ASSESSMENT AND OPTIMIZATION OF SCHEMES FOR TRACKING AND ROUTING TO MOBILE USERS IN PACKET-BASED NETWORKS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Cristina A. Hristea
August 2002

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Fouad Tobagi
(Principal Adviser)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

John Hennessy
(Secondary Adviser)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Mary Baker

Approved for the University Committee on Graduate Studies.

Abstract

Although a lot of progress has been made in recent years. supporting mobility in the Internet is still a difficult challenge.

The Internet today consists of multiple networks. interconnected by IP routers. In each network multiple subnetworks may coexist. also connected via routers. One subnetwork may connect multiple LAN segments by means of MAC switches. For scalability purposes. IP addresses are hierarchical. which allows routing decisions to be performed using address aggregates. Due to this addressing structure. the IP address of a mobile user needs to change dynamically as the user moves from one subnetwork to another (or one network to another) to match addressing at the new subnet (or the new network). Unlike IP addresses. MAC addresses are flat. as the scale of the problem is smaller. Switches inside a subnetwork track individual users and forward packets to them.

If a source sends packets to a destination that is moving and changing IP addresses. the contents of the packet may refer to a destination address that is stale. and the packet may be discarded. or redirected by some special-purpose entity in the network known as an agent. The source may be informed of the change in the IP address of the destination as well, so that future packets can be sent directly to its new location. As users move inside a subnetwork. changes need to take place at one or more switches tracking that user. in an attempt to maintain connectivity to the user at all times.

Tracking updates may take a long time to arrive at the agents. end-hosts and switches. which can result in a temporary loss of connectivity to the user. This becomes particularly noticeable when users are engaged in streaming multimedia applications. and may even result in the abortion of TCP sessions. Furthermore. current technologies significantly limit the number of individual addresses that can be tracked by a device built at a reasonable cost. Therefore. any solution to support mobility must deal with inherent delays caused by distances between the moving user and entities in the network that track and assist the user. as well as limitations of the current technology in terms of cost and performance.

In this thesis we assess. design and optimize schemes to support mobility in the Internet. These schemes exploit techniques called address-lookahead. packet n-casting. transparent learning and light-weight explicit registration. Via numerical simulation. we demonstrate considerable improvements in the user-perceived quality of applications. at no significant increase in cost. For example. 40% more voice calls can experience a high level of quality. at the cost of only a 3% increase in the bandwidth consumption in the Internet.

Acknowledgments

This thesis would not have been possible without the help of many people. First. I would like to thank my husband. David Seibert for all the love and support he has given me during the thesis effort. I am grateful to my adviser. Fouad Tobagi for the freedom of exploration in my research and for his continued support. Thanks go to my sponsors at the Stanford Networking Research Center. particularly Detlef Clawin. Atsushi Shionozaki and Joseph Kim. I am grateful to my associate adviser John Hennessy for the many inspiring chats we have had. and to Mary Baker for her comments on this thesis. I would like to thank my family, particularly my dad who. as a Ph.D. has motivated me to attain the highest level of learning in my field. Finally. I would like to thank my research group at Stanford (Benjamin. Athina. Chuck. Wael. Lola. David H., Amit. Jiang, Christophe. David O.) and many friends in my life who have made the Ph.D. experience a fun and enjoyable one.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the broadest sense. the term mobile networking refers to a networking technology that allows users to maintain uninterrupted end-to-end connectivity while moving (at any speed) over a geographical area (building. campus. metropolis or even the entire world). Such technology is often associated with wireless communication.

Mobile networking already exists today for voice communication services offered with cellular telephony. however it relies on circuit-switching. which. in general. is appropriate only for handling voice traffic. With the growing interest in providing data communication services and access to the Internet to wireless mobile users. there is a need to seek a packet-based mobile networking solution to support any communication service. Indeed. packet switching and the Internet protocol are quickly becoming the networking standards towards which all networks are converging. including those used by the traditional telecommunication sector. as less appropriate circuit-switched technologies are abandoned. Interest in having mobile users access the Internet is evidenced by the increase in cellular phones with built-in Internet access. PDAs and laptops with wireless interfaces and car navigation systems with Internet connectivity.

Although a lot of progress has been made towards supporting mobility in the Internet, this problem still poses difficult challenges. To understand these challenges, let us first review some basics of Internet networking.

The Internet today consists of multiple networks, interconnected by IP routers. In each network multiple subnetworks may coexist, also connected via routers. One subnetwork may connect multiple LAN segments by means of MAC switches. For scalability purposes, IP addresses are hierarchical, which allows routing decisions to be performed using address aggregates. Due to this addressing structure, the IP address of a mobile user needs to change dynamically as the user moves from one subnetwork to another (or one network to another) to match addressing at the new subnet (or the new network). This type of movement is generally referred to as *macro-mobility*. Unlike IP addresses, MAC addresses are flat, as the scale of the problem is smaller. Switches inside a subnetwork track individual users in order to route packets to the appropriate LAN segment for delivery. User tracking takes place using the transparent learning protocol, which allows switches to learn about the location of users implicitly, by listening to the traffic the users generate. Movement inside such a network with a flat address space is usually referred to a *micro-mobility*.

In the rest of this chapter, we will describe the challenges and approaches for supporting macro-mobility and micro-mobility respectively. We will then describe one final approach to supporting macro-mobility that relies on a network infrastructure built using the techniques for micro-mobility support proposed in this thesis.

1.1 Macro-Mobility Support

To handle macro-mobility, one or more entities in the Internet are required to track changes in the IP address of mobile user, so that packets can be properly forwarded to the new address. For example, Mobile IP exploits IP agents that redirect any traffic sent to the old IP address of a mobile user to its new address [21]. Binding updates may be sent to each corresponding host, CH, of a mobile, to inform them of the new address [36, 21]. Future packets sent by these hosts will be directed to the new address of the mobile user.

Unfortunately, the acquisition of a new IP address can take a very long time (on the order of hundreds of milliseconds [38]). Moreover, updating the tracking entities can take time, due to propagation delays in the Internet. Consequently a mobile can experience blackouts at times of movement, during which no packets can be received. This can result in packet loss, unless packet buffering is performed in the Internet to assist the user. However, this technique can lead to increased packet delay, which may result in loss of interactivity, affecting the overall user-perceived quality of interactive applications. Handover delay can even affect reliable applications such as those employing the TCP-IP protocol. Blackouts may lead to packet re-transmissions, however there is a maximum number of re-transmissions allowed, after which the TCP sessions must be aborted.

Given these delay overheads, it becomes important that the network act in advance of user movement (instead of reacting to user movement), to enable the mobile host to acquire the new IP address ahead of time. Indeed, various proposals have been made in the context of Mobile IP to allow a mobile host to perform address lookahead while still at the old subnet, by exploiting hints from the lower layers [19, 20]. Moreover, packet n-casting may be performed at the agents to allow traffic to be received at any of the IP addresses associated with the user at a given time [22].

Apart from Mobile IP, a number of solutions exist to support macro-mobility [36, 40], but these also suffer from the same delay impairments caused by the address acquisition process and propagation delays in the Internet. Hence in our work, we propose that address lookahead and packet n-casting be also exploited in the context of these schemes for enhanced user-perceived quality of Internet applications.

Thus, in the first part of this thesis (Chapter 2) we assess the contribution of techniques known as address lookahead and packet n-casting to optimizing the performance of various schemes for supporting macro-mobility in the Internet. We show that the reliability of TCP/IP applications can increase to almost 100%, while up to 40% more voice calls experience a high level of quality.

1.2 Micro-Mobility Support

Over the past decade, we have witnessed tremendous developments in LAN technologies, such as increases in switch processing by a few orders of magnitude, and increases in link bandwidth and distances covered (owing to the fiber optics technology). These advances resulted in an increase in the size of LANs, and more recently, the deployment of such technologies in metropolitan areas (MANs). When users roam inside MANs of this type, they may depart one LAN segment and join another. Tracking information at one or multiple switches in the subnetwork can become stale due to this movement. Hence, a packet that is received at the switch for the mobile user will be sent to the wrong LAN segment, where it will be discarded. This can result in a temporary loss of connectivity to the user, which becomes particularly noticeable when users are engaged in streaming multimedia applications.

To prevent packets form being mis-routed or even lost, packet flooding may be employed, however at the expense of increased bandwidth consumption in the network. Another choice is to employ lightweight control-based mechanisms for user tracking, however this can lead to larger databases at the switches. Scalability problems can arise as current technologies significantly limit the number of individual addresses that can be tracked by a switch built at a reasonable cost.

Thus, in the second part of this thesis (Chapter 3), we propose to devise optimal schemes for tracking users in networks that handle user micro-mobility. We show that the transparent learning protocol, currently used in switched LANs, cannot be used efficiently to track fast moving users. Additional tracking mechanisms are necessary, such as based on explicit control messaging. Furthermore, we show that, when implementing these mechanisms, the selection of parameters can drastically affect performance, and thus must be done carefully to take into account the design constraints, as well as the application and user mobility characteristics.

1.3 Macro-Mobility Support using a Micro-Mobility Infrastructure

We have seen how macro-mobility handoffs can be slow because they involve detecting that a new subnet is reached, acquiring a new address at that subnet and propagating this information to some entities in the network. To mitigate these problems we use anticipation techniques at the lower layers that permit address lookahead and packet n-casting to take place at the higher layers. Unfortunately, in some networks anticipation cannot be implemented. For example, in wireless LANs, handoffs are hard and forward, which means that connectivity with the existing wireless cell must be discontinued prior to establishing connectivity at the new cell. For these networks, we need to investigate a different method to optimize support for macro-mobility.

To this end, in the third part of this thesis (Chapter 4), we investigate the idea that a mobile user maintain a fixed IP address while roaming inside a large region that spans many IP subnets from before. This is equivalent to building a large network to support micro-mobility, within which switches track users through their IP or MAC address. Using numerical simulation, we show that this technique can result in the elimination of voice calls with poor quality.

