the destination MH moves. When MHs are in separate regions, the SHs handle this routing.

2. Mobile Support Stations (MSS) — A base station in this system is called an MSS. An MSS contains a static table of all groups to which it belongs. In addition, it joins IP multicast group associated with each of these MSS groupings. The MSS also maintains a list of all MHs that reside in any cell that belongs to one of its groups.

3. Supervisor Hosts (SH) — An SH is responsible for a region of the wireless network containing several MSSs. It performs the routing of packets from outside this region to the different MHs inside the area. To perform this routing it maintains several tables. The first table contains the list of MSSs in each group. The second table contains the current location and direction of motion of each MH in the region. This allows the SH to determine which group an MH belongs. The last table contains the information the SH needs to route data to MHs in other regions. For each connection to a remote MH, this table contains:

    - The virtual connection ID in its own region

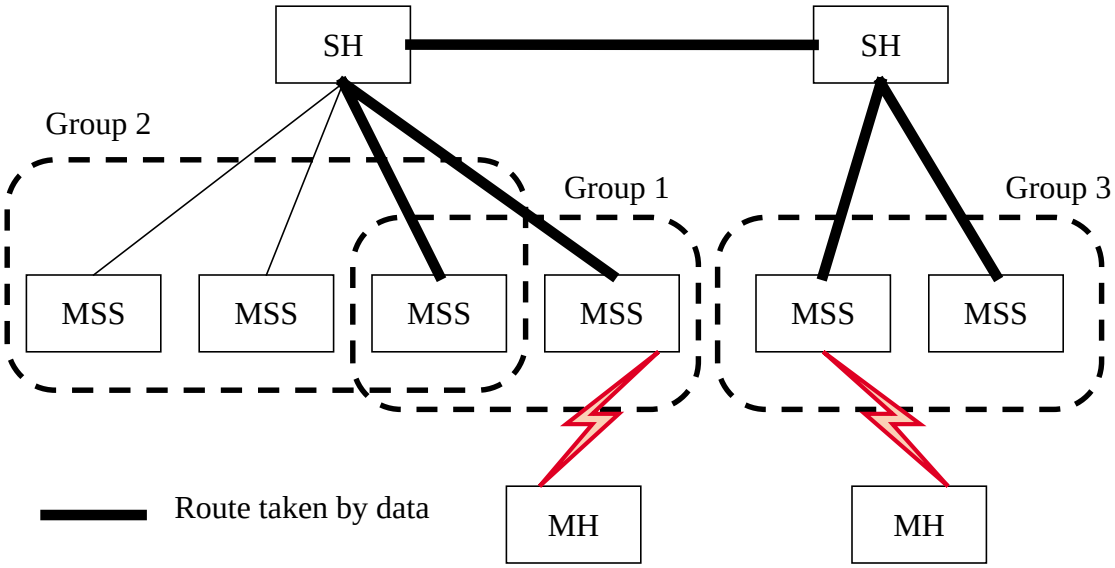


Figure 2.2 Groups-Based Routing

- The virtual connection ID in the remote region
- The id of the remote SH
- The group associated with the local MH

The interconnection of and routing between these entities is shown in Figure 2.2

2.2.3.1 Connection Establishment

When creating a connection to another MH, the source MH requests the SH in its area to find the destination MH. If the source and destination MHs are in the same region, the SH notifies the MH immediately. The MHs must connect directly to each other without the aid of the SH. As long as they remain in the same region, each must keep track of the others's current group. If the desired MH is not in the region, the SH broadcasts a request to all other SHs to search for the destination MH. Once the destination MH is found and its associated SH responds to the search request, the source SH creates a connection to the SH of the destination MH.

2.2.3.2 Data Routing

If the destination and source MH are in the same region, the source transmits packets directly to the destination's current group ID. This delivers the packet to the MSS currently responsible for the MH who forwards the packet. This transmission may also deliver the packet to several MSSs nearby the MH. These MSSs buffer the packet for some time and then discard it. If the source and destination MHs are in different regions, the source MH sends any packets to its SH. The source's SH uses its connection to the destination MH's SH to forward the packet. The destination's SH routes the packet by forwarding it to the MH's current multicast group. This results in the same delivery as above.

2.2.3.3 Connection Maintenance

When an MH moves within a region, it must update the SH and all corresponding MHs in the same region. The receivers of this information update their tables with the new group ID of the MH. This update comprises the handoff. When an MH moves between regions, it first transmits the ID of its old SH and the ID of all its connections. The new SH notifies the old SH of the movement and "adopts" all the active connections. The old SH adds a forwarding link to route incoming data to the SH of the new region.

2.2.3.4 Analysis

The groups based routing uses some interesting techniques to provide continuous communication to the mobile host. However, it has the following drawbacks:

1. Real-world wireless networks, especially in-building networks, are subject to very complicated propagation characteristics. This results in cells of all different shapes and sizes. Using velocity vector based handoff prediction to determine groups is not the best solution. A better solution is to use dynamic measurements at the mobile host or base stations to determine groupings.

2. Propagation characteristics in wireless networks and movement characteristics of users tend to be dynamic. The statically assigned groups do not provide adequate support for this dynamic nature.
3. Since groups memberships can not adjust dynamically, data may be multicast in a much wider area than is necessary for a mostly stationary user. This results in an unnecessary use of backbone network bandwidth.
4. During handoff, a roaming host must notify each MH in the area that it has connection to. Since communication has a high degree of physical locality, this policy results in a large number of updates.
5. The system assumes only MH to MH communication. This allows them to modify all sources of data. This is unlikely in many data networks.
6. The SHs use a broadcast-based search to find the mobile host. This is clearly non-scalable and unsatisfactory.

Only an analysis of the algorithms is presented. Although many of the concepts are interesting, some make unrealistically optimistic assumptions about real-world systems.

2.2.4 Summary of Connection-Oriented Routing

None of these connection-oriented systems meets all our goals for support of handoff. The cellular system design does not support high user densities or high handoff rates. As a result, it uses very centralized techniques to perform routing updates. The VC tree approach examines uses a similar centralized approach as AMPS to perform routing of multiple connections to a mobile host. However, the VC tree system performs optimizations to reduce the load on the connection establishment system. Despite this, it too is not extremely scalable. The Groups-based routing presents an interesting application of multicast to the problem. However, it makes a number of assumptions about wireless technology that are not necessarily valid.

2.3 Datagram-Based Network Routing [Perk95b, Myle93]

Much of the work in providing datagram routing support for mobile hosts has been performed as part of the Internet Engineering Task Force (IETF) working group developing the official mobile-IP standard. Mobile-IP is an extension to the existing IP [Stevens94, rfc1058, rfc1122] routing protocol to support delivery of packets to mobile hosts. The delivery of packets from mobile hosts to fixed location hosts is already dealt with by the standard IP routing. Each host in an IP network has a unique IP address and host name that are the network uses to route packets to this host. Routers in an IP network forward each packet based on the destination address in the packet header and a routing table containing the next network hop in the route to all different IP addresses. The fact that the IP-based Internet has a hierarchical structure allows routers to combine entries in a routing table. This combined with the fact that hosts usually remain in fixed locations result in small, infrequently updated routing tables on most routers. Unfortunately, as mobile hosts (MHs) become more popular these two characteristics will no longer be true. If IP networks use the standard routing protocols for mobile hosts, Internet routers will need to store routes to each mobile host and to update these routes in response to mobile handoffs. Clearly, the existing IP routing solution will not scale to large numbers of mobile hosts. Several proposals have been made to The IETF Mobile-IP group in the past. The objective of these proposed protocols is to deliver packets addressed to the mobile hosts “home” IP address to the mobile host’s current location. We will summarize these past proposals here as well as the current state of the Mobile-IP standard. We will describe the specific weakness of each protocols in the associated subsection. We will present weakness common to several protocol or mechanism at the end of the section.

2.3.1 Columbia MHP [Ioan91,Ioan93a,Ioan93b]

The basic technique used by Columbia MHP is the creation of a virtual mobile subnet. The scheme assigns each mobile host an IP address within this subnet. The mobile subnet routers (MSRs) advertise routes to this subnet to the rest of the network. This creates the illusion that the virtual mobile subnet is a real subnet connected to the rest of the network

with the various MSRs as gateways or routers. The MSRs also perform the function of a base station by routing packets between the wireless and wired networks. In addition, the MSRs keeps track of the MHs in its cell and caches the location of MHs in other cells.

2.3.1.1 Data Routing

When a packet is transmitted to a mobile host, the normal IP routing delivers it to the MSR nearest the source host. If the location of the MH is either the nearest MSR's cell, it delivers the packet directly. If the MSR has cached the location of the MH, it encapsulates the packet within another IP packet addressed for the MSR serving the destination MH and forwards it. However, if the MH's location is unknown, the MSR queries the other MSRs for the current location of the MH. Once a MSR responds, the nearby MSR caches the location information and forwards the packet.

2.3.1.2 Handoff

When an MH moves, it searches for a new MSR and registers with it. The MH also notifies its previous MSR of its new location. The previous MSR updates its MH location cache with this information.

2.3.1.3 Analysis

The above technique works well for campus area mobility. Since the protocol uses a broadcast search to find mobile hosts, it does not support a large number of MSRs. In addition, if the hit rate for the MSRs mobile host location cache is not high, a large number of broadcast searches may be necessary. Fortunately, most systems should experience a good hit rate due to the locality of source-destination communication.

To support wide area mobility, Columbia MHP defines the concept of popup operation. If the MH is out of its home region, it acquires a temporary address and registers it with one of the MSRs in its home campus. This MSR responds to all MH location searches and intercepts packets for the MH. It then encapsulates these packets and transmits them to the

MHs temporary address. This results in sub-optimal routing for MHs away from their home campus.

2.3.2 IBM [Perkins93a, Perkins93b, Rekhter93]

The IBM MHP relies on a machine, called the Mobile Router (MR), that resides on the subnet associated with the MH's home address. The MR is responsible for keeping track of the current location of its associated MH and performing the routing of packets. The MH must connect to and register with a base station (BAS) to have access to the remainder of the network.

2.3.2.1 Data Routing

Normal IP routing delivers a packet destined for an MH to its home network. When the mobile host is not at its home network, the MR responsible for the MH intercepts the packet. The MR uses its location information and the IP loose source routing (LSR) option to route the packet to the MH through the BAS serving it. In addition, RFC1122 [Brad89] specifies that if a host receives a packet with the LSR option it must use the reverse route for any replies. Therefore, the protocol requires the MHs to attach the LSR option to all packets transmitted. All return packets from a corresponding host will use the reversed LSR route instead of passing through the MR on the home network. This allows hosts corresponding with the MH to eventually obtain a more optimal route to the MH than the path through the home network.

2.3.2.2 Handoff

When a mobile host moves between cells it connects to a new BAS. It also notifies the previous BAS and its MR of its new location. The old BAS forwards any future packets to the new BAS. Eventually, new packets from the mobile host with LSR option will arrive at the different corresponding hosts. This will update the routes used and the old BAS will stop receiving packets for the MH.

2.3.2.3 Analysis

This proposal has many promising features. It provides a simple mechanism to provide optimal routing between corresponding and mobile hosts. It also prevents data loss, albeit with the introduction of additional delays, through the use of forwarding. Unfortunately, the shortcoming of this protocol is the poor implementation of the LSR option in most currently deployed routers and hosts. This prevents this protocol from being a viable option. As a result, the IBM group later submitted a modified version of this proposal that uses encapsulation and temporary addresses. In the modified protocol, the MR forwards the packets destined for the MH encapsulated inside packets addressed to the MH's temporary address. The BAS decapsulates the packet and delivers it to the MH. This modification eliminated some of the important features such as optimal routing from the protocol. In addition, other modifications have suggested splitting the MR into a Location Directory (LD), to store current MH locations, and a Redirector (RD), to perform the routing function of the MR.

2.3.3 Internet Packet Forwarding Protocol (IPTP) [Wada92, Wada93]

The IPFP proposal from Matsushita has many features in common with the IBM proposal. A basis of this approach is the introduction of special routers called Packet Forwarding Servers (PFS). A PFS host on the MH home subnet performs many of the same functions as the MR in the IBM proposal. In addition, all PFS also perform the functions of a base station (like the BAS in the IBM proposal) Like the MR, a home network PFS keeps track of its mobile host's current location. It also intercepts packets for this MH and forwards them to the MH's current location. The crucial difference between the IBM and IPTP approaches is the mechanism used to deliver the packet to the MH's current location. Whereas the IBM approach used loose source routing, the IPFP approach uses the concept of temporary address and encapsulation.

2.3.3.1 Data Routing

In addition to a home address, this scheme assigns each MH a temporary address that reflects its current location. This temporary address changes each time the MH changes location. To perform the necessary routing the MH must always inform the PFS of its current temporary address. When the home PFS receives a packet for the MH, the PFS encapsulates it within a packet destined to the current temporary address of the MH. In addition, the IPTP protocol supports the concept of route optimization. Any corresponding host may route packets directly to the MH by encapsulating packets and transmitting them to the MH's temporary address.

2.3.3.2 Handoff

During a handoff to a new PFS, an MH notifies the home PFS and its previous PFS of its new temporary address. The home PFS updates its encapsulation and the old PFS caches the MH's location to provide temporary forwarding service. In addition, all corresponding hosts using the route optimization must also discover the new temporary address. If a corresponding host transmits a packet to the old temporary address, the old PFS responds with an MH location update message. If the old PFS has the MH's current location cached, this message contains the new temporary address of the MH. Otherwise, the old PFS sends the home address of the MH.

2.3.3.3 Analysis

The use of temporary addresses for routing from the home network creates several advantages and disadvantages in comparison to loose source routing. Each network that supports mobility must allocate enough temporary address as needed to support the maximum number of MHs expected. This may consume a large number of IP address in the already crowded IPv4 address space. However, the use of temporary addresses provides a simple mechanism to perform the same function as LSR while avoiding the lack of support for LSR in routers.

2.3.4 CMU [John93b, John93a]

A protocol proposed by Johnson also uses the concept of a special host on the MH home subnet. The CMU proposal calls this host the Location Server (LS). Like the home agents in the other proposals, the LS also tracks the current location of the MH while it is outside the home area. When an MH is roaming outside its home area, it must find a local BS willing to accept its packets. Each MH must continuously provide the LS with its current BS's address. The CMU approach also introduces the concept of a Location Cache (LC). A location cache is any host or router that keeps track of a mobile host's current location. An LC uses this information to perform route optimization.

2.3.4.1 Data Routing

The data packets are routed towards the home network of the mobile host as in the IBM and IPTP approach. Instead of always being delivered to the LS on the home network, the first LC encountered that knows the location of the MH intercepts the packet. Either the LC or LS encapsulate the IP and transmit it to the MH current location. Instead of using temporary addresses or LSR, the CMU approach encapsulates the packets and transmits them the BS responsible for the MH. The BS decapsulates the packets and forwards them across the wireless network to the mobile host.

2.3.4.2 Handoff

When a mobile host connects to a new BS, it notifies its LS and previous BS. The previous BS uses the information to forward any packets that may arrive. The scheme uses a separate mechanism to update the contents of the different LCs in the network. When a packet passes through an LC with old information, the LC incorrectly forwards the packet to a previous BS. If this previous BS still has the MH's current location cached, it forwards the packet to the new BS. Otherwise, it forwards the packet back to the home network. The encapsulated packet header records the address of the source LC. The new BS or home LC transmit an ICMP message to the originating LC to update the MH information.

2.3.4.3 Analysis

By not using temporary addresses or LSR, the CMU proposal avoids the two major problems with the other similar approaches. In addition, the LC concept provides a mechanism to improve the efficiency of routes between any Internet host and the mobile host. However, the LCs in the Internet would require very large tables since they store routes to each individual MH.

2.3.5 Sony [Teraoka91, Teraoka93]

The Sony proposal presents a more aggressive approach to incorporating mobile hosts into the Internet. This approach assigns each MH a pair of IP addresses, a virtual network (VIP) address and a physical network (PIP) address. The PIP address corresponds to the current location of the mobile host. The VIP address does not change over time. Upper layer protocols and applications use the VIP address to communicate with the MH. The mobile host keeps a machine on its home network, called the home gateway (HG), notified of its current PIP address. Each router and gateway in the network maintains a cache of VIP-PIP mappings. To update these caches, the Sony protocol uses a technique called the *Propagating Cache Method*. Since the current VIP-PIP mapping of the source host for a packet is include in the header, the routers and hosts in the network can update their cache by snooping on packets transmitted by the mobile host.

2.3.5.1 Data Routing

As a packet passes through the network, the intermediate routers examine it to identify VIP address they have cached. If the intercepting router has a more recent VIP-PIP mapping for the MH than the one in the packet, it redirects the packet to the proper location by changing the mapping. If no router enroute to the home network has a recent VIP-PIP mapping, the packet propagates to the HG. The HG always has a valid mapping for the MH.

2.3.5.2 Handoff

When a mobile host moves to a new cell, it sends a notification to its home network. This updates the cached VIP-PIP mappings in the HG and each of the routers between the MH's current location and home network. Similarly, when an MH disconnects from the network, it attempts to delete cache entries for its VIP address. A disconnect notification that partially floods through the network performs the cache update. In addition, timeouts eventually invalidate entries that are not deleted.

2.3.5.3 Analysis

The Sony approach has many significant weaknesses. The use of temporary addresses forces each region to allocate enough addresses for the maximum expected density of users. Like any other scheme using temporary addresses, this results in an inefficient use of IP addresses. The mechanism used to perform handoff assumes that the control messages are not lost. Any loss of a handoff control message can result in long periods during which inefficient or incorrect routing results. These problems prevent this approach from being a viable alternative.

2.3.6 IETF Mobile IP Draft 12 [Perk95b]

The IETF Mobile IP protocol combines many of the concepts that were put forth in the IBM, CMU and IPTP proposals. Like these previous proposals, the IETF protocol uses the concept of a home agent (HA) that resides on the home network of the MH. The home agent keeps track of a care-of address for each MH. This care-of address may either be associated with the mobile host (i.e., a temporary address) or a foreign agent (i.e., a base station). When the MH is away from its home, the HA intercepts packets for it and forwards them to the MH's care-of address.

2.3.6.1 Data Routing

When a packet is transmitted to a mobile host, the network delivers it to the home network where a home agent intercepts it. The home agent uses IP-in-IP encapsulation [Perk95a] to forward the packet to the current care-of address for the MH. If the care-of address corresponds to a foreign agent (FA), it must decapsulate the packet and deliver it to the MH. If the care-of address is the MH itself, the normal IP routing delivers the packet to the MH.

2.3.6.2 Handoff

When a mobile host moves, its care-of address changes. The MH must simply notify the home agent of the new address to update the routing. Since the protocol does not use forwarding links, packets delivered to the previous care-of address are dropped. As a result, a handoff causes any packets “in-flight” to be lost. For reliable communications, a higher-layer protocol must retransmit any lost packets.

2.3.6.3 Analysis

The IETF draft allows much adaptability in the specifics of implementation. For example, the protocol support the use of either pure encapsulation or temporary addresses to deliver packets from the home network to the MH. To maintain backward compatibility with current routers, the protocol does not use concepts such as location caches in routers. This prevents the development of a simple mechanism to optimize routes from unmodified Internet hosts. A separate IETF draft presents mechanisms to add forwarding from previous foreign agents and route optimization from modified Internet hosts [John95]. To add this support, foreign agents and corresponding hosts must maintain a binding cache containing the home and care-of address for mobile hosts. In addition, these hosts must encapsulate and transmit packets directly to the MH’s care-of address. When a mobile host obtains a new care of address, it notifies its previous foreign agent. The old foreign agent updates its binding cache and forwards any future packets. To support route optimization, the home agent deduces that the source of any packet it intercepts does not have a binding cache entry. The home agent may then transmit a binding update to the source of the

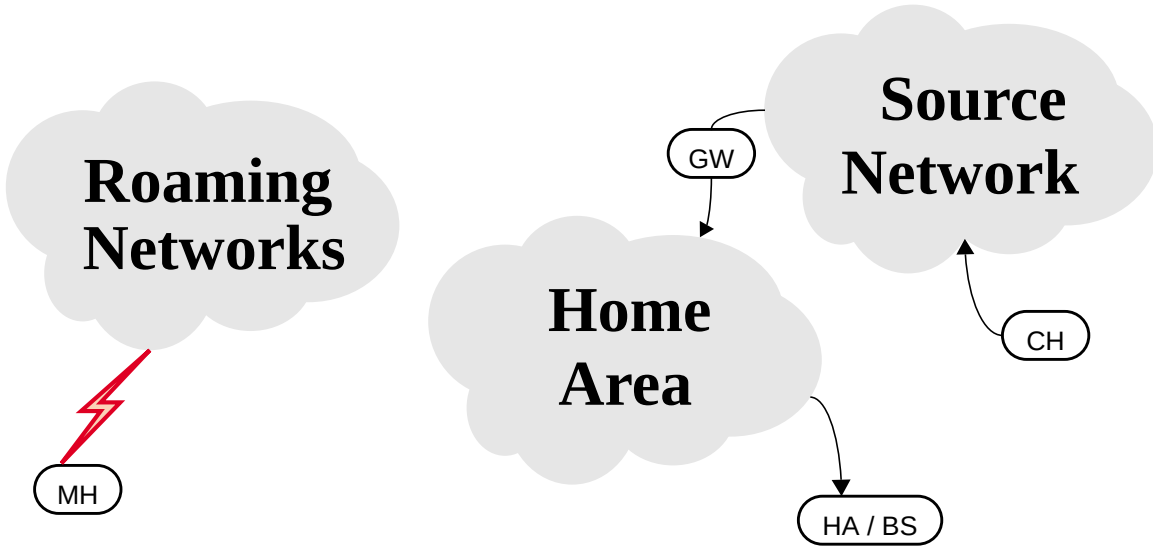


Figure 2.3 First Stage of Various Mobile-IP Protocols

> The protocols route packets to a special location that stores the current location of the mobile host. This location may be any nearby BS, a home agent, or a gateway in the network.

packet. If the source of the packet supports route optimization, it updates its binding cache and routes future packets to the care-of address. After a handoff occurs, packets from a route optimizing host trigger binding warning messages. This causes the corresponding host to request a new binding update from the home agent. This form of route optimization is very complicated and only supports slow updates.

2.3.7 Datagram Routing Summary and Analysis

In general, the basic routing used by the different Mobile IP proposals consists of two stages. In the first stage, shown in Figure 2.3, the protocols route packets to a special location that stores the current location of the mobile host. This location may be any nearby BS, a home agent, or a gateway in the network. The second stage of the routing, shown in Figure 2.4, uses the information in the special machine to deliver the packet to either the MH or a machine near the MH. All routing protocols that use this form of routing suffer from a problem known as triangle routing (Figure 2.5). The route of packets from the cor-

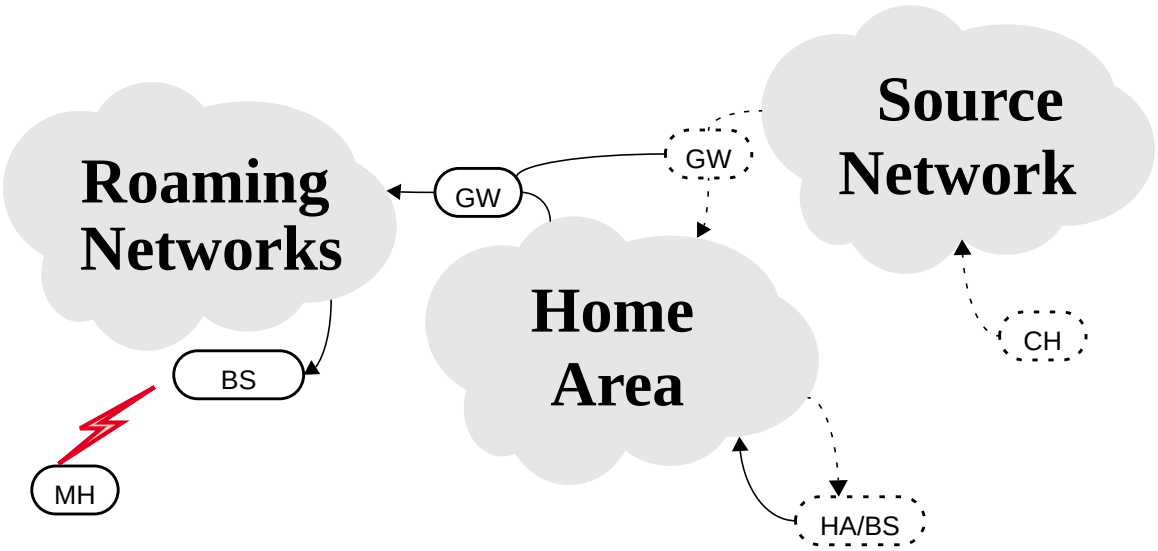


Figure 2.4 Second Stage of Various Mobile IP Routing

The various special machines use the location information to deliver packets to either the MH or a machine near the MH

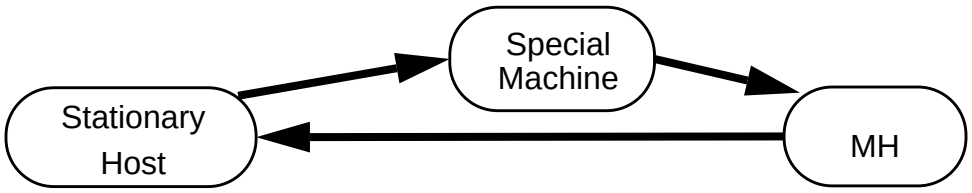


Figure 2.5 Triangle Routing

responding host to the mobile host must pass through the special machine. If the special machine does not reside on the optimal path between the two, an inefficient routing occurs. The different proposals use different techniques to avoid triangle routing. The specifics of the special machine, mechanism to deliver packets to the MH's location, and route optimization are summarized for the different protocols in Table 2.1.

In most of the proposals, the mobile host can not receive packets until it contacts the old base station through the new base station. During this period, the old base station discards any packets destined for the mobile host. Contacting the old base station initializes the for-

Proposal	Special Machine	Normal MH Address	Final Delivery Method	Optimal Routing	Weakness
Columbia (Sept. 91)	every BS	special subnet	encapsulation	no	poor wide area support
Sony (Sept. 91)	all network routers	home network address	temporary address	by router caching	poor router cache updates
IBM I (July 92)	home network	home network address	loose source routing	via reversal of LSR	poor LSR support in Internet
Matsushita (Nov. 92)	home network	home network address	temporary address	by CH	inefficient use of IP addresses
IBM II (Jan 93)	home network	home network address	temporary address	no	inefficient use of IP addresses
CMU (July 93)	home network	home network address	encapsulation	by location caches	non-scalable location caches
IETF	home network	home network address	encap. (can be temp addr)	optional - by CH	optional forwarding

Table 2.1 Summary of Mobile IP Proposals

warding of packets to the mobile host. While packets are being forwarded, they experience additional delays due to a longer route. Eventually, the protocol notifies machines on the home network and corresponding hosts of the mobile host's new location. At this point, the handoff completes, the routing of data stabilizes and end-to-end performance

returns to normal. When the route switches between forwarding and the final route, the mobile host is likely to receive a few out-of-order packets. This form of routing updates is unsatisfactory for many Internet applications. Multimedia applications and reliable protocol like TCP require mostly in-order delivery and reasonably consistent end-to-end performance. Violations of these assumptions will not result in failures but severely degraded performance. This form of routing was developed because the Mobile IP proposals did not want to use information from lower layers of the network stack. In addition, the several proposals mainly aimed to support users who use wired network interfaces while mobile (i.e., a user who visits another location and plugs into a network). We believe most mobile users will use wireless interfaces and that the datagram routing algorithms should take advantage of information given by the typical wireless network.

2.4 State Transfer During Handoff

When a mobile host moves between cells, the new cell's base station must take over all responsibilities of the previous base station. The most important of these responsibilities is to route packets destined for the mobile host. As a result, much of the previous work in supporting mobility has concentrated on the updating of routes in networks. However, a handoff protocol must also transfer other responsibilities, such as retransmitting lost packets and providing QOS guarantees across the wireless link, to the new base station. The transfer of responsibilities may require the transfer of a significant amount of state to the new base station. For example, to perform retransmissions of lost packets, a base station may maintain a copy of each packet that has not been acknowledged. For retransmissions to continue after a handoff, this cache of unacknowledged packets must also exist on the new base station. The system must transfer this state quickly and efficiently to avoid adding significant delay to the handoff process. One system that performs this state transfer is I-TCP.

Reliable transport protocols such as TCP [Post81b, Stev94, Brad89] have been tuned for traditional networks made up of wired links and stationary hosts. TCP performs very well

on such networks by adapting to end-to-end delays and packet losses caused by congestion. It provides reliability by maintaining a running average of estimated round-trip delay and mean deviation, and by retransmitting any packet whose acknowledgment is not received within four times the deviation from the average. Due to the relatively low bit-error rates over wired networks, all packet losses are correctly assumed to be caused by congestion.

In the presence of the high bit-error rates, intermittent connectivity and handoffs characteristic of wireless environments, TCP reacts to packet losses as it would in the wired environment: it drops its transmission window size before retransmitting packets, initiates congestion control or avoidance mechanisms (e.g., slow start [Jaco88]) and resets its retransmission timer (Karn's Algorithm [Karn87]). These measures result in an unnecessary reduction in the link's bandwidth utilization, thereby causing a significant degradation in performance in the form of poor throughput and very high interactive delays [Cace94]. Recently, several reliable transport-layer protocols for networks with wireless links have been proposed [Bakr94, Bakr95, Cace94, Yava94] to alleviate the poor end-to-end performance of unmodified TCP in the wireless medium. These proposals fall into two classes of approaches: the split-connection approaches and the link-layer reliability approaches.

2.4.1 The Split Connection Approaches [Badrinat93, Bakre95]

The Indirect-TCP (I-TCP) protocol [Bakr94, Bakr95] was one of the first protocols to use this method. It involves splitting a TCP connection between a fixed and mobile host into two separate connections at the base station — one TCP connection between the fixed host and the base station, and the other between the base station and the mobile host. Since the second connection is over a one-hop wireless link, there is no need to use TCP on this link. Rather, a more optimized wireless link-specific protocol tuned for better performance can be used [Yava94]. During a handoff, the I-TCP connections must move from the base sta-

tion currently forwarding data to another one. This is done by transferring the state associated with the two connections to the new base station.

The advantage of the split connection approach is that it achieves a separation of flow and congestion control of the wireless link from that of the fixed network and hence results in good bandwidth at the sender. However, this design results in very poor handoff performance. Since a base station acknowledges all packets to the sender, any packets not yet delivered to the mobile host and other TCP state it has is “hard” state. There can be no errors in this state and during a handoff it must be transferred reliably to the new base station. The state maintained at a base station in I-TCP consists mainly of a set of socket buffers. Since this state must be moved to another base station during handoff, the latency of handoff is proportional to the size of these socket buffers.

2.4.2 Link-level Retransmissions [Paul95]:

In this approach, the wireless link implements a retransmission protocol coupled with forward error correction at the data-link level. This approach improves the reliability of communication independent of the higher-level protocol. If a link-layer protocol attempts to provide complete reliability a significant amount of state may need to be moved during handoff. However, even a link layer that does not provide complete reliability must transfer some state in order to provide any improvement to reliability after a handoff.

2.4.3 State Transfer

In summary, the existing approaches to combat high-bit error rates maintain a large amount of “hard” state at the base station. This makes it difficult to perform handoff quickly in systems using these techniques. During handoffs, these protocols must move any packets that have not been acknowledged to the new base station. Among these systems, only I-TCP currently has handoff support that is described in the literature.

During a handoff, the I-TCP system moves its connections from the base station currently forwarding data to another one. This is done by transferring the state associated with the

two “split” connections to the new base station. The method used to perform the state transfer is to encapsulate and forward the state to the new base station after a mobile host has entered a new cell. The duration of this transfer is directly related to the amount of state present in a base station and inversely related to the wired network bandwidth available between nearby base stations.

The state maintained at a base station in I-TCP consists mainly of packets unacknowledged by the mobile host and packets not yet transmitted across the wireless network. The number of such packets is proportional to the TCP socket buffer size used. Since this state must be moved to another base station during handoff, the latency of handoff is proportional to the size of these socket buffers. The I-TCP handoffs range from 265 ms for empty socket buffers to 1430 ms for 32KByte socket buffers [Bakr95].

Clearly, this is an unsatisfactory solution. We must explore alternative mechanisms to perform this state transfer more quickly.

2.5 Summary of Related Work

Unfortunately, prior work in this area has not provided a scalable solution that addresses the issue of providing consistent end-to-end performance.

Among routing protocols, much has been accomplished in the context of providing support for the Internet protocol suite in a wireless environment [Perk95b]. However, these systems tend to have significantly degraded performance during handoffs. Conversely, the cellular telephone network uses circuit switching to provide connection-oriented services. It maintains these connections during handoffs through highly centralized knowledge and control by the mobile telephone switching offices. Other routing systems, such as VC Trees and Group-based routing, also do not meet our requirements for a handoff protocol. The VC Tree approach does not scale well and the Group-based approach makes unrealistic assumptions about the wireless network.

In addition, very little work has addressed the problem of distributing non-routing state during a handoff. The systems that do attempt to perform this function use the naive approach of isolating all state and transferring it at handoff time. This provides unsatisfactory handoff performance.

In order to support existing protocols and multimedia applications, we need handoffs that complete quickly and do not affect end-to-end performance. As part of exploring support for this low-latency handoff, we present alternative routing update algorithms for connection-oriented and datagram-based networks. In addition, we also describe methods to perform the state transfer quickly and in advance of the actual user movement between wireless cells.

Chapter 3

Analysis of Connection-Oriented Rerouting

The important first steps in developing low-latency handoff schemes were our examination of the various possible rerouting algorithms and quantifying their differences. This chapter presents an analysis of three important classes of rerouting algorithms, Full Re-establishment, Incremental Re-establishment and Multicast-based Re-establishment, in a generic switch-based connection-oriented network.

3.1 Introduction

The primary goal of the low-latency handoff is to support the transport of multimedia to mobile hosts. Multimedia applications typically have strict constraints on network parameters such as delay, delay jitter, throughput, and reliability bounds. Most real-time network services provide guarantees on these performance parameters to applications that request them. In order to provide these performance guarantees to individual conversations, approaches such as the Tenet Real-Time Protocol Suite [Ferr92] rely on connection-oriented networks and resource reservation. In this chapter, we examine and analyze various methods of supporting handoff in protocols such as Tenet.