To make building such a network feasible, we address the scalability aspect of extended LANs and their potential deployment in large metropolitan areas to serve large populations of mobile users. We note that the scalability of such infrastructures can be obtained by partitioning the switches in the infrastructure into subsets that play different roles with respect to mobile hosts and the hosts communicating with them. More specifically, we limit the set of switches in the infrastructure that track a mobile user. To reach such a mobile host, packets originating at a communicating host are broadcast over a set of switches which intersect with the tracking subset. Using numerical simulation, we show that by using network partitioning, the traffic load at the switches can be reduced by more than 50%.

In conclusion, in this thesis we address the problem of supporting micro-mobility and macro-mobility in the Internet, with the goal of improving the user-perceived quality of Internet applications for mobile users. Specifically, we assess, design and optimize various schemes for tracking mobile users that account for the state of technology, and are optimized for a variety of user mobility rates and application characteristics.

Chapter 2

Assessing the Impact of Handover Delay

2.1 Introduction

With the recent explosion in wireless mobile communication, a great deal of work is taking place to solve the problem of handling mobile users in the Internet.

Due to address prefixing at the IP routers, the IP address of a user changes dynamically as the user moves from one IP subnet to another (macro-mobility). To handle macro-mobility, some entities in the Internet are required to track the change in the IP address of the mobile user, so that packets can be properly forwarded to the new address. For example, Mobile IP exploits IP agents that redirect any traffic sent to the old IP address of a mobile user to its new address [21]. Binding updates can be sent to each CH of a mobile, to inform them of the new address. Future packets sent by these hosts will be directed to the new address of the mobile [36, 21].

Unfortunately, the acquisition of a new IP address can take a very long time (on the order of hundreds of milliseconds). The reasons for this delay are propagation delay at the local network, request processing, address conflict checking and other processing at the entities that distribute the IP addresses [38]. Moreover, updating the tracking entities can take time, due to propagation delays in the Internet.

Consequently a mobile can experience blackouts at times of movement, during which no packets can be received. This can result in packet loss, unless packet buffering is performed in the Internet to assist the user. However, this technique can lead to increased packet delay, which may result in loss of interactivity, affecting the overall user-perceived quality of interactive applications. Handover delay can even affect reliable applications such as those employing the TCP-IP protocol. Blackouts may lead to packet re-transmissions, however there is a maximum number of re-transmissions allowed, after which the TCP sessions must be aborted.

Given these delay overheads, it becomes important that the network act in advance of user movement (instead of reacting to user movement), to enable the mobile host to acquire the new IP address ahead of time. Indeed, a number of proposals have been made in the context of Mobile IP to allow a mobile host to perform address lookahead while still at the old subnet, by exploiting hints from the lower layers [19, 20]. Moreover, packet n-casting may be performed at the agents to allow traffic to be received at any of the IP addresses associated with the user at a given time [22].

Apart from Mobile IP, a number of solutions exist to support macro-mobility [36, 40], however they also suffer from the same delay impairments caused by the address acquisition process and propagation delays in the Internet.

In the work described in this chapter, we assess the contribution of address lookahead and packet n-casting to optimizing the performance of various schemes for supporting macro-mobility in the Internet. We use numerical simulation to assess the effectiveness of various macro-mobility solutions in supporting interactive multimedia and data traffic. Our simulations exploit statistical models for user mobility and application traffic and user-perceived quality measures for a realistic assessment of performance. In the context of Mobile IP, the reliability of TCP IP applications increases to 100%, while voice calls with quality that is less than excellent are almost entirely eliminated.

Furthermore, we propose that address lookahead and packet n-casting be also exploited in the context of schemes that are not based on Mobile IP. In this context, we show that the reliability of TCP IP applications can increase to 99%, while up to 40% more voice calls experience a quality level that is acceptable or better.

The chapter is structured as follows: in the next section we discuss various schemes to support macro-mobility in the Internet. In Section 2.3 we describe our simulation environment and the results we gather using these simulations. Finally, in Section 2.4 we conclude the chapter.

2.2 Related Work

2.2.1 The Mobile IP Scheme

In Mobile IP [21], a mobile host (MH) is assigned a permanent home address and a home agent (HA) in its home subnet. The domain name service (DNS) maps the domain name of the host to its home address. When the MH moves to a foreign subnet, it acquires a temporary care-of-address (COA) from some entity in the network - dynamic host configuration protocol (DHCP) server [17], an advertised foreign agent

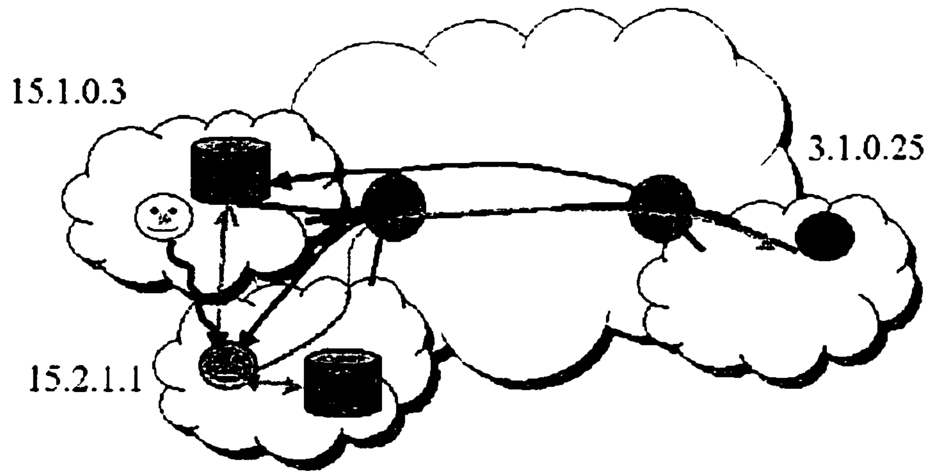


Figure 2.1: Local agent infrastructure in Mobile IP tunneling traffic for a mobile user. The MH moves from one subnet, where its COA is 15.1.0.3 to another subnet where its COA becomes 15.2.1.1. The agent at the old subnet tunnels the traffic for MH to the new subnet.

(FA) in the subnet or via autoconfiguration -, and it registers its new address with the HA. As we can see, the HA acts as a dynamic directory for that MH. However, instead of simply serving queries from a CH concerning the location of the MH, the HA actually participates in the routing of packets sent by each CH to the MH. The HA captures all IP packets addressed to the home address of the MH (e.g. via gratuitous proxy address resolution, ARP [18]) and redirects them to the COA of the MH (e.g. by using IP-in-IP tunneling). The process of sending packets to the MH via the HA is known as triangle routing.

If the HA happens to be located far way from the location of mobile users, the HA may be slow at receiving location updates for those users, affecting their ability to communicate in a timely manner. To address this issue, Mobile IP uses a special-purpose infrastructure of agents that reside close to the MH and can be used to serve the MH for tracking and routing purposes. This is shown in Figure 2.1. Agents in

Mobile IP (FAs) can be chained for the purpose of localizing user tracking. Instead of informing the HA, the FA at the new subnet has the option of sending binding updates to the FA at the subnet that the user visited previously. The FAs need to map the old COA to the new COA of mobile users.

The chaining of FAs can lead to long, inefficient routes in the case of mobile hosts frequently changing their points of attachment to the Internet. To mitigate this problem, a number of optimizations can be used. A first such mechanism is route optimization, which allows the CH to learn the COA of a mobile host by means of so called binding updates. Then, a CH with enhanced networking software can send the packet directly to the mobile host, thus avoiding redirection at the Mobile IP agents in the Internet. A second mechanism is to exploit the organization of FAs into a tree hierarchy with a small number of levels such that only a few agents are traversed at all times, regardless of the user movement characteristics [30].

Mobile IP was originally designed without any assumptions about the underlying link layers over which it operates so as to achieve the widest applicability. This approach has the advantage of facilitating a clean separation between the link layer and IP layers of the protocol stack, but it has negative consequences for handoff latency. The strict separation between the link and IP layers results in delay overhead at times of handoff, as the registration process cannot begin until after the lower layer handoff has completed.

An optimization has been recently proposed [19, 20] which works as follows: a moving host receives hints from the lower layers that it is about to cross IP subnets. At this time, the mobile host prepares for crossing by first performing an address lookahead at the new subnet. Using the new address that it acquires, it sends an

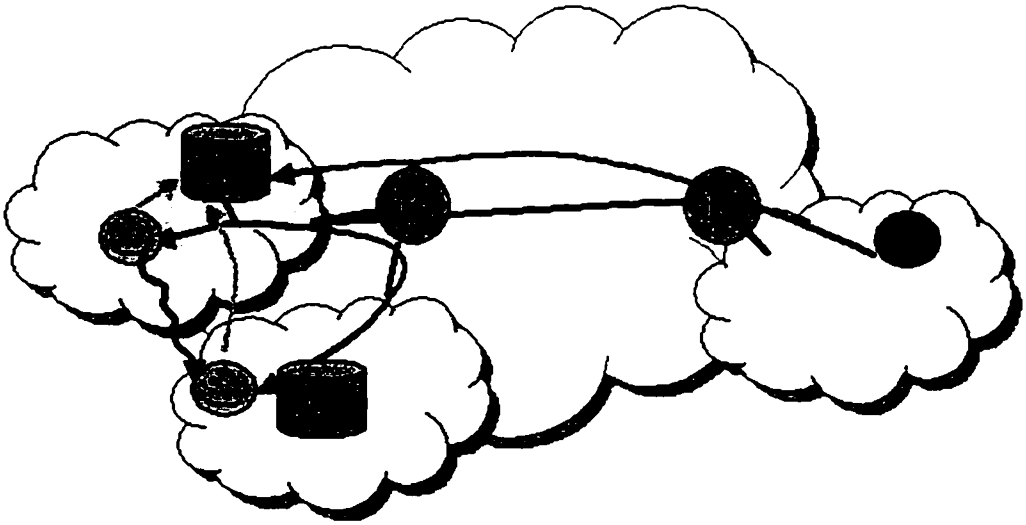


Figure 2.2: Optimizing Mobile IP by using address lookahead and n-casting at the agents. The MH sends a pre-update to the agent at the old subnet upon acquiring the new address. Once moved to the new subnet, the MH sends a post-update to the agent at the old subnet.

update to the agent at the current subnet. Then, the mobile crosses to the new subnet just in time to receive the first packet sent by each CH.