These protocols use per-connection resource allocation in a connection-oriented network to ensure that guarantees will not be violated under conditions of heavy network congestion. In connectionless network environments, congestion typically causes decreased throughput, increased delay, and even an increased probability of packet loss. These methods, which are designed for wired networks, do not directly address the mobility of hosts. To support mobile hosts, these protocols would have to support rapid re-routing of connections in progress. Although the performance guarantees are necessary for the delivery of multimedia data to mobile hosts, the primary concern of this work is not providing these guarantees, but rather maintaining the connections in use as a host moves within the network.

The effectiveness of the rerouting performed during handoff processing is measured by several criteria. In particular, it is desirable to minimize the service disruptions (such as delayed arrival or loss of packets) and overheads (such as latency, MH and BS buffering, and excess reservation of network resources). Two straightforward solutions to the challenge of connection-oriented rerouting are forwarding data from the original BS to the new BS and establishing a new connection between the multimedia source and the new BS. These approaches incur considerable overheads: the connection re-establishment overhead is proportional to the distance between the source of data and the BS and the forwarding overhead is proportional to the range of movement by the user. We propose two lower overhead algorithms that modify the existing connections by partially re-establishing them to perform the rerouting. This localizes the rerouting to near the mobile host. One protocol capitalizes on the logical locality of geographically adjacent cells to partially re-establish the connections during handoff. The other uses multicast facilities to provide connectivity to the new base station when the host moves. Since many wireless networks provide advance warning of handoff, we also examine the use of information or “hints” from the wireless network to aid handoff processing.

In this chapter, we present a quantitative comparison of these strategies using analytically derived formulas for connection-setup latency, required buffering, and excess resource

reservation. We compare the algorithms and evaluate their feasibility, given the physical limitations imposed by current wireless network technology. Our basis for comparison is the case in which connections are fully re-established to the source. Our analysis indicates that the two new schemes can provide significantly better rerouting performance. In particular, the multicast-based algorithm is especially promising. It completes the rerouting more quickly, requires less buffering in base stations and uses a similar amount of network resources as the other algorithms. We also found that "hints" from the wireless network about impending handoffs improve the performance of rerouting considerably. These hints provide a mechanism for trading-off wired network bandwidth for reduced handoff latency.

The remainder of this chapter is organized as follows. Section 3.2 describes the assumptions we make about the environment for our analysis. We introduce the different algorithms in Section 3.3. In Section 3.4, we analyze and compare the performance of the various algorithms for rerouting. In Section 3.5, we examine the implications the rerouting algorithm performance has on the layout of base stations in a network. We present our conclusions about these rerouting algorithms in Section 3.6.

3.2 Environmental Assumptions

The physical and logical aspects of the networking environment play a large role in determining the nature and performance of the rerouting algorithms and their requirements. In this section, we examine the assumptions we make in our analysis about portable computers, wireless networks, and routing protocols.

As mentioned in Section 1.2, typical mobile hosts are much less capable than their desktop counterparts. However, in this analysis we assume that the mobile hosts contain enough processing power to run applications and the necessary communication protocols. We also assume that the MHs will support various multimedia services such as the playback of audio and video [Shen92].

The wireless and wired network technologies available today define various aspects of mobile communication. We assume a nano-cellular radio network (with cells less than ten meters in diameter) covering the interior of a building. Based on current radio technology (Section 1.1), we make the following assumptions:

- Each MH can communicate with at most one BS at a time.
- There is generally considerable overlap between the cells of the network.
- Each host in a radio network knows which BSs are "in range" and the radio signal strength from those BSs.
- The strength of the radio signal strengths gives the MH an indication of when it is close to a new cell. In non-radio wireless and in some radio networks, other techniques to obtain these hints are used. These techniques are described in Section 1.4.

Finally, we assume that the BSs can be programmed to support our rerouting algorithms. Due to these characteristics of wireless networks, our approach uses information about an impending cell transition as a hint to improve performance, rather than as an integral component of the handoff algorithms.

Many multimedia services, such as audio-video conferencing or video playback, have associated with them performance requirements that must be met to guarantee acceptable service to the users. [Ferr90a] describes the requirements that some typical applications place on networks. The Tenet Real-Time Protocol Suite [Ferr92] is one approach to providing these *real-time performance guarantees* in packet-switching networks. This approach relies on resource reservation in a connection-oriented network environment. The Tenet suite is used as the basis of the wired network for our analysis.

Before communication can take place in the Tenet system, the involved hosts must establish a real-time channel (a connection with performance guarantees). Channel establishment involves a single source-routed round-trip pass of control messages from the source to the destination of the connection and back.[1] On the forward pass, the network performs

admission control tests to ensure that it will be able to satisfy this channel's performance requirements without violating the guarantees of existing real-time channels. Assuming the tests succeed, the nodes tentatively allocate resources (such as bandwidth), pending the acceptance of this channel at all other participating nodes. On the return pass of a successful establishment, the nodes commit the resources to the channel. [Mah93] describes in detail a mechanism and protocol by which real-time channels can be established and managed.

We also assume that all connections have at most one MH as an endpoint, i.e., at least one endpoint is a fixed host on the wired network. This assumption somewhat simplifies the design of the algorithms, as they do not need to consider the case where both ends of a connection move simultaneously.

The algorithms that we present modify existing connections to perform rerouting. We call the point where a connection is modified the *crossover point* (see Figure 3.1). Choosing this crossover point requires some knowledge of the network topology. We assume, however, that control entities have only local knowledge of the network, such as next-hop routing information; this allows them to function without needing large status updates from their peers. This assumption also prevents a single entity from making rerouting decisions independently. Therefore, multiple control entities must collaborate using a distributed algorithm to determine the crossover point.

3.3 Algorithms

In this section we discuss three different schemes for rerouting in connection-oriented mobile networks. We present a Full Re-Establishment scheme as a basis for comparison. We then introduce two new algorithms, Incremental Re-Establishment and Multicast-Based Re-Establishment. The algorithms all rely on a number of pre-existing connections

1.The Tenet real-time channels are simplex unicast, but it is a fairly simple matter to support duplex real-time channels. Ongoing work is aimed at providing real-time multicast network services.

in the network, to be used primarily for control purposes. We assume that any two network nodes connected by a link have a control channel between them. Every pair of base stations responsible for physically adjacent cells also has a control connection between them. (We note that these base stations may not be directly connected by a physical link.) In addition, these algorithms require the BSs to buffer a bounded amount of data that has already been transmitted to the mobile host. This ensures that no data loss occurs when the mobile host moves to an adjacent cell. The algorithms make no attempt to re-order out-of-order packets. They will, however, attempt to prevent data from becoming out of order as the result of a handoff operation.

For each algorithm, we give a short description, followed by a brief example of its application re-rerouting a single channel. We also present the optimized case which takes advantage of the cell overlap hint. In each of the examples, the MH moves from the cell whose point of control is BS 1 into the cell corresponding to BS 2. Thus, in all cases, BS 1 is the “old BS” and BS 2 is the “new BS”. The point at which the connections to the old and new BS diverge is known as the crossover point. The location of this crossover node significantly affects the rerouting algorithm performance. Figure 3.2 shows the before and after situation of a handoff. The examples use a tree topology to simplify the description of each algorithm. We also only present examples where a hint is available from one adjacent BS. The algorithms also support situations where handoffs to multiple cells are predicted. Numbered lines correspond to control messages exchanged between network entities to complete the rerouting.

3.3.1 Full Re-Establishment

The Full Re-Establishment (FR) algorithm executes rerouting by establishing a set of completely new channels between the MH and the servers. In this algorithm, the time to reroute a connection is proportional to the number of network hops between the MH and server. Since this distance is not easily bounded, we expect this algorithm to perform

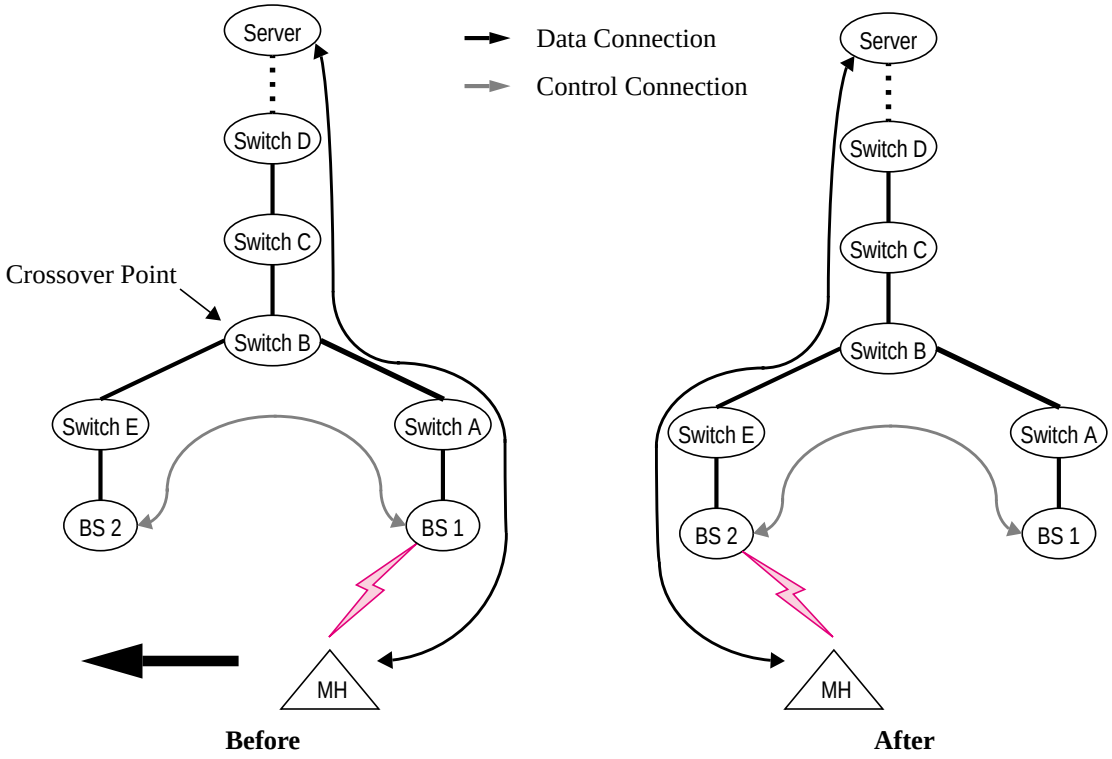


Figure 3.1 Rerouting Examples

poorly. We use it for comparison by identifying the performance the other algorithms gain by modifying only part of the connection.

3.3.1.1 Full Re-Establishment without Hints

The “Without Hints” part of Figure 3.2 depicts the communication necessary to perform the FR scheme in the general case where the MH has no advanced warning that a handoff from BS 1 to BS 2 is about to occur. Once the MH enters the new cell, it identifies itself to the new cell’s BS, BS 2, and requests that its connections be rerouted through the new BS (message labeled “1” in the diagram). Included in this greeting message is an identifier for the old BS, BS 1, as well as a list of the identifiers for the various connections originating or terminating on the MH. After BS 2 acknowledges this greeting with message (2), data transmission from the MH may begin.

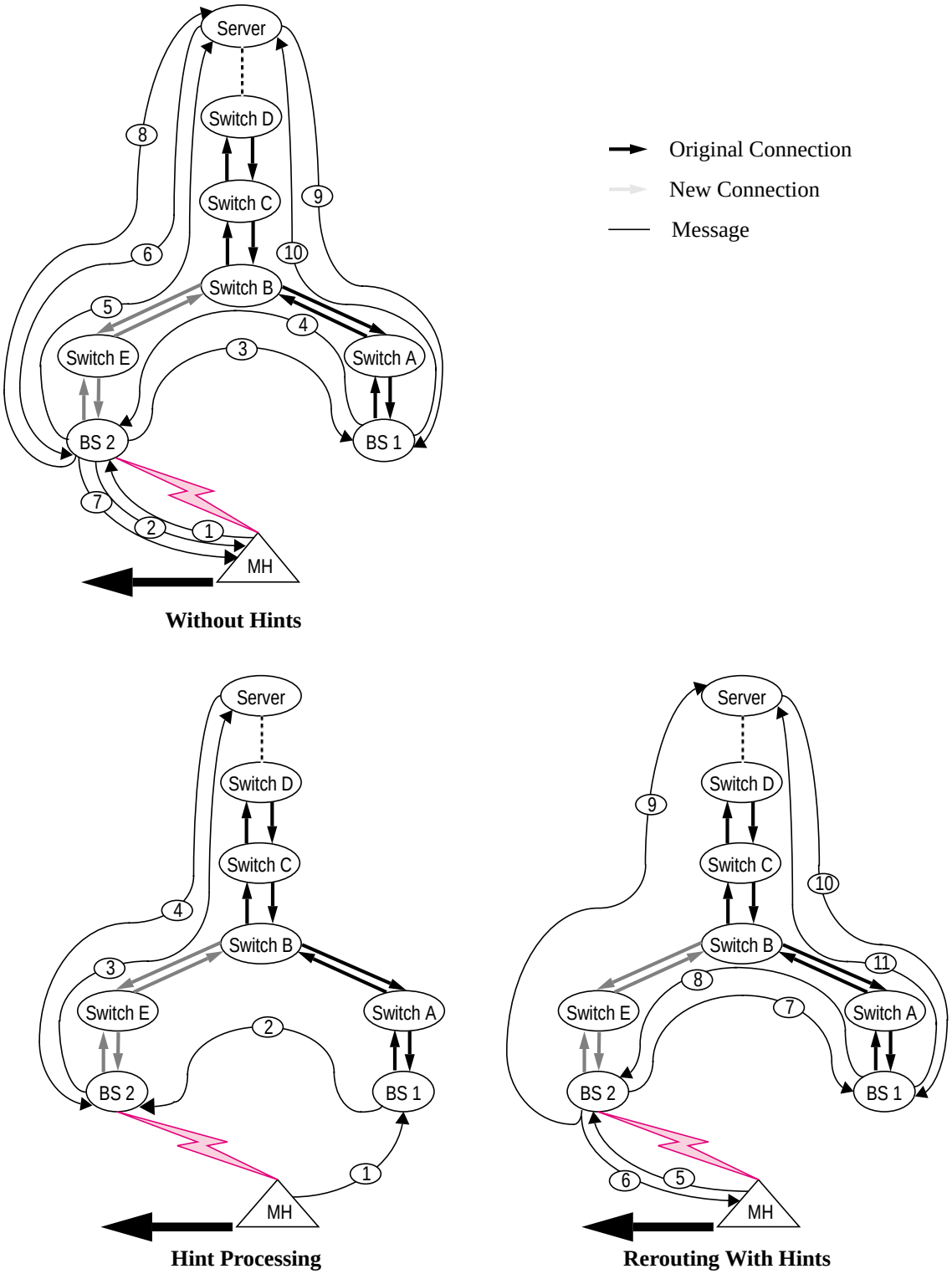


Figure 3.2 Full Re-Establishment

In order to avoid an interruption in data service, BS 2 immediately uses the pre-existing BS-to-BS control channel to request that downlink data from each of the MH's connections be forwarded from the original BS (3). Message (3) also requests that BS 2 be allowed to forward uplink data from the MH through BS 1 to the Server. BS 1 acknowledges these requests, and may begin forwarding data to BS 2 just after (4). Once (4) has been received, BS 2 may begin forwarding data transmitted by the MH through BS 1.

While this data forwarding is occurring, BS 2 begins establishing connections to each of the appropriate Servers on the network (5). Once a connection is fully established (6), the new BS informs the MH (7) and sends a message (8) to the Server requesting that it redirects all data transmissions down the newly established downlink. Step (9) then tears down the downlink portion of the connection between Switch B and BS 1. BS2 buffers the redirected downlink data, and the remainder of the data forwarded through BS 1 to the new BS. This ensures in-order delivery of data to the MH.

Once the new portion of the connection has been established, data from the MH can flow directly to the Server over the new connection. To preserve in-order delivery of the data from the MH to the Server, BS2 initially buffers this uplink data until the uplink data forwarded through BS 1 drains from the old portion of the connection. This buffering is proportional to the difference between the transmission delays along the new path and the forwarding or old path. Once all messages forwarded through BS 1 have been delivered to the Server, the uplink portion of the old connection is torn down (10) and uplink data may flow directly through BS 2. The rerouting is completed once all of the new connections have been created and the old connections destroyed.

3.3.1.2 Full Re-Establishment with Hints

In a radio network, the MH can take advantage of the overlap between adjacent cells to detect that it is entering a new cell before it loses contact with its current BS. This "hint" allows the MH to request that the current BS establish in advance new connections between the Servers and the new BS. This scenario is shown in the "With Hints" portion

of Figure 3.2. At the hint time, the MH requests (1) that its current BS (BS 1) send a list (2) of active connections to the new BS (BS 2). The new BS may then establish each of the connections to the appropriate network Server (3). A successful establishment is indicated to BS 2 by the acknowledgment in step (4).

During this pre-establishment, the MH ceases communications with the old BS and begins communicating with the new BS (5). As in the more general algorithm, this greeting is then acknowledged (6). Depending on how much time has elapsed, the new connections may or may not have been established to the new BS. In the best case, establishment for each of the new connections has been completed, and message (6) also notifies the MH of which connections have been established. As in the more general algorithm without hints, BS 2 then initiates a forwarding request for all of the data transmitted during the radio communication switchover period on a given connection from the Server to BS 1 (7, 8). Downlink data can then be forwarded across the pre-existing channel by BS 1. However, the uplink data is not forwarded, but sent only on the newly established connection to the Server. To ensure in-order delivery of the data from the MH to the Server, this uplink data must be initially buffered at BS 2 until all of the data that was transmitted by the MH (while in the old cell) can be sent along the old connection. This buffering is proportional to the difference between the transmission delays along the two paths, taking into consideration the time for the MH to make contact with the new BS.

While downlink data is being forwarded, the new BS sends a message (9) to the server requesting that data be rerouted to the new connection. Once the Server switches active connections, it begins deleting the channel to the old BS (10). After all uplink data has been transported to the Server, the old connection is completely torn down (11).

If the MH arrives at the new BS before its connections have been established (but after the hint processing has begun), forwarding for both the down and uplink is initiated as in the general case with no hints. The MH is then notified as each connection is established. At this point, the new BS sends a message to the Server indicating that it may begin sending

downlink data over the newly established channel. The teardown of the old channel proceeds as in the general case with no hints.

If the MH does not end up fully entering the cell it was approaching but instead eventually moves out of its range, it must send another message to its original BS revoking the hint. This results in a tear-down of the connections that had been set up in anticipation of the arrival of the MH. Such incorrect hints result in the temporary allocation of resources that are never used to perform a handoff.

3.3.2 Incremental Re-Establishment

In contrast to the FR algorithm, the Incremental Re-Establishment (IR) scheme attempts to re-use as much of an existing connection as possible, creating only the portion between the crossover point and the new cell's BS. The corresponding portion of the original connection is then torn down. We expect the performance of the IR scheme to be directly related to the distance to the crossover point and therefore the distance between base stations. As a result, this scheme should be much more scalable than the Full Re-establishment scheme.

3.3.2.1 Incremental Re-Establishment without Hints

The "Without Hints" portion of Figure 3.3 illustrates the application of the IR scheme in the general case where the MH has no advanced warning of the impending handoff. When the MH first arrives in a new cell, it sends a greeting message (1) to the new BS, BS 2, which contains a list of connections to be rerouted as well as the identity of the old BS. Data transmission from the MH to BS 2 begins after BS 2 acknowledges this greeting (2). As in the FR scheme, BS 2 requests that BS 1 forward the downlink and uplink data from each of the MH's connections across the BS-to-BS connection (3, 4).

In order to reuse a portion of the existing connection, message (3) also implicitly requests on behalf of the new BS that the old BS invoke the distributed crossover point location process. This location algorithm is initiated by the old BS because the new BS has no

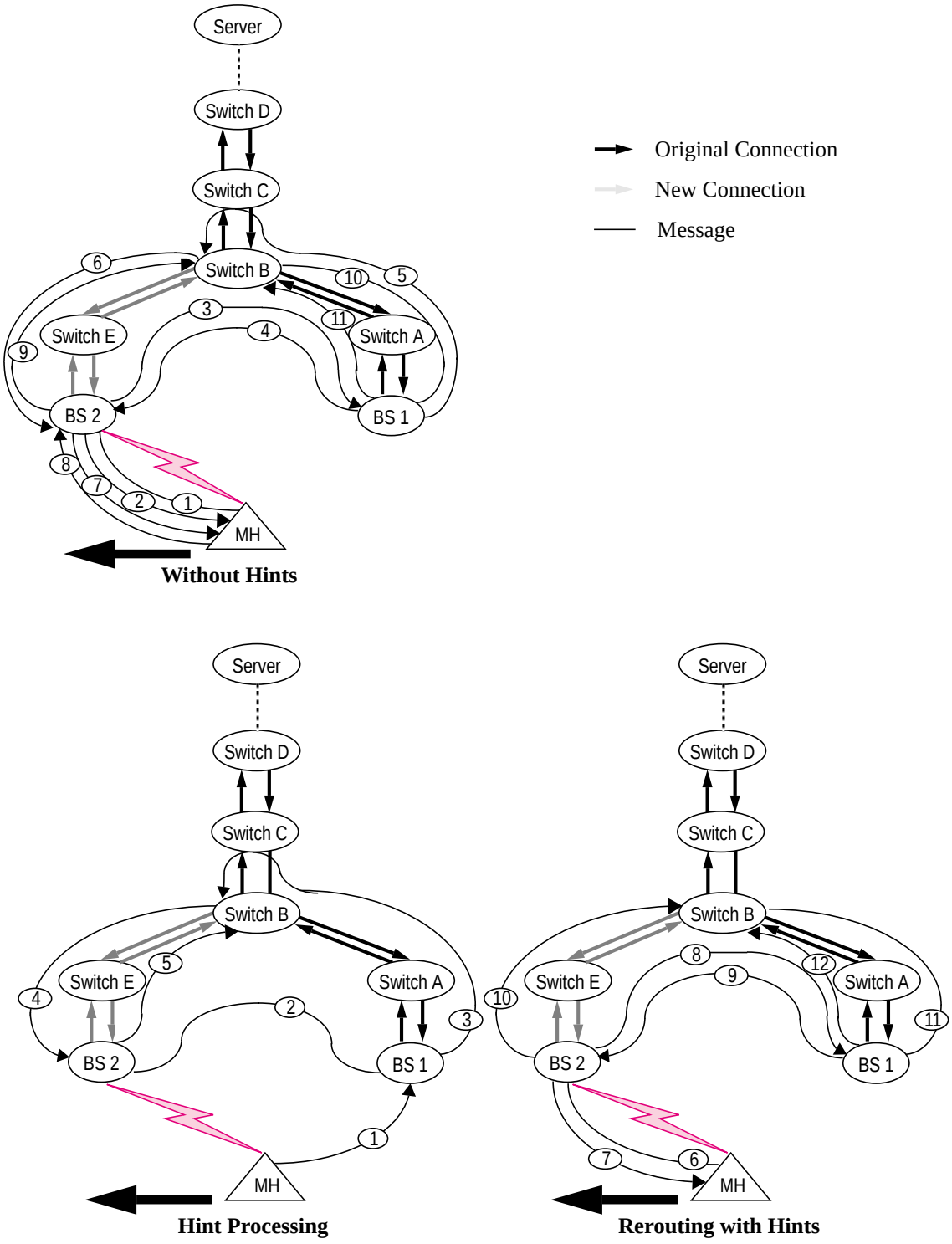


Figure 3.3 Incremental Re-Establishment

knowledge of the connection path to the old BS and possesses only local topological knowledge. BS 1 invokes this decision process by backtracking along the original connection route one hop at a time (5). Each switch along this route decides whether it knows the appropriate crossover point by examining its routing tables. If the switch uses different ports to reach the old and new BSs, it continues to forward this reroute message (Switches A and B in step 5). If the switch uses the same port to reach both the old and new BSs (Switch C), it knows that the switch one hop in the downlink direction (Switch B) is the crossover point. Switch C then communicates this fact to Switch B.

Once the crossover point has been identified, the new portion of the connection must be established and the old portion torn down. The first step, establishing the new partial connection between the crossover point (Switch B) and the new BS (BS 2), is performed as it would be in a wired network (6). This connection is established to the MH using message (7). Assuming establishment is successful, acknowledgments are sent from the MH to BS 2 (8) and from BS 2 back along the path to Switch B (9).

Message (9) is also an implicit request for Switch B to redirect all data transmitted by the Server down the newly established downlink. As in the FR algorithm, buffering at BS 2 is used to allow in-order delivery of downlink data to the MH. The downlink portion of the connection between Switch B and BS 1 is then torn down in step (10). After step (9), data from the MH can flow directly to the Server through BS 2. This uplink data is buffered at BS 2 to allow the uplink data forwarded through BS 1 to be delivered to the crossover point. Once all messages forwarded through BS 1 to the Server have been delivered to Switch B, the uplink portion of the old connection is torn down (11) and uplink data may flow over the new connection. When all new connections have been created and old connections destroyed, the rerouting is complete.

3.3.2.2 Incremental Re-Establishment with Hints

The “With Hints” portion of Figure 3.3 shows how the cell overlap information can be used to facilitate the rerouting. The MH informs the current BS, BS 1, of the potential new

BS, BS 2, and requests that new partial connections be established through the appropriate crossover points to the new BS (1). BS 1 communicates the list of connections about to be re-established to the new BS (2), and then invokes the crossover location algorithm (3). Once located, the crossover point, Switch B, begins to establish a new connection to the new BS (4). A successful establishment is indicated to Switch B by the acknowledgment in step (5).

While the connection to the new BS is being pre-established, the MH moves completely into the new cell, where it identifies itself (as in the more general algorithm) to BS 2 (6) and is acknowledged (7). Analogous to the FR scheme with hints, in the optimal case where all connections are established, message (7) contains identifiers for the already-established connections. Similarly, BS 2 initiates downlink data forwarding from BS 1 (8, 9) and then requests that the crossover point redirect downlink data over the new connection (10). Again, uplink data is not forwarded through the old BS. Buffering to ensure in-order delivery of both downlink and uplink data is performed as in the FR scheme with hints, except that the endpoint of the two paths under consideration is the crossover point rather than the server. Finally, suboptimal cases are handled in a manner analogous to that of the FR scheme with hints.

3.3.3 Multicast-Based Re-Establishment

To support video conferencing and other distributed applications, some networks support multicast connections with dynamically changing memberships. Rerouting can be implemented using a small layer of functionality on top of the multicast facility. During a handoff, data from the source host is multicast to both the new and old base stations. Once the handoff is complete, the old base station is removed from the multicast channel. The use of multicast has several interesting ramifications. Because data for the downlinks are transmitted simultaneously to multiple base stations during the interim, the actual switchover can be fairly quick, with decreased buffering. It also allows for a much better use of the location hints provided by the wireless network. Since data is transmitted over

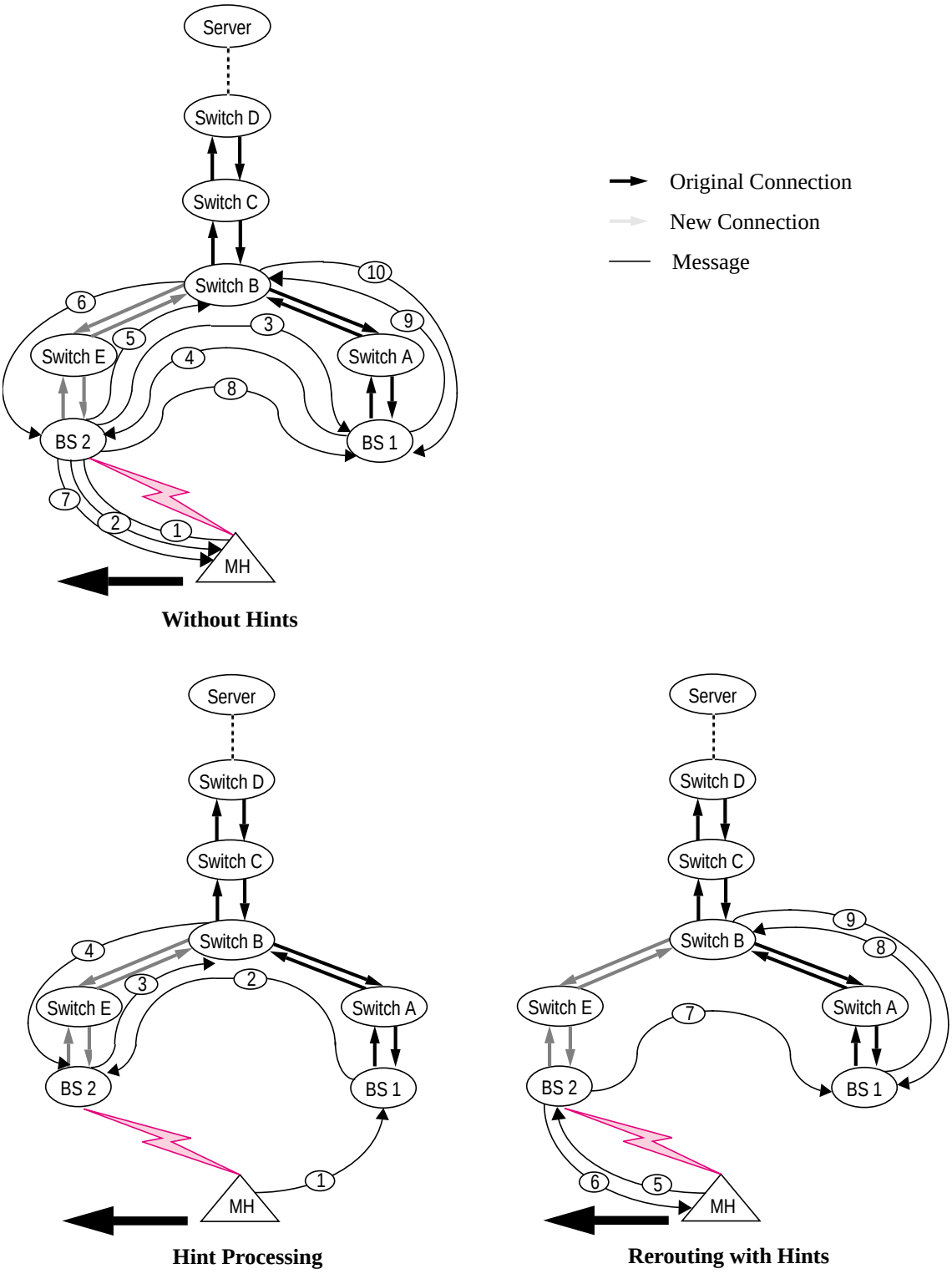


Figure 3.4 Multicast-Based Re-Establishment

multiple paths during the handoff, we expect the MB scheme to consume significantly more network resources than either the FR or IR algorithms.

3.3.3.1 Multicast-Based Re-Establishment without Hints

The “Without Hints” diagram in Figure 3.4 illustrates the operation of the MB rerouting algorithm. The MH detects that it is entering a new cell, so it acquires a wireless channel to the new BS, BS 2, and sends it a greeting message (1). This message contains an identifier for the old BS, BS 1, as well as a list of the identifiers for the various multicast channels originating or terminating on the MH. This greeting is immediately acknowledged by BS 2 (2). BS 2 then sends this channel list to the old BS along the pre-existing BS-to-BS connection (3), requesting that all data for each channel be forwarded to the new BS. In addition, BS 2 requests that it be allowed to forward data from the MH through BS 1. Upon receipt of an acknowledgment (4), BS 2 begins transferring forwarded data from BS 1 to the MH and forwarding MH data destined for the server.

Concurrently with the forwarding requests, BS 2 executes a multicast join operation (which establishes a new branch connecting BS 2 with the existing channel) for each existing channel to add itself to the multicast channel (5). Upon receiving an acknowledgment indicating a successful establishment (6), the new BS notifies the MH of the successful join operation (7). BS 2 synchronizes the data arriving down the new branch with the data being drained from the old BS. To preserve in-order delivery of the data from the MH to the server, this data must be initially buffered at BS 2 until the data forwarded through BS 1 is drained from the old portion of the connection.

When all of the multicast joins have completed and all necessary data have been obtained from the old BS, the new BS sends a completion message to the old BS (8) over the control connection. At this point, BS 1 can assume that it has no more responsibilities to the MH, and so it executes a multicast leave operation for each of the channels associated with the MH (9, 10). This operation frees node resources along the path from the BS to the

first node in common with another branch of the multicast channel. The rerouting is complete after the last branch has been torn down.

3.3.3.2 Multicast-Based Re-Establishment with Hints

The “With Hints” diagram in Figure 3.4 illustrates the advance setup the network performs in response to the hint and a best case scenario for an ensuing handoff. The hint is delivered as a message to BS 1 (1) identifying the potential new BS, BS 2. BS 1 then notifies BS 2 (2) that it should initiate multicast join operations to all of the MH’s channels in anticipation of a handoff (3, 4). When the MH loses contact with BS 1 and enters BS 2’s cell, it identifies itself with BS 2 as before (5). Due to the hint, BS 2 will have already initiated joins for all of the channels associated with the MH. When the joins have been successfully completed, BS 2 notifies the MH (6) and drains any data buffered for the MH. Finally to complete the rerouting, BS 2 sends a completion message to BS 1 (7), which then executes a multicast leave operation for each MH channel (8, 9). Buffering requirements on the BSs are similar to the FR and IR algorithms with hints.

3.4 Analysis

In this section, we describe our analysis of the algorithms. We use an analytical model of the algorithms and a characterization of the network to compute the values of various metrics describing the overheads involved in connection rerouting.

3.4.1 Parameters and Metrics

We use a set of technology-dependent parameters, shown in Table 3.1, to characterize the network. We assume no contention for link bandwidth or other network resources. Third generation wireless networks are capable of supporting megabit per second speeds, and current experimental LANs support gigabit per second bandwidths. Protocol processing time is assumed to be constant except in the case where admission control tests need to be performed, as Tenet-style admission control tests are processing intensive (e.g., 25 ms on

a RISC workstation like a DEC 5000/240).[1] Current workstation technology is approximately 10 times faster than a DEC 5000; therefore, we have reduced our admission control processing time to 2ms for our analysis. We have placed an upper bound of 50 Bytes on control messages, as they generally contain a small, fixed amount of information. (Channel establishment messages may be much larger, however.) The maximum data packet size is chosen to be 8 KBytes, which is an average packet size in many typical local area networks. Finally, the time taken to acquire a wireless channel is dependent on the actual wireless network system in use.

Symbol	Definition	Value
BW_{wl}	Bandwidth of the wireless link.	2 Mbps [ATT]
BW_{w}	Bandwidth of the wired backbone network.	155 Mbps [Schr92]
L_{wl}	Latency of the wireless link, including data link and network layer processing.	2 ms [ATT]
L_{w}	Latency of a link in the wired backbone, including data link and network layer processing.	500 μs [Zhan92]
PPT_{fixed}	Protocol processing time for control messages.	0.5 ms
PPT_{adm}	Protocol processing time for steps where admission control is to be performed, in excess of fixed protocol processing time.	2 ms
S_{ctrl}	Upper bound on the size of a control message.	50 bytes
S_{data}	Maximum size of a data packet.	1024 bytes
$T_{acquire}$	Time for an MH to acquire a wireless channel to a base station.	5 ms [Brod93]

Table 3.1 Technology-Dependent Network Parameters

Each network connection to and from a mobile host at the time of rerouting can be characterized by the set of parameters listed in Table 3.2. These parameters are dependent on the locations of the mobile host and its multimedia servers, as well as the topology of the net-

1.The values for protocol processing times were derived from measurements of the Tenet Real-Time Channel Administration Protocol (RCAP) running on DECstation 5000/240 workstations.

work. By varying these parameters, we can examine the overheads associated with each rerouting algorithm over varying lengths of connections.

Symbol	Definition
H_{new}	Number of hops to the crossover point from the new (destination) BS.
H_{old}	Number of hops to the crossover point from the old (original) BS.
H_{ctrl}	Number of hops between the old and new BSs along their control channel.
H_{server}	Number of hops between an MH and its multimedia server.

Table 3.2 Parameters Characterizing Network Connections

We can measure the performance of the various rerouting algorithms using the metrics listed in Table 3.3. The values of these metrics, evaluated for a given set of technology-dependent parameters and various parameters of network connections, will give some idea of the effectiveness of the different schemes. $T_{disrupt}$ corresponds to the service disruption time, that is, the time during which the MH cannot receive data on its downlink. (This disruption may manifest itself as a pause in the playback of video.) B_{mh}, $B_{bs,\,down}$, and $B_{bs,\,up}$ correspond to the buffering required on the MH and the BS to prevent data from becoming out of order as a result of a rerouting operation. The bandwidth-space-time products are one measure of the added network resources required to do the rerouting. They are essentially the product of the bandwidth provided to a connection, multiplied by the length of the connection and the amount of time that the connection is in existence. P_{excess} measures the network resources that are allocated to inactive connections (for example, a channel or portion of a channel that has been established but is not being used to carry data). P_{fwd} is a measure of the network resources reserved for the forwarding of packets between adjacent BSs during rerouting.

3.4.2 Derivation of Metrics

We now present the derivation of the metrics listed in Table 3.3 for the Incremental Re-Establishment algorithm, in the case that there are no hints regarding advance warning of a handoff. The metrics for the other rerouting algorithms are derived in a similar way.

Symbol	Definition
$T_{disrupt}$	Service disruption time, the time during which the MH cannot receive data on its downlink.
$T_{complete}$	Rerouting completion time, the time after which the rerouting processing has completed and the network has stabilized.
B_{mh}	Buffering required on the MH for buffering of uplink data during rerouting.
$B_{bs,\,down}$	Buffering required in the BSs for buffering of downlink data during rerouting.
$B_{bs,\,up}$	Buffering required in the new BS for buffering of uplink data during rerouting.
P_{excess}	Excess bandwidth-space-time product used by inactive channels during rerouting.
P_{fwd}	Bandwidth-space-time product used by data being forwarded during rerouting.

Table 3.3 Metrics for Comparing Handoff Algorithms

This analysis assumes perfect delivery of control messages. We also assume that the maximum throughput for a connection is the throughput of the bandwidth of the wireless link. All of our computations for buffering requirements are derived on a per-channel basis, with the worst case being that in which the channel's throughput is equal to the wireless link's total bandwidth.

We compute the time taken for each of the scheme's control messages to be transmitted, forwarded, and if necessary, processed. Message numbers for this section are those from Section 3.3.2.1 and Figure 3.3. Message (1) is a greeting from the MH to the BS in the new cell. The latency needed for this message is the sum of the time to acquire a wireless channel, the transmission time of a control message on that channel, the propagation time on the wireless link, and the fixed protocol processing time on the new BS.

$$T_1 = T_{acquire} + \left(\frac{S_{ctrl}}{BW_{wl}} \right) + L_{wl} + PPT_{fixed} \tag{1}$$

Message (2) is an acknowledgment of the greeting back to the mobile host; its total delay is simply the sum of the transmission and propagation times, plus processing time in the mobile host.

$$T_2 = \left(\frac{S_{ctrl}}{BW_{wl}}\right) + L_{wl} + PPT_{fixed} \quad (2)$$

Immediately after sending Message (2), the new BS sends a request (3) for forwarding of both uplink and downlink data to the old BS, using the control channel between the two (physically) adjacent base stations. Its total latency is the end-to-end delay through the control channel plus the fixed protocol processing time in the old BS.

$$T_3 = \left[\left(\frac{S_{ctrl}}{BW_w}\right) + L_w\right] H_{ctrl} + PPT_{fixed} \quad (3)$$

The next message, Message (4), is an acknowledgment of the forwarding request, sent from the old BS to the new BS. Its total latency is computed similarly to that of Message (3).

$$T_4 = T_3 = \left[\left(\frac{S_{ctrl}}{BW_w}\right) + L_w\right] H_{ctrl} + PPT_{fixed} \quad (4)$$

Immediately after sending Message (4), the old BS begins the distributed crossover point location process (5). The messages needed to find the crossover point will incur a latency (T_5) equal to the transmission and propagation time along each hop, plus the fixed control protocol processing time in each switch along the path from the old BS to the crossover point. We note that the PPT_{fixed} term in each hop's latency is necessary because, in contrast to Messages (2) and (3), the switch controller at each hop needs to do some processing of the control message (specifically, to determine whether it knows the crossover point).

$$T_5 = \left[\left(\frac{S_{ctrl}}{BW_w}\right) + L_w + PPT_{fixed}\right] H_{old} \quad (5)$$

Message (6) is a partial channel establishment from the crossover point to the new BS. Admission control tests need to be executed at each switch along this path, which implies a delay of PPT_{adm} at each hop in addition to the normal PPT_{fixed} required for normal control message processing.

$$T_6 = \left[\left(\frac{S_{ctrl}}{BW_w}\right) + Lw + PPT_{fixed} + PPT_{adm}\right] H_{new} + PPT_{adm} \quad (6)$$

Message (7) completes the partial channel establishment from the new BS to the mobile host; therefore the bandwidth and link latency are those of those of the wireless link.

$$T_7 = \left(\frac{S_{ctrl}}{BW_{wl}}\right) + L_{wl} + PPT_{fixed} + PPT_{adm} \tag{7}$$

Message (8) is the acknowledgment of the partial channel establishment across the wireless link.

$$T_8 = T_2 = \left(\frac{S_{ctrl}}{BW_{wl}}\right) + L_{wl} + PPT_{fixed} \tag{8}$$

Acknowledgment of the partial channel establishment (back to the crossover point) is completed by Message (9). This control message retraces the path of the partial establishment, hop by hop.

$$T_9 = \left[\left(\frac{S_{ctrl}}{BW_{w}}\right) + L_{w} + PPT_{fixed}\right] H_{new} \tag{9}$$

Message (10) begins the teardown of the partial connection from the crossover point to the old BS. As with the Message (5), it must travel hop by hop between switch controllers.

$$T_{10} = T_5 = \left[\left(\frac{S_{ctrl}}{BW_{w}}\right) + L_{w} + PPT_{fixed}\right] H_{old} \tag{10}$$

A final message, Message (11), is sent from the old BS to the crossover point to complete the teardown of the old leg of the connection. The delay incurred by this message is computed identically to that of Message (10).

$$T_{11} = T_{10} = \left[\left(\frac{S_{ctrl}}{BW_{w}}\right) + L_{w} + PPT_{fixed}\right] H_{old} \tag{11}$$

Using the time spent transmitting, forwarding, and processing each of the control messages. we can compute the values of the various metrics. $T_{disrupt}$, the amount of time during which network service on the downlink to the MH is disrupted, is the time the MH needs to acquire a wireless channel, have the new BS arrange for data forwarding, and for the MH to receive the first forwarded data packet.

$$T_{disrupt} = T_1 + \left(\frac{S_{ctrl}}{BW_{wl}}\right) + T_3 + T_4 + \left(\frac{S_{data}}{BW_{wl}}\right) + L_{wl} \quad (12)$$

$T_{complete}$, the amount of time for the all the rerouting to complete, is the time from when the MH acquires a channel to the time at which the connection to the previous BS is torn down. All events except for the acknowledgment transmissions occur sequentially in an IR rerouting. Therefore, the completion time is the sum of the times for each of the events during rerouting.

$$T_{complete} = T_1 + \left(\frac{S_{ctrl}}{BW_{wl}}\right) + T_3 + \left(\frac{S_{ctrl}}{BW_{w}}\right) + T_5 + T_6 + T_7 + T_8 + T_9 + T_{10} + T_{11} \quad (13)$$

The amount of buffering required by the MH is determined by the amount of time during which the MH cannot transmit data on its wireless uplink. This includes the time for the MH to greet the new BS and the time for the new BS to acknowledge the greeting.

$$B_{mh} = (T_1 + T_2)\, BW_{wl} \quad (14)$$

The amount of buffering required on the old BS for downlink buffering is determined by the amount of data that will need to be “replayed” to the MH through the forwarding channel and the new BS. The data buffering must be sufficient to cover the period during which the new BS arranges for data forwarding from the old BS and waits for the corresponding acknowledgment. The buffers must also contain the data that was being transmitted along the wireless link from the old BS to the MH when the MH moved into the new cell. We assume that the first forwarded data packet immediately follows the acknowledgment of the forwarding request (Message 4).

$$B_{bs,\,down} = \left(T_1 + \left(\frac{S_{ctrl}}{BW_{wl}}\right) + T_3 + \left(\frac{S_{ctrl}}{BW_{w}}\right) + L_{wl}\right) BW_{wl} \quad (15)$$

The amount of uplink buffering needed on the new BS is influenced by two terms. The first is the time to establish forwarding minus the time taken to send the acknowledgment for the MH’s greeting and the time needed for the first data packet to be arrive at the new BS. The other term is the difference in the delay between the paths going through the old

and new base stations. In the case where a network offers deterministic delay bounds [Ferr92], it would be possible to use those values in this derivation.

$$B_{bs,\,up} = max\left(T_3 + T_4 - (2L_{wl}),\ \left\lceil L_w + \left(\frac{S_{data}}{BW_w} \right) \right\rceil max\,(H_{ctrl} + H_{old} - H_{new},\, 0) \right) BW_{wl} \tag{16}$$

The amount of excess bandwidth-space-time product is given by the amount of network resources allocated but not in use. This includes the bandwidth used by the channel between the new BS and the crossover point during establishment of the new partial connection and subsequent acknowledgments and the bandwidth used by the channel between the old BS and the crossover point after the switchover and before that connection has been completely deleted.

$$\begin{aligned} P_{excess} = \Bigg(& \left\{ \frac{H_{new}^2 + H_{new}}{2} \right\} \left(\frac{S_{ctrl}}{BW_w} + L_w + PPT_{fixed} + PPT_{adm} \right) + \\ & (T_7 + T_8 + T_9)\, H_{new} + \\ & T_{10} H_{old} + \\ & \left(\frac{H_{old}^2 + H_{old}}{2} \right) \left(\frac{S_{ctrl}}{BW_w} + L_w + PPT_{fixed} \right) \Bigg) BW_{wl} \end{aligned} \tag{17}$$

The bandwidth-space-time product required for forwarding data from the old BS to the new BS during rerouting is dictated by the amount of data that needs to be forwarded.

$$\begin{aligned} P_{fwd} = \Bigg\lceil & T_1 + \left(\frac{S_{ctrl}}{BW_{wl}} \right) + T_3 + \left(\frac{S_{ctrl}}{BW_w} \right) + T_5 + T_6 + T_7 + T_8 + T_9 \\ & + \left(\frac{S_{data}}{BW_w} + L_w \right) H_{old} + \left(\frac{S_{data}}{BW_{wl}} + L_{wl} \right) \Bigg\rceil BW_{wl} H_{ctrl} \end{aligned} \tag{18}$$

3.4.3 Performance of Schemes

We have derived formulae for the performance metrics for the FR, IR, and MB algorithms, with and without the availability of hints regarding advance warning of a handoff. Substituting the values listed in Table 3.1 into these formulae provides the framework for our analysis. We have varied values from Table 3.2 to obtain values for the performance metrics listed in Table 3.3. H_{server} was fixed at six hops, while both H_{new} and H_{old} were varied

from one to six hops, to simulate the choice of a crossover point anywhere along the path from the base stations to the server. Depending on the size and organization of the switches, the diameter implied by such a network (six hops) may be as large as a UC, Berkeley sized campus. For simplicity, the network topology was assumed to be tree like. In addition, H_{new} and H_{old} were chosen to be equal for each experiment. H_{ctrl} was chosen to be twice the distance between the crossover point and the old or new BS. For the cases where the algorithms use cell overlap hints, we assume that the MH sends the hint to the old BS one second before it enters the new cell. This value is based on the frequency of beacons in some systems and is consistent with the relatively slow movement of users. We first examine the effects of topology on the performance of different algorithms and then compare the relative performance of the algorithms.

3.4.3.1 Topology Effects

We present a discussion of the performance results of the FR, IR and MB algorithms, both with and without hints. Our analysis of the algorithms using hints assumes the best case, where all connections have been established by the time the MH enters the new cell. For the best case situation to occur, a hint about handoff must be available approximately the completion time of an “un-hinted” rerouting before the actual transition between cells.

Figure 3.5 shows the service disruption time in the delivery of downlink data. For all schemes except MB with hints, this metric is dependent on the time to set up data forwarding, it is highly dependent on H_{ctrl}, the number of hops in the path between the old and new BSs. The service disruption time is the same for both the case with and without hints since both request forwarding of downlink data. In the case of MB with hints, which does not request downlink forwarding, there is a constant service disruption time, which is shorter than the disruption time for the other algorithms.

Figure 3.5 shows the completion time for the different rerouting schemes. For the situation where hints are unavailable, the completion time is closely related to the number of connection hops that are being modified. The number of hops modified is constant, H_{server}, or

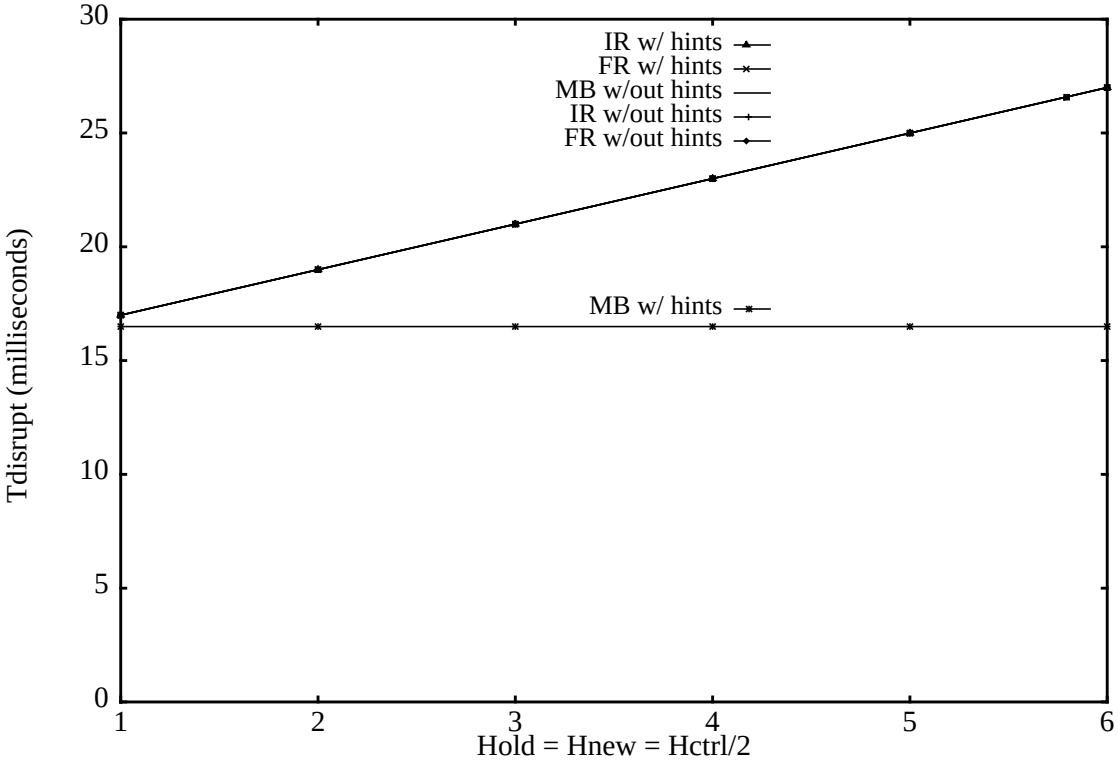


Figure 3.5 Service Disruption Time

FR and is equal to the distance to the crossover point, H_{new} and H_{old}, for MB and IR. When hints are available, the completion times for the FR and IR schemes are closely related to the time to set up forwarding and to tear down the old connection. Therefore, it is highly dependent on the number of hops in the path between the old and new BSs, H_{ctrl}, and on the number of hops torn down, H_{server} for FR and H_{old} for IR. For the MB algorithm with hints available, the completion time mostly dependant on H_{old} since it does not use forwarding.

The mobile host buffers temporarily stores uplink data during the time which it is out of communication. For all algorithms, this time includes only the greeting message and acknowledgment and is therefore constant. Because of its definition, the MH buffering is a constant 20800 bits for all of the rerouting algorithms and for all combinations of topology and hint availability.

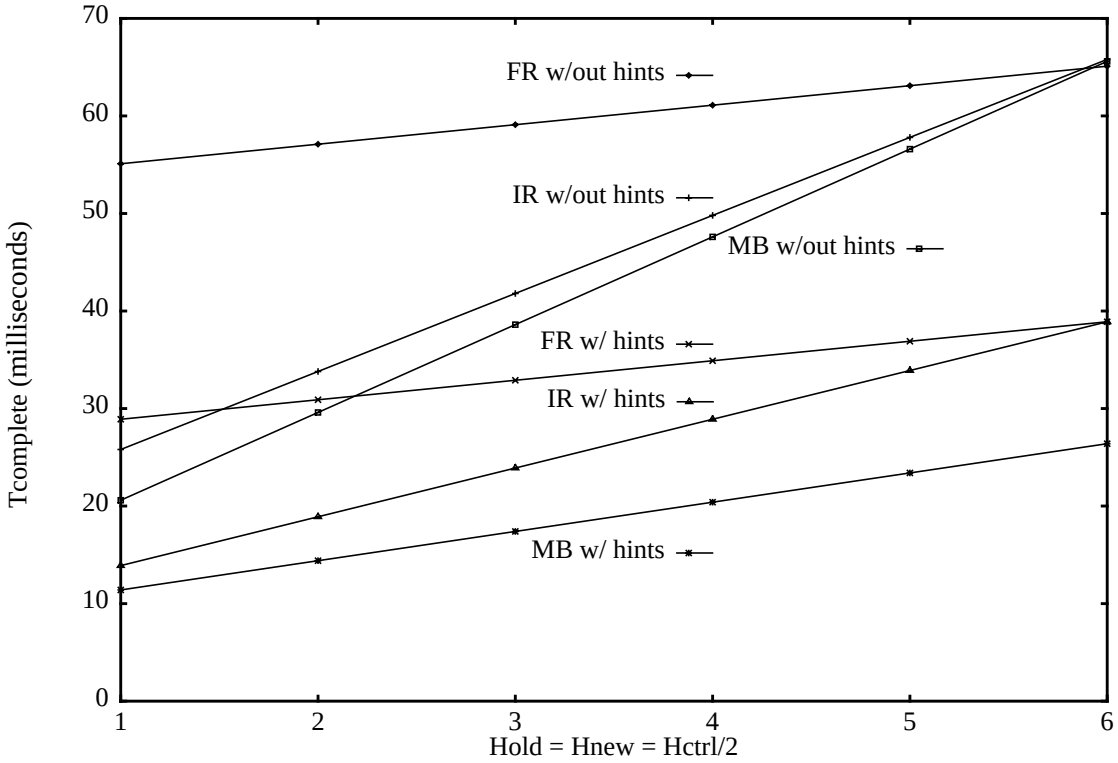


Figure 3.6 Rerouting Completion Time

Figure 3.7 illustrates the buffering required on the base station for downlink data. As in the case for service disruption, this metric is highly dependent on the time required to initiate data forwarding from the old BS to the new BS. Because each scheme (with the exception of MB with hints) requests this forwarding, the buffering requirements shown in Figure 3.7 apply to all of these schemes. The constant downlink BS buffering required for MB with hints (to buffer data sent over the radio communication switchover period) is several thousand bits less than that required for the case of $H_{old} = H_{new} = H_{ctrl}/2 = 1$.

The buffering required on the base station for uplink data is shown in Figure 3.8. For the case when hints are unavailable all the algorithms perform similarly. The buffering quantity is determined by the maximum of the time to set up forwarding and the difference in propagation times to the crossover point along the uplink forwarding path and the new uplink path. For a small number of hops between BSs, the difference in propagation time

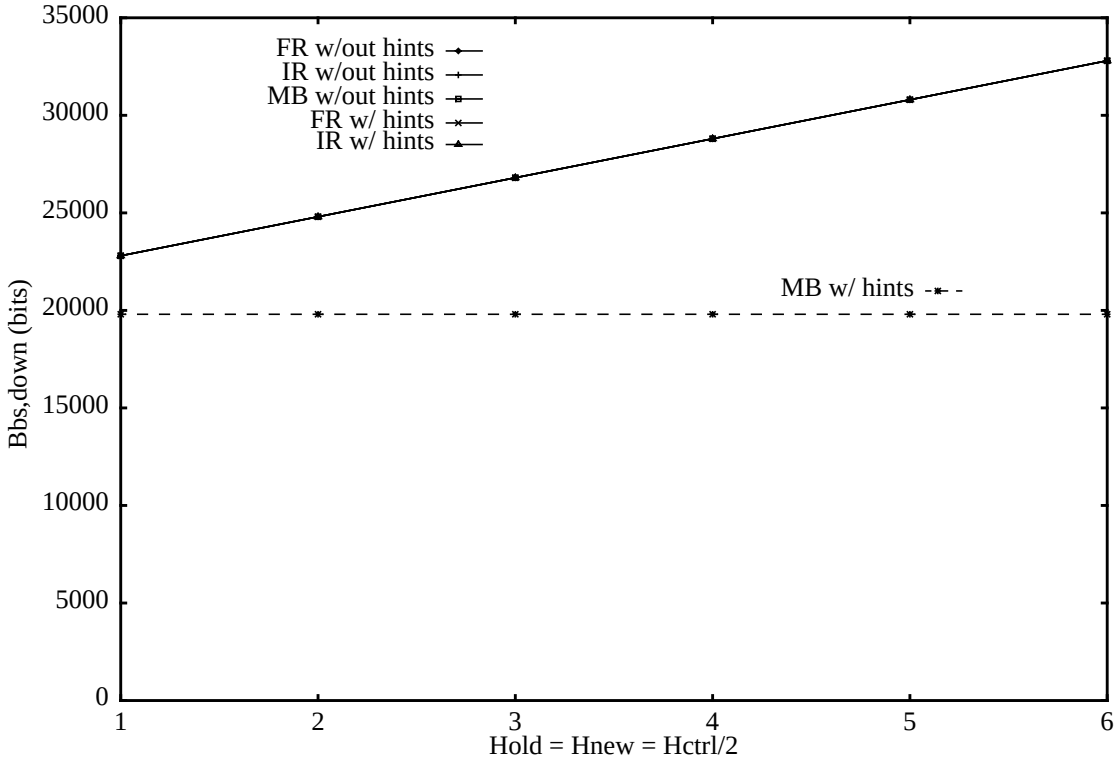


Figure 3.7 Base Station Buffering per Stream Requirements (Downlink)

dominates and for more than 3 hops between BSs, the forwarding set up is the limiting factor. Thus, the uplink buffering requirements are directly proportional to the number of hops along the forwarding path between the old and new BSs.

With the availability of hints, none of the algorithms forward uplink data; as a result, the uplink buffering requirements with hints are dependent only on the difference in propagation times to the crossover point along the old and new uplink paths, including the time for the MH to switch cells. In our example tree topology the old and new uplink paths have equal lengths. Hence, there is no buffering required for uplink data in the hinted IR, FR, and MB cases. However, buffering is necessary for these algorithms in many other topologies.

Figures 3.9 and 3.9 diagram the amount of bandwidth over all links consumed by channels established, but not in use, during a rerouting. This value also reflects the duration for

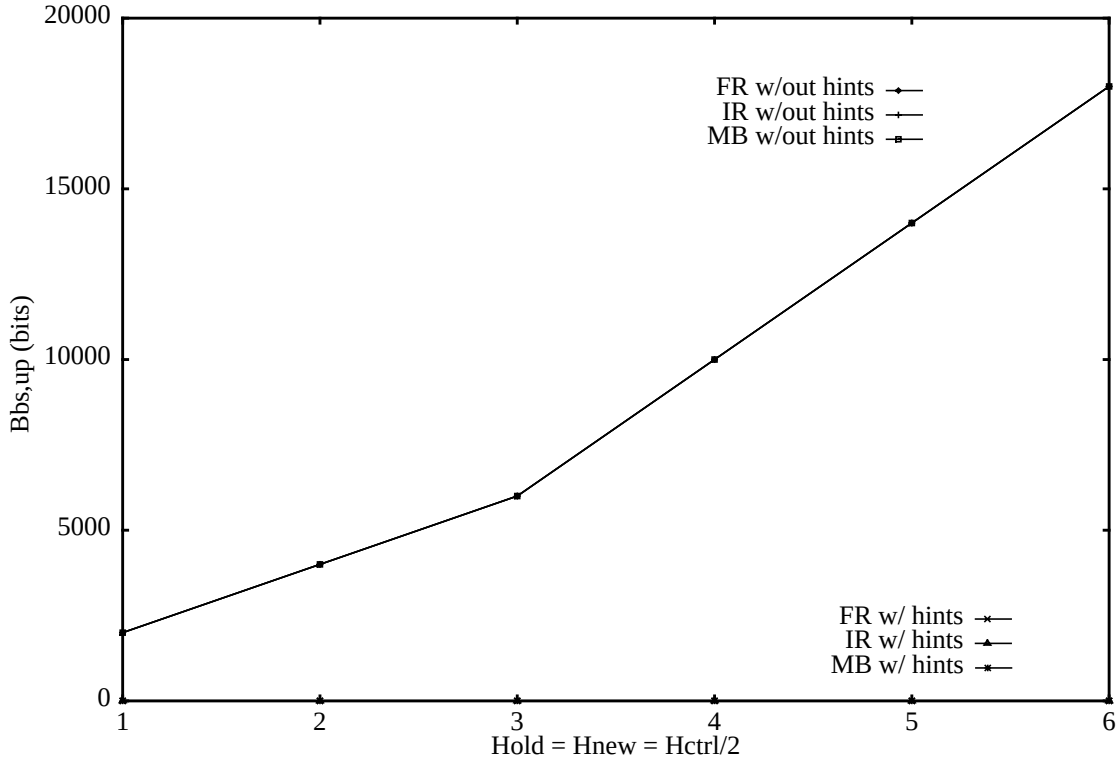


Figure 3.8 Base Station Buffering Requirements (Uplink)

which these resources are held. As expected, the excess resources consumed for the case where hints are available exceed those used for the non-hinted case, because the new channel is established in advance through the use of the hint. For both the IR and MB algorithms, this effect is magnified as the distance from the crossover point increases because the amount of excess resources held grows as the square of both H_{old} and H_{new}. Because the resources reserved for the FR scheme are dependent only on the distance between the BS and the server, the plot of P_{excess} for FR is constant for both the non-hinted and hinted cases.

The use of hints in the algorithms results in a nearly ten-fold increase in the resources reserved, but not used, in performing a rerouting. This factor is highly dependent on the choice of the time value representing how far in advance the hint is received before the MH actually moves into the new cell.

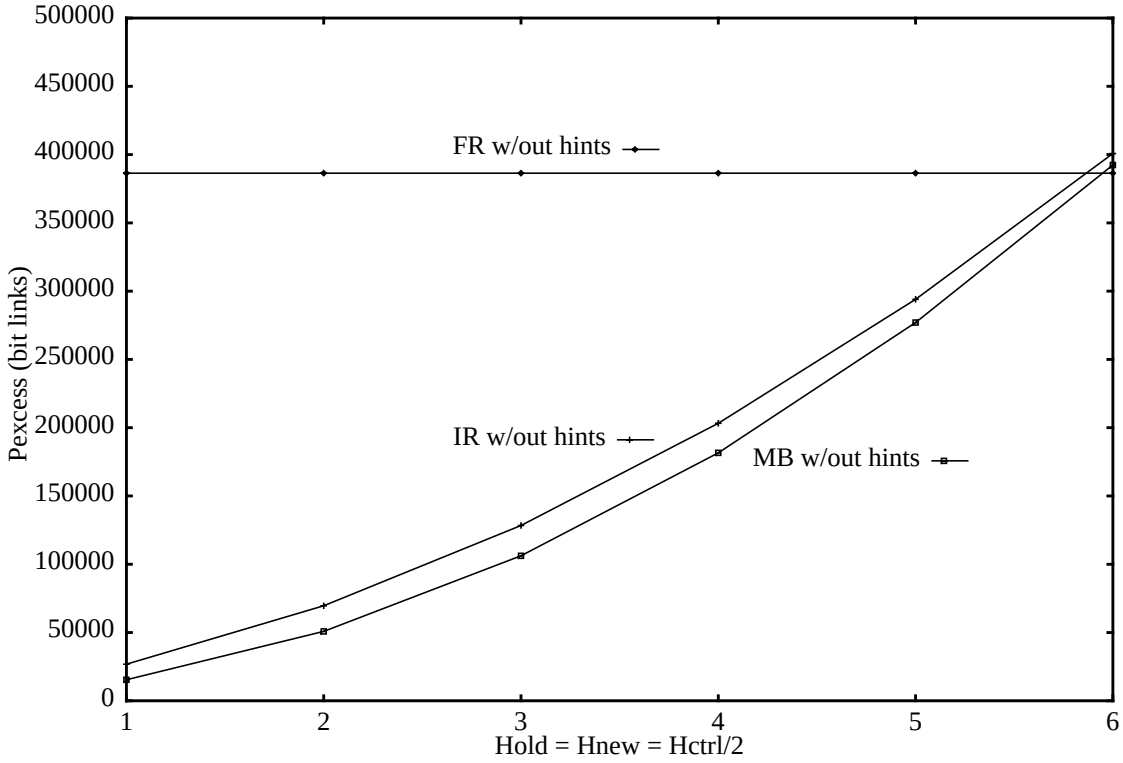


Figure 3.9 Excess Bandwidth-Space-Time Utilization (no hints)

The amount of bandwidth-link-time resources consumed to perform forwarding is shown in Figure 3.11. No data is forwarded for the hinted MB scheme. For the other schemes, the plots for the cases both with and without hints grow as the distance from the crossover point increases. This behavior is due to the fact that as H_{ctrl} increases, the number of links used to perform forwarding grows. In addition, the amount of data to be forwarded (or time that the resources will be consumed) grows as the time to set up the forwarding increases. The resources used in forwarding data for the case with no hints exceed those used in the case with hints because data must be forwarded during connection establishment for the former scheme. The use of hints results in a decrease in the resources consumed by a factor of two (for small values of the network topology parameters) to three (for larger hop count parameters). This behavior is also exhibited for the FR and MB schemes without hints and the FR scheme with hints.

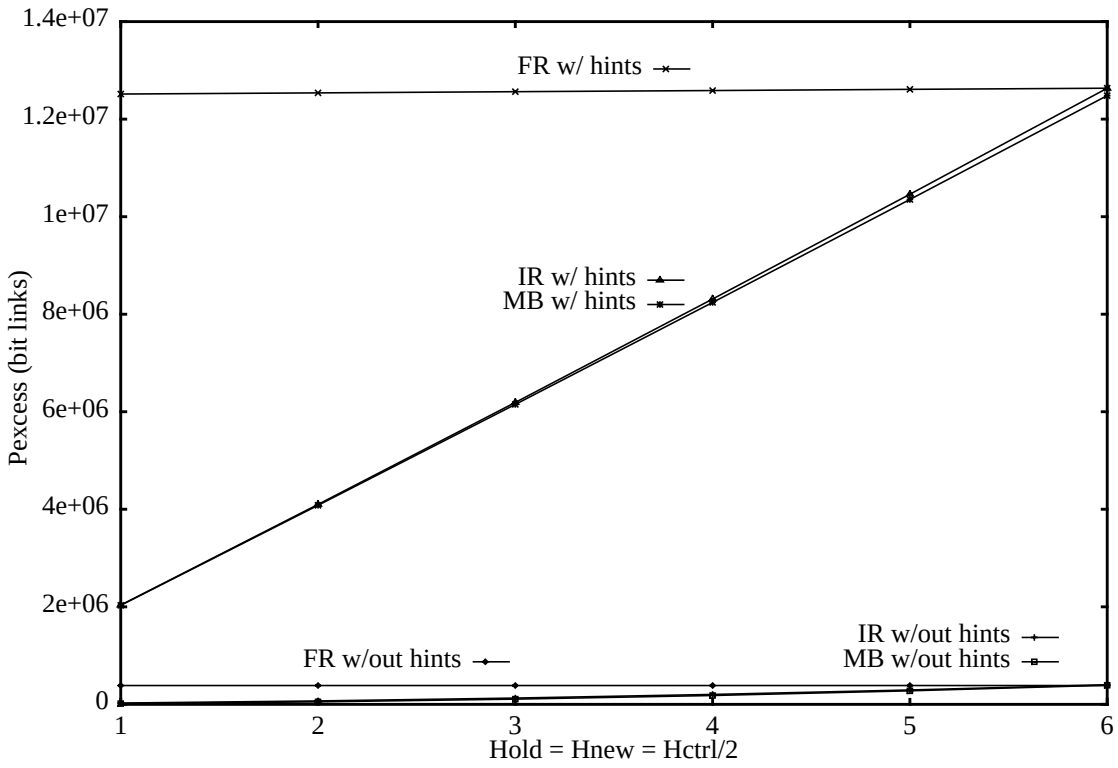


Figure 3.10 Excess Bandwidth-Space-Time Utilization (all algorithms)

3.4.3.2 Comparison of Schemes

The objective of the FR and IR schemes is to localize the route processing to as near the mobile host and base station as possible. This localization should reduce the impact of handoff on the network and improve handoff performance. To see if the algorithms meet their objectives, we compare their performance to the FR scheme, which is not localized.