Unfortunately, problems arise with this optimization due to lack of synchronization between user movement and the updates taking place at the various agents. For example, it is possible that the updates take place well in advance of the mobile actually crossing to a new subnet. Packets that are sent to the mobile are intercepted and forwarded by the agent to the new subnet where the mobile was supposed to move. Since the mobile is not yet at this subnet, the packets are either discarded, or buffered by the new agent, possibly for a long period of time, after which the packets become useless and need to be discarded.

To mitigate this problem, another optimization was proposed based on packet bicasting (or more generally, n-casting) [22]. This is shown in Figure 2.2. As before, the mobile performs an address lookahead while still at the old subnet. It then proceeds to send a pre-update to the agent at the current subnet. When packets are received by this agent, they are bi-casted (or more generally, n-casted) to all the addresses associated with the mobile user at that time. The user can then receive the packet wherever she happens to be at that time. Once at the new subnet, the mobile sends a post-update to the old agent, telling it to discontinue casting packets to the old IP address.

2.2.2 End-to-End Schemes

Special purpose agent infrastructure may be expensive to deploy. Because of this, researchers have investigated methods to support user mobility at the end hosts. In these schemes, binding updates are exchanged between the end hosts of a connection. Once acquiring a new IP address, a mobile sends a binding update messages to each existing CH, informing them of its new IP address. This is shown in Figure 2.3. At that point, a binding entry is created in the cache, mapping the permanent address or the domain name of that host to some temporary COA. When packets are about to leave, either of two options are possible: 1) the IP layer encapsulates them with a header containing, in the destination field, the most-recent COA of the destination host [21]. 2) All the connections between that CH and MH are migrated from the old IP address to the new IP address [36].

When a HA is used in Mobile IP, the home address of a host is the invariant, consequently every packet may be redirected. This results in triangle routing, which can add delay overhead. Instead, one can take advantage of the fact that host-name lookups are ubiquitously done by most applications that originate communication

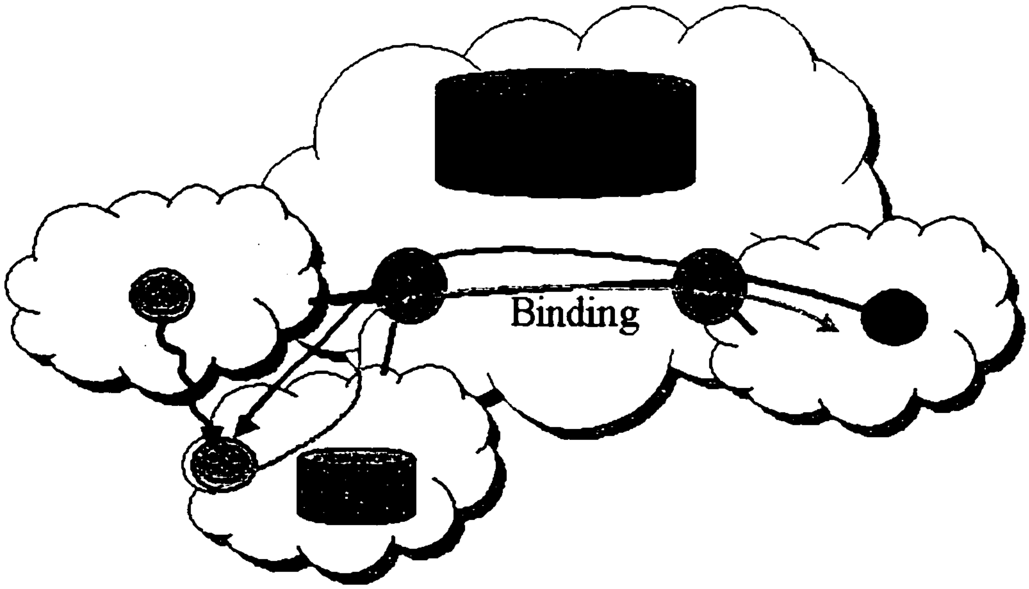


Figure 2.3: End-to-end binding between hosts. Once moved to a new subnet, the MH sends a binding update to each CH. DNS is updated as well.

with a network host, and use the DNS name as the invariant. For example, in [36], when a MH changes its IP address, it proceeds to update its DNS entry by issuing a secure DNS update. Then, hosts wishing to contact the mobile host obtain the most up-to-date address of that host, and thus can reach the mobile host directly at its new address.

Nonetheless, as before, a number of delay overheads exist that can seriously affect performance. The address acquisition can take a long time, furthermore a CH may be far away from the mobile host, resulting in binding updates that take a long time to arrive. Another problem with this solution is that the two ends of a session may move simultaneously such that neither will receive the binding update, hence communication can be entirely lost between the two hosts.

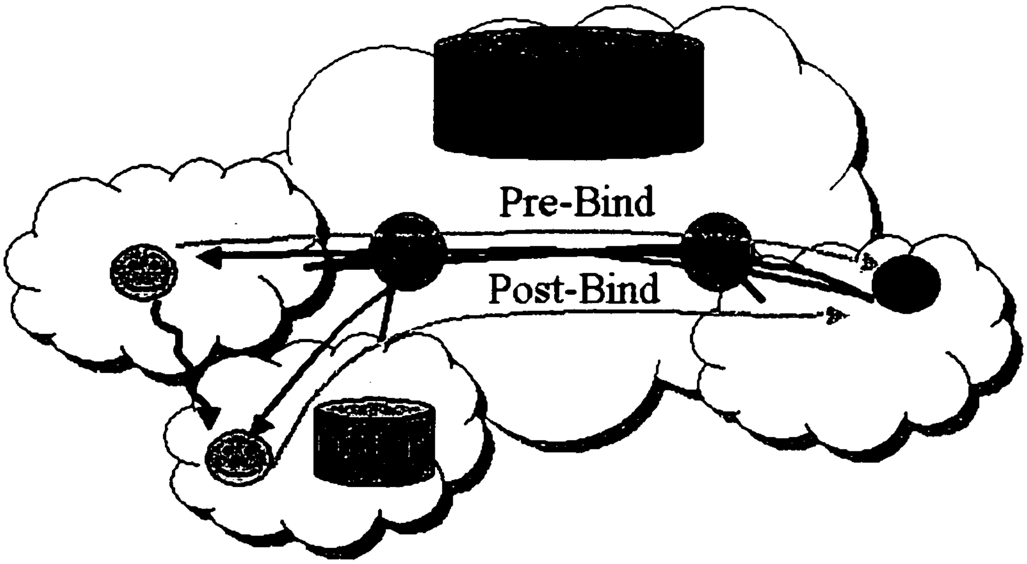


Figure 2.4: Optimizing end-to-end schemes by using address lookahead and packet n-casting at the end hosts.

2.3 Proposed Work

Mobile IP advocates that agent-based network infrastructure be deployed in the Internet for the purpose of redirecting traffic sent to incorrect IP addresses. Such infrastructure can be used to make user movement transparent to the end hosts and DNS. This is important if one cannot upgrade the hosts and, or DNS servers in the Internet to support mobility and thus to ensure compatibility with the existing Internet and its hosts. Another reason for exploiting network infrastructure for mobility support is that this infrastructure can be local to the mobile host irrespective of the movement and location of that host. Hence, handshakes between the mobile host and the network infrastructure can take place much faster than between the MH and the CH.

Supporters of end-to-end schemes advocate doing away with special-purpose, agent-based infrastructure for supporting macro-mobility in the Internet. Instead, they propose that intelligence be placed solely at the end hosts, an arguably simpler and perhaps less costly solution. The disadvantages of the end-to-end approach are its lack of compatibility with the existing Internet protocols, and the complexity of modifying the end-hosts to support the additional required functionality.

Regardless of which scheme one choses to employ, both Mobile IP and end-to-end schemes suffer from impairments caused by Internet delays. To mitigate these delays, address lookahead and packet n-casting can be exploited, albeit at an increase in the complexity of the network.

In the first contribution of this chapter, we propose to mitigate handover delay in the context of end-to-end schemes by exploiting the said optimizations, originally proposed in the context of Mobile IP. For the new scenario, we implement the optimizations in the following way:

As shown in Figure 2.4, upon acquisition of a new IP address, the MH sends a pre-binding update to each of its active CH. Upon receipt of the binding update, the CH is required to send copies of packets to the old and the new IP addresses associated with a MH, every time it wishes to send an IP packet to that user. When the wireless link gets established at the new subnet, the MH sends a post-binding update to every CH. At this time, the CH stops sending packets to the old IP address for that user. Note that, while packet loss can be reduced, it cannot be eliminated completely for the following reasons: 1) If the CH moves before it receives a pre-binding update ("simultaneous" mobile host and CH movement), the two ends will lose contact. 2) Sessions started after the pre-binding update is sent do not benefit from n-casting. 3) DNS lookups may return stale addresses, due to delay in updating DNS.

A second contribution of this chapter is assessing the impact of the optimization techniques on the performance and cost of the network. Indeed, one important disadvantage of these techniques is the increase in network cost stemming from the following: 1) the need to support address lookhead at the lower layers. 2) Data packet duplicates sent by the CH (or the agents) to the old and new MH locations. 3) The use of an additional binding control message. Important tradeoffs need to be investigated that balance minimizing network cost with maximizing network performance.

In the rest of this chapter, we describe the experiments we conduct to measure the impact of exploiting address lookahead and packet n-casting on the user-perceived quality of applications and network bandwidth consumption in the context of both Mobile IP and end-to-end solutions.

2.4 Experiments

2.4.1 Simulation Description

Our simulation environment is based on the Scalable Simulation Framework (SSF) package distributed by researchers at the Dartmouth University [14]. Four schemes for supporting mobile users are implemented as follows: 1) E2E, the basic end-to-end solution. 2) E2E_Bicast, the E2E scheme enhanced with address lookahead and packet bi-casting, a simple form of n-casting where n equals two. 3) MIP, the basic Mobile IP scheme, using chained foreign agents, route optimization and low-latency optimization. 4) MIP_Bicast, MIP enhanced with packet bi-casting at the agents.

The network consists of 64 subnets which are grouped hierarchically in a two-level hierarchy, consisting of 8 top routers and 64 lower-level routers. This is shown in Figure 2.5. When the Mobile IP schemes are used, the low-level routers also perform

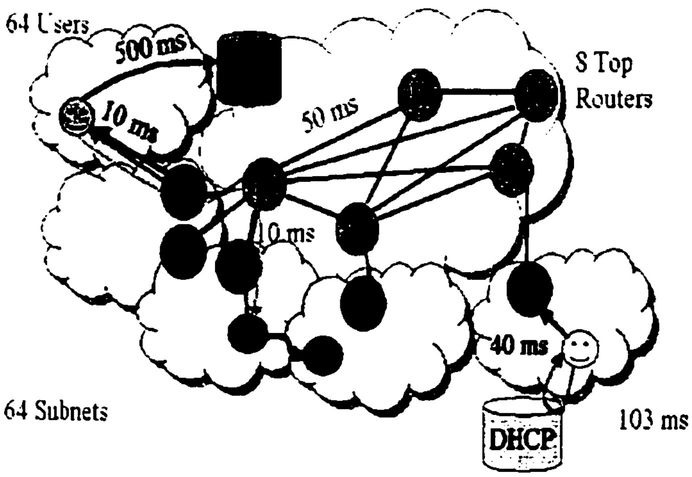


Figure 2.5: Simulation model.

the function of agents. There is no buffering of traffic at the agents, to minimize delay overhead and reduce implementation complexity at the agents and mobile users.

We tune various parameters in the model according to data from the literature [25, 38]. The top-level routers are connected via links with a traversal latency of 50 ms[1]. The top-level routers are connected to the second-level routers via links with a traversal latency of 10 ms. The hosts are connected to the second level routers via links with a traversal latency of 10 ms. The address acquisition delay is 103 ms, the FA update takes 40 ms and the DNS update completes in 500 ms.

Using SSF abstraction, we implement a user model with various mobility and application usage characteristics. At the beginning of the simulation, users are uniformly distributed across the network. We allocate one user per subnet, for a total of 64 users across the entire network. Evidently, as users move around, the distribution of users across the network changes. We use a mobility model inspired by existing literature [1] that works as follows: A user is stationary at a certain node for some

[1] For simplicity, link delays are assumed to be fixed in these simulations, although in reality, some variability may exist due to occurrences such as network congestion.

period of time that is randomly drawn from an exponential stochastic process. When this period elapses, the user moves to one of its network neighbors (movement to any neighbor is equally likely) where it again resides for some random period of time, given by the exponential stochastic process. In our simulations, we vary the rate of the exponential process, thus affecting the rate of user movement. We consider three such rates to be of interest: once every 10 seconds (fast), once every 100 seconds (medium), once every 1000 seconds (slow). To understand the choice of these rates, consider a geographical area populated with 802.11 wireless cells, one or more of which form an IP subnet. The high mobility rate corresponds to the rate at which subnets of size 300mX300m (between 10 and 50 cells) would be crossed by a car moving at about 110 km hr. The medium rate corresponds for example, to pedestrians moving at 11 km hr, crossing 300mX300m subnets. Finally, the last rate corresponds, for example, to pedestrians crossing subnets of size 3000mX3000m (or cars crossing 30kmX30km subnets, or airplanes, moving at 440km hr, crossing 60kmX60km subnets).

Unicast communication takes place among users in the network. We model two types of applications using stochastic processes: 1) TCP IP application 2) Streaming voice application.

The TCP IP application is modeled as a reliable connection, where data packets are acknowledged. A host sends data packets that are identified by sequence numbers. Upon receiving a packet, the corresponding host sends an acknowledgment in which it specifies the next sequence number it expects to receive. When a sending host does not receive an acknowledgment for a given sequence number, it re-sends the data packet corresponding to that sequence number after some timer expires or after a binding update is received, suggesting that the corresponding host has moved and may have lost packets. This process is repeated up to X times. By then, if the packet is still not acknowledged, the session is aborted at the sender. At the receiving end, the session is subsequently aborted, after some timer expires (this timer takes into account X and the timeout period at the sender). If all the packets of a session are properly acknowledged, the session is considered to be complete. In most TCP IP implementations, the number of re-transmissions X is set to 3, however we vary X to study its impact on performance.

The sliding-window protocol (with a maximum window size of 16) is used for flow control. As proposed in the recent literature, the TCP IP model does not employ congestion avoidance, since loss in our experiments is entirely caused by movement and not by network congestion [36]. Note that, in the TCP IP implementation of today, packet loss that is not caused by network congestion may still trigger the congestion control mechanisms of TCP IP. Such mechanisms can be highly undesirable in mobile scenarios where bandwidth is a scarce resource, given they can further reduce throughput to the mobile users. To prevent the triggering of congestion control mechanisms in TCP IP, one solution proposed in the literature is to distinguish loss that is caused by network congestion from other loss (such as caused by movement or fading at the wireless links) [28].

In our simulations, each user can be engaged in up to 10 simultaneous TCP sessions. For a given session, the number of packets sent by a host is chosen stochastically from a Pareto distribution with a mean of five hundred. Off times (when hosts are not engaged in sessions) are drawn from an exponential process with a mean of 25 seconds.

To measure the performance of the TCP/IP application, we count the number of aborted sessions and the number of completed sessions at the sender and receiver. Evidently, as a user moves, it may lose packets. The higher the mobility rate, the higher the number of aborted sessions and the lower the number of completed sessions we expect. A higher value of X can help increase the chance that the packet is eventually received and the session resumed.

For the streaming application, packets are issued at time intervals of 10 ms, which is a typical interspacing for packets during phone conversations. The duration of a conversation and the off-time between conversations are taken from exponential processes with variable means (from a few to a few tens of seconds). To measure performance, we use the E-model [23], which computes the user-perceived quality of calls by taking into account the average packet loss and the average packet delay of the call. The quality of a call is quantified to be one of four possible choices, depending on the amount of packet loss and packet delay incurred: excellent, good, acceptable and bad. The reader is encouraged to explore [23] for more details on this evaluation model.

To measure the impact on the cost of the network, we also measure the packet amplification caused by the bi-casting technique. The packet amplification is derived from the total number of packet transmissions aggregated over all the routers in the network.

2.4.2 Results

TCP/IP Application

In Figure 2.6 we plot the aborted sessions (on a log scale) as a percentage of the total attempted sessions during a simulation. We see that the worst performing scheme

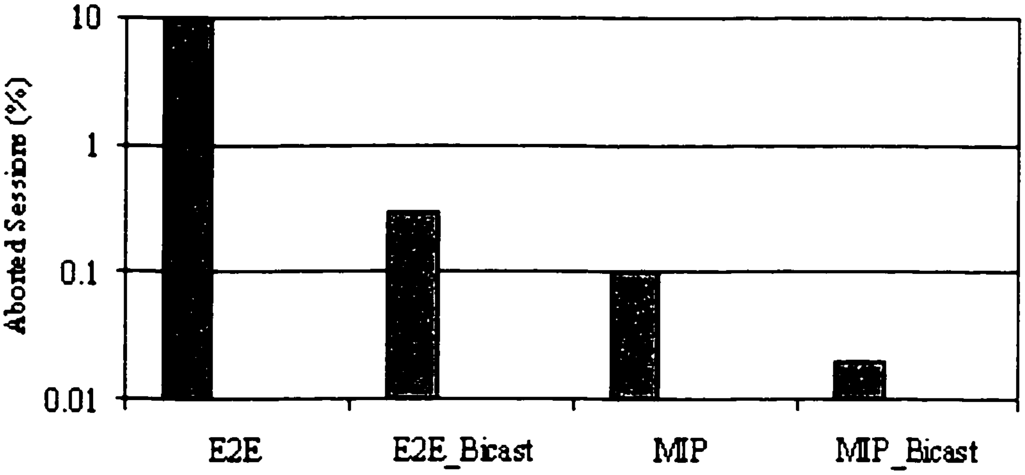


Figure 2.6: Aborted sessions for various schemes.

is E2E, resulting in almost 10% aborted sessions. E2E_Bicast shows considerable improvement in reliability as the percentage of aborted sessions decreases to less than 1%. The best performing scheme is MIP_Bicast, with a reliability close to 100%.

We identified five factors that can lead to the abortion of a session as follows:

1. The delay in updating DNS. The session cannot be initiated, due to the fact that a stale IP address is used to reach the destination. The source queries DNS while a DNS update (for the destination) is in progress and before the DNS entry gets updated. The longer the DNS update time, and the more frequent the user movement, the more likely that this scenario occurs.

2. Delay between the end hosts. Packets are sent by one end of a session during the time that a binding update travels from the other end of that session toward the sending user. If the solution does not employ agents, these packets are lost. If X is greater than 0, this packet may be re-transmitted. However, the re-transmitted packet(s) may be lost if the destination moves again. This can lead

to the abortion of the session.

3. Simultaneous user movement. (Movement is "simultaneous" when one end moves while a binding update is in progress sent by the other end). The longer the distance between the two ends of a session and the higher the frequency of user movement. the more likely that this scenario occurs. When agents are not employed. this movement leads to the abortion of the session. irrespective of the value of X.