We first compare the backbone network utilization, P_{excess} and P_{fwd}, for the three rerouting algorithms. As noted before, these metrics measure the amount of bandwidth used over all of the links and the duration of that bandwidth's use. The excess resources used by the FR scheme depend only on the distance between the BS and the server and do not vary significantly with other topology parameters. The IR and MB algorithms, however, require increasing amounts of excess bandwidth as the paths from the base stations to the cross-

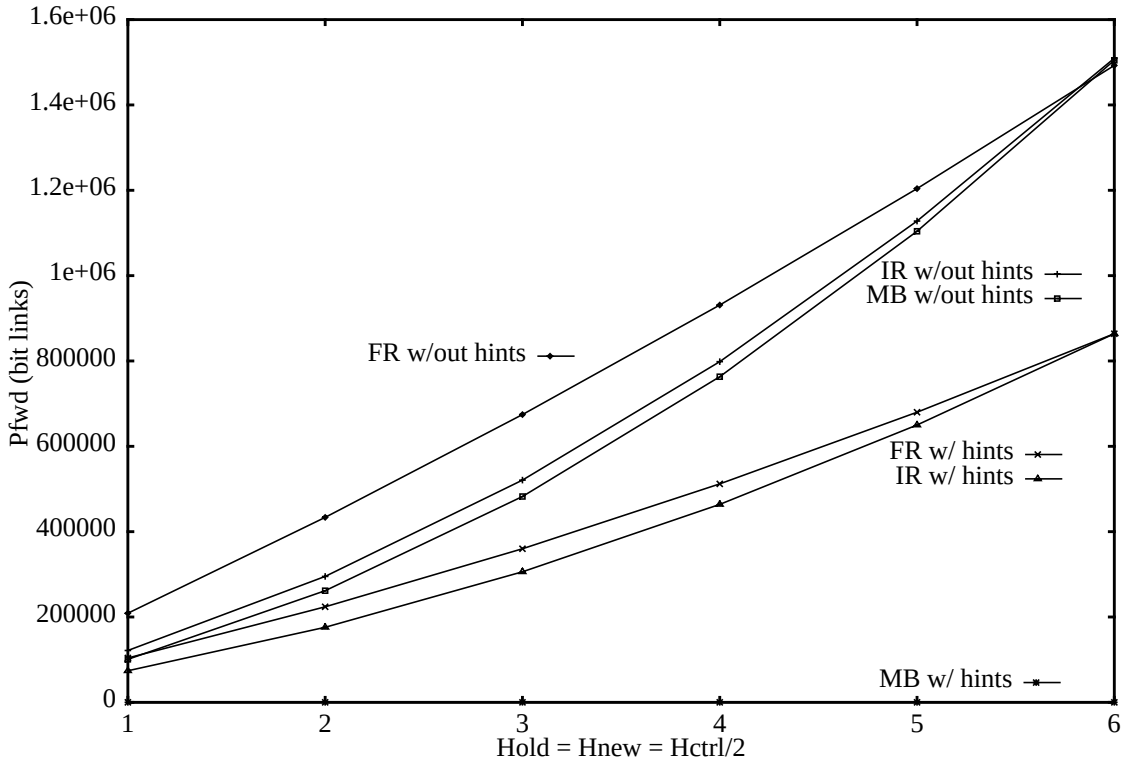


Figure 3.11 Forwarding Bandwidth-Space-Time Utilization

over point gets longer resulting in a nearly linear increase in P_{excess}. The maximum utilization of the MB and IR algorithms is equal to the normal utilization of the FR scheme. As shown in Figures 3.9 and 3.9, a five-fold or more reduction in the resources consumed is possible with the use of the IR or MB algorithms instead of FR. This reduction is only seen for topologies with crossover points close to the BSs.

In Figure 3.11, the values of P_{fwd} is shown for all algorithms. In all situations except for the MB scheme with hints, the length of the forwarding channel is equal to the distance between base stations. The MB scheme eliminates the need for forwarding when hints are available. The duration for which the forwarding connection is used is related to the rerouting completion time. The completion time of a FR rerouting is nearly constant. As a result, the forwarding resources needed to support the FR algorithm varies linearly with distance between base stations. The completion time of the IR and MB schemes varies

with the distance to the crossover point and is always less than the completion time for FR. Therefore, P_{fwd} for the IR and MB schemes increases roughly with the square of the length of the control connection. In addition, the use of IR or MB over FR decreases the amount of resources consumed in forwarding data. The decrease in scenarios without hints is 30 to 50 per cent for cases when $H_{old} = H_{new} = H_{ctrl}/2$ is one or two. In the cases where hints are available and the crossover point is close to the BSs, IR offers a 12 to 15 per cent decrease in resource consumption over FR. The MB algorithm consumes the least forwarding resources when hints are available since no forwarding of data is done.

When compared on the basis of service disruption time, MH buffering, or BS buffering, the FR, IR, and MB algorithms are not significantly different. This is because most of these schemes use the same forwarding mechanism and these metrics are mainly controlled by the forwarding of data. One exception is the case where MB algorithm uses hints to perform a rerouting. Since data is simultaneously sent to the old and new BSs, the MB algorithms eliminates the need for forwarding of data, downlink data buffering and it reduces the service disruption time by 40% when BSs are 6 hops apart.

The effects of hints on the performance of the algorithms was striking. Depending on how far in advance a hint is available before the MH moves, the excess resources necessary to complete the rerouting may be significantly greater than those needed in the case where hints are not available. In our analysis with hints available one second in advance, resource consumption is approximately 10 times greater than when hints were unavailable. However, the use of hints can result in a two- to three-fold decrease in the resources necessary to forward data and allow rerouting to complete up to a factor of three faster.

These results also show that the effects of network topology are important. If the network is constructed such that the paths between the BSs and the crossover points are short, significant reductions in the resources required for handoffs may be realized for the IR and MB algorithms over the more naive FR algorithm. This effect is also dependent on the distance between physically adjacent BSs, as control data and forwarded data must travel

along this path. These results suggest that it is advantageous to place physically adjacent base stations logically close together in the wired network topology used to support mobile hosts.

Overall, the results indicate that the IR and MB algorithms both effectively localize rerouting to reduce the impact of handoff on network utilization. The use of forwarding in all the schemes is the primary mechanism that minimizes the service disruption caused by rerouting. This is because forwarding localizes the changes necessary before an MH is reconnected. As a result, the IR algorithm and MB algorithm without hints do not reduce the handoff disruption seen by the user. Only the MB algorithm with the availability of hints improves the service disruption performance by using a mechanism other than forwarding to reconnect the MH quickly.

3.5 Optimal Base Station Layout

Our analysis of rerouting overhead has shown that base station layout is an important factor in handoff performance. The relationship between wired topology and rerouting suggests that there are optimal ways to connect base stations in a wireless system. This layout would be determined based on the wired network limitations, the rate of handoffs between different cells and the typical network communication load of each cell.

Wired networks typically consist of broadcast-based networks, such as ethernet, connected together using routers or switches. There are two critical limited resources in such networks: bandwidth and switch ports. It is these limited resources that prevent all adjacent base stations from being placed close together in the backbone network. The bandwidth requirements of a cell determine how many base stations may be placed on a single broadcast network. Similarly, the fan-out of routers determines how many broadcast networks can be directly connected to each other.

The problem of optimally connecting base stations while meeting the requirements of the wired network can be modeled using a graph. Each node in the graph corresponds to a

base station and the weight of the node corresponds to its expected bandwidth. The edges between nodes correspond to the adjacencies between cells in the physical world. The weight of each edge is equal to the rate of handoff between the two associated cells. Since the topology of a network cannot be dynamically changed, this graph does not take into account the dynamic traffic patterns of users. The node and edge weights are based on long duration averages for the various measures. This sets up a graph containing the all the information needed from the wireless network. To determine the optimal base station lay-out, the nodes of this graph must be aggregated together into sets while not violating the requirements of the wired network. An example graph and aggregation (not necessarily optimal) is shown in Figure 3.12. In this example, many MHs reside and move between the cells of BS1 and BS2. There are also several MHs that normally reside in BS6. The corresponding edges and nodes in the BS graph are given higher weights (related to actual bandwidth consumed by MH and handoff rates between cells). BS1 and BS2 should be as close together as possible since the handoff rates between them are high. Since BS1, BS2 and BS6 each consume a significant amount of network bandwidth, no more than two of them can share a single network with another BSs (i.e. two of them consume the entire network bandwidth). The other BSs require significantly less bandwidth and can all be placed on a single network. A possible aggregation and network topology of BSs that meets are requirements is shown.

The first step in the general algorithm to determine the layout is to determine which base station should be on the same broadcast networks. The limitations of the network imply that in combining nodes together, the total weight of the set must not exceed the bandwidth of the broadcast link. The objective of aggregating the nodes together is to minimize the total weight of edges between the sets. This minimizes the amount of handoff traffic between networks. This step is known as the Graph Partitioning problem and is NP-Complete [Gare79]. The next step of the algorithm is to map the graph back to an actual network topology while taking the restrictions placed by router fan-out into account. This can be done using a greedy algorithm. Take the set of nodes with the highest degree and use as

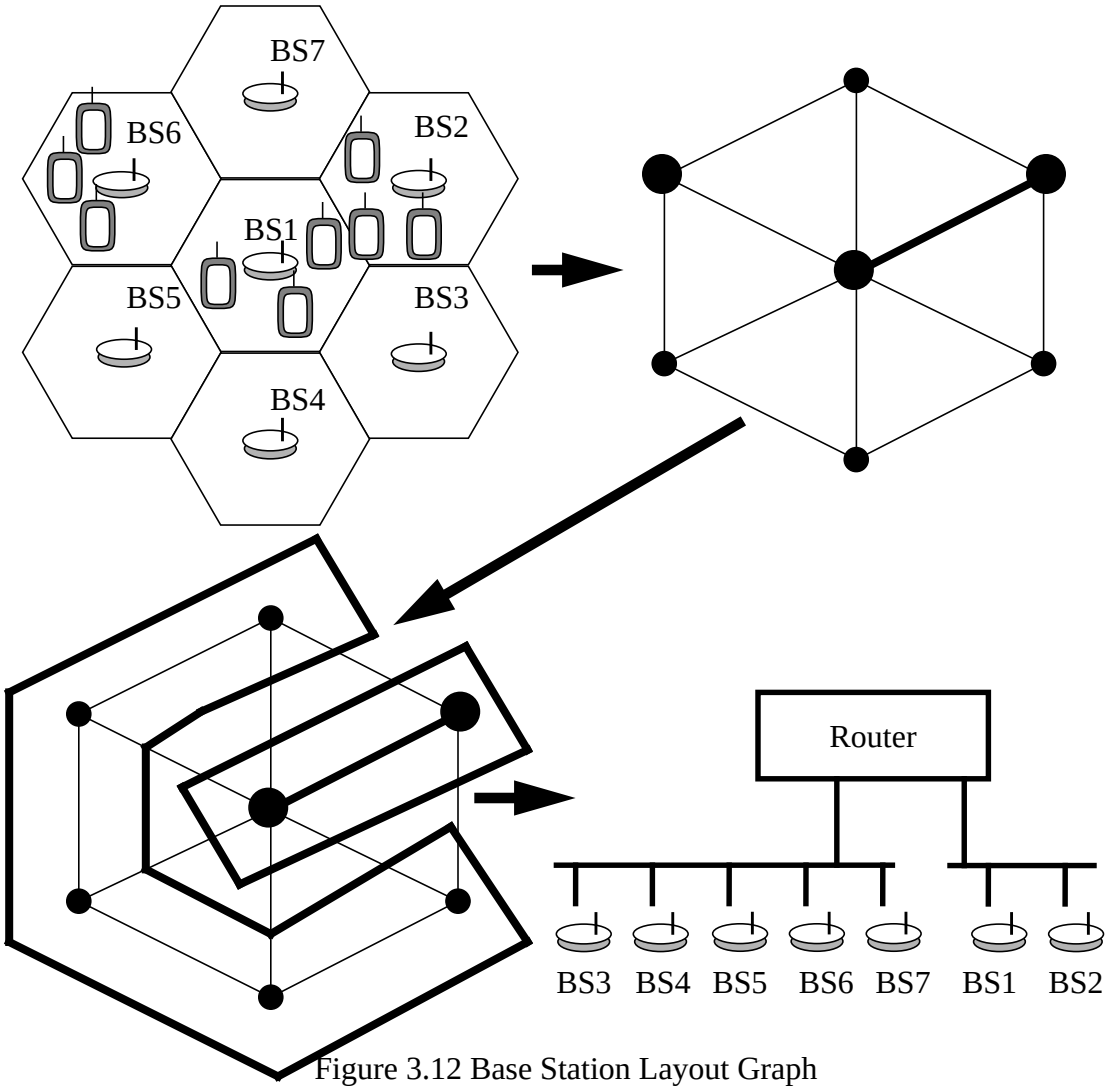


Figure 3.12 Base Station Layout Graph

many routers as needed to implement the edges. This is repeated with the remainder of the graph.

This algorithm produces an network layout that provides optimal rerouting performance. However, it may use an unnecessarily large number of routers. Some routers can be eliminated with only minimal impact on handoff latency and overhead. For most situations, this algorithm should provide a layout in which most handoffs occur between base stations that are one or two hops apart.

3.6 Summary

A combined multimedia, mobile computing environment poses new problems in networks. Multimedia applications typically require certain qualities of service from network services; real-time network services often use connections in order to provide real-time performance guarantees. In order to provide such services in a network with mobile hosts, the network must reroute connections as hosts move between cells.

We present and analyze several schemes for supporting connection-based services in mobile networks. As a basis for comparison, we use a Full Re-Establishment algorithm, which establishes entirely new connections every time a host moves. In contrast, we present the Incremental Re-Establishment algorithm, which modifies an existing connection by establishing only the portion of the channel between the base station and the node where the old and new channels would diverge. Our Multicast-Based scheme relies on network-layer multicast to deliver data to more than one base station during a rerouting.

We have performed an evaluation of these rerouting schemes using a simple analytical model. The analysis of these algorithms implies that our new schemes for re-establishing connections in mobile networks can provide improved performance compared to naive algorithms. By localizing the handoff processing, our algorithms reduce the handoff completion time by a factor of 2—3 and reduce the backbone network utilization by a factor of 4—5. In particular, the use of multicast services combined with the use of cell overlap information provide the greatest benefits. Our results also show that network topology is an important consideration in the design of mobile, connection-based networks. In particular, it is advantageous if physically adjacent base stations can be located with logical locality in the network.

3.7 Implications for Other Network Technologies

Most networks in use today do not use switch-based connection-oriented networks. We must especially examine how some of the concepts analyzed apply to datagram-based net-

works and broadcast links. Routing in a datagram-based network is similar to connection-oriented routing in several important ways. For example, datagram-based network routers also maintain tables used to route packets. The crucial difference is that the tables are indexed by destination instead of by connection. However, many of the concepts of localizing handoff processing by using multicast or incremental re-establishment apply to datagram networks as well. The introduction of broadcast links, such as Ethernet, creates one important change in our analysis: it reduces the cost of multicast greatly. Multicast delivery to base stations on the same broadcast link incurs no additional overhead. This makes multicast more promising in such networks.

In the remainder of this dissertation, we describe various implementations that utilize the concepts of hints, multicast, and intelligent buffering in environments consisting of broadcast links and datagram-based networks. These implementations were performed on the most common, easily available and best supported network technologies. This was to allow wide use of the resulting implementations and the demonstration of these handoff techniques in existing environments. Despite some of the differences in technology, the results of this analysis and the performance of the implementations indicate that the concepts are valid in a variety of environments.

Chapter 4

Support for Mobility in an IP-based Environment

The analysis in Chapter 3 showed the importance of multicast, hints and intelligent buffering in implementing low-latency routing update support. Based on the analysis, we implemented a routing protocol that dramatically reduces route-update latency by using the multicast infrastructure already available for IP networks. We choose to implement the concepts in a IP network since IP is the most common routing protocol and is the basis of the rapidly growing Internet. This implementation operates on the Daedalus testbed [Dae95], which uses off-the-shelf portable computers and wireless network hardware to create an IP-based mobile environment. This chapter presents the details and performance of this implementation.

4.1 Introduction

As mentioned in Section 2.3, the introduction of mobile hosts in the IP based Internet introduces many new problems. A number of solutions to this problem have been proposed and are described in that section. In most proposed solutions, including IETF Mobile IP [Perk95b], packets traversing the network during a handoff are lost or experi-

ence unusually long delays. The IETF Mobile IP system assigns a home network to each mobile host. When the route to a mobile host changes, the IETF protocol notifies a machine on this home network of the new route. This form of routing updates is similar to the full re-establishment algorithm described in Chapter 3. Since routing updates must propagate to the home network, the duration of the update is proportional to the network distance, or round trip time, between the mobile host and its home. The efficiency of this mobility scheme depends on either infrequent repositioning or roaming only in the home area.

Although mobility far from the home agent and frequent handoffs are not common today, this form of mobility is likely to occur more often in the future. While a route is being updated, the IETF protocol delivers packets enroute to the mobile host to the incorrect location. Since the mobile host has moved, the last hop router for these packets may either drop them or forward them to the mobile host's current location.

If the packets are dropped, the handoff results in the loss of all data enroute to the mobile host. This amount of loss is related to the product of the sustained bandwidth to the mobile host and the delay for changing the route of data. Typical bandwidths range from 2Mbps (wireless LAN) to 10Kbps (wireless WAN) and the minimum possible delays range from 10 milliseconds (near home mobility) to 100 msec (cross-country mobility). These values greatly depend on distance from the home agent, the connectivity of the MH and the implementation of Mobile IP. For wireless LANs, which will likely have the most frequent handoffs, we optimistically estimate losses to range between 4—40 Kbytes. This will cause many TCP connections to lose an entire window of data during such handoffs and video applications to lose several frames of updates. Alternatively, forwarding data after the routing stabilizes causes the data to be delivered to the mobile host after a long additional delay.

Both high delay jitter and variation in data loss are unsatisfactory for many standard multimedia applications and reliable protocols (such as TCP). Both multimedia applications

and reliable protocols adapt to long-term estimates of delay and packet loss between the data source and mobile host. However, they do not perform well when rapid variations violate their estimates. To support these applications and protocols, an IP-routing protocol for mobile hosts must provide communication with consistent data loss and delay. For example, the TCP Reno implementation can tolerate the loss of a few packets per window (typically a round-trip time) and the delay of an individual packet by approximately double the round-trip time of a connection. Delays or losses greater than these limits cause the TCP protocol to invoke congestion-avoidance procedures. This severely degrades the performance for some time after the handoff. Although other applications may have stricter requirements, we use the TCP tolerance as a design goal.

In this chapter, we describe the design and implementation of our IP routing protocol for mobile hosts. In our routing protocol, we apply the concepts developed for connection-oriented handoff to the support of IP routing to mobile hosts. The analysis of connection-oriented handoff showed that use of multicast and “hints” can provide the fast, data-loss free handoff we need. The hints provide the handoff protocol with location or movement information about the mobile host. The protocol obtains these hints from sources such as wireless network physical-layer measurements, base station (BS) layout and user movement prediction. Using hints, the handoff protocol can realize that the mobile host is nearing another cell and can begin to multicast data destined for the mobile host to other base stations nearby. This localizes the routing changes in the network and allows much of the routing to be set up in advance of the mobile host entering the nearby cell. We have designed and implemented a routing protocol that incorporates these concepts. It typically performs handoffs in 5—15 msec and nearly eliminates all handoff-related data loss.

The remainder of this chapter is organized as follows. In Section 4.2, we describe the basic algorithms of our protocol and highlight the difference between it and the IETF Mobile IP protocol. We describe the environment of our implementation in Section 4.3 and the details of our implementation in Section 4.4. Section 4.5 describes the measure-

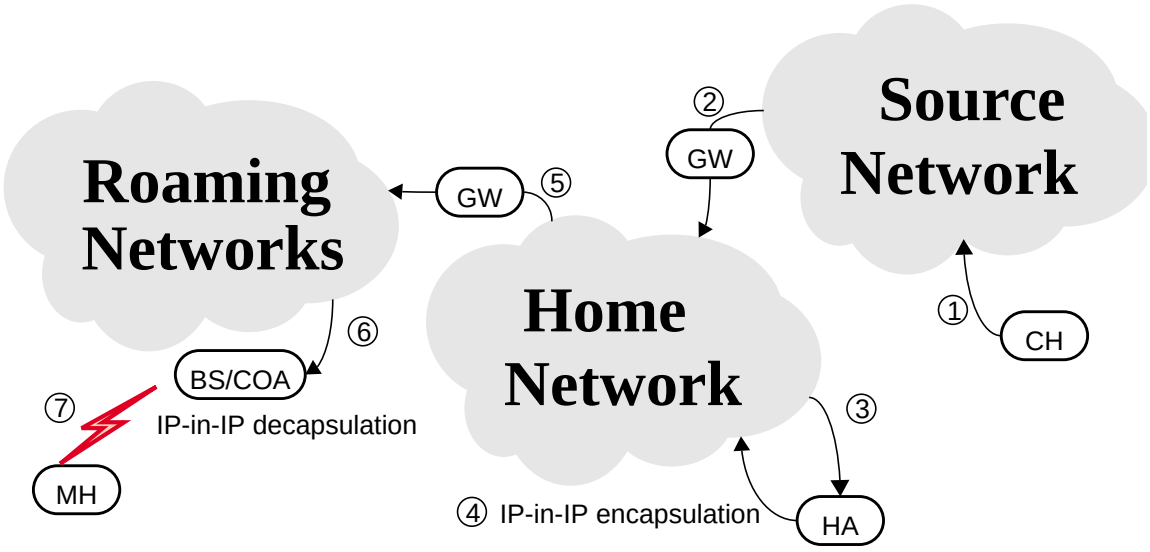


Figure 4.1 IETF Mobile IP Routing Encapsulation.

Numbering indicates order of links traversed in route of packet from CH to MH.

ments taken on our implementation. We present our conclusions about this protocol and implementation in Section 4.6.

4.2 Algorithm Overview

The problem of IP support for a mobile host splits into two parts: routing data from the mobile host and routing data to the mobile host. Fortunately, packets from mobile hosts to corresponding hosts (CHs) can use the normal IP routing system. As a result, the proposed mobile IP routing protocols concentrate on the delivery of packets to the mobile host. In most mobile IP proposals, there are 2 basic stages to the route of a packet to a mobile host. In the first stage, the normal IP routing delivers the packet to a special machine or agent that is capable of routing packets to mobile hosts. In the second stage of the routing, the agent forwards the packet to the mobile host's current location. In the IETF proposed protocol, a home agent uses IP-in-IP encapsulation to forward the packet. Figure 4.1 shows the routing used by the IETF Mobile IP protocol. Our mobile routing protocol uses the same first stage as the IETF Mobile IP protocol. However, our routing protocol differs

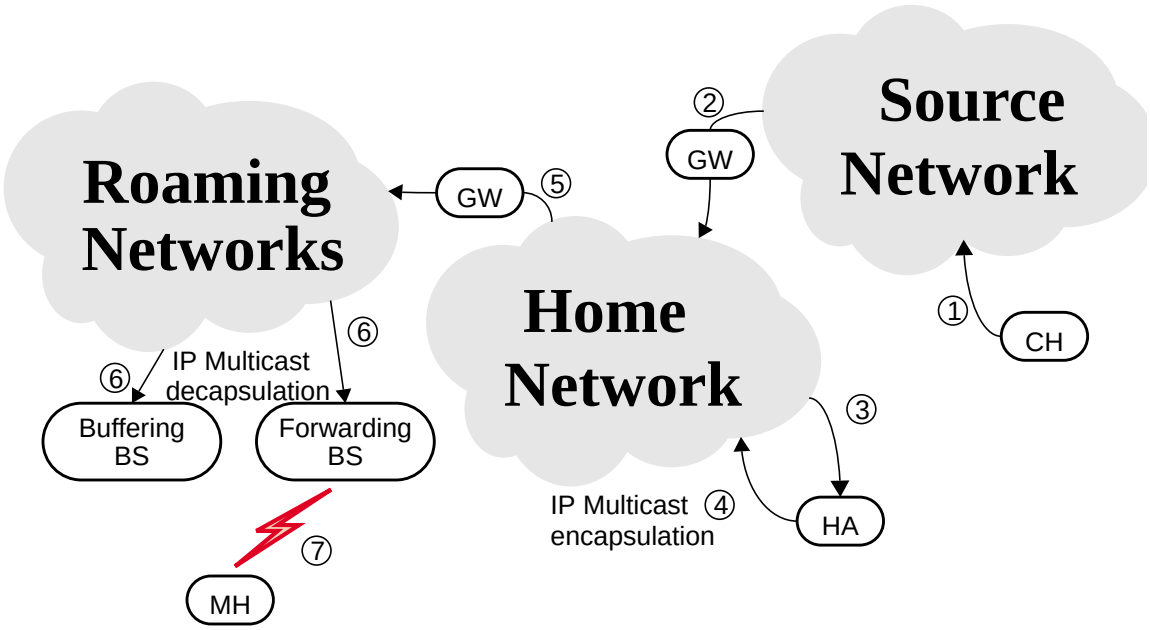


Figure 4.2 Multicast-based mobile IP Routing Encapsulation.

Numbering indicates order of links traversed in route of packet from CH to MH.

from Mobile IP in the second stage in order to support low-latency handoff and to reduce packet loss and delay variation during handoff. In this stage, our system uses multicast [Deer91], hints and intelligent buffering in nearby base stations to eliminate data loss and provide the consistent performance that applications and protocols require [Keet93]. We chose to use this form of second-stage routing based on our analysis of connection-oriented routing. The analysis indicated that multicast effectively reduces routing update time by localizing the changes and that, with the availability of hints, buffering eliminated the possible packet loss without the need for forwarding. Figure 4.2 shows the routing in our protocol.

In our scheme, there are three basic parts to the routing system:

1. Delivery to the home agent: Our scheme must provide must provide a mechanism that hides the mobile protocol and routing updates.

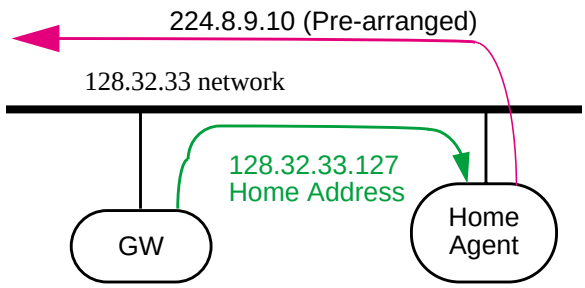


Figure 4.3 Home Agent Encapsulation.

2. Locating the mobile host: The routing system must determine the physical location of the mobile host. Our location algorithm must also provide the routing protocol with information about when and to where handoff is likely.

3. Delivery to the mobile host: The routing system must support the delivery of packets from the home agent to the mobile host.

Like the IETF proposal, we use the normal IP routing to route a packet to the mobile host's home agent (HA). This allows standard Internet machines to communicate with the MH. Our scheme assigns each MH a long-term IP address (home address) associated with its home location. The MH's home agent (HA) intercepts any packets transmitted to this home address using a proxy ARP (Address Resolution Protocol) mechanism. The HA is responsible for forwarding these packets to one or more base stations in the vicinity of the mobile host. In our implementation, we chose to use IP multicast to perform this delivery. Each MH is also assigned a temporary IP multicast address. There is a one-to-one mapping between multicast groups and mobile hosts to avoid confusion between mobile hosts. The home agent and mobile host must jointly choose the multicast address when the mobile user leaves the home network. The home agent encapsulates packets destined for the MH (when the MH is not at its home location) and forwards them to its associated multicast group. The members of this multicast group include the base stations in the vicinity of the mobile host, but the mobile host itself does not join the group. An example of this interception and encapsulation is shown in Figure 4.3. In the example, the multicast address chosen when the mobile host leaves its home area is 224.8.9.10.

The second task of the routing system is to determine the current location of the MH. Each BS periodically broadcasts a beacon message to all the MHs in range of it. Each MH keeps track of all the recent beacons it has received to approximate its current location and motion. The MH uses statistics such as the received signal strength of the beacons and communication quality to identify which BSs are nearby. The MH also determines which wireless network cell it should join as well as to which cells it is likely to handoff in the near future. Based on this analysis, the MH configures the routing between the HA and the various BSs.

The delivery of packets from the HA to the BS utilizes the dynamic routing provided by IP multicast. The BS responsible for the cell containing the MH joins the IP multicast group. This BS, the primary, forwards to the MH each packet transmitted from the HA on the multicast group. At any instant of time, there is at most one primary base station, also called the forwarding base station, in the system for a given mobile host. In addition, the MH requests BSs that are identified as likely handoff targets to join the multicast group. These BSs do not forward the packets from the multicast group to the wireless network. Instead, each of these BSs buffer the last several packets transmitted from the HA. Typically, handoffs occur to cells whose BSs have been primed for an MH. When an MH enters such a cell, the new primary BS begins transmitting packets from its buffer of packets. Since the data “in-flight” from the CH is delivered directly from the new BS without having to forward it from the previous BS, this handoff scheme has minimal data loss during handoff and incurs no delays due to data transfer. The routing of data between the HA and the MH is shown in Figure 4.4.

Handoffs are mobile-initiated and occur when the mobile host discovers a base station with a stronger signal than the current one. The sequence of events typical of a handoff is shown in Figure 4.5. In the figure, an MH is moving from BS1’s cell to BS2’s cell. The handoff begins when the MH receives a new beacon measurement. Based on its beacon measurements and some hysteresis, the MH computes that BS1 should be buffering packets and that BS2 should be forwarding packets. Using hysteresis avoids very frequent and

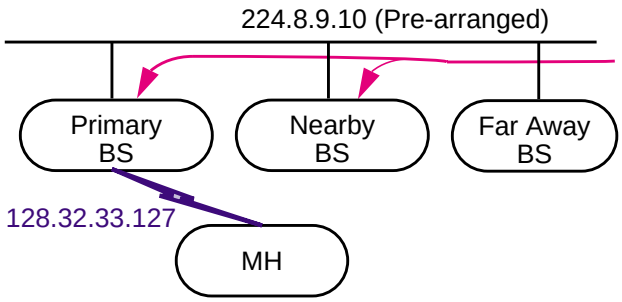


Figure 4.4 Home Agent to Mobile Host

unnecessary handoffs. After comparing the desired state for each of the BSs in the area with their current state, the MH transmits a set of control messages to the various BSs. These control messages request each BS to either begin or end forwarding and buffering of packets. At most one BS is in forwarding state at any instant of time for a given MH. While these requests are being delivered and processed, packets continue to arrive from BS1. The control message to activate forwarding on BS2 also includes the list of the last several packets received by the MH. This allows the new primary BS, BS2, to determine which packets in its buffer have already been delivered to the MH by the previous forwarding BS. After synchronizing its buffer, the BS2 begins forwarding packets from the buffer and the multicast group to the MH. We define the duration of the handoff as the time between the first request to the new base station and the arrival of the first packet at the mobile host.

Setting up the routing in advance greatly reduces the actual duration and amount of data loss of handoff. Most handoffs in our system complete in 5—15 msec and cause no data loss. This latency is independent of the number of network hops between the base stations, home agent or source. This allows handoff to complete without affecting the performance of data transfers using TCP or other protocols.

4.3 System setup/architecture

We have implemented the handoff protocol on a testbed consisting of IBM ThinkPad laptops and IBM PC compatible base stations running BSD/OS 2.0 from BSDI, communicat-

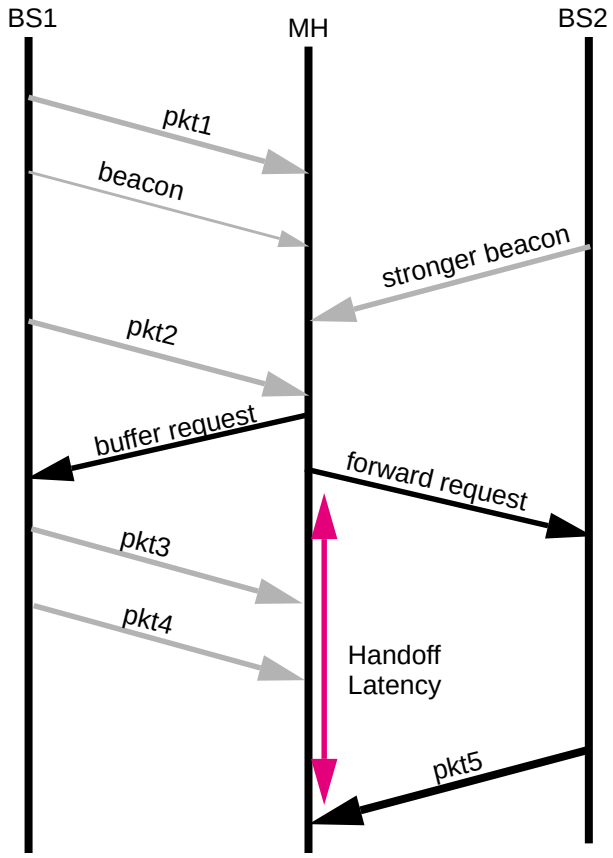


Figure 4.5 Typical handoff messaging.

ing over an AT&T WaveLAN. BSD/OS 2.0 is a BSD 4.4 derived operating system. We made several modifications to the network modules of the operating system to support the mobile routing protocols. In addition, we added special support for the low-delay IP TOS (type of service) option by adding a special low-delay packet queue for each network interface. When sending data to a network interface card (NIC), the operating system transmits data from this fast queue first. This allows packets requesting low-delay service to be delivered in advance of other packets.