4. The delay in address acquisition. There is a blackout time that may occur while the user acquires a new IP address. This affects solutions which do not use address look-ahead. Packets that arrive during this blackout time are lost and may be re-transmitted (for X ·0). however the packet may still not reach its destination if it moves again. This may result in the abortion of the session.

5. The delay in updating the local FA. There is a blackout time that may occur while the user is registering its new address with the local FAs. This affects solutions that do not use address lookahead and pre-binding updates. or suffer from lack of synchronization between movement and updates. Packets that arrive during this blackout time are lost and may be re-transmitted (for X ·0). however the packet may still not reach its destination if it moves again. This may result in the abortion of the session.

To assess the impact of each factor presented above. we vary the DNS update latency. the FA update latency and the address acquisition time. The results for the E2E scheme are plotted in Figure 2.7. The mean mobility rate is 0.1 moves/sec (the FA update time is irrelevant for this scheme). We can see that, when there is no delay in

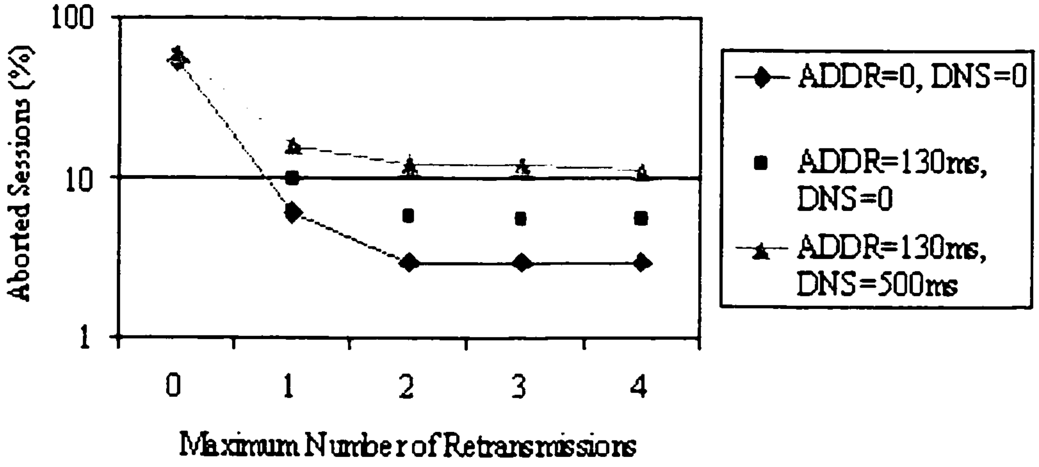


Figure 2.7: The percentage of aborted sessions when the address acquisition is 0ms or 130ms and the DNS update time is 0 ms or 500 ms.

address acquisition and DNS updates, the percentage of aborted sessions goes down to 3%. This suggests that the contribution of user mobility (cases 2 and 3) to the total number of aborted sessions is about 3% (when the user mobility rate is 0.1 moves sec). The contribution of the address acquisition, when set to 130ms (case 4) is about 3% as well. Finally, the contribution of the DNS delay, when set at 500ms (case 1) is about 6%. Case 5 does not affect the E2E scheme.

There is an optimal value of X which gives the highest throughput. Furthermore, the mobility rate as well as the scheme employed for tracking mobile users can affect the overall throughput as well.

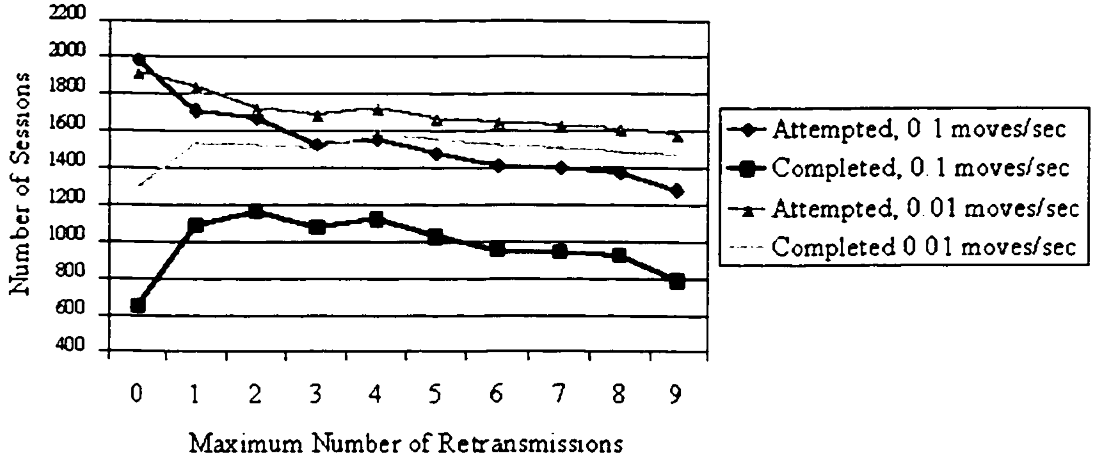
To understand these comments, let us look at Figure 2.8, where we plot the total number of sessions attempted (including sessions that did not get initiated or that were aborted midstream) and the total number of sessions completed for various values of X, for various schemes and mobility rates. The maximum number of active TCP sessions per user is 1, the scheme employed is E2E and the mobility rate are 0.1 moves/sec and 0.01 moves/sec in Figure 2.8a). The mobility rate is 0.1 moves sec and the schemes employed are E2E and MIP in 2.8b).

As we can see from Figure 2.8a), there is an increase in the number of sessions completed as N increases from 0 to 2, and a decrease as N increases further. There are about 120% more sessions completed when N=2 compared to N=0 but 30% fewer sessions completed when N=9 compared to N=2 for the mobility rate of 0.1 moves sec. The reason for the early rise is the increase in the reliability of a given sessions as increased amounts of packet re-transmissions are allowed. The reason for the latter observation is that sessions progress much more slowly when N is large, due to the large number of packet re-transmissions, thus fewer sessions can be started within some fixed amount of time of one simulation.

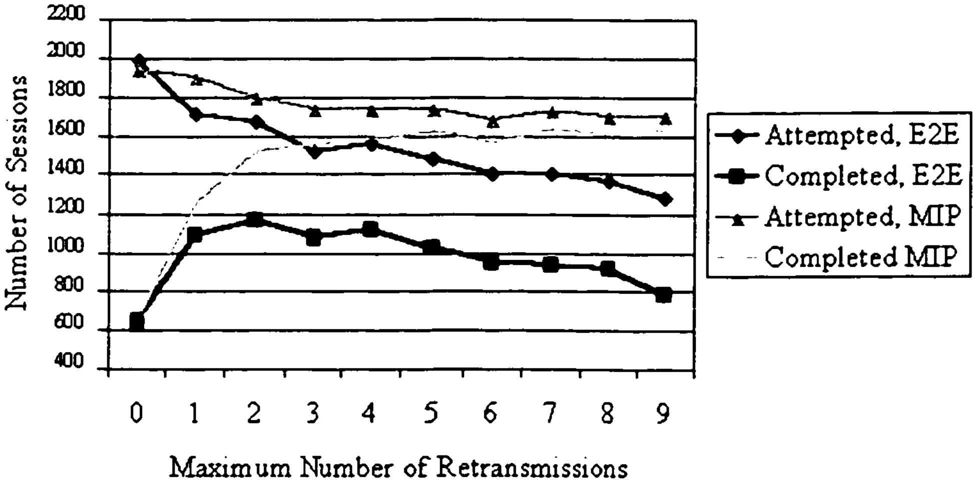
By selecting the points where the number of completed sessions is maximized, we conclude that the optimal value of N for the mobility rate of 0.1 moves sec is 2, while for the mobility rate of 0.01 moves sec is 4. Note that these values are slightly different from those used in common implementations of TCP IP, where N is usually set to 3. Nonetheless, the throughput achieved when N equals 3 is very close to the optimal, thus this choice is justified.

The second observation we made is that the mobility rate affects the overall throughput. In fact, the larger mobility rate the lower the throughput. We see that there are about 50% more sessions completed and 50% fewer sessions aborted when the mobility rate is 0.01 moves/sec versus 0.1 moves sec.

Finally, we claim that the scheme employed to handle mobility can also affect the results. Indeed, as can be seen from Figure 2.8b), throughput is better for MIP than for E2E. Also, the optimal values for N differ from N=2 for E2E to N=5 for MIP.



(a) Mean mobility rate 0.1 moves sec



(b) Mean mobility rate 0.01 moves sec

Figure 2.8: The total number of sessions for various values of X (maximum number of re-transmissions).

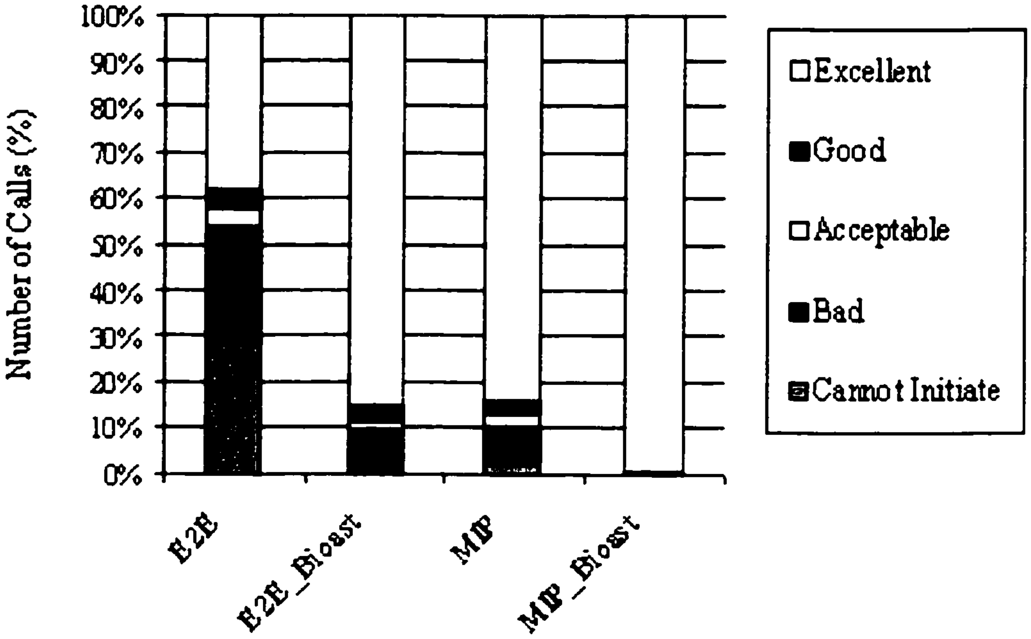


Figure 2.9: The breakdown of calls according to the user perceived quality for various schemes.

Voice Application

In Figure 2.9 we plot the breakdown of calls according to the user perceived quality for various schemes. The worst performing scheme is E2E where more than 50% of calls have quality that is less than acceptable. The E2E_Bicast scheme shows a considerable improvement in performance as only fewer than 10% of calls have performance that is less than acceptable. The best performing scheme is MIP_Bicast where calls have excellent performance across the board.

To understand these results, let us look at the average packet loss experienced during calls, which is one determining factor for call quality. In Figure 2.10 we plot the percentage of calls that exhibit the following average loss: greater than 20%, less than 20% but greater than 10%, less than 10% but greater than 5%, less than 5% but greater than 1% and less than 1%. We see that, in the case of E2E, about 30%

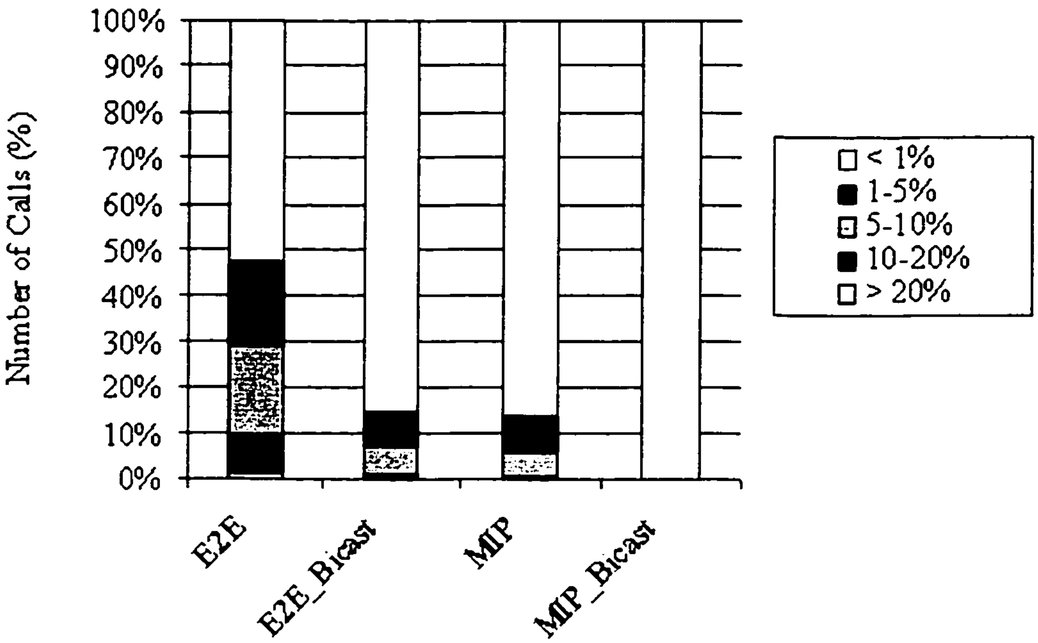
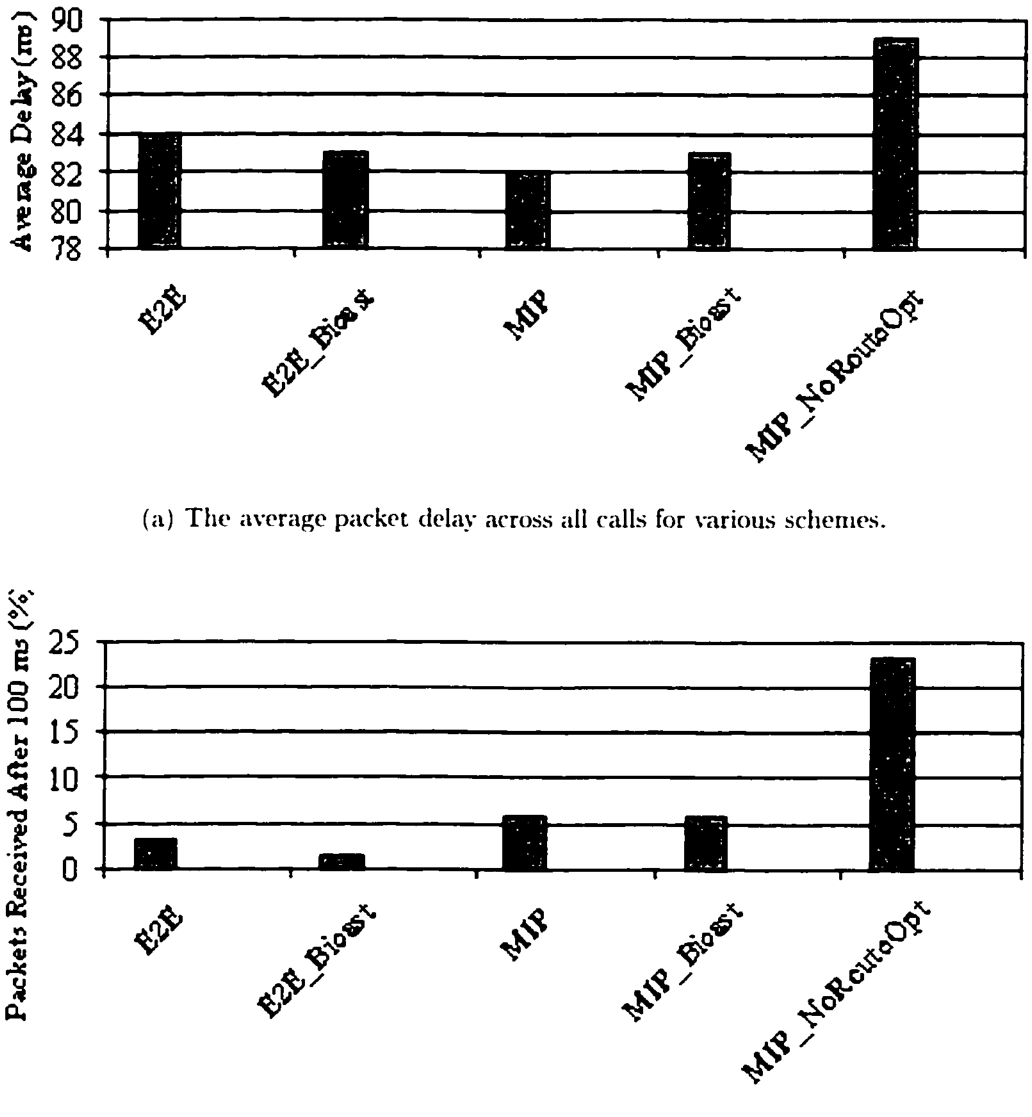


Figure 2.10: The breakdown of calls according to the average packet loss experienced for various schemes.

of calls have a packet loss greater than 5%. This is due to the delay in receiving binding updates at the CH and DNS. In E2E_Bicast, only about 10% of calls display this loss behavior. Note that packet loss in not eliminated. This is due to situations where the pre-binding update is not received at the CH, hence the scheme degenerates to the E2E scheme. In MIP, also about 10% of calls encounter more than 5% loss, due to lack of synchronization at the agents. In MIP_Bicast, loss is almost entirely eliminated.

The second determining factor for quality is packet delay. In Figure 2.11 we plot the average delay incurred during a call and the percentage of packets with a delay greater than 100 ms (which represents the threshold at which humans start to perceive loss of interactivity), as observed for various schemes. The average packet delay is relatively constant for the four schemes previously studied. The reason for this is that



(a) The average packet delay across all calls for various schemes.

(b) The percentage of packets with a delay greater than 100 ms.

Figure 2.11: Packet delay and various macro-mobility schemes.

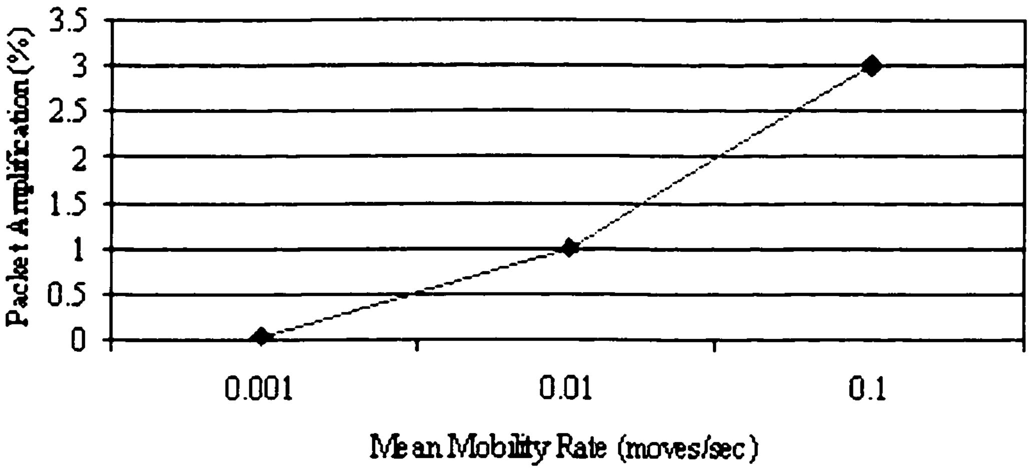


Figure 2.12: Packet overhead, as a percentage to user data traffic for various user mobility speeds.

all schemes use route optimization, which allows traffic to flow across the most direct path between the source and destination of that packet. For comparison, we also plot the average delay exhibited by the MIP scheme when no route optimization is supported. We notice that packet delay increases drastically by almost 25%. We observe that about 23% of packets incur an average delay greater than 100 ms suggesting that lack of route optimization support can lead to some loss of interactivity.