The WaveLAN is a 915 MHz, wireless, direct-sequence spread-spectrum local-area network. It provides a shared, Ethernet-like link with a raw bandwidth of 2 Mb/s. Since WaveLAN is a broadcast-based network, all machines within radio transmission range share the link bandwidth. The WaveLAN network interface card provides an interface to

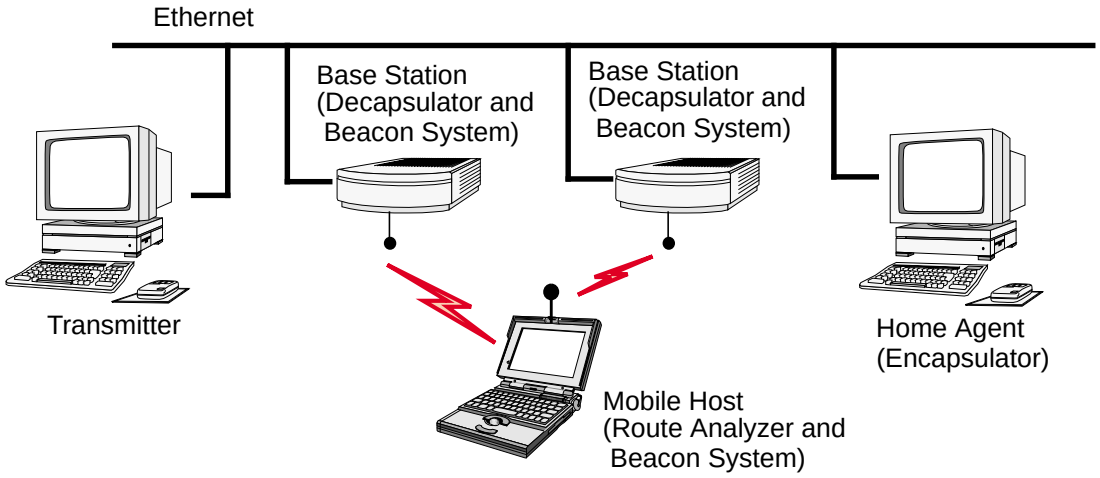


Figure 4.6 Network topology for experiments.

retrieve the radio signal strength, silence level and signal quality observed during the reception of a packet.

The network topology is shown in Figure 4.6. There are currently 4 PCs in our experimental system. One fixed location host is the source and sink of data during our experiments and another is a home agent for the mobile hosts. In our experiments, the mobile hosts handoff between the 2 PCs base stations on an Ethernet. We moved the base stations to other Ethernets for some experiments. The implementation on this testbed currently supports transfers to and from mobile hosts and supports smooth handoffs.

4.4 Implementation

The implementation consists of four distinct modules: the encapsulator, the beaconing system, the decapsulator and the route analyzer. The beaconing system allows the mobile host to identify its current location. The encapsulator at the home agent and decapsulators at the base stations perform the actual routing of packets between the corresponding host and the mobile host. The route analyzer uses the information provided by the beaconing system to configure the routing of packets. Figure 4.6 shows the location of these four

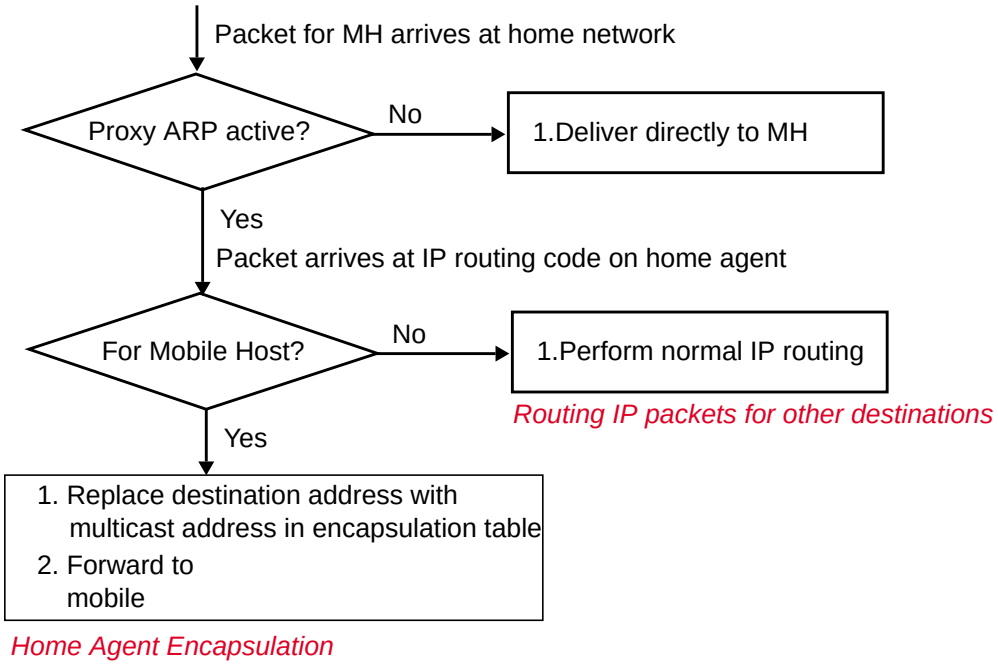


Figure 4.7 Flowchart for encapsulator

modules in our network testbed. In the following sections, we describe these various modules in more detail.

4.4.1 Encapsulator

The first stage of routing between a corresponding host and a mobile host is to deliver the packet to an entity that understands mobile routing, the home agent. When a mobile host leaves its normal home location, it must initialize the home-agent encapsulation. The home agent begins to intercept all packets destined for the mobile host and forward them to the IP multicast address assigned the mobile host. Encapsulation code added to the IP routing modules and a set of calls to configure the encapsulation and interception of packets implement the home agent in our system. The processing of packets by the home agent is shown in Figure 4.7.

When a mobile host is away from its home locations, a home agent uses the proxy ARP mechanism to respond to all ARP requests for the MH. However, the home agent returns

its own ethernet address to the transmitter of the ARP request. As a result, all packets destined for the mobile host are instead delivered to the home agent.

The IP routing code receives any packets that arrive at the home agent and are not addressed to itself. Once passed to the IP routing module, the home agent must identify that the packet is destined for a mobile host and encapsulate the packet. To support this processing, we made the following significant changes to the kernel of the home agent:

1. A home agent kernel contains an encapsulation table containing information about each mobile host for which it is responsible. Each entry in the list contains the IP address of a mobile host, the IP Multicast address to use in encapsulating packets for that mobile host, and the time to live (TTL) for encapsulated packets.

2. A pair of newly added IP level socket options, `ADD_ENCAP` and `DEL_ENCAP`, control the encapsulation. Both socket options take the home address of mobile host and the IP Multicast group to forward the packet to as passed values. The `ADD_ENCAP` call also requires the calling process to pass the TTL to use on the IP Multicast group. When a mobile host leaves his home address, the home agent joins the MH's IP Multicast group and uses the `ADD_ENCAP` socket option to add an entry to its encapsulation table. This initializes the encapsulation and forwarding. If the mobile host returns to its home network, the home agent uses the `DEL_ENCAP` to delete an entry from the table and has the base station leave the associated IP Multicast group. This socket option call combined with the deletion of the associated proxy ARP entry allows the mobile host to receive packets directly at its home address.

3. The IP routing module contains a filter to identify packets destined to mobile hosts for which this home agent is responsible. The filter searches the encapsulation table for the destination address of a packet. If the address matches that of a mobile host, the filter passes the packet to the mobile encapsulation code.

4. The mobile encapsulation module uses the information contained in the encapsulation table and the destination address of the packet to perform the forwarding and encapsulation. The module replaces the destination address field in the IP packet with the IP Multicast address associated with the mobile host. Finally, the home agent uses the normal IP packet output routines to forward the packet.

Once the packets are output, the IP Multicast routing delivers them to the base stations listening on the appropriate multicast groups. The route analyzer requests only the base station that is forwarding packets to the mobile host and base stations that are likely targets for handoff to listen on the mobile host's multicast group.

4.4.2 Beacon system

The beaconing system runs on the mobile hosts and base stations. Its primary responsibility is to provide the route analyzer with information about the location of the mobile host. The beaconing system passes the route analyzer the IP address and communication quality of each base station that is within range of the mobile host. The route analyzer uses this information to identify which base stations should listen on the multicast group associated with a mobile host.

The WaveLAN NICs (network interface cards) provide several useful measurements associated with the reception of a packet. They give the signal strength of the packet transmitter, the background noise at the time of reception and the signal quality of the received packet. However, these measurements must be retrieved upon the reception of a packet from the NIC. To simplify the collection of this data, a process at the base station periodically sends beacon packets. At the mobile host, we added a set of calls to help identify these beacon packets and retrieve the associated measurements.

At the mobile host, two new `ioctls`, `ADD_BEACON` and `DEL_BEACON`, control the retrieval of radio measurements from the WaveLAN NIC at the mobile host. The information passed to these calls include the IP address and WaveLAN physical layer address of a

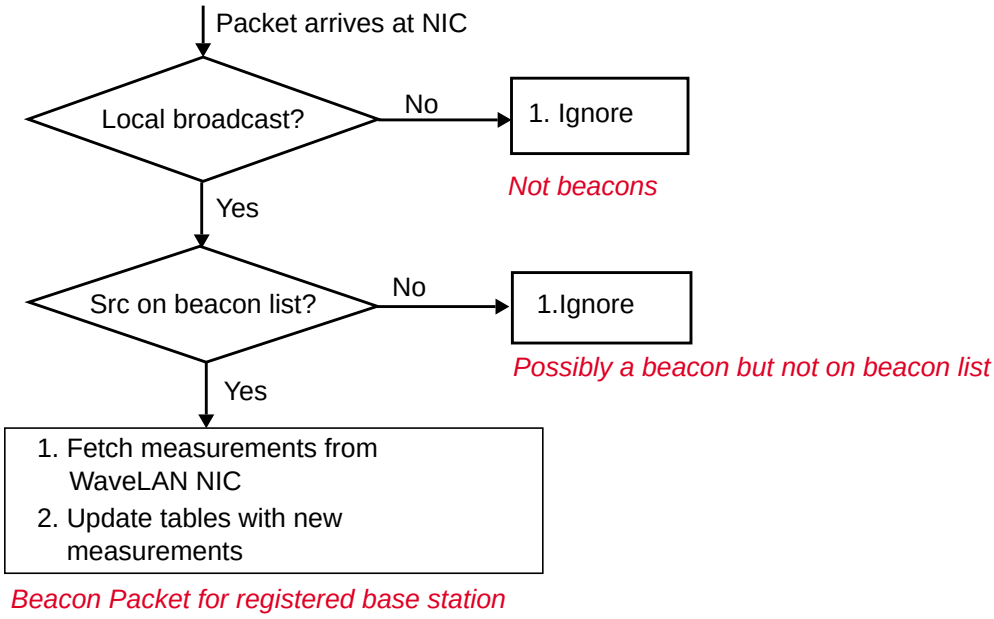


Figure 4.8 Flowchart for beacon processing.

base station. The `ADD_BEACON` `ioctl` adds the passed address to a table of base stations in range; whereas, the `DEL_BEACON` call removes an entry from this list. When a WaveLAN broadcast packet arrives from one of the base stations in the table, the device driver retrieves the signal measurements from the NIC and stores it. The device driver currently stores the last 3 samples for each base station. Figure 4.7 shows the filtering and retrieval of physical layer measurements.

A user level process can retrieve these radio measurements using the BSD kernel memory interface, `kvm`. We wrote a simple library that mapped the entire table of base stations and measurements into the memory of a user level program. The library also provides support for retrieving the samples of a specific base station.

The beacon packets contain the IP addresses, for wired and wireless interfaces, of the source base station and a time stamp. These beacons are UDP packets transmitted to the WaveLAN broadcast address and the beacon service UDP port. A beaconing daemon at each base station transmits a beacon packet once per second. At this beaconing rate, we

observe approximately a 10% degradation in the aggregate bandwidth of the WaveLAN network.

At the mobile host, the route analyzer uses the `ADD_BEACON` and `DEL_BEACON` ioctls to update the list of base stations that are in range of the mobile host. This allows the mobile host to have relatively fresh information about the communication quality to each of the nearby base stations.

4.4.3 Decapsulator

Decapsulators that run at each base station perform the final stage of the routing of a packet from corresponding host to a mobile host. A route analyzer on a mobile host requests one or more decapsulators in its vicinity to receive packets for a mobile hosts. To receive the packets, the requested base stations join the IP Multicast group associated with the mobile host. The route analyzer chooses a single base station in its area to be its primary. The decapsulator on this base station forwards the decapsulated packets across the wireless network to the mobile host. The other decapsulators that are receiving packets for the mobile host buffer the last few packets. The decapsulator consists of two major sections: a set of changes to the IP routing code and a user-level daemon that receives and processes requests from the mobile hosts. The division of responsibility between these modules is shown in Figure 4.9.

We modified the IP routing module to perform the decapsulation, forwarding and buffering. The IP routing code receives any packets that arrive at the base station and have an IP Multicast address. The decapsulator scans all multicast packets to identify any that are destined for a mobile host. It then processes the packet based on the current state of decapsulation (either primary forwarding or nearby buffering) for the mobile host. Figure 4.7 shows the processing of multicast packets at the decapsulators. To perform the necessary tasks, we made the following significant changes to the kernel of the home agent:

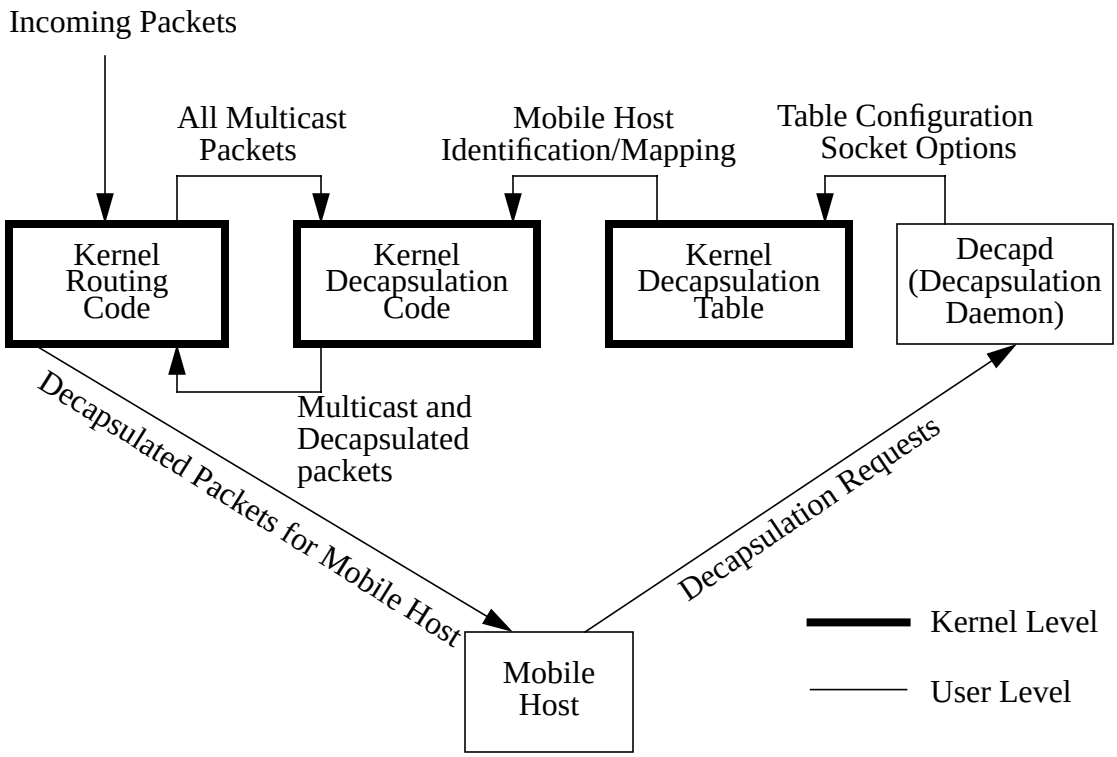


Figure 4.9 Decapsulator Layering

1. A decapsulation table in the base station kernel contains information about each mobile host for which the base station is responsible. An entry for a mobile host in the list contains its associated IP Multicast group, home IP address, and the current state of decapsulation from this base station. The decapsulator on a mobile host's primary base station is in the forwarding state. The other base stations that have been requested to join the multicast group associated with the mobile host have decapsulators in the buffering state. While in the buffering state, a decapsulator keeps the last 12 packets it has received that are destined for the mobile host. We chose to buffer 12 packets based on our measurements of handoff times. The decapsulator uses this cache of packets to prevent data loss of packets "in-flight" during handoff.

2. A set of newly added IP level socket options, `ADD_DECAP`, `DEL_DECAP`, `ENA_DECAP` and `DIS_DECAP`, control the decapsulation. The information passed to these socket options include the IP Multicast group and the home IP address associated

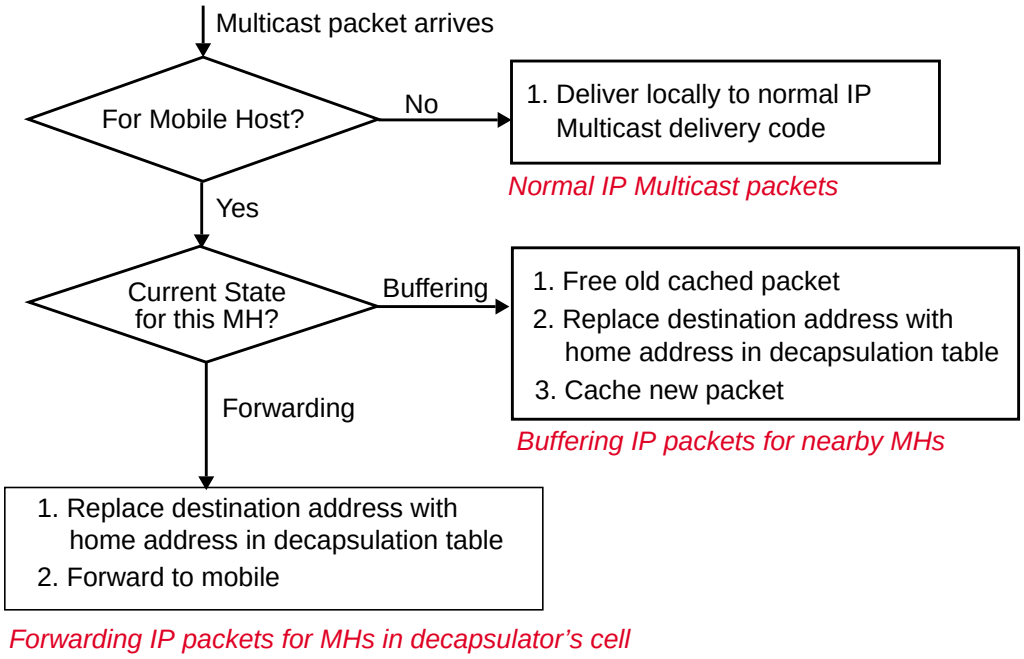


Figure 4.10 Flowchart for decapsulator.

with a mobile host. When a mobile host enters a base station's vicinity, the decapsulator uses the ADD_DECAP socket option to add an entry to decapsulation table and to join the base station to the MH's IP Multicast group. The initial state of decapsulation is to only buffer packets for a mobile host. When the mobile host enters the base station's cell (i.e., chooses the decapsulator as its primary forwarder), the system uses the ENA_DECAP call to change the state of decapsulation from buffering to forwarding. During the change to forwarding state, the base station forwards to the mobile host any packets that were stored while the decapsulator was in buffering mode and have not yet been delivered to the mobile host. This eliminates any loss of packets enroute to the mobile during handoff. To identify which packets to transmit from the cache, the system passes the IP IDs of the last 3 packets received by the mobile host. The ENA_DECAP call performs a linear search through the buffer of packets for the IP IDs. The socket option discards all packets older than those matching the IP IDs and transmits the remainder to the mobile host. Once the mobile host leaves the cell, the decap-

sulator returns to the buffering state using the `DIS_DECAP` call. Finally, the `DEL_DECAP` call deletes the decapsulation entry from the table and has the base station leave the associated IP Multicast group when the mobile leaves the area.

3. The IP routing module contains a filter to identify packets destined to mobile hosts for which this decapsulator is responsible. The filter searches the decapsulation table for the multicast address of a packet. If the address matches that associated with a mobile host, the filter passes the packet to the decapsulation code.

4. The decapsulation module uses the information contained in the decapsulation table and the destination multicast address to perform the final delivery of a packet. The module replaces the destination address field in the IP packet with the home IP address of the mobile host. The base station must have a host route entry in its routing table for the home IP address of the mobile host. This entry forces the packet to be routed across the wireless network to the actual mobile host. We accomplish this by giving each mobile host a temporary IP address in the region's wireless network and setting up a route entry for the home address through this temporary address. This allows the base station to use the normal IP packet output routines to forward the packet to the mobile host.

A user level daemon, `decapd`, performs many of the control functions needed in the decapsulator. When a mobile host comes within range of a base station, the route analyzer on the mobile host creates a TCP control connection between the route analyzer and the decapsulation daemon. When the TCP control connection is created, the `decapd` process adds a routing table entry between mobile host's home and temporary address. `Decapd` accepts four different forms of requests on these connections. The requests map directly to the four different decapsulation socket options available on a base station. The route analyzer uses the information provide by the beaconing system to decide which base station should be forwarding packets and which stations should be buffering packets. It communicates these decisions to the decapsulators across these TCP control connections. By pro-

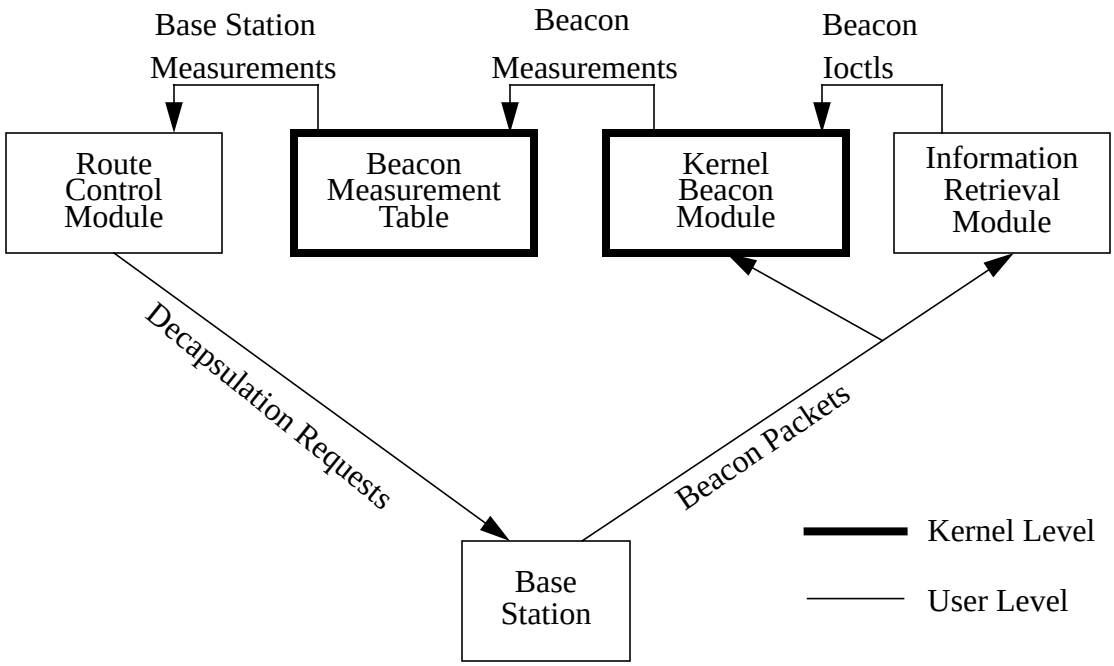


Figure 4.11 Route Analyzer Division

cessing these packets a decapsulator puts the base station into the state desired by the mobile hosts.

4.4.4 Route Analyzer

The route analyzer uses the information and functionality provided by the decapsulators and beaconing system to determine and control the route of packets to the mobile host. There are two major sections to the route analyzer: a module that interacts with the beaconing system and a module that interacts with the decapsulators. The responsibilities of these modules are shown in Figure 4.11.The following sections describe these modules in detail.

4.4.4.1 Information Retrieval Module

The information retrieval module is responsible for identifying what base stations are nearby and retrieving any information the wireless network can provide about these base stations. This module is the collector of "hints" about user mobility available from the system. In the WaveLAN based system, these hints come from the signal measurements available from the NIC. The beaconing system provides a mechanism to filter and retrieve measurements. The information retrieval module chooses which measurements to filter and retrieve.

At the mobile host, the information retrieval process receives all packets destined for the beacon service UDP port. When a beacon packet arrives, the process attempts to read the in kernel radio measurements associated with the transmitting base station. If the WaveLAN device driver has the base station in its list of nearby base stations, the process retrieves the beacon measurements. Otherwise, the process performs an `ADD_BEACON` `ioctl` to register this base station as one that is in range of the mobile host. It also creates a TCP control connection to this base station for use by the route control module. If a beacon has not been received from a base station for 5 seconds, the module determines that the base station has gone out of range of the mobile host and uses the `DEL_BEACON` `ioctl`. This allows the mobile host to maintain relatively fresh information about the communication quality to each of the base stations nearby. In addition, it also provides the mobile host with a reliable communication channel to each base station in range.

4.4.4.2 Route Control Module

The route control module uses the data gathered by the information retrieval module to identify the mobile host's connectivity. It determines which base station is best for the mobile host to route data through as well as to which base stations the mobile host is likely to handoff in the near future. It uses this information to configure the routing of the nearby base stations.

Each time a beacon arrives at the mobile host, the information retrieval system provides the route-control module with a new list of nearby base stations. This route control module sorts the lis in order of decreasing signal strength. Normally, the route control module chooses the base station with the best signal strength as the primary. However, the module uses hysteresis to avoid changing primary base stations too frequently. The strongest signal must be at significantly better than the current primary base station's signal before the route module initiates a handoff. We tuned the amount of hysteresis based on measurements in the building and experience with the radio network. The module also identifies several of the base stations with the strongest signals as likely handoff targets. The module requests these base stations to buffer packets for the mobile host. The module has three different policies that it can use to determine the number of buffering base stations. The first scheme requests all base stations in range to buffer packets for the mobile host. Another model has a fixed number of base stations buffering at all times. The last model adapts the number of buffering base stations to the rate of the mobile host's motion. The module estimates the motion of the mobile host using the total magnitude of signal measurement changes in the past few seconds. There are three definable thresholds that categorize the motion as none, slow or fast. For each class of motion a different number of base stations buffer for the mobile host. The choice of model should be based on several factors including: the characteristics of user movement, the availability of backbone network resources for multicast, the cost of multicast and the desired "hit-rate" of handing-off to base stations that have routing already set up. We discuss the tuning of these parameters in Section 4.5.5

Once the route control module has determined which base stations it desires to forward or buffer packets, it must communicate these decisions to the decapsulators on the base stations. The module sends a control packet to each base station that needs to change state across a TCP control connection. In order to deliver this packet quickly, the module transmits this packet with the low delay IP TOS flag set. This message contains the new state the base station should be in and the IP IDs of the last three packets received by the mobile

host. At the base station, the `decapd` process reads these messages and performs the appropriate decapsulation socket options. The state changes at the different base station comprise the handoff of a mobile host from one cell to another and set up the routing desired by the mobile host.

4.5 Measurements

There are two important goals of our routing protocol: avoid data loss due to handoff and prevent delay jitter due to handoff. In addition, the routing should not result in:

1. End-to-end performance restrictions
2. Inefficient routing
3. Significant overhead in routers or other machines

We have taken several measurements to identify if the multicast-based routing meets these goals.

To measure the performance of the system, we developed a benchmark tool called `net-perf`. This program allows us to output UDP/IP, TCP/IP and UDP/IP Multicast packets of a fixed packet size. The program can either limit its output rate or transmit at the maximum rate possible by the sender. We used the program to generate traffic between the corresponding host and the mobile host. The program outputs the average throughput observed for a transfer as well as intermediate throughput measurements during the transfer. In addition, we obtained measurements by using `tcpdump` [McCa93] on another machine connected to the wireless network. This provided information about the end-to-end performance of our systems as well as detailed timing of events.

To isolate the impact of handoff on performance, we performed transfers between fixed and mobile hosts while regularly spaced handoffs between two base stations occurred. As mentioned in Section 4.4.2, each base station sends out a beacon signal once per second. In our experiments, the arrival of a beacon at the mobile host did not trigger an analysis of

the signal strengths of the different base stations; instead, the mobile host used the time elapsed since the last handoff to determine if a handoff should occur. In order to stress the performance impact of the handoff scheme on end-to-end performance, we performed several tests, varying the time between handoffs from 1 to 10 seconds.

The parameters we are interested in measuring or analyzing are the overall end-to-end throughput (and degradation from the maximum), the handoff latency, the number of dropped packets, the network utilization and the buffer requirements at the base stations. Although the overall end-to-end throughput is what applications perceive, understanding end-to-end effects requires a closer analysis of what happens to individual packets during a handoff. This motivates our choice of the handoff latency and number of lost packets as performance metrics. We describe the results of using both UDP and TCP as higher-level transport protocols.

To identify the disruption to both UDP and TCP transfers, we examined the sequence of events during handoff. We also examined UDP transfers with and without handoffs occurring to identify the data loss and latency of messaging during a handoff. Finally, we examine the end-to-end performance of our protocol by comparing TCP performance of transfers using our protocol with transfers using normal IP routing. We examined three variants of the multicast-based handoffs:

1. *Multicast routing with buffering base stations:* This is the best case for our handoff scheme. Here, the new base station that the MH transfers to has already joined the multicast group and is receiving and buffering packets for the MH prior to the handoff. In practice, this happens if the MH has a good idea of its location and sets up the state of the decapsulation at the new base station in advance. In this case, we expect the best handoff latencies (relative to the other experiments) and almost no packet loss.

2. *Multicast routing with no buffering:* In this case, the new base station joins the multicast group corresponding to the MH but does not buffer packets on its behalf. We did this experiment to isolate the effects of route updates and buffering. In practice, this may happen when the MH decides that requiring the BS to buffer packets for it in advance would unnecessarily consume buffer resources in the network.

3. *No multicast, no buffering:* This is the pathological worst case of the algorithm and could be caused by highly unpredictable movement by the MH. We did this set of experiments mainly to illustrate the benefits of buffering packets in advance and multicast-based route management.

In observing the low-level timing of packets to different base stations, we observed some peculiar media-access delay variations in the WaveLAN network. When multiple WaveLAN devices are attempting to access the media, unfair access patterns occasionally appear. One of the devices grabs the media and successfully transmits many packets. Even when it has completed transmission, the other devices may wait several hundreds of milliseconds before transmitting. This poor multiple access performance occasionally appears as part of the beaconing and handoff messaging. This results in some unusually delayed beacons and prolonged handoffs. We believe that this phenomenon also causes the peak bandwidth of transfers to a mobile host to drop 10%, from 1.6 Mbits/sec to 1.45 Mbits/sec, in the presence of beacons. To identify the actual performance of the handoff protocol without the impact of this unfair media-access protocol, we have removed some of the affected measurement samples from our runs and analysis.

4.5.1 Handoff latency

We measured the disruption caused by our routing protocol by examining the sequence of handoff events during UDP transfers. We obtained the timings of events from analyzing `tcpdump` traces of handoffs during UDP runs. Handoffs were injected every 10 seconds during the runs. Transfers were rate controlled to 1.4Mbits/sec. In the first set of runs the base stations were located on the same Ethernet. In addition, the two base stations were

always either forwarding or buffering packets for the mobile host. This is the fastest handoff possible in our system and is expected to be typical of most handoffs. In other runs, we placed the base stations 1, 2 and 3 network hops apart. We also examined latencies for handoffs to base stations that had joined the multicast group but were not buffering packets for the mobile host and base stations that had not joined the multicast group. Table 4.1 summarizes the handoff latencies observed for these different situations.

Handoff method	Distance between BSs	Handoff latency
Multicast-based with buffering	0 hops	8—15 msec
	1 hop	8—15 msec
	2 hops	8—15 msec
	3 hop	8—15 msec
Multicast-based and no buffering	0 hops	8—15 msec
	1 hop	10—15 msec
	2 hops	10—15 msec
	3 hops	10—15 msec
No multicast and no buffering	0 hops	15—20 msec
	1 hop	18—23 msec
	2 hops	20—25 msec
	3 hops	25—30 msec

Table 4.1 Handoff latencies (msec)

A time chart of the different stages of handoff processing during the best case handoff runs is shown in Figure 4.12. The x-axis shows the observed range of times between events. The variation in delays is mainly due to network queueing at the MH and BS and delays in the MAC layer of the Wavelan network. During idle periods, handoffs complete on the low end of the time ranges listed, with handoff latencies on the order of 8—10 msec. The

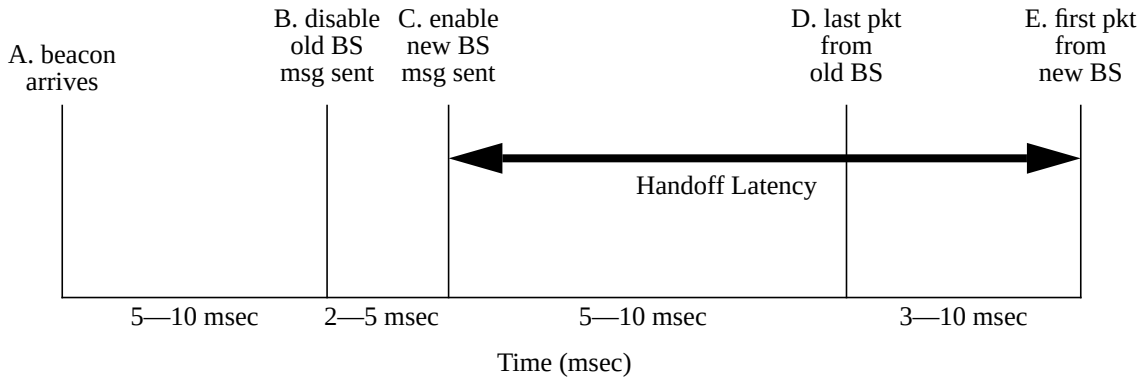


Figure 4.12 Timing of typical/best case handoff events

handoff latency grows to around 15 during busier periods. The mobile host only experiences a disruption in communication between events D and E. Therefore, the typical disruption is only 3 msec and the peak disruption is about 10 msec. We observed very similar handoff performance for the situation where the new base station has joined the multicast group but is not buffering for the mobile host.