Bi-Cast Overhead

We have seen how packet bi-casting improves performance. However, this does not come without a price. Evidently, there is bandwidth overhead which arises for two reasons: 1) two copies of a data or control packet may be distributed during some window of time between the receipt of the pre-binding update and the receipt of the post-binding update. 2) two binding updates have to be sent per move instead of one. In Figure 2.12, we plot the percentage increase in the total number of packets (including control packets) that are routed inside the network in the case of the optimized end-to-end scheme with pre post binding updates. for various mean mobility speeds. As expected. there is an increase in the number of packets which increases with the mobility speed. The overhead amounts to 3% at the highest user mobility rate. We believe this is a small amount which can be easily accommodated in the networks of today. even at the wireless segments. as 3% represents only a few Kb s (typical) to a few Mb s (highest speed 802.11a wireless LANs).

2.5 Conclusions

Schemes to support macro-mobility in the Internet can be impaired by various delay overheads arising from the address acquisition process. and propagation delays among various network entities. Two techniques based on address lookahead and packet n-casting have been proposed to hide these delay overheads. In this work. we assess the impact of these optimization on user-perceived performance and network cost via numerical simulation. We demonstrate that such techniques can be very effective at improving the user-perceived quality of voice calls and the reliability of TCP-based applications. at no significant increase in cost.

Chapter 3

Optimizing User Tracking in LAN-based MANs

3.1 Introduction

Since the primary function of a communication network is to provide communication facilities among users and between users and services in the network. one of the key problems such a network faces is the need to locate the whereabouts of various entities in it. This problem becomes noticeable especially in large networks with mobile users. and is handled by components such as forwarding (routing) tables residing at various nodes in the network. The location problem manifests itself to the fullest extent when the network consists of a large number of nodes distributed all over the network. and users are allowed to move from one network site to another frequently. In this case. it is necessary to have a set of distributed and dynamic mechanisms enabling the nodes to keep track of users in a continuous manner and reach them at all locations and points in time. Furthermore, real-time applications such as voice depend on packets being delivered at a constant rate and within a certain time budget. If at times of user movement, the network cannot ensure the timely delivery of packets, they become irrelevant and would need to be discarded. Diversity in packet routing may help reduce the packet latency, however it can lead to significant bandwidth congestion inside the network if employed when routing bandwidth intensive applications such as video and TCP. The purpose of this work is to design appropriate tracking and routing mechanisms to support users moving at any speed, and using applications with a wide range of characteristics.

Networks with mobile users have become widely used over the last decade. One example are the wireless networks such as the 802.11 local area networks (LANs). Wireless communication usually involves a base station (known as an access point or AP in the 802.11 LANs) and a wireless user. One base station may support between a few tens to a few thousands of users. In wireless networks, communication regions controlled by one base station are known as cells. The 802.11 standard supports the bridging of multiple wireless cells (known as a Basic Service Set or BSS) into an extended network (known as an Extended Service Set or ESS), via some network components which we refer to as switches. While the 802.11 standard gives precise and specific guidelines on the implementation of the physical and medium access layer of the BSS, a standard for the design of the ESS switch has not yet been defined. According to the 802.11 standard, the ESS switch may be created from many technologies including current IEEE 802 wired LANs. IEEE 802.11 does not constrain the ESS switch to be data link or network link based, nor does it constrain it to be centralized or distributed in nature. The switch may reside with the AP, or connect multiple APs together. By integrating the switch functionality with the

AP functionality however, one may reap the benefits of cost reduction and a simpler process for the deployment of the ESS.

Given the convenience of untethered communication, we believe that the future will bring the deployment of 802.11 networks over large areas, such as across cities, anywhere from small to large and educational and industrial campuses and plants. As 802.11 networks become larger, supporting many fast moving users and increased application demands, a scalable design of the ESS switch becomes paramount for ensuring that high quality of service is delivered to each individual user and their applications.

Two major tasks of this network are as follows: user tracking, enabling a user to move from one location to another and routing, enabling packets to reach their intended destination. Appropriately balancing the complexity of each of these tasks can be challenging, as illustrated by the following two scenarios.

In the first scenario, a "full information" strategy is used, which requires every switch in the network to maintain a complete routing table containing up-to-date information on the whereabouts of every user. This may make packet routing cheap but user tracking very expensive since it is necessary to update the routing tables of all the switches every time a user moves inside the network. However, this strategy may be appropriate if the number of users is small and users frequently converse with each other.

In the second scenario, a "no-information" strategy is used, where no updates are performed following a user move or at any time, thus abolishing altogether the concept of routing tables and making the track operation cheap. However, routing becomes very expensive as it requires either the flooding of all the packets over the entire network. or some global search for the destination of each packet over the entire network.

In between these extremes one may design a "partial-information" strategy that may perform well for a wide range of communication and movement characteristics. making the costs of both tracking and routing relatively cheap. The difficulties here arise in how to partition the network properly and assign responsibilities for tracking and routing at the entities in the network as to achieve optimal results.

At the heart of the problem are two important issues as follows: On one hand. packets may need to be routed at some switches in the absence of tracking information. This can lead to inefficiencies that greatly affect the network cost and performance. For example. imagine a large network supporting ten million users. of which 10% are actively engaged in a TCP applications consuming an average of 1 Mb s. If tracking information is available for only 9 out of 10 users and the packets are flooded when tracking information is not present. every switch in the network will see the combined load from 10% of the users in flooded traffic. which amounts to an average of 100 Gb s (on top of the bandwidth used for forwarding the other traffic). Handling such large amounts of traffic can be prohibitively expensive given the current link and switch technology. This suggests that tracking information may need to be distributed to as large a number of switches as possible.

On the other hand. current technologies significantly limit the number of users that can be tracked by a switch built at a reasonable cost. One way to implement the tracking databases is as a direct-lookup memory. This implementation scheme allocates entries for every address in the address space. Lookups are performed using the user address as an index into the database and usually take one clock cycle to complete. Unfortunately, because it allocates an entry to every possible address in the network (even if it is never used), this solution can require very high database capacity.

Another possibility is to design the tracking database as a content-addressable memory or CAM. CAMs allow single cycle lookups without the constraint of allocating an entry for every possible user that may exist in the network. This is done by decoupling the user address from the index into the database. Entries for users are placed at arbitrary locations. When a request arrives for a particular user, a search is performed to find the index for that user in the database. Parallel searches can take place at many indexes of the database if enough memory bandwidth is available, or if comparator logic is built at multiple indexes (directly in hardware). Trade-offs can be made between the length of the search and the cost of the implementation.

Recent work on designing CAMs using DRAM technology has shown that a single chip CAM can store up to 64,000 entries and can be accessed at clock frequencies of a few tens of MHz [24]. This can place significant scaling limits on the number of users supported inside the network. According to Moore's law, the size of the memory will quadruple in the next three years, making it feasible to store up to 256,000 user entries, which represents still a stringent constraint on the design of the tracking scheme. Furthermore, access speed will advance much more slowly, hence even if a lookup search takes only one cycle, this memory chip will allow only up to a few tens of million of packets to be routed every second. In turn, this places a stringent constraint on the routing scheme employed in the network.

To increase the processing capacity at the switches, a faster memory technology may be considered (such as SRAM), that can operate at Ghz speeds, hence forwarding up to 1 billion packets per second. However it in turn further limits the database size to up to a few tens of thousands of entries, making the design of the tracking scheme even more difficult.

A hierarchical memory design using both DRAM and SRAM technology could potentially allow the benefits of both a large DRAM storage and fast SRAM lookups to be exploited. Then, assuming a high cache hit ratio in the SRAM, hundreds of thousands of users can be tracked by a single dual memory chip forwarding many hundreds of million of packets per second. Stringent constraints are still placed on the design of the tracking and routing algorithms.

Evidently, at an increase cost in the overall solution, more than one chip may have to be used for tracking mobile users at the switches of a large network. Nonetheless, to minimize cost, it remains crucial that the tracking and routing schemes operate efficiently to reduce the number of tracking entries and the bandwidth consumption at the switches in the network.

As we will see in the following section, there is a wealth of literature that deals with the problem of tracking and routing in networks for mobile users. Unlike the previous solutions before us, we propose a comprehensive set of schemes for user tracking and routing that take into account a wide range of mobility speeds, different application characteristics and the state of technology. Each solution is designed with specific application and mobility characteristics in mind, and made suitable to specific technologies based on their cost and performance capabilities.

3.2 Related Work

In LAN networks, switches distributedly forward data packets to users. For this purpose, switches learn about the location of users via a transparent learning scheme (also known as snooping), which works as follows: switches listen to traffic originated from users and record in their forwarding databases the pair of the address of that user and the port via which the traffic was received. A timer is associated with this pair which ensures that entries in the database do not become stale in face of changes in the network or user states. A packet addressed to a user that does not currently appear in the forwarding tables of a switch will be broadcast across all the ports of that switch (other than the port via which the packet was received). It is then assumed that this destination user will reply to the broadcast and that future packets addressed to it from the same source will not be broadcast.

Correct operation of the forwarding scheme just described requires that the network have only one path between any pair of switches (the network does not have any cycles). This restriction is a major reason large switched LANs have not been widely deployed. In order to build large capacity or highly reliable networks, one must allow switches to function in arbitrarily connected graphs. A solution to this problem has been proposed and is currently under IEEE standardization [16]. This solution allows the network to be decomposed into one or more spanning trees, and that each spanning tree meets the topological constraints for the basic switch forwarding scheme. Then, packets traveling on a network with cycles can be forwarded correctly by first decomposing the network into a set of numbered spanning trees, and then marking each packet as traveling on a single spanning tree. Furthermore, the forwarding tables at each switch are expanded to add a tree number to each entry.