The worst case for our routing algorithm occurs when the target base station of a handoff is several network hops away and has not been buffering packets or joined the multicast group for the mobile host. In this situation, the mobile host must first request the base station to join the appropriate multicast group before requesting it to forward packets. The time taken for the new base station to join the multicast group depends on the number of network hops between it and the nearest member of the group. Typically, the number of hops is the same as the distance between the new and old base stations. Our measurements have shown that a join of a multicast group takes between 2 and 5 msec per hop. The timing for this handoff is shown in Figure 4.12. In the case of base stations 3 hops apart, the hand off takes between 15 and 30 msec to complete.

4.5.2 Packet Loss

When a handoff occurs, the mobile host transmits the IP IDs of the last three packets it has received to the new primary base station. If this base station had been buffering packets for the mobile host, it attempts to prevent data loss by transmitting some of the packets

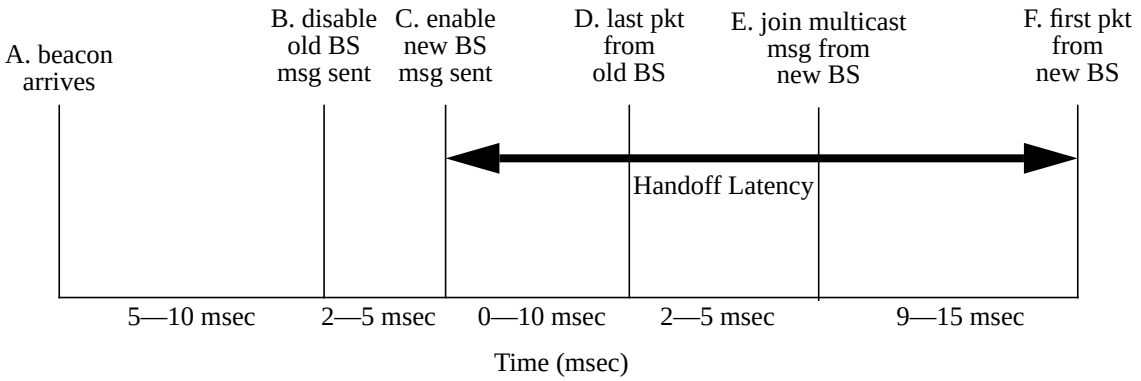


Figure 4.13 Timing of worst case handoff events

that arrived after the packets identified by the IP IDs. Therefore, packet loss only occurs when either an insufficient number of packets are sent from the cache or the base station had not been buffering packets. This mechanism also creates occasional duplicate packets. The packet loss for different situations is shown in Table 4.2.

We can characterize the first source of losses by examining packet losses for UDP transfers in which the mobile host periodically hands-off between two base stations. The purpose of forwarding data from the cache at the base station is to prevent the loss of packets that are enroute to the mobile hosts between events D and E in Figure 4.12. The maximum time between these events is 10 msec. Since a minimum sized packet takes 2 msec to transmit on the WaveLAN, we felt that buffering a maximum of 12 packets was more than sufficient to have all “in-flight” packets in the cache. The route analyzer at the mobile host retrieves the IP ID data between events B and C in the time chart. The decapsulators use this information to synchronize the base station cache at approximately the time of event E. In lightly loaded situations, the base station uses the IP IDs about 5 msec after acquisition and the packets identified are usually in the cache. The base station needs to forward few packets when the traffic is light since few packets are “in-flight”. When under heavy load, the IP IDs can be up to 15 msec out of date and some of the packets after those identified may have already been transmitted to the mobile host. It is difficult to distinguish between the light and heavy load situations at the base station. Therefore, the system must be tuned to either incorrectly identify packets as delivered during light load periods or as

Handoff method	Distance between BSs	No. of lost packets
Multicast-based with buffering	0 hops	0
	1 hop	0
	2 hops	0
	3 hops	0
Multicast-based and no buffering	0 hops	2—3
	1 hop	3—4
	2 hops	3—4
	3 hops	3—4
No multicast and no buffering	0 hops	2—3
	1 hop	3—4
	2 hops	4—5
	3 hops	4—5

Table 4.2 Number of packets lost during handoff

undelivered during heavier load. In addition, there are occasions where WaveLAN media access delays under very heavy load cause the IP IDs to be very old. As a result, the packets identified are usually not in the cache. In these situations, the system should forward several packets to avoid data loss. From experience with the system, we choose to forward the entire cache when the IP IDs were not found in the buffer (very heavy load situations) and forward packets that arrived after those identified when they were found (light or heavy load). This technique resulted in very few packet losses and occasional transmission of duplicate packets during heavy load. We performed several 1.0 Mbits/sec UDP transfers of 1024 byte packets to the mobile host. For handoffs to base stations that are buffering packets for the mobile host, the decapsulator forwards 2—3 packets from the cache of packets resulting in no packet loss during most handoffs. Under a heavier load, 1.4Mbits/

sec, the new base station needs to send 3—4 packets need from the cache. The “staleness” of the IP IDs result in 4—5 packets being transmitted. Therefore, heavy load handoffs usually cause one duplicate packet to be transmitted on average by the new base station. The addition of a simple duplicate detection mechanism at the mobile host can prevent the delivery of these duplicate packets to applications.

The second source of errors, handoffs to non-buffering base stations, is much more difficult to characterize. User mobility patterns and the policy used to identify likely handoff targets determine the frequency of handoffs to base stations that are not in buffering mode. Unfortunately, user mobility patterns in our environment are not well understood and as a result, the frequency of these handoffs cannot be calculated. We can identify how many packets are lost by examining the handoff delays. The duration of the handoff messaging and the time needed for the new base station to join the multicast group determines the number of packets lost. As for the case with buffering, approximately 3—4 packets are lost during handoff due to the overhead of the messaging. This loss is typical of handoffs to base stations that have already joined the multicast group. For the worst case handoffs, packets are also lost during the period the base station is joining the multicast group, 2—5 msec per hop to the multicast group. This results in the loss of about 1 additional 1 KByte during handoff for every 2 hops in our tests. The peak loss in our tests was 5 KByte and occurred when the base stations were 3 hops apart. Such a loss corresponds to about 30 msec of transfer time or one video frame or data.

4.5.3 End-To-End Performance

To isolate the impact of handoff on end-to-end performance, we examined TCP transfers with regularly spaced handoffs between two base stations. In order to stress the impact of the handoff scheme on end-to-end performance, we performed several tests, varying the time between handoffs from 1 to 10 seconds. The results for different handoff rates are shown in Table 4.3. These measurements show that even frequent handoffs have very little impact on performance. In a real environment, handoffs are likely to occur much less fre-

Time Between Handoffs (seconds)	Throughput (Mbits/second)	Standard Deviation (Mbits/second)
1	1.42	.011
2	1.43	.016
3	1.43	.012
5	1.43	.014
8	1.44	.012
10	1.43	.012
∞	1.45	.011

Table 4.3 Throughput received by the mobile host at different handoff frequencies

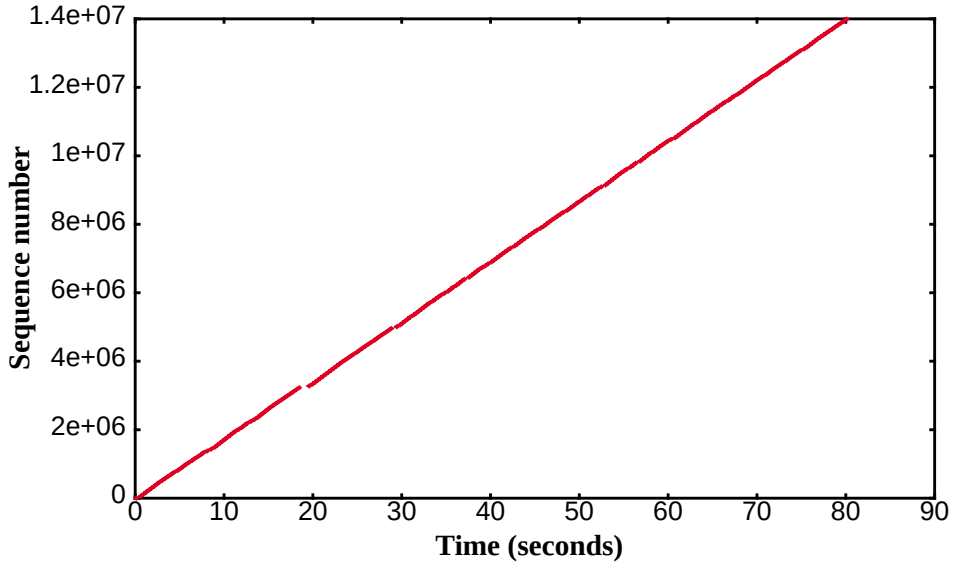


Figure 4.14 Sequence numbers for transfer to mobile host over channel with handoffs every 10 seconds.

quently than once per second. The behavior of a connection experiencing handoff is shown in Figure 4.14. The figure plots the sequence numbers of a TCP connection to a mobile host with handoffs occurring every 10 seconds. We see that the data transfer progresses without any significant interruptions despite the presence of the handoffs. The throughput during this transfer was consistently about 1.4 Mbits/s. The significant pause

observed at the 20 second mark is a result of the WaveLAN media access problems mentioned before.

4.5.4 Implementation Complexity

The breakdown of line counts for different modules is shown in Table 4.4. Although the

Module	User Line Count	Kernel Line Count
Beacon Daemon (`beacond`)	200	
Route Analyzer Daemon (`wrouted`)	1800	
Decapsulator Daemon (`decapd`)	500	
Kernel Routing Changes		150
Socket Option Code		500
WaveLAN Driver Beacon Code		200
Misc. Kernel Code		350
Misc. User Libraries	2000	
Totals	**4500**	**1200**

Table 4.4 Lines of code in different modules

line counts for the kernel modules are lower, these sections introduced far more implementation complexity than the user level code. The kernel code was difficult to debug and common execution paths, especially in the routing code, had to be carefully tuned to reduce overhead. Most of the user level code performed control functions and could provide adequate performance without optimization.

4.5.5 Overhead Analysis

To identify the overhead caused by our network protocol, we needed to identify the best possible performance. We set up routing tables in our network such that packets could be

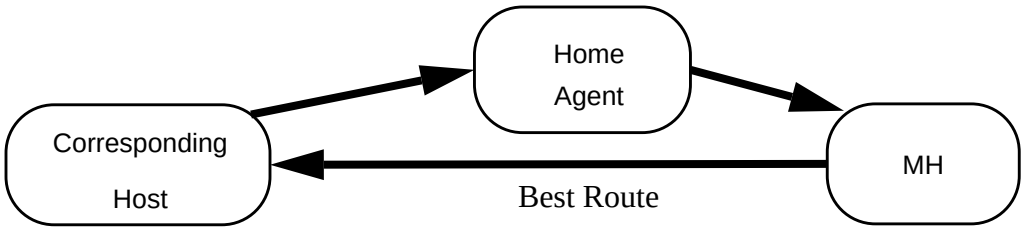


Figure 4.15 Example of Triangle Routing

delivered to one of our laptop computers using normal IP routing. The system has a peak end-to-end throughput of 1.6 Mbits/sec using either TCP or UDP. However, the activity of the beaconing system reduces this throughput to 1.45 Mbits/sec. We believe that this 10% degradation is a result of the media access problems mention earlier. From our end-to-end performance measurements (Section 4.5.3), we see that the routing protocol is capable of providing the full bandwidth (1.45 Mbits/sec) available. This indicates that the basic encapsulation and decapsulation do not impose a significant overhead on transfers. There are two basic sources of inefficiency in the protocol: the routing of packets to the base stations and the overhead of buffering packets at several base stations.

Our routing uses a home agent and IP Multicast to deliver packets to the mobile host. This introduces two well-known sources of inefficiency into our routing.

1. Triangle Routing: Since all packets must pass through the home agent, packets may not follow the best route between source and destination. This problem is known as triangle routing (Figure 4.15) because of the triangle created by the various paths. When the home agent resides along or near the best route between the mobile host and corresponding host, the route taken by packets to the mobile host is reasonably efficient. However, packets may experience high overheads when the home agent is far from the best route. To avoid this problem, the corresponding host must also be able to transmit packets to the forwarding and buffering base stations of a mobile host. This type of route optimization is difficult as long as we use IP Multicast to deliver packets to multiple base stations. For example, if a route optimizing corresponding host transmitted

packets to the IP Multicast group of the mobile host, IP Multicast routing would deliver the packets to all other members of the multicast group, including the home agent and other route optimizing hosts. We need a multicast mechanism that distinguishes between transmitters and receivers to perform the route optimization. This would allow the different senders, the home agent and route optimizing corresponding hosts, to transmit packets directly to the different base stations forwarding and buffering packets for a mobile host.

2. We chose to use IP Multicast to deliver packets to multiple base stations. However, many other mechanisms can be used to perform the necessary delivery of packets. The drawbacks of the IP Multicast currently used, DVMRP [Deer89], include inefficient support for wide area, sparse membership groups and high overhead of maintaining multicast routing tables. The delivery of packets from the home agent to a few, possibly far away base stations is not well suited to DVMRP based IP Multicast. Future multicast protocols may be better suited to the task. However, we can also support the delivery necessary with a much more limited multicast protocol designed specifically for mobility support.

Another significant form of overhead in the protocol is the buffering of data at base stations. The measurements of the routing protocol indicate that packet loss and handoff latencies are significantly lower for handoffs to buffering base stations. However, buffering consumes both base station memory and network bandwidth. Therefore, we want to avoid buffering at base stations that the mobile host does not visit. We refer to the percentage of handoffs that occur to buffering base stations as the “hit rate” of the mobility hints and the average number of base stations buffering for a mobile host as the “consumption rate”. There are several important factors that affect these rates:

1. Physical layout of base stations: This determines which base stations a mobile host can hear and the relative signal strength of the reception. As a result, this affects the number of base stations that may be buffering for a mobile host. If base stations have a high degree of overlap in cell coverage (i.e., they are close together in comparison to the cell

radius), a mobile host will have a difficult time identifying the primary handoff targets. If there is insufficient overlap, holes will appear in the wireless network coverage, adversely affecting connectivity.

2. The mobility patterns of users: This impacts the “hit rate” of handoffs. Certain mobility patterns may need many more buffering base stations for reasonable performance. If a user moves between cells as often as the location hints are updated, he or she is more likely to have handoffs to secondary or tertiary handoff targets. If an environment contains a large number of such users, the system should increase the number of buffering base stations to improve the hit rate of hints. However, this improved hit rate must be balanced with the resulting increase in consumption.

3. The route analyzer buffering base station policy: The route analyzer determines how many base stations are buffering for a mobile host at one time. Matching the policy to the type of user mobility in the system can result in the use of very few buffering base stations to provide a high hit rate for the hints. For example, if the typical user has a high mobility or handoff rate, the system should force many of the base stations that are in range of a mobile host to buffer for it. If the user community is more heterogeneous, an adaptive strategy will work better.

4. The wired network layout of base stations: If base stations are on the same physical, broadcast-based networks, the multicast overhead is very low. Base stations that tend to buffer for the same mobile hosts should be placed close together in the network. The actual overhead of multicasting data to multiple base stations is analyzed in Chapter 3. To reduce the overhead, the placement of base stations should follow the algorithms presented in that chapter. This placement places base stations that often handoff to each other on the same networks. This localizes the multicast necessary to support the mobility hints.

User mobility traces are necessary to tune each of these factors. Unfortunately, our system is still too early in its operational use to provide these traces, and other existing systems have not made them available. However, once the system is tuned, the overhead of buffer-

ing at a base station simplifies to the use of bandwidth (equal to the wireless link bandwidth) on a few wired network hops between nearby base stations and the use of small buffers (12 packets) in the buffering base stations. Given the increase in bandwidth of wired networks and the modest magnitude of the buffering memory required, this overhead is quite reasonable for the handoff performance gains.

The use of IP Multicast makes the current implementation of the multicast-based routing protocol most suitable for campus-area mobility. To scale to wider area mobility, we must replace the IP Multicast encapsulation with a multicast delivery more suitable for mobility. The characteristics desired for this delivery include: sparse, wide area group support and a separation of transmitters and receivers.

4.6 Summary

The introduction of host mobility presents a significant new challenge in IP packet routing. Several solutions have been proposed to provide support for the delivery of packets to the mobile host. Unfortunately, these solutions do not consider the serious impact they have on the performance of multimedia applications and TCP-like transport protocols. These applications and protocols depend on relatively consistent end-to-end communications to provide reasonable performance. In order to seamlessly integrate mobile hosts into the Internet environment, a new routing protocol that allows these standard applications and protocols to operate efficiently is needed.

In this chapter, we have presented, measured and analyzed a new protocol that meets the requirements for a IP routing protocol for mobile hosts. This protocol applies the concept of multicast that was developed for connection-oriented networks to the IP routing problem. Our implementation measurements show that this routing protocol can perform most handoffs in 5—15 msec without adversely affecting packet loss or end-to-end throughput. The protocol also prevents data loss during handoff through the use of intelligent buffering and synchronization during handoffs. Our experience also indicates that protocols such as

TCP and multimedia applications such as vic [McCa95] work well with this routing protocol.

Chapter 5

State Distribution For Handoff

In the previous chapters, we examined mechanisms to update routing in the backbone network when handoff occurs. However, there is often additional state present in a base station that must be handled properly during a handoff. In this chapter, we present a method to transfer this state to a nearby base station quickly during a handoff. This mechanism leverages the multicast used by handoff to reflect state updates at the base stations.

5.1 Introduction

When a mobile host moves between cells, the new cell's base station must take over all responsibilities of the previous base station. The most important of these responsibilities is to route packets destined for the mobile host. However, a handoff protocol must also transfer other responsibilities, such as retransmitting lost packets and guaranteeing QOS across the wireless link, to the new base station. This transfer of responsibilities may require a significant amount of state to be moved to the new base station during the handoff procedure. For example, to perform retransmissions of lost packets, a base station may maintain a copy of each packet that has not been acknowledged. For retransmissions to continue

after a handoff, this cache of unacknowledged packets must also exist on the new base station. The system must transfer this state quickly and efficiently to avoid adding significant delays to the handoff process.

The most common method to perform the state transfer is to encapsulate and forward the state to the new base station after a mobile host has entered a new cell. The duration of this transfer is directly related to the amount of state present in a base station and inversely related to the wired network bandwidth available between nearby base stations. The I-TCP system uses this form of state transfer in their handoff [Bakr95]. In their system, the presence of a significant amount of state at the base station lengthens the handoff duration from 245 msecs to approximate 1435 msecs.

A better solution is to anticipate repositioning, performing as much of this state transfer as possible before the actual entry of the mobile host into the nearby cell. However, this is difficult since the state may change between the time it is transferred to the nearby base station and the handoff occurs. This forces the nearby base station to mirror all state changes that occur at the primary base station after the initial transfer of state. To perform this mirroring, we must identify all events that cause state changes at a base station and ensure that these events also occur at the nearby base stations. In many situations, we can perform the mirroring by using the multicast delivery of packets and placing careful restrictions on base station state. We have implemented a local retransmission protocol that uses the concept of state mirroring to reduce the latency of handoff. Handoffs with the transfer of the retransmission protocol state take approximately 8—15 msecs, the same duration as handoffs without the state transfer. The variations in handoff time result from variations in media access delay and not from the amount of state present in the base station.

The remainder of this chapter is organized as follows. In Section 5.2, we describe the basic techniques used to mirror state at nearby base stations and the limitations placed on state at the base station to support fast handoff. We present the snoop protocol, that uses

the mirroring techniques, in Section 5.3. Section 5.4 describes the measurements taken on our implementation of the snoop protocol. We present our conclusions about these techniques and implementation in Section 5.5.

5.2 Limitations/Requirements

In order to have two machines mirror state transitions, they must both observe the same state changing events. There are many possible causes for these state changes, including the passage of time, delivery of packets, and measurement of resources. Unfortunately, the nearby base station can not replicate some events, such as page faults and randomized delays. In addition, some of the state changing events, including packet arrival and fine-grained timers, occur much more frequently than others, such as mobile activation or TCP connection establishment. The frequently occurring events are likely to be the cause for the state changes that occur during the period between the transfer of state to the nearby base station and the handoff of the mobile host. The difficulty of replicating state changing events and the presence of events that may occur frequently during the handoff process make it impossible to exactly duplicate all possible state transitions at the nearby base stations. However, we can replicate the needed events between base stations by placing some restrictions on the state. In order to perform state replication, we placed the following restrictions on the protocols that place state at the base station:

1. The protocols must use “soft state” at the base station. This implies that the protocol is resilient to errors in the state at the base station. An example of soft state can be seen in link-layer retransmission protocols. If a protocol guarantees that a packet will be reliably transmitted by a link, the transmitter of a packet must ensure that it maintains a perfect copy of the packet until the receiver acknowledges reception. Since the state must be maintained perfectly, it is “hard state”. If a link layer only promises to improve reliability, the transmitter can support occasional errors in the copies of unacknowl-

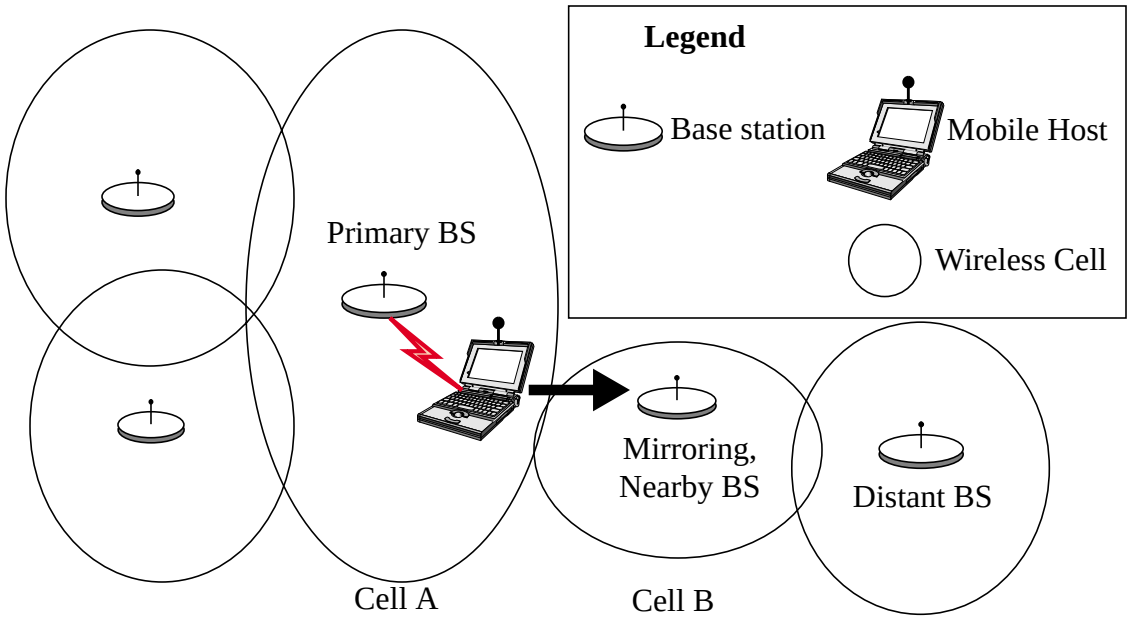


Figure 5.1 Typical State Mirroring

edged packets (“soft state”). However, this forces the transmitters and receivers to support occasional packet loss on the link. In addition, the errors in soft state usually result in degraded performance.

2. All state transitions that are likely to occur during the state mirroring period must result from the delivery of packets to a base station. This allows us to replicate these events at the nearby base stations.

3. The state at the base station must have as short a “memory” as possible. This allows the protocol to recover to recover quickly from errors in state. In addition, this short memory implies that the entire state can be reconstructed quickly if necessary.

Figure 5.1 illustrates a situation in which the state mirroring is desired. In the figure, the mobile host is currently in cell A but moving towards cell B. Its primary base station routes packets and maintains state for it. After identifying that a handoff to cell B is likely, cell B’s base station is primed for handoff. As part of our multicast-based routing protocols, it begins receiving packets for the mobile host. The nearby base station also begins mirroring the state transitions occurring at the primary base station.

Restricting rapid state transitions to result from packet delivery simplifies the problem of duplicating state to the problem of delivering the same packets to multiple base stations. Fortunately, the multicast-based routing protocols perform this delivery of packets as part of the handoff algorithm. When a mobile host is near a cell boundary, the routing protocol delivers packets destined to the mobile host to the primary and likely handoff target base stations. At the beginning of the period during which handoff is likely, some state can be transferred to the nearby base stations. During the period, the nearby base stations attempt to mirror the primary base station's changes in state using in the delivery of packets for the mobile host.

Our multicast-based routing protocol does not guarantee that an identical set of packets is delivered to the primary and nearby base stations. We could utilize a reliable multicast algorithm to guarantee identical packet streams. However, these algorithms tend to be expensive or inefficient. The network may lose random packets enroute to a base station due to congestion or other causes. Since the mirroring base stations are likely to be geographically nearby and topologically close in the network, the congestion-related losses to the different base stations should be similar. Therefore, the set of packets delivered to the mirroring and primary base stations may not be identical but will likely be similar. Since we require the use of "soft" state, the slight differences in packet delivery will result in a slight performance degradation of the protocols creating state at the base station.

There may also be situations where a handoff has not been predicted and the mobile host moves into a cell in which the base station has not been mirroring state. Since holes and errors in soft state usually result in degraded performance, the mobile host will experience poor performance for some time after a handoff. The lifetime of incorrect state at the base station limits the duration of this degraded performance. The short memory requirement on state allows this unprepared base station to create the state it needs relatively quickly.

In general, the errors and missing data ("holes") in the state result in degraded performance in protocols using soft state. Therefore, our handoff protocol must attempt replicate

state as closely possible to provide the best possible performance. We have designed and built a local retransmission protocol, called snoop [Bala95a, Bala95b], that shows how a protocol can be built to meet these requirements for low-latency handoff. This protocol is described in detail in the next section.

5.3 Case Study — Snoop Protocol Handoff

In Section , we described some of the problems TCP has in wireless networks. Reliable transport protocols such as TCP [Post81b, Stev94, Brad89] have been tuned for traditional networks made up of wired links and stationary hosts. However, this tuning results in poor performance over wireless links with high bit-error rates. We have developed a protocol, the snoop protocol, that alleviates the problems caused by high bit-error rate while supporting low-latency handoffs.

Recently, several reliable transport-layer protocols for networks with wireless links have been proposed [Bakr94, Bakr95, Yava94, Paul95] to alleviate the poor end-to-end performance of unmodified TCP in the wireless medium. These proposals are summarized in Section 2.4. In summary, the existing approaches to combat high-bit error rates maintain a large amount of “hard” state at the base station. This makes it difficult to perform handoff quickly in systems using these techniques.

5.3.1 The Snoop Protocol

A primary goal of the snoop protocol is to alleviate the end-to-end TCP performance problems caused by the high bit-error rates of wireless links while providing support for low-latency handoffs. Since TCP is already in wide use, it is also desirable to achieve the goal of improving its performance in our network without changing existing TCP implementations in the fixed network. The only components of the network we can expect to have administrative control over are the base stations and the mobile hosts.

For transfer of data from a fixed host to a mobile host, we make modifications only to the routing code at the base station. These modifications include caching unacknowledged TCP data and performing local retransmissions based on a few policies dealing with acknowledgments (from the mobile host) and timeouts. By using duplicate acknowledgments to identify packet loss and performing local retransmissions as soon as this loss is detected, the module shields the sender from the vagaries of the wireless link. In particular, transient situations of very low communication quality and temporary disconnection are hidden from the sender.

For transfer of data from a mobile host to a fixed host, we detect missing packets at the base station and generate negative acknowledgments for them. These negative acknowledgments are sent to the mobile host (the sender), which then processes them and retransmits the corresponding missing packets. This requires modifications to both the base stations and mobile hosts.

These mechanisms together improve the performance of the connection in both directions, without sacrificing any of the end-to-end semantics of TCP or modifying host TCP code in the fixed network. In addition, the snoop protocol meets the three requirements we placed on state in the base station.

1. **Soft State** — The state at the base station mainly consists of packets unacknowledged by the mobile host. If there are any errors or holes in this cache of packets, the original sender will retransmit the packet. The state at the base station is used only to improve performance not to ensure end-to-end reliability. Errors or missing packets in the state prevent the snoop protocol from performing local retransmissions of packets. As a result, the end-to-end TCP throughput to the mobile host drops.
2. **State Transitions** — All state transitions at the base station are a result of a TCP packet from the corresponding or mobile host.

3. **State Lifetime** — A packet remains in the cache at the base station until it is acknowledged by the mobile host. This lifetime is at most the round trip time between the mobile and corresponding host. Therefore, the state for a connection at the base station can be recreated in a single round trip time.

A preliminary design of a protocol based on these ideas appeared in [Amir95]. Simulations of the protocol indicated that it was capable achieving the same throughput as unmodified TCP at 10 times higher bit-error rates. These promising results were verified by an implementation described in [Bala95a]. This implementation was built on the same infrastructure as described in Chapter 4. This allowed us to experiment with the interaction of the handoff protocol described in that chapter and the snoop protocol.

5.3.2 Snoop and Handoff

The snoop protocol relies on a cache of unacknowledged packets to improve end-to-end TCP performance. Typically, the size of this cache is proportional to the TCP window size. After a handoff, the new base station must have the current set of unacknowledged packets in its cache to provide improved performance based on local retransmissions.

When a handoff is requested by the mobile host or anticipated by the base station, the nearby base stations join a multicast group to update the routing. They receive all packets destined for the MH and the snoop modules here attempt to mirror the state present in the primary base station for the different TCP connections. Using these packets, the snoop module builds up its cache for various connections to the mobile host. However, during this period, packets and acknowledgments *from* the MH continue to pass only through the primary BS. Therefore, the nearby buffering base stations cannot snoop on any acknowledgments for the caches they are mirroring. This prevents the snoop module from freeing up packets that have safely reached the MH. As a result, packets are only freed using a FIFO scheme when snoop runs out of buffer space. In normal operation, the buffering BSs have a superset of the packets contained at the forwarding BS.

Once the handoff occurs, acknowledgments begin to pass through the new forwarding BS. The first acknowledgment the BS receives identifies which packets have been received by the MH on a connection. The BS uses this information to clean and synchronize the contents of the snoop cache to the last state of the previous primary BS.

In reality, the states of the new and previous BSs are not likely to be identical after the synchronization. The new primary BS may be missing several packets from its snoop cache. This is because it may have either missed several packets while it was in buffering mode (due to congestion) or buffered packets for only a short period of time before handoff. Since the snoop protocol does not change any of the end-to-end semantics of TCP, it is resistant to these gaps in its state (i.e., this state is soft). As the connection progresses, the holes in the cache are either filled or ignored. In the worst case, these holes result in a slight performance degradation for a short period after handoff. Our experience has been that it takes only a few packets before the snoop module at the new base station reaches the current state.

The addition of the snoop protocol to the base station resulted in a significant change in the buffer synchronization of the handoff protocol in Chapter 4. When a handoff occurs, the new base station must forward a portion of its cache of buffered packets to the mobile host. The objective is to prevent any packet loss due to handoff. The mobile host transmits the IP IDs of the last 3 packets received along with the request to perform the handoff. The base station uses the IP IDs to identify which packets to transmit from its cache. Unfortunately, the IP IDs tend to be out of date by the time base station actually forwards the packets. This buffer synchronization strategy resulted in a few duplicate packets and did not necessarily generate the acknowledgments the snoop module needs to clean its cache. However, the snoop module at the base station provides a simple mechanism to avoid loss and duplication for TCP packets. After a handoff, the snoop module transmits the few most recent packets on each active connection to the mobile host. The mobile host responds with a set of TCP acknowledgments that identifies the last packet received on each connection. This causes the snoop module to clean and synchronize its cache of

packets, and begin retransmitting packets as appropriate from its cache. The handoff protocol notifies the snoop module of a handoff so that it may perform this processing. However, the handoff protocol still performs the IP ID based buffer synchronization and transmission during handoff for non-TCP packets.

The duration of a handoff is short since no state is explicitly transferred between BSs. The scheme avoids state transfer because the new primary BS has attempted to mirror the state of the previous primary BS prior to handoff. This handoff scheme attains low-latency for TCP streams since multicast provides a simple, lightweight way to mirror state changes at the BSs and because the snoop protocol meets the three basic requirements for multicast-based state mirroring.

5.4 Measurements

We performed several experiments with the snoop and handoff protocols on our wireless testbed and compared the resulting performance with unmodified TCP. In the presence of the routing system's beacons, this peak throughput drops to about 1.45 Mbits/sec. We present the results of various data transfers from a fixed sender to a mobile receiver. We first examine the performance the snoop protocol alone to determine what performance improvements it provides. We then examine the performance of the snoop protocol when handoffs are occurring to identify overheads and weaknesses of the state distribution. We used the network configuration shown in Figure 5.2. The sender TCP stack was based on TCP Reno, an implementation that supported fast retransmissions upon the arrival of three duplicate acknowledgments. The maximum possible window size for the connection was 64 KBytes and the maximum TCP segment size was 1460 bytes.

In order to measure the performance of the implementation under controlled conditions, we used a Poisson-distributed bit error model. We generated a Poisson distribution for each bit-error rate and changed the TCP checksum of the packet at the base station if the error generator determined that the packet should be dropped at the receiver, before forwarding the packet over the wireless link. The same operation was done for packets

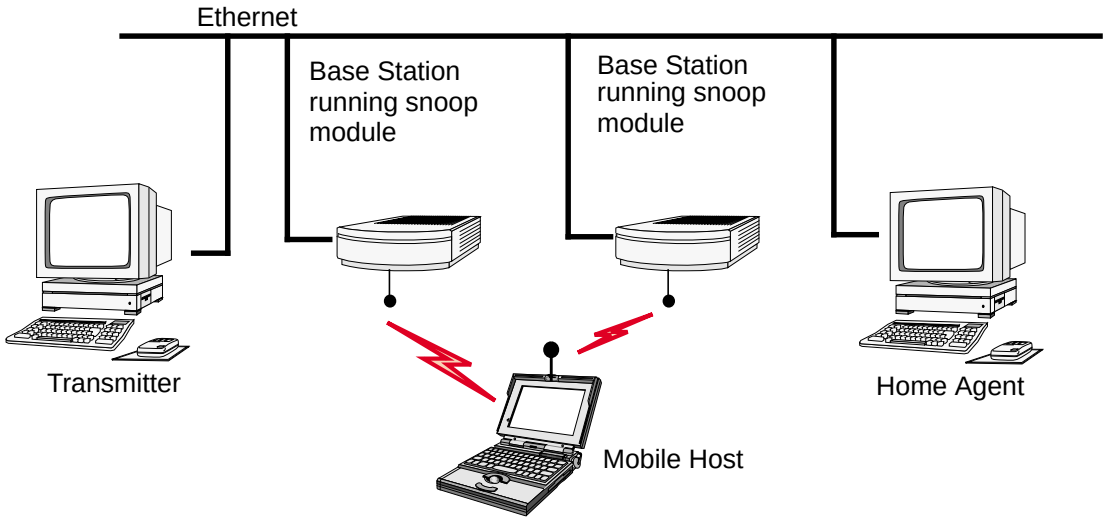


Figure 5.2 Network topology for experiments.

(acknowledgments) from the mobile host. We also experimented with using a two-state Markov error generator that more accurately modeled the wireless channel [Linn95]. The two states corresponded to periods of good connectivity and periods of poor connectivity. Poisson-distributed errors were generated at different rates in each state. We varied the transition probability from the good to the poor connectivity state from 1% to 0.1%. The reverse transition probability was varied from 80% to 99%. Throughput measurements converged very slowly when using this error model. Also, it was difficult to interpret the implications of the results. The Poisson error model measurements show how the snoop protocol would perform in either the good or bad channel state.

The performance of the snoop protocol is most important for bulk transfers. Bulk transfers also provide a good test for the state mirroring since they cause snoop to create large amounts of state at the base station. To test snoop, we performed runs involving a 10 MByte transfer and repeated them ten times at each error rate. Figure 5.3 compares the throughput of a connection using the snoop protocol with that of a connection using an unmodified TCP implementation, for various Poisson-distributed bit-error rates shown on a log scale. The vertical error bars in the figure show the standard deviation of the receiver throughput.

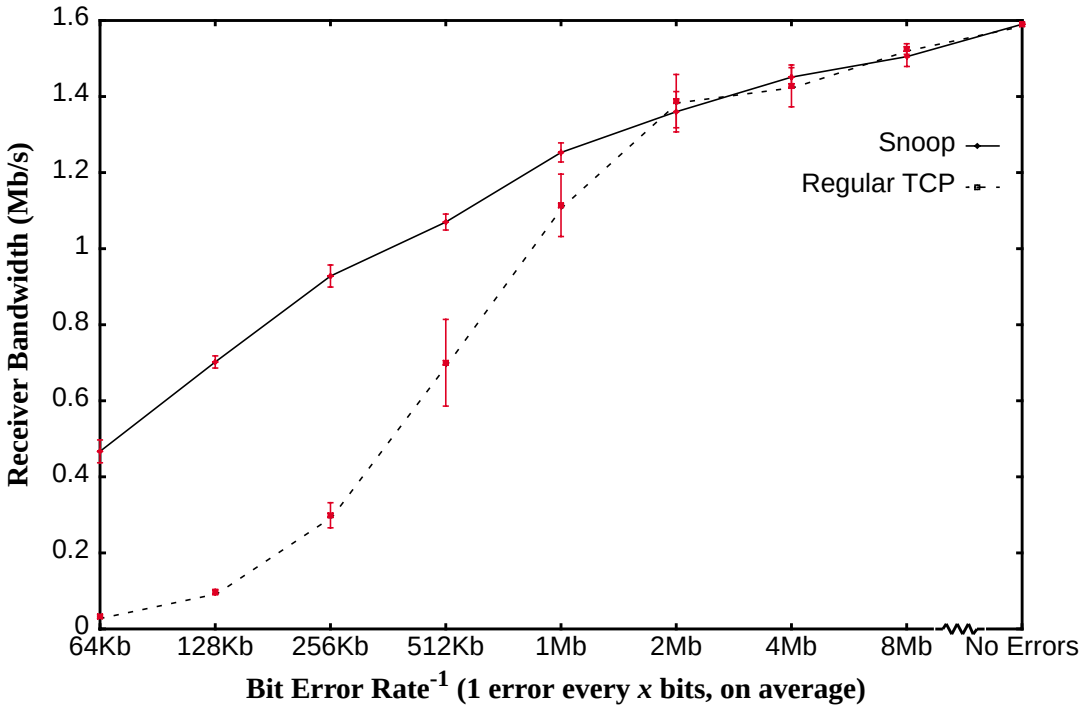


Figure 5.3 Throughput received by the mobile host at different bit-error rates

X-axis is a $\log_2$ scale and the vertical error bars show the standard deviations of the measurements.

We see that for error rates of over $5x10^{-7}$ (close to the 2 Mb point on the x axis of the graph) the snoop protocol performs significantly better than unmodified TCP, achieving a throughput improvement factor of 1 to 20 depending on the bit-error rate. In fact, the snoop protocol is robust and completes the run at even high error rates. For similar error rates, the regular TCP connection does not make any progress. Under conditions of very low bit error rates ($< 5x10^{-7}$), we see little difference between the snoop protocol and unmodified TCP. At such low rates there is typically less than one error per transmitted window and unmodified TCP is quite robust at handling these. At these low error rates, snoop behaves as if it is not present, which ensures no degradation in performance.

To isolate the impact of handoff on normal TCP performance in the absence of bit-errors, we examined the performance of TCP with regularly spaced handoffs between two base

Figure 5.4 Throughput received by the mobile host at different bit-error rates with handoffs every 5 seconds. ($\log_2$ scale)

stations. As part of the routing protocol each base station sends out a beacon signal once per second. In order to stress the performance impact of the handoff scheme on end-to-end performance, we performed several tests, varying the time between handoffs from 1 to 10 seconds. The results for different handoff rates is discussed in Section 4.5.3 and is shown in Table 4.3. The measurements show that even frequent handoffs have very little impact on end-to-end TCP throughput. They also show that data transfers progress without any significant interruptions despite the presence of the handoffs.

Since the basic handoff mechanisms introduce no additional overhead to basic TCP performance, the performance obtained from snooping base stations while handoffs occur should indicate how well the state mirroring performs. Any performance degradation compared to that presented in Figure 5.3 indicates errors or holes present in the soft state of the new base stations. Figure 5.4 shows the performance obtained from the combination of the snoop and handoff protocols in the presence of bit errors. The figure plots the receiver throughput seen over a 10 Mbyte transfer. The connections experienced different

error rates, shown on the x axis, and handoffs at a fixed rate, one every 5 seconds. At low bit-error rates, $< 1x10^{-6}$ (1/1Mbit), we obtain close to the peak performance available in the presence of beacons, 1.45 Mbits/s. At higher bit-error rates, we see performance comparable to transfers through snoop to a mobile host not experiencing handoff. This indicates that the handoffs do not adversely affect the snoop processing and that the snoop protocol state was effectively mirrored at the buffering base stations.

5.5 Summary

Many modern protocols introduce state in routers and gateways to improve their performance or implement their policies. This state creates a significant new problem in the presence on mobility. Most current approaches to supporting mobile networking have concentrated on routing. The solutions proposed for supporting state in routers and mobility have relied on explicit transfer of state during handoffs. This results in handoff durations that are directly proportional to the amount of state maintained in the base stations. In order to support low latency handoffs a mechanism to mirror this state at the new base station without an explicit transfer is necessary.

In this chapter, we have proposed a mechanism to support the light-weight mirroring of state between base stations. This mechanism uses two important concepts:

1. It places specific restrictions on the type of state at the base station.
2. It uses the multicast delivery of packets provided by our mobile routing algorithms.

We have implemented the snoop protocol that improves end-to-end TCP performance in the presence of a high bit-error rate. It does this through the use of state at the base station, including a fairly large cache of packets. The snoop protocol uses the state distribution mechanism to support very low latency handoffs. We have shown that the snoop protocol provides the same end-to-end TCP performance improvements when handoffs occur as when they do not.

Chapter 6

Support for Mobility in the Infopad Environment

The technology used to provide mobility and portability places restrictions on the capabilities of mobile host and network communication. For example, battery-powered operation usually implies slower computation, less storage and reduced wireless transmission power. In addition, wireless networks provide much lower bandwidth and higher bit-error rates than conventional wired networks. In this chapter, we present a system design that combats these problems and the implications it has on handoff processing.

6.1 Introduction

The objective of the InfoPad project to create a system that provides ubiquitous information access with a low powered portable terminal. To support access to multimedia information, the portable unit must provide high bandwidth network connectivity and possibly a significant amount of computational resources. The InfoPad system uses a number of innovative techniques to provide these capabilities while taking into account the limitations imposed by portable, wireless operation. In this chapter, we examine the impact

some of these techniques have on the handoff support we have described in the previous chapters.

The basic approach taken by the InfoPad project is to move all power hungry tasks off the mobile host. This results in a low-power, light-weight, wireless, multimedia terminal (the pad) that contains no general-purpose computational resources. The computation necessary to make the pad more than simple display device occurs on compute servers connected to a nearby backbone network. This design places three unique requirements on routing of data to the mobile host.

1. Unlike most mobile hosts, the pad can not perform any of the routing analysis and decision making. This computation must be performed by the backbone network servers.
2. The routing protocol need not support delivery of packets from arbitrary hosts to the pad; only the special compute servers transmit packets to a pad. The routing algorithm should take advantage of this to provide more efficient routes from these sources.
3. Packets must be delivered to the pad a timely and consistent manner to provide adequate interactive performance to the user.

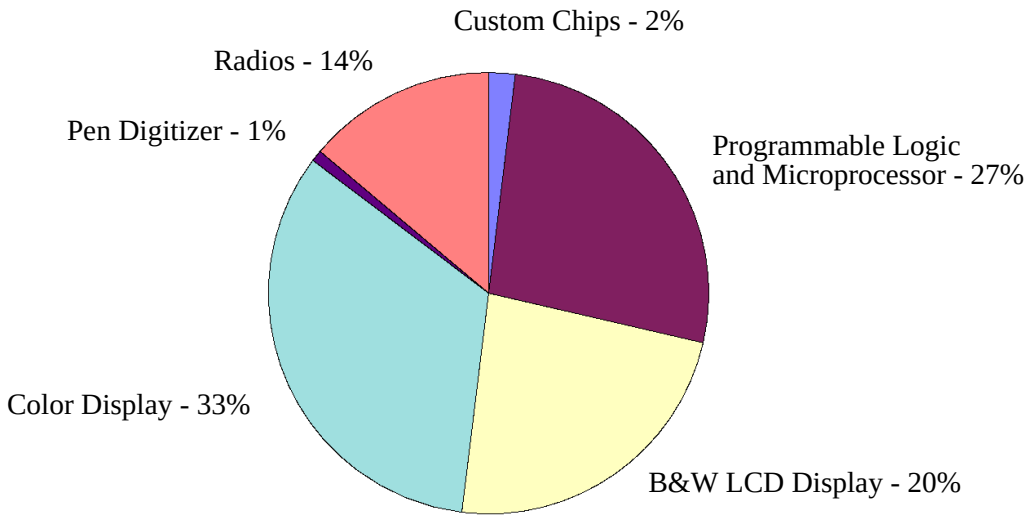
The existing routing protocols fail in several important ways. Mobile IP and the Daedalus multicast routing protocol both expect the mobile host to perform a significant amount computation for route analysis and negotiation with routers. The Mobile IP protocol supports optimal routing between the computer servers and the mobile host through the use of the protocol's route optimization option [John95]. This optimal routing allows corresponding host to transmit packets directly to a mobile host without the use of a home agent. However, the Mobile IP standard does not adequately support the real-time nature of data streams to the pad since packets during a handoff are unacceptably delayed or lost. The multicast routing protocol eliminates these delays and losses. However, it must be modified to support more efficient routing from the compute servers. In this chapter, we describe how the handoff algorithms of the previous chapters can be adapted for this constrained environment.

This remainder of this chapter describes is organized as follows. Sections 6.2 and 6.3 provide the background necessary to understand the routing in the InfoPad system. Section 6.2 presents an overview of the InfoPad system. It describes the motivation for the system design and the architecture chosen. In Sections 6.4 and 6.5, we describe the details of routing and handoff in the InfoPad system. We present an analysis of the InfoPad routing in Section 6.6 and we conclude with a summary in Section 6.7.

6.2 InfoPad System Overview

The main limitation of current portable computers is short battery life. Unfortunately, battery technology is improving at the relatively slow rate of 30% every 5 years [Eage92] with no breakthrough technologies expected. The typical approach to improving battery life is to re-engineer chips and systems for low-voltage operation. Beyond exploiting this approach in the InfoPad system, a much more dramatic step is being taken by eliminating as much as possible of the electronics and local computation in the portable. This is done by migrating the power hungry tasks that typically occur in a portable terminal to servers on the backbone network. This approach allows the design of a very inexpensive, lightweight terminal with a long battery life. The total power consumption of the mobile terminal is shown in Figure 6.1.

As a result of moving power hungry tasks elsewhere, the InfoPad system's mobile host is a low power, lightweight, wireless, multimedia terminal [Chan94] that operates in indoor environments and supports a high density of users. Although the portable terminal, called the pad, contains no general purpose computational resources, it does contain a sophisticated low power chipset that displays text and graphics, plays audio and compressed video, records audio, and captures pen input. The pad also contains a wireless network interface that connects it to a wired backbone network. All computation needed to make the pad more than a display device occurs on servers that reside on the wired backbone. These servers transmit data such as screen updates, video streams and audio samples across the wireless link to the pad. These streams are resistant to errors since they contain



TOTAL Power - 9 Watts

Figure 6.1 Pad Power Consumption

only display data and no control information. By transmitting such data streams across the wireless link, the design gains resistance to errors in wireless transmission and provides access to better computational power for applications in the fixed infrastructure.

The InfoPad design gains these benefits at the expense of transmitting streams requiring high data bandwidth across the wireless link. However, to deliver multimedia data to the mobile hosts already requires high bandwidth wireless network. For example, NTSC quality video delivery to the pad requires approximately 1.5 Mbits/sec. The delivery of audio and screen updates does not place a significant additional requirement on the wireless network. Through the use of a pico-cell networking architecture and wideband spread spectrum radios, the InfoPad system aims to provide up to 2 Mbits/sec of bandwidth to each of 50 users in an area the size of a typical classroom.

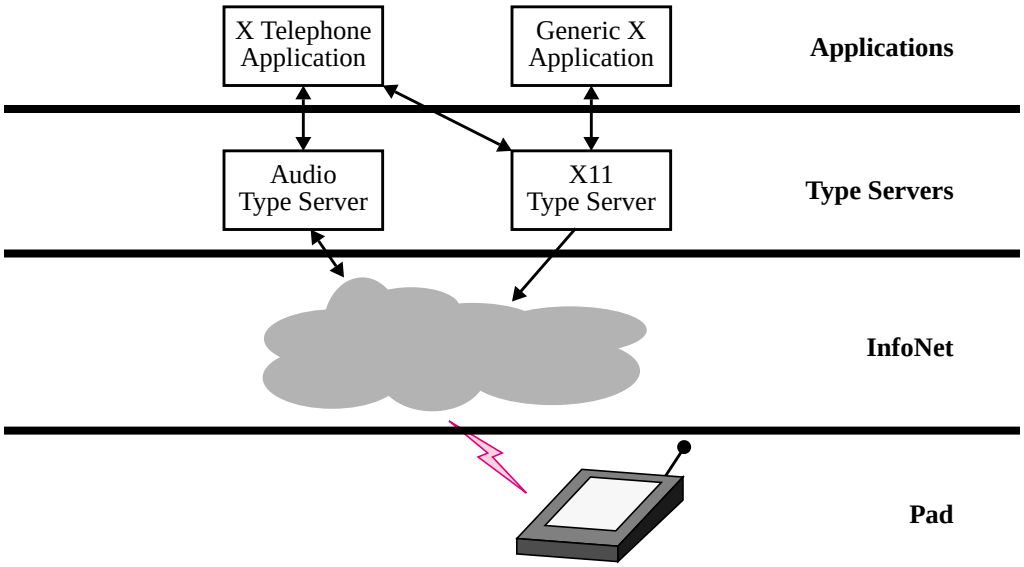


Figure 6.2 InfoPad System Layering

The resulting InfoPad system architecture consists of the four layers shown in Figure 6.3. The layers perform the following functions:

- *Pad*: A low power, portable multimedia terminal capable of displaying text and graphics, playing audio and compressed video, recording audio, and capturing pen input.
- *InfoNet:* A network management and routing system that supports mobility and distributed processing.
- *Type servers:* A set of network-transparent servers that shield applications from knowledge of the mobile environment and terminal hardware.
- *Applications*: A set of applications that execute upon backbone network compute servers and are optimized for use on the pad.

The hardware in the pad supports the operation of the wireless link, and the acquisition and display of multimedia data. A set of application specific integrated circuits (ASICs) designed specifically for low power dissipation [Chan94] perform much of the processing in the terminal. The embedded software on the pad currently executes on a combination of a low power processor (ARM 610) and special purpose programmable logic. This soft-

ware initializes and controls the ASICs, implements wireless link level and reliable transport protocols, and monitors the quality of the wireless channel. As mentioned previously, the design also significantly reduces power consumption by migrating computationally intensive tasks from the pad to the backbone network. The pad *only* contains the hardware necessary to control the communication link and display or record multimedia data.

The InfoNet layer presents the software layer above it, the type servers, with an abstraction in which each pad appears as a stationary, network connected multimedia terminal. The InfoNet software also provides support for low latency, high bandwidth communication between the type servers and the pad. To accomplish this, the InfoNet contains a network management and routing system that hides the mobility of the pad and manages the wireless network resources (e.g., allocation of frequencies to pads).

In the InfoPad system, there is a type server associated with each device available on or emulated by the pad. The InfoNet system presents the type servers with the view of the pad as a stationary, closely connected terminal. The various type servers take this representation and provide the applications with an interface similar to that of a stationary machine with local computation. To bridge the gap between these abstractions, the type servers hide the specifics of the pad hardware and provide a device independent, network transparent interface for the applications to use. The type servers follow established standards as far as possible to allow compatibility with existing applications. For example, instead of developing a new graphics system InfoPad uses a specially modified X11 server that runs on a compute server to support the display of text and graphics on the pad. Standard X clients can connect to this server to display windows.

The type servers provide compatibility with standard workstation applications, allowing many off-the-shelf software packages to work unmodified on the InfoPad system. However, many of these applications are not well suited to the InfoPad environment. They may not operate well with the small screen size nor take advantage of the characteristics of

handwritten or spoken input. The InfoPad group built several new applications to demonstrate the viability of the system.

These four layers combine to provide the user with a portable workstation system. Unlike many other portable systems, the InfoPad system provides mobile users with compute performance similar to their desktop environment. In the remainder of this chapter, we examine the design and implementation of the layer responsible for routing in the system, InfoNet.

6.3 InfoNet

In addition to performing routing, the InfoNet layer performs several additional functions that influence its design and implementation. To make the pad appear to be a stationary, closely connected terminal, InfoNet performs the following functions:

- Packet Routing

  The InfoNet software is responsible for routing packets from the type servers to the mobile terminal. Each type server interfaces to a pad as it would interface to a host on a fixed network. This allows the mobility of the pad to be completely transparent to the type servers and applications. To provide this support, InfoNet must also perform the handoff of communication between the cells of the wireless network in a quick and efficient manner.

- Radio Resource Management

  The wireless network divides the building in which the InfoPad operates into pico-cells with a typical radius of 10 meters. The InfoNet software is responsible for allocating the frequency hopping sequences and determining when handoff occurs.

To meet the demands of applications and users, we have designed the InfoNet to also support the following:

- Support Varying Qualities of Service

  Applications supported by InfoPad have widely varying bandwidth and error rate requirements. For example, video requires high bandwidth and tolerates low reliability, while user input tolerates low bandwidth and requires high reliability. To support these requirements, InfoNet design provides the ability to request and guarantee different Qualities of Service (QOS) on the wired and wireless networks.

- Low Latency, Consistent Performance Routing

  Applications and user interface software execute on computers connected to the backbone network. As a result, the observed performance of the terminal is tightly linked to the latency of communication between the compute servers and the mobile terminal. To provide the best possible, consistent terminal performance, InfoNet must provide low latency, consistent performance routing between the mobile hosts and type servers as efficiently as possible.

Unlike the systems described and analyzed in the previous chapters, the InfoNet layer must perform the above functions without the use of any processing at the mobile host. The requirements divide the InfoNet state into two associations: radio cell and pad. In order to improve efficiency, scalability and modularity, we also decided to separate the functionality of the control and data paths. This resulted in an InfoNet architecture consisting of three logical modules: *gateway, cell server* and *pad server.* The partitioning of functions among these modules is shown in Figure 6.3.

For each pad in the system there is a single corresponding pad server running on the backbone network. The pad server is a control entity responsible for managing access to the terminal. It allocates the terminal bandwidth and resources (speaker, microphone, etc.) among the different type servers and applications. To perform its duties, the pad server maintains information about the pad’s state, including radio bit-error-rate (BER), received radio power level and cell location.

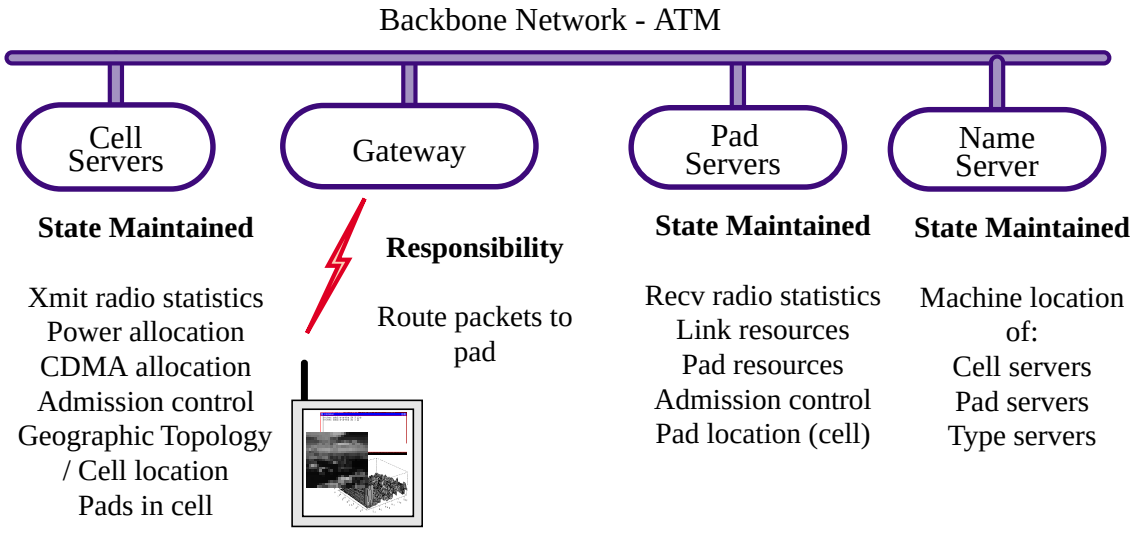


Figure 6.3 InfoNet Partitioning

Each radio cell in the system has an associated gateway and cell server. These two modules together provide the functionality of a conventional base station. The gateway performs the data path operations associated with base stations while the cell server provides the associated control functionality. The gateway routes packets between the wired and wireless networks and performs the protocol conversions necessary between the two networks. The cell server controls the allocation of resources, such as radio frequencies or CMDA codes, among the pads within the cell. Since spread spectrum wireless links are interference limited, resource consumption within a cell often affects nearby cells. As a result, each cell server must negotiate resource allocations with the neighboring cell servers. Each cell server maintains information about the current status of its cell, such as radio bit-error-rate (BER) and available frequencies.

6.4 Routing in InfoNet

The primary responsibility of the InfoNet layer is to provide routing between the type server and the mobile pad. A key performance goal of the InfoNet implementation is to minimize latency while providing the full bandwidth of the wireless network to the type servers and applications. The pad server, cell server, gateway, and name server are imple-

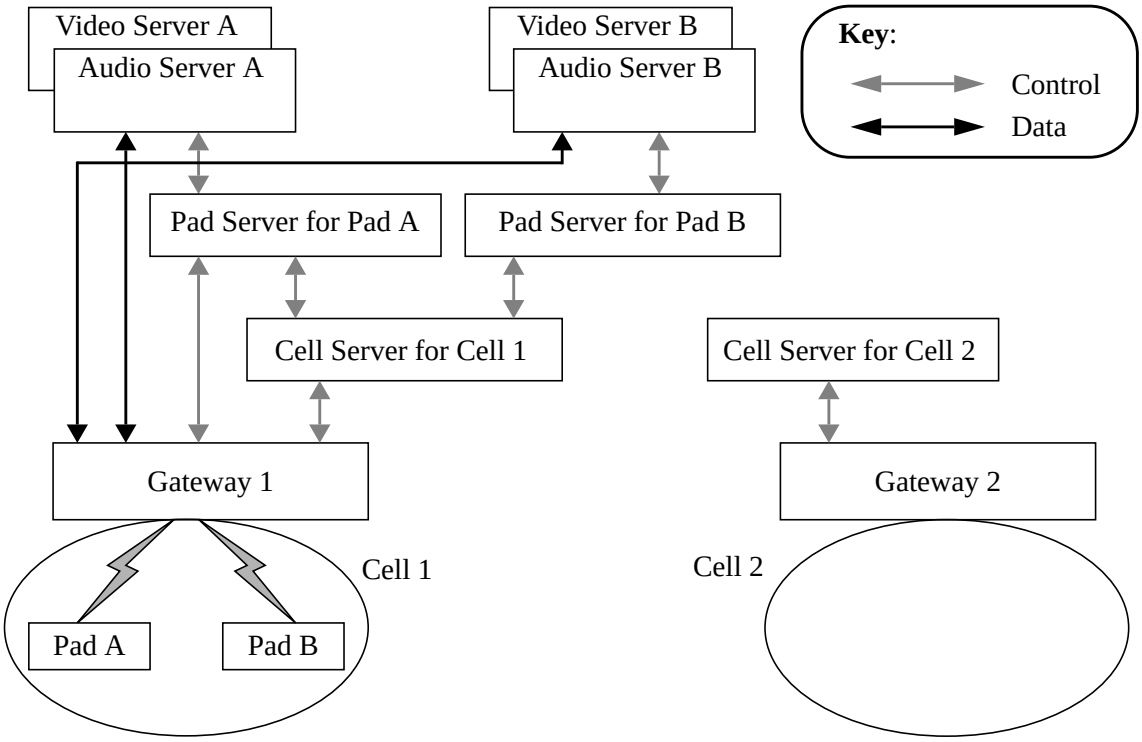


Figure 6.4 InfoNet Data Routing.

Cell 1 has two pads, A & B, in its coverage area whereas cell 2 has none. Each pad has a video and audio type Server running. Arrows depict routing of data.

mented as UNIX user level processes. The routing of data between the pad and the type servers (through the InfoNet servers) is shown in Figure 6.4. The data connections carry the multimedia display information between the type servers and pad. Since this data is error-resistant and delay sensitive, the data connections use UDP. The control connections are used to request services and to set up and maintain the data connections. These connections use TCP since control messages must be transmitted reliably.

Once a type server is initialized, it begins transmitting data to the pad. InfoNet is responsible for routing these data packets to the gateway associated with the pad. When the pad moves from one cell to another, the system must update the routing of packets quickly to avoid data loss. Several existing schemes, such as Mobile IP and our multicast-based rout-

ing, can perform routing to mobile hosts. However, these protocols do not meet or take advantage of the three unique routing requirements of the InfoPad system:

1. Unlike most mobile hosts, the pad cannot perform any of the routing analysis and decision making; this analysis must be performed by the backbone network servers.
2. The routing protocol need not support delivery of packets from arbitrary hosts to the pad; only the special compute servers transmit packets to a pad. The routing algorithm should take advantage of this to provide more efficient routes from these sources.
3. Consistent performance, low latency, high bandwidth delivery of packets to the pad. This is necessary to provide adequate interactive performance to the user.

Most existing mobile routing protocols have been developed for use by devices that can execute applications and process a full network protocol stack locally. As a result, they violate our first requirement by assuming that decision making and error handling can be performed on the mobile host. Some existing routing protocol, including Mobile IP, provide support for modifying the source of data to understand mobile routing. This results in a more optimal route between the source and mobile host. However, many of these schemes, including Mobile IP, do not meet our third requirement. As a result, they are not well suited to the multimedia nature of the pad. They operate by updating the routing tables and appropriately forwarding packets that are already in flight. However, this introduces additional latencies that severely degrade the quality of video and audio received at the pad. To perform seamless handoffs, the new gateway must begin receiving packets for the pad prior to the handoff. Multicast-based handoff provides a natural, low overhead mechanism to deliver the same packets to both the new and current gateways. Since no existing protocols dealt with all three unique factors of the InfoPad system, we chose to modify the IP-based multicast routing scheme to add support for route optimization and compute-server based route analysis.

The InfoNet routing system operates by assigning each pad in the system a unique IP Multicast [Deer91] address. Each type server for a pad transmits packets to a different port of

this IP Multicast group. The gateway responsible for the pad is also a member of the associated multicast channel and forwards packets it receives from any type server across the wireless network. This routing allows the type servers to transmit packets without knowing the current location of the pad.

Since the pad does not support the processing necessary to perform the analysis and decision making necessary, we modified the multicast-based routing protocol to use the pad servers to configure routing. The pad server tracks the pad's current cell location and the wireless link quality between the pad and gateways near the pad. It analyzes this information to identify the best gateway to use. The pad server also requests gateway to leave or join the multicast group to configure the routing and initiate handoff. However, parts of the error handling procedure must remain in the pad. These procedures handle situations in which communication between the pad and the compute servers is disrupted. This results in two forms of handoff in our system: the normal form (requested handoff) and the disconnected form (unexpected handoff).

When the pad server identifies that a different gateway should be used, it initiates a requested handoff. The pad server requests the gateway of the desired cell to join the corresponding IP Multicast group. As a result, all the processing and routing changes necessary for handoff complete before the pad enters the adjacent cell. This enables handoffs to complete quickly and not affect the delivery of multimedia data. This form of handoff has been implemented and measured in a similar system that uses conventional laptops as mobile devices. As seen in Chapter 4, handoffs of this type take between 8 and 15 msecs. to complete independent of the amount of data in flight.

In the other form of handoff, unexpected handoff, the link between the pad and gateway degrades before the pad server can identify any problems and initiate a requested handoff. The pad loses contact with its current gateway and servers by entering into a new cell. At this point, the pad executes the normal pad initialization procedure to establish communication with the new gateway. This procedure reconnects the pad to its servers in the back-

bone network. Unexpected handoff takes much longer to complete than requested handoff. Fortunately, we expect requested handoff to occur much more often than unexpected handoff in typical use. The messaging for these two forms of handoff is described in the next section.

6.5 Handoff Messaging

The following sections describe the two forms of handoff that occur in the InfoPad system. The first form of handoff, requested, is the most likely. In the requested handoff procedure, the pad server sets up the routing in advance to smooth the transition between cells. The second form of handoff, unexpected, is identical to the activation of a new pad. The initialization procedure either (1) creates an environment that allows the users to login and request applications to be run, or (2) reconnects a pad to its previously existing environment. For each of the forms of handoff, the following sections describe how the basic blocks of Infonet software architecture interact to produce this desired result.

6.5.1 Requested Handoff

The pad server periodically queries the pad to retrieve any radio measurements. The pad server uses these measurements to identify when a pad is reaching the edge of a cell. Once the pad server has identified that a handoff would be beneficial, it initiates a requested handoff. The communication involved in requested handoff is shown graphically in Figure 6.5. The communication is described below:

1. For a requested handoff to occur, some part of the system must determine that a handoff will improve the communication quality. The information necessary to perform this decision is the BER of data received by the pad and the received signal strength of transmissions in nearby cells. The pad server queries the pad for this information (message 1). Characteristics of RF communication (such as local fades) result in short lived drops in communication quality. To avoid performing a handoff in these situations, successive queries must indicate that a handoff is desirable.

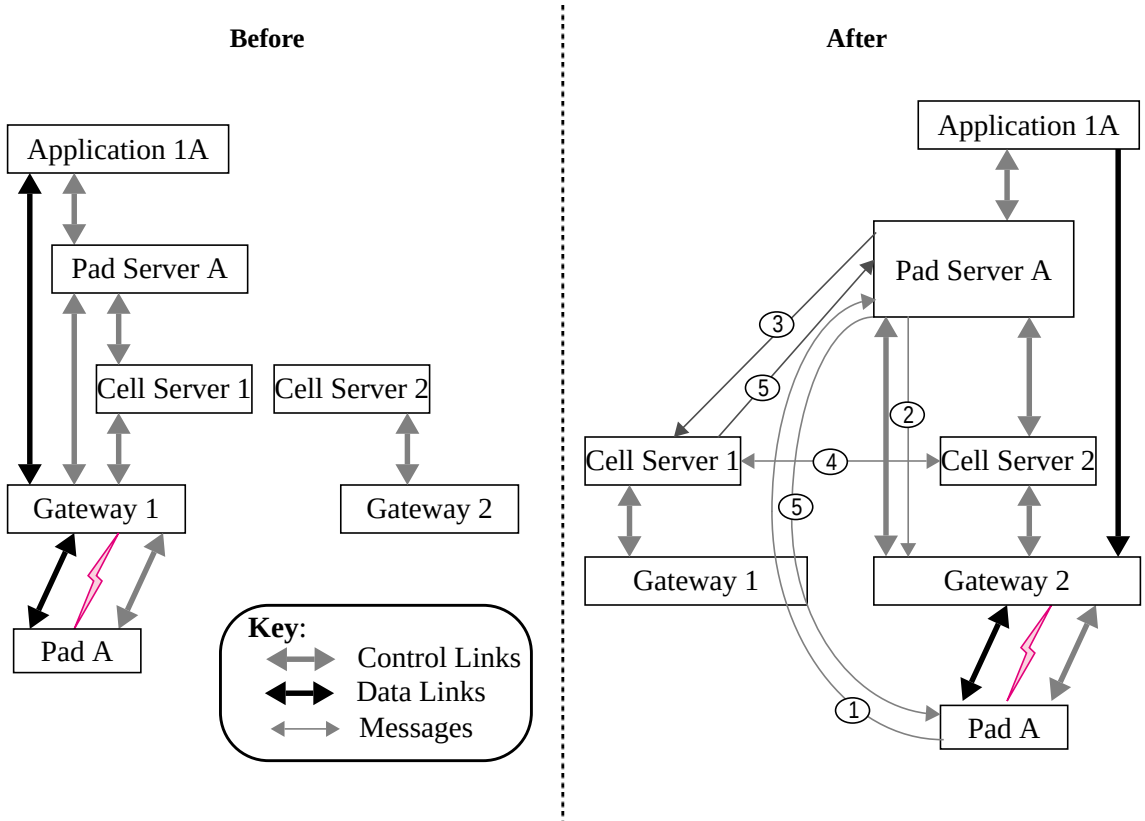


Figure 6.5 Requested hand-off.

2. After these queries, the pad server connects to the likely target gateway (message 2). This causes the new gateway to join the IP Multicast group associated with the pad. Since the pad is not in this gateway's radio cell, the gateway discards any incoming packets from the multicast group.

3. The pad server then informs the cell server to begin a requested handoff (message 3). This handoff request contains the information the pad server has about the signals received by the pad.

4. The cell server uses this information to begin negotiating with the servers of the appropriate adjacent cells (message 4). During this negotiation, the cell server attempts to obtain the resources needed by the pad in the adjacent cell (radio channel, bandwidth, etc.).

5. Once the necessary resources are allocated, the pad server and pad are notified of the new cell and gateway (messages 5).

6. At this point, the pad modifies its radio configuration to communicate with the new gateway and the pad server disconnects from the previous gateway. The previous gateway leaves the multicast group. This results in the desired routing of data to the pad and completes the requested handoff.

6.5.2 Pad Activation / Unexpected Handoff

The separation of the handoff decision making process, the pad server, from the mobile unit, the pad, creates a second form of handoff, unexpected handoff. Unexpected handoff occurs when a mobile host loses contact with its current gateway. This disconnects the pad from its pad server. Some form of error handling process at the pad must recognize this problem and reconnect to the pad server. This reconnection process is identical to a power-on procedure for a pad. At power on time, the InfoNet system either creates a new environment that allows the user to login and request services or reconnects a user to his previously in use environment. To perform the pad activation, a new pad server must be started or an existing pad server must be reconnected to the pad. Also, the system needs to create the necessary control connections between the pad server, gateway and cell server. Figure 6.6 illustrates the communication involved in this task. The communication is described below:

1. Each pad in the system contains a unique ID. When a pad is powered up or loses connection to its gateway, it transmits the ID to the nearest gateway across a wireless control channel (message 1).

2. The gateway recognizes this as a request to resume operation of a pad or to activate a pad for the first time. The gateway forwards this ID to the associated cell server (message 2).

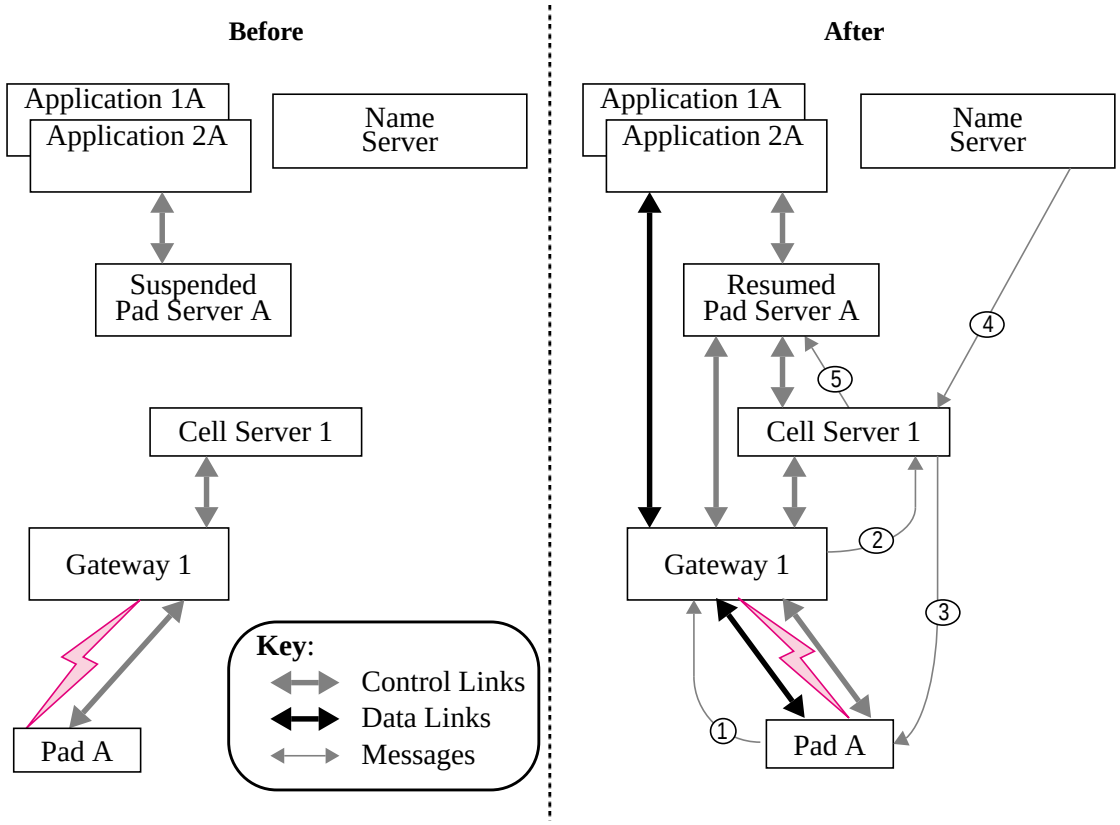


Figure 6.6 Pad Activation

3. The cell server allocates a wireless channel, bandwidth, and associated resources for the pad to operate within the cell. The cell server transmits information about this resource allocation to the pad (message 3).

4. The cell server then queries the name server to locate a previously running pad server (message 4).

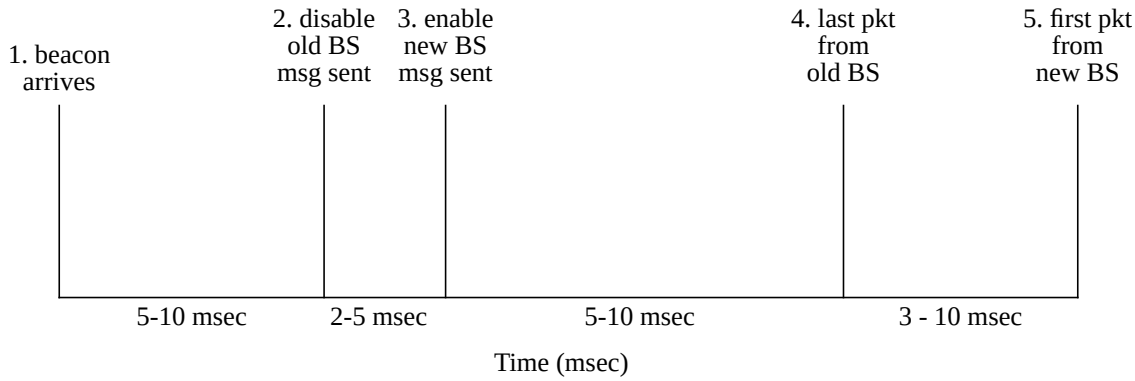
5. If a pad server is already associated with this pad, the cell server informs the pad server of the pad's current state and location (message 5). This allows the pad and pad server to reconnect and resume operation.

6. If no pad server is present in the network, the cell server must initiate the start-up of a new pad server. The pad server executes the applications to authenticate the user. After authentication has been successful, the user may start other applications. This authentication may be done by providing a login prompt and using handwriting recognition.

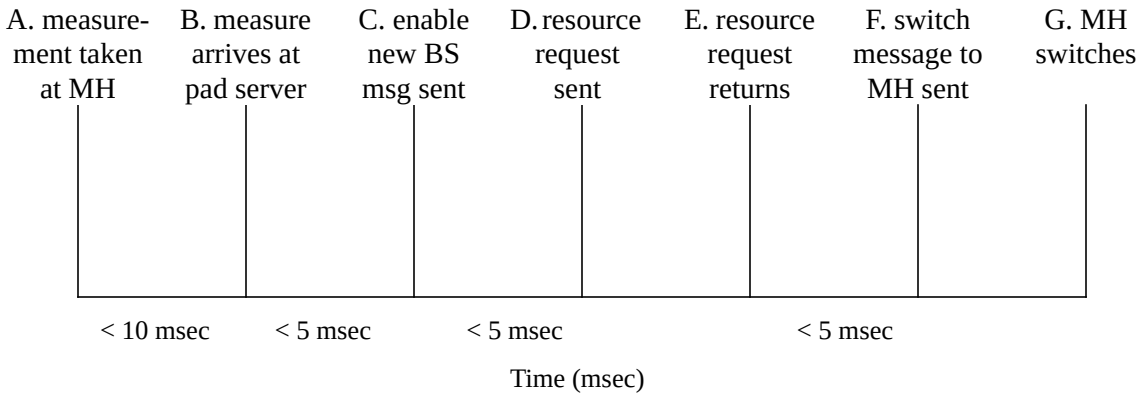
6.6 Analysis

The design of the InfoPad system makes it very difficult to obtain measurements of handoff performance. Since the mobile host is simple, it does not provide enough support for debugging or event timing. In addition, the current state of the radio system design makes it difficult to observe handoff events from a separate monitoring machine. However, we can analyze the messaging required for handoff and compare it with Daedalus handoff scheme of Chapter 4.

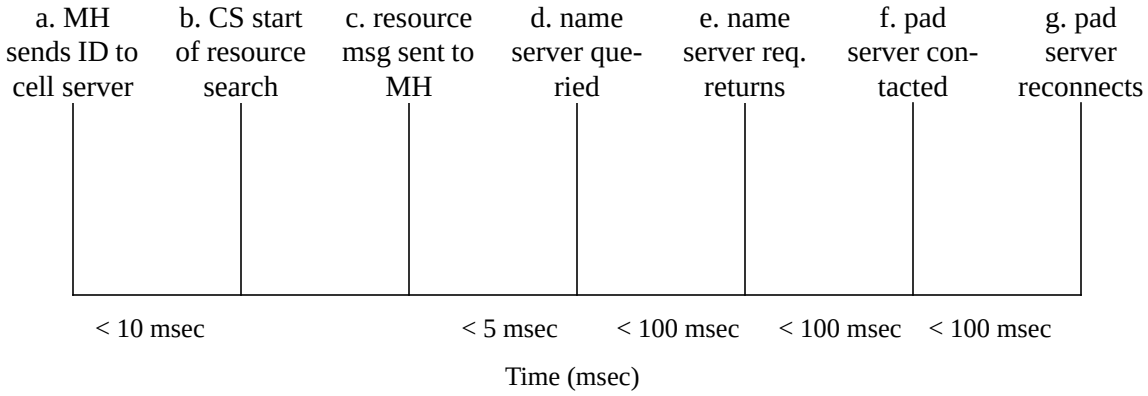
The sequence of events in a InfoPad Requested Handoff and a Daedalus handoff is shown in Figure 6.7. In the Daedalus system, much of the variation in handoff times resulted from variation in the wireless network media access delays. The InfoPad wireless network uses a channel based radio system that provides consistent access delays. As a result, we expect much less variation in handoff times. In the Daedalus system, the mobile host initiates a handoff when it receives a beacon (event 1). The InfoPad requested handoff occurs when the pad server receives a new radio measurement from the mobile host (event B). This measurement arrives at the pad server a single server to pad latency (estimated to be < 10 msecs from current measurements of the system) after mobile pad took it (event A). Since the measurement analysis in InfoPad (between events B and C) occurs on a fast compute server (currently a SparcStation 20), we expect it to take somewhat less time than the analysis in the Daedalus system (between events 1 and 2). As in Daedalus, the mobile host transmits a pair of packets after the analysis completes (between events C and D). However, these packets are transmitted across the wired network and should take less time than their Daedalus counterparts. These packets request the new base station to prepare for handoff. Unlike Daedalus, this may result in a distributed resource allocation process (between events D and E). This resource allocation has not been implemented and it is unclear how long it will take. Once the resource allocation completes, the pad server must analyze the results (between events E and F) and transmit the final instructions to the mobile host. This analysis should be less time consuming than the initial measurement analysis. Finally, the mobile pad must receive and implement the instructions given by the



Daedalus Handoff Timing Measurements



InfoPad Requested Handoff Timing Estimates



InfoPad Unexpected Handoff Timing Estimates

Figure 6.7 Timing of handoff events

pad server (between events F and G). It is unclear how quickly the radio system can switch base stations. Since Daedalus uses a single broadcast based media (WaveLAN) for all its base stations, there is no comparable delay in its handoff timing. These estimates indicate that the requested handoff should complete in at most 25 msecs + time to perform cell resource allocation + time to switch radio to new cell. Since we are using multicast-base routing, the pad should only be out of communication during the period in which its radio is switching cells. The unexpected handoff duration is dominated by the several instances of TCP connection establishments and control handshakes (between events d & e, e & f, and f & g). Each of these will take approximately 100 msecs, combining to result in a handoff taking over 300 msecs. During this period, the mobile host receives no screen updates or other information. This results in a serious disruption in observed pad performance. This indicates that identifying impending handoffs is much more important in the InfoPad environment than in systems with more intelligent portables. Fortunately, we believe the information from the InfoPad radio system and the high degree of cell overlap will prevent unexpected handoff from occurring often.

6.7 Summary / Status

Portable operation places several constraints on the capabilities mobile hosts. These restrictions result from:

- **Battery Powered Operation**: Techniques employed to improve battery life have resulted in portable computers with significantly less compute power and storage than their desktop counterparts. Since battery technology is improving slowly, this disparity is unlikely to disappear in the future.
- **Wireless Networks:** To provide network connectivity to mobile hosts, we must employ wireless networks. Unfortunately, wireless networks have lower bandwidth and higher error rates than wired networks.

The InfoPad system uses an innovative design to combat these problems. In order to deliver multimedia data to mobile hosts, a high bandwidth wireless network had to be

developed. The InfoPad design uses this high bandwidth wireless network to combat the problems of high bit error rate and battery life. The design migrates all power-intensive computation to backbone network servers. These servers transmit only display information across the wireless network. As a result, this design relies heavily on the quality of communication between the network servers and mobile pad.

In this chapter, we have described the implications of this design on the network software and routing algorithms of the previous chapters. To provide the performance necessary without disruptions during handoff, we used a variation of the multicast-based mobile routing protocols as presented earlier in this dissertation. We modified the protocols to account for two of the unique factors of the InfoPad environment: lack of computation support in the mobile host and the limited number of possible sources of data transmitted to the pad. The differences were accounted for by moving route decision making to a backbone network server while leaving some error handling in the mobile host. The modified protocol also takes advantage of the observation that only a small set of specialized servers may transmit data to the mobile host. These modifications resulted in the formation of two forms of handoff: requested handoff and unexpected handoff. Analysis of the messaging required shows that requested handoff should take slightly longer, 25 msecs + cell resource allocation time, than handoff in the Daedalus testbed. This is because additional handshakes with the pad server, cell server and pad are necessary. However, unexpected handoffs, which are expected to take on the range of 300 msecs to complete, impose an unacceptable performance penalty on the system. Fortunately, we believe our handoff prediction will be sufficiently accurate to avoid unexpected handoff. An implementation of this protocol is currently in use on the InfoPad system.

Chapter 7

Conclusion and Directions for Future Work

In this chapter, we conclude this dissertation with a summary of research contributions and examination of areas for future work.

7.1 Research Contributions

In this dissertation, we have examined various methods to allow mobile users to roam without interruption to their network communication. The objective of this work has been to create a networking environment in which mobile users can use continuous media, such as audio and video, as well as more traditional network services. We have developed and used several new techniques to support the communications handoff without adversely affecting loss, bandwidth or end-to-end delay.

In future ubiquitous information access systems, the most common form of mobility related handoff will involve cellular wireless networks. As a result, any network protocols designed for mobility must be optimized for this common case. Two unique aspects of cellular networks that can be exploited to improve handoff performance are the geographical

layout of cells and the location information provided by many cellular technologies. This information can be combined to predict likely handoffs. In this dissertation, we have introduced the concept of these predictions, which we call “mobility hints,” and shown how they may aid handoff processing.

In examining how hints may improve handoff, we have shown that multicast-based handoff algorithms best take advantage of the knowledge provided by the handoff predictions. The multicast routing allows us to perform the equivalent of pre-fetching and caching for the routing of packets to mobile hosts. Before the re-routing of data is actually requested, we establish a route to the new base station through the use of multicast. Similarly, the base stations that are part of the multicast group compose our cache for handoff targets. As with memory caching, requests complete more quickly if they have been correctly predicted. In our system, handoffs complete more quickly when the target base station has already joined the multicast. The primary cost of this solution is the use of backbone network bandwidth. Two important factors that limit this cost are:

1. The bandwidth of wireless networks is low, typically 1 — 2 orders of magnitude less than similarly complex wired networks.
2. By carefully placing base stations on the backbone network topology, we can place base stations likely to be involved in a handoff near each other.

This limits the scope of the necessary multicast by localizing the handoff targets. The multicast routing of data also enables some new approaches to solving the handoff problems of state distribution and data loss.

The most common technique that has been used to prevent data loss during handoff is data forwarding. However, data forwarding has several important drawbacks, including:

1. Inefficient routing of data resulting in temporary increase in end-to-end delays and possible decrease in bandwidth.
2. Possible out-of-order delivery of packets while routes stabilize.

3. The danger of forwarding loops.

In this dissertation, we have introduced intelligent buffering techniques that dramatically reduce the data loss without the drawbacks of forwarding. These techniques are made possible by the use of hints and multicast-based routing protocols.

In addition, past efforts have used techniques similar to forwarding to transfer base station state during a handoff. This adds considerable delays to the handoff processing. We have shown how state can be replicated in likely handoff targets with the use of multicast packet delivery.

To summarize, we have developed the following techniques to improve handoff:

1. *Hints*, which are location information from the cellular wireless system, to predict handoff.
2. *Multicast*, based on these hints, to efficiently establish routing in advance of handoff.
3. *Intelligent buffering*, enabled by the multicast of data, to reduce data loss without the use of complicated forwarding.
4. *State replication*, enabled by the multicast, to avoid explicit state transfers during the handoff processing.

We have successfully demonstrated each of these techniques in the mobility support implemented for the Daedalus system. Our experience in this testbed has helped to identify areas for continued work in the area of handoff and new topics for exploration in mobile computing.

7.2 Future Work in Handoff

Although this research project has explored many areas in improving handoff processing, there remain many open areas of research. These areas include the development of special

multicast algorithms especially suited for mobility support and the analysis of user mobility in different environments.

7.2.1 Special Multicast Support

The use of DVMRP-base IP Multicast limited our implementation of handoff in the Daedalus testbed to campus area support. The drawbacks of the DVMRP [Deer89] include inefficient support for wide-area, sparse membership groups and high overhead of maintaining multicast routing tables. The delivery of packets from the home agent and route optimizing corresponding hosts to a few, possibly far away base stations is not well suited to DVMRP-based IP Multicast. Future multicast protocols may be better suited to the task. However, we can also support the delivery necessary with a much more limited multicast protocol designed specifically for mobility support. The necessary multicast needs to support the following:

- *Distinction between sources and sinks* — Home agents and route optimizing hosts do not want to receive packets from each other. They only desire their packets to be delivered to the base stations near the mobile host.
- *Wide-area, sparse membership support* — Home agents and route optimizing hosts may be far away from the mobile host. In addition, the only members of each multicast group are the home agent, a few route-optimizing hosts, and a some of the base stations near the mobile host.
- *Scalable routing tables* — Current multicast routing tables scale proportionally to the number of multicast addresses times the number of sources. Sources for mobile multicast may be able to provide routing hints, such as the regional area of the mobile, to aid in reducing the size of multicast routing tables.

7.2.2 Exploitation of Mobility Traces for More Effective Support

We implemented our algorithms on testbeds with few mobile hosts and relatively few users. This combined with the unavailability of movement traces in commercial systems

prevented us from analyzing user mobility. As a result, the handoff prediction algorithms are currently based on measurements from the wireless networks and the geographic layout of cells. The accuracy of predictions has room for improvement, especially in networks without in-cell position information. One promising technique is to employ user movement patterns. Our intuition indicates that the following is likely true about user movement:

- Users have long periods during which they never move between cells.
- Individual users may have unique movement characteristics making user profiles advantageous.
- Indoor mobility patterns are very different from outdoor patterns.
- Group meetings and migrations may be common and introduce highly transient and localized handoff loads.

As wireless systems mature more information will become available about typical user patterns. This information may verify the above intuition and may identify other attributes that are important characteristics of user mobility.

7.3 New Directions

At the beginning of this thesis, we stated the goal of this work as supporting future ubiquitous information access systems. We accomplished a large measure of this goal by making mobility transparent to current network services. However, there are other important differences that prevent access to current services in this new environment. These include inherently lower bandwidth and higher error-rates. In addition, portable computers contain less storage, perform computation slower and possess smaller screens. Much work needs to be done to hide these differences from applications as well.

7.3.1 Agents for Mobiles

To compensate for the deficiencies of portable computers, applications for mobile hosts must be written to adapt their resource demands. One possibility is to split applications, such that a portion remains on the mobile host and another part, called the agent, runs on a more capable machine. The agent portion performs the bandwidth and computationally intensive portions of an application and transmits a low bandwidth stream of information to the mobile host. The portion of the application that remained on the mobile host performs any final, simple operations necessary to provide the desired service to the user. For this form of support, each application must be modified. The agents also create new problems for supporting mobility. For example, an application may start its associated agent near its current location. However, if the user moves while the application is still active, the separation between agent and application will grow and performance may degrade. The development and analysis of agents for mobile computing are important areas for future work.

7.3.2 Transport Protocols

Transport protocols also make many assumptions about their operation environment. For example, TCP is a reliable transport protocol tuned to perform well in traditional networks where congestion is the primary cause of packet loss. However, networks with wireless links incur significant losses due to bit errors. This environment violates many of the assumptions made by TCP, causing degraded end-to-end performance. Several proposals [Bala95b] for improving end-to-end TCP performance have appeared in recent publications. There remains work to be done in modifying existing protocols for operation in the ubiquitous information access environment. In addition, systems such as satellite communications, which provide highly asymmetric bandwidth and long round-trip times, will need entirely new protocols. We need new protocols to support the wide variety of new network technologies that are emerging.

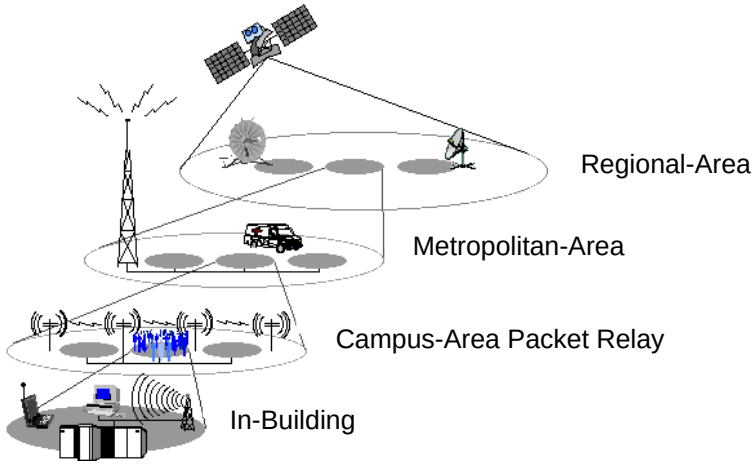


Figure 7.1 Wide-area Overlay Networks

7.4 Overlay Networks / Adaptive Systems

As new wireless technologies emerge, we have realized that each provides unique performance advantages and disadvantages. Given this variance in performance, it is likely that future mobile hosts may contain multiple network devices. As a result, wireless networks of the future are likely to be heterogeneous with each host will be simultaneously connected to different wireless networks. Figure 7.1 shows a set of different *overlay* networks a single portable may be capable of accessing. These overlay networks are characterized by different coverage, bandwidth, latency, and bit-error rates. Future network management software and application support services must allow mobile computers to adapt dynamically to the state of their network connectivity.

For example, the development of mobile hosts with access to multiple wireless technologies creates a new form of handoffs called *vertical handoffs*. The vertical handoffs, which are handoffs between cells of different technologies, introduce some new twists to the problem of handoffs. For example, in the handoff systems described in this thesis, relatively simple techniques are used to compare connectivity in different cells. To support vertical handoff, we need more sophisticated techniques for comparing the connectivity available in cells using different technologies.

Future projects should examine the problems of supporting routing and handoff in heterogeneous networks, characterizing network performance, and supporting applications that are aware of the quality of their network connections and adapt to changes in this quality [Kat94]. The goal of a project should be to allow users to access the same applications and information despite significant changes in the quality and state of their network connectivity.

References

[Abra70] N. Abramson. The aloha system – another alternative for computer communications. In *Fall Joint Computer Conference, AFIPS Conference Proceedings,* volume 37, pages 281 – 285, 1970.

[Acam94] A. S. Acampora and M. Naghshineh. An Architecture and Methodology for Mobile-Executed Handoff in Cellular ATM. *IEEE Journal on Selected Areas in Communications*, 12(8):1365–1375, October 1994.

[Amir95] E. Amir, H. Balakrishnan, S. Seshan, and R. H. Katz. Efficient TCP over Networks with Wireless Links. In *Proc. HotOS-V,* May 1995.

[Anon93] Anonymous. Cellular Magic. *2600 Magazine*, pages 4–12, Spring 1993.

[ATT] American Telephone and Telegraph. *WaveLAN: PC/AT Card Installation and Operation*.

[Bakr94] A. Bakre and B. R. Badrinath. I-TCP: Indirect TCP for Mobile Hosts. Technical Report DCS-TR-314, Rutgers University, October 1994.

[Bakr95] A. Bakre and B. R. Badrinath. Handoff and System Support for Indirect TCP/IP. In *Proc. Second Usenix Symp. on Mobile and Location-Independent Computing*, April 1995.

[Bala95a] H. Balakrishnan, S. Seshan, E. Amir, and R.H. Katz. Improving TCP/IP Performance over Wireless Networks. In *Proc. 1st ACM Conf. on Mobile Computing and Networking*, November 1995.

[Bala95b] H. Balakrishnan, S. Seshan, and R.H. Katz. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. *ACM Wireless Networks*, 1995. To appear.

[Bals93] D. M. Balston and R. C. V. Macario. *Cellular Radio Systems*. Artech House, Inc., Boston, MA., 1993.

[BL95] T. Berners-Lee. *Hypertext Markup Language - 2.0*. MIT Laboratory for Computer Science, 545 Technology Square, Cambridge, MA, Nov 1995. RFC-1866.

[Brad89] R. T. Braden. *Requirements for Internet Hosts – Communication Layers*. Information Sciences Institute, Marina del Rey, CA, October 1989. RFC-1122.

[Brod93] R. Brodersen. Personal communication, June 1993. InfoPad Radio Design.

[Cace94] R. Caceres and L. Iftode. Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments. *IEEE JSAC*, 13(5), June 1994.

[Chan94] A. Chandrakasan, A. Burstein, and R. Brodersen. A Low Power Chipset for Portable Multimedia Applications. In *Digest of Technical Papers, ISSCC 1994*, pages 82–83, February 1994.

[Dae95] The Daedalus Project Home Page. http://daedalus.CS.Berkeley.EDU/, 1995.

[Deer89] Steve Deering. *Host Extensions for IP Multicasting*. RFC, SRI International, Menlo Park, CA, Aug 1989. RFC-1112.

[Deer91] S. E. Deering. *Multicast Routing in a Datagram Internetwork*. PhD thesis, Stanford University, December 1991.

[Eage92] J. S. Eager. Advances in Rechargeable Batteries Spark Product Innovation. In *Proc 1992 Silicon Valley Computer Conference*, pages 243–253, August 1992.

[Erik94] H. Eriksson. Mbone: The multicast backbone. *Communications of the ACM*, 37(8):54–60, 1994.

[FCC95] Federal Communications Commission Home Page. http://fcc.gov/, 1995.

[Ferr90a] D. Ferrari. Client requirements for real-time communication services. *IEEE Communications Magazine*, 28(11):65–72, November 1990.

[Ferr90b] D. Ferrari and D. Verma. A scheme for real-time communication services in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368–379, April 1990.

[Ferr92] D. Ferrari, A. Banerjea, and H. Zhang. Network support for multimedia a discussion of the tenet approach. Technical Report 92/072, Computer Science Division, Univ. of California at Berkeley, November 1992.

[Gare79] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman and Company, New York, 1979.

[Ghai94] R. Ghai and S. Singh. An Architecture and Communications Protocol for Picocellular Networks. *IEEE Personal Communications Magazine*, 1(3):36–46, 1994.

[Ioan91] J. Ioannidis, D. Duchamp, and G. Q. Maguire. IP-based Protocols for Mobile Internetworking. In *Proceedings of SIGCOMM '91*, pages 235–245, 1991.

[Ioan93a] J. Ioannidis. *Protocols for Mobile Internetworking*. PhD thesis, Columbia University, 1993.

[Ioan93b] J. Ioannidis and G. Q. Maguire. The Design and Implementation of a Mobile Internetworking Architecture. In *Proc. Winter '93 Usenix Conference*, San Diego, CA, January 1993.

[Jaco88] V. Jacobson. Congestion avoidance and control. In *SIGCOMM 88*, August 1988.

[Jaco92] V. Jacobson, R. T. Braden, and D. A. Borman. *TCP Extensions for High Performance*. RFC, May 1992. RFC-1323.

[John93a] D. Johnson. Ubiquitous Mobile Host Internetworking. In *Proceedings of Fourth Workshop on Workstation Operating Systems*, October 1993.

[John93b] D. B. Johnson. Transparent Internet Routing for IP Mobile Hosts. IETF Draft, July 1993.

[John95] D. B. Johnson and C. Perkins. Route Optimization in Mobile IP. IETF Mobile-IP Draft, July 1995.

[Jubi87] J. Jubin and J. Tornow. The darpa packet radio network protocols. *Proceedings of the IEEE*, 75(1):21 – 32, January 1987.

[Karn87] P. Karn and C. Partridge. Improving Round-Trip Time Estimates in Reliable Transport Protocols. In *SIGCOMM 87*, August 1987.

[Keet93] K. Keeton, B.A. Mah, S. Seshan, R.H. Katz, and D. Ferrari. Providing Connection-Oriented Service to Mobile Hosts. In *Proc. 1993 USENIX Symp. on Mobile and Location-Independent Computing*, August 1993.

[Lein87] B. Leiner, D. Nielson, and F. Tobagi. Issues in packet radio network design. *Proceedings of the IEEE*, 75(1):6 – 20, January 1987.

[Linn95] J. P. Linnartz. Personal communication, August 1995. Wireless Communications Networks/Radio Propagation Tutorial.

[Mah93] B. A. Mah. A Mechanism for the Administration of Real-Time Channels. Technical Report 93/735, Computer Science Division, Univ. of California at Berkeley, March 1993.

[MBo95] MBONE Information Web. http://www.best.com/ prince/techinfo/ mbone.html, 1995. MBone WWW Page.

[McCa93] S. McCanne and V. Jacobson. The BSD Packet Filter: A New Architecture for User-Level Packet Capture. In *Proc. Winter '93 USENIX Conference*, San Diego, CA, January 1993.

[McCa95] S. McCanne and V. Jacobson. vic: A Flexible Framework for Packet Video. In *Proc. ACM Multimedia '95*, November 1995. to appear.

[Myle93] A. Myles and D. Skellern. Comparing four ip based mobile host protocols. *Computer Networks and ISDN Systems*, 26:349–355, 1993.

[Net95] Welcome to Netscape. http://home.netscape.com/, 1995. Netscape's WWW Page.

[Paul95] S. Paul, E. Ayanoglu, T. F. LaPorta, K. H. Chen, K. K. Sabnani, and R. D. Gitlin. An Asymmetric Link-Layer Protocol for Digital Cellular Communications. In *Proc. InfoComm '95*, 1995.

[Perk95a] C. Perkins. IP Encapsulation within IP. IETF Mobile-IP Draft, October 1995.

[Perk95b] C. Perkins. IP Mobility Support Draft 12. IETF Mobile-IP Draft, 1995.

[Post81a] J. B. Postel. *Internet Protocol*. RFC, Information Sciences Institute, Marina del Rey, CA, September 1981. RFC-791.

[Post81b] J. B. Postel. *Transmission Control Protocol*. RFC, Information Sciences Institute, Marina del Rey, CA, September 1981. RFC-793.

[Schi91] D. Schilling. Broadband CDMA for Personal Communications Systems. *IEEE Communications Magazine*, 29(11):86–93, Nov 1991.

[Schr92] M. Schroeder. Personal communication, October 1992.

[Sesh94] S. Seshan, M. T. Le, F. L. Burghardt, and J. Rabaey. Software Architecture of the InfoPad System. In *Mobidata Workshop*, Rutgers University, N.J., November 1994.

[Shen92] S. Sheng, A Chandrakasan, and R. Brodersen. Portable Multimedia Terminal. *IEEE Communications Magazine*, 30(12):64–76, Dec 1992.

[Stev94] W. R. Stevens. *TCP/IP Illustrated, Volume 1*. Addison-Wesley, Reading, MA, Nov 1994.

[Yava94] R. Yavatkar and N. Bhagwat. Improving End-to-End Performance of TCP over Mobile Internetworks. In *Mobile 94 Workshop on Mobile Computing Systems and Applications*, December 1994.

[Zhan92] H. Zhang and T. Fisher. Preliminary Measurement of the RMTP/RTIP. In *International Workshop on Network and Operating System Support for Digital Audio and Video*, San Diego, CA, November 1992.