Unfortunately snooping works on the assumption that information gathered in the past can be used for some period of time in the future. However, when users in the network are mobile, this time period can become very small, hence the timer (lifetime) value of snooped information has to be set conservatively to a low value. For the snooping scheme to remain effective, data packets need to be sent at a high rate by the mobile users in order to update the tracking databases at the switches along the paths to their communicating partners. Given the characteristics of wireless devices, this may not be possible. Consider for example that the user is a portable device with a small battery life. In this case, the user application is likely to limit the length and duration of transmissions by the user device to conserve battery power. Since transmitting packets is significantly more expensive than receiving packets, it is likely that the user applications will be tailored to favor receiving packets over transmitting them. Consequently, snooped entries for such users may not get refreshed as frequently as desired.

A different method for forwarding in switched LANs, which is used only for multicast applications, is to explicitly register users with the appropriate switches via a registration scheme designed especially for that purpose, known as the Generic Attribute Registration scheme or GARP [15]. Users spread across the switched LAN can join a multicast group and can receive packets that are sent to that group. A message sent to a multicast group does not embed the address of the destination users. Instead, a multicast address is used to represent a group of member users and switches assume the burden of mapping that address to all the group members and delivering the group packets to them. Switches learn about the existence of the members via explicit JOIN messages sent by the members. Like in the transparent learning scheme. timers are associated with each forwarding entry to ensure that the databases do not become stale in the event that member users are no longer alive or the network configuration has changed. Unlike in the transparent learning scheme however. tracking here happens pro-actively. even in the absence of any data communication. Also. all users are tracked at all the switches in the active topology of a LAN. including switches that do not belong to the path that connects users with their communicating counter-parties. These features are necessitated by the nature of the application which supports the open host group concept [15]. One drawback of this design is the large size of the tracking databases when many populous multicast groups are simultaneously active.

In [33] Sincoskie at al discuss the issue of supporting mobile users in large switched LANs. They argue that a primary challenge is reducing the excessive bandwidth consumption caused by broadcasts. To address this challenge. they propose the following scheme:

1. The network is converted into a rooted graph

2. The edges are partitioned into two sets: downward edges and upward edges

3a. Packets arriving from an upward edge are routed in the usual way

3b. Packet arriving from a downward edge are transmitted on the other downward edges; after some delay. if the packet is perceived as not having reached its destination. it is transmitted on the upward edge as well. A packet is perceived at a switch as not having reached its destination if no packet arrives to that switch that was sent by that destination.

In essence. this scheme performs a search for a given destination inside the network. The benefit of this scheme is that bandwidth is saved when the packet does not need to be forwarded on the upward edge. The disadvantage of this scheme is that the other packets experience an increase in latency. Furthermore, this optimization depends strongly on the network topology and communication patterns among users inside that network. Evidently, this technique works best when there is strong locality in communication between the switches of the network. Let us assume that this large network is deployed to support connectivity across a city. While locality may be present, say in the context of phone communication in a business district (e.g. high frequency of calls among the employees of the same firm), it may not be true in a residential district where Internet surfing may be the predominant application.

3.3 Schemes

A scheme to support mobile users in a given network must include mechanisms for tracking the users and for routing packets to them inside that network. As we have learned from the previous section, a number of techniques already exist for tracking and routing to users in various networks. For example, two techniques for tracking are transparent learning and explicit registrations, and two techniques for routing are flooding and searching. The question arises concerning the selection of the optimal scheme to support mobile users traveling at various speeds and using applications with a variety of characteristics. Clearly, the answer to this question depends not only on the over-arching characteristics of the techniques employed, but also on how they are combined, implemented and tuned for a given set of user application and mobility characteristics. In this chapter, we propose to answer this question. Various schemes are investigated and optimized to support mobile users in a large, distributed network with many switches.

In the rest of this section, we describe each scheme in detail. In Section 3.4 we describe the simulation environment we use to compare the performance of the schemes for different mobility and application characteristics. In Section 3.5 we present our results and in Section 3.6 we conclude the Chapter.

3.3.1 FLOOD

This solution is similar to the "no-information" strategy for tracking described in Section 1. However, in the FLOOD scheme, a user is tracked at exactly one switch in the network, known as the "local" switch[1]of that user. The reason for this is to prevent packets from being broadcast unnecessarily on a wireless link via which users and their local switch may be connected. By knowing who these users are, a switch can avoid sending packets across the wireless link when the destination of those packets is not among those users. This is important for conserving bandwidth across the wireless link, which can be scarce.

Routing can be performed in one of two ways: via flooding as currently done in LANs or via searching, where selective broadcasting is employed to limit the number of packets in the network at one given time. An implementation of the search scheme is discussed in [33] which we summarized in Section 2. The benefit of flooding over searching is that a given destination may be reached more quickly, which can reduce packet latency. For these schemes to work, a spanning tree must be used to avoid infinite loops as described in Section 2. In our implementation, we choose to use the flooding mechanism because it is more widely known and used.

In summary, the advantage of the FLOOD scheme is that the size of the tracking

[1] For example, if the switch is built as part of the AP in a 802.11 ESS, the local switch is the AP with which the user is associated at a given time.

databases at the switches can be very small, provided that users are not concentrated heavily at a few locations in the network. However, the problem with this solution is that each switch may see the aggregated load from all the users in the network, which can pose scalability problems to the network.

3.3.2 SNOOP

This solution embodies the forwarding scheme used in LAN switches, which we described in Section 2. The idea here is that switches can learn about the location of users by transparently listening to the data traffic emitted by the users. By doing so, switches can unicast packets to users whose location they have learned. This reduces the need for data broadcasts, which in turn can reduce the bandwidth consumption in the network. This strategy works the best when users send data traffic frequently, allowing switches to learn the location of users on a more frequent basis. Evidently, the down-side of this scheme is that it results in larger tracking databases at the switches than in the case of the FLOOD scheme.

In order to avoid maintaining stale entries at the switches in face of user movement or power down, timers are associated with the entries created via the learning algorithm. The timers are set to some maximum lifetime value and decremented in time. When the timers expire, the entries are no longer valid. For the SNOOP scheme to work, the maximum lifetime value has to be properly selected. Here the user mobility rate is of crucial importance, for the following reason: The higher the mobility rate, the higher the chance that a user moves while an entry for that user is still valid at a switch,with information for that user that is no longer correct. If the user does not happen to be sending data packets that can reset that entry, a packet sent to this user may be routed using incorrect information at the switch.

We can see that setting the maximum lifetime value too high relative to the user mobility rate (and the application usage) can result in stale entries at the switches, which causes packet mis-routing or even loss. However, a high maximum lifetime value can result in more snoop entries being present at the switches in the network at any given time. This allows more packets to be unicast instead of broadcast throughout the network, which may save bandwidth consumption. The key question becomes at what values should the timers for the snooped entries be set so to achieve minimal packet loss and still avoid the need for packet broadcasting as much as possible? We identify this value by running numerical simulations for various user mobility and application characteristics.

3.3.3 SNOOP_PU

General Description

Unfortunately, snooping may not work well when users move at fast speeds, frequently changing their points of attachment to the network. The reason for this is that the timers associated with the snooping entries need to be set to very low lifetime values, which causes them to time-out more frequently, possibly resulting in more data broadcasts. This becomes an issue when users do not send enough data traffic to allow the snoop entries to be reset more frequently so as to accommodate the smaller lifetime values used.

To fix this problem, we use explicit control messages to update the snoop entries at some switches in the network. The questions that arise concern which switches are to be updated and under what circumstances. A user can send refresh messages to some or all the communicating hosts that are actively sending data to the user at the time the refresh message is to be sent. The switches along the paths between the user and the communicating hosts to which the refresh messages are sent can be updated in this fashion. However, sending refresh messages to all the communicating hosts may not be efficient if many such hosts exist. To deal with this case, a user may choose to send a single refresh message to be flooded across some spanning tree of the network. Another possibility is that the refresh message be sent to some rendez-vous point in the network (like the home agent in Mobile IP) which other hosts must then contact prior to establishing a session with the mobile user.

Refresh messages may be sent every time a user moves[2], or periodically at some frequency F. The first solution may result in fewer refresh messages than the latter solution, especially when F is large. Sending fewer refresh messages can save control bandwidth and processing overhead in the network. The latter solution may result in more up-to-date entries being maintained at the switches, especially when F is large. Evidently, selecting an appropriate value for F depends on the user mobility rate. When tuning F, one must also take into account the control bandwidth overhead and the savings in data bandwidth consumption from keeping more up-to-date entries at the switches.

The SNOOP_PU scheme may be designed to take into account the user state. For example, refresh messages may be sent when the user is in one state, but not in another. One possible categorization of user states is as follows: 1) The user is on and it is actively engaged in both the sending and receiving of packets from other users.

[2] Note that the refresh message may be sent in advance of the actual change in location to expedite the hand-off process. This is possible only if the user can communicate with both the old and new switch simultaneously (this may require for example that the wireless cells overlap).

2) The user is on and it is actively engaged in sending packets to other users but it is not receiving any packets. 3) The user is on and it is actively engaged in receiving packets from other users but it is not sending any packets. 4) The user is on but it is not active. 5) The user is off. Refresh messages may be used if a user is in state 3. and not used if the user is in any other state. While this feature can increase the complexity of various entities in the network. it may save control bandwidth overhead and reduce the size of the tracking databases at the switches inside the network.

Our Implementation

In our implementation. a user sends refresh messages when it changes location. to all the communicating hosts that are actively sending data to that user at the time of move. No differentiation in treatment is made based on the user state. Although this scheme is simple. it can still result in performance gains over the SNOOP scheme when users do not send data to their communicating hosts frequently (relative to the user mobility rate). The SNOOP_PU scheme is identical to the SNOOP scheme in all other respects. Like for the SNOOP scheme. we identify the optimal lifetime value by running numerical simulations for various user mobility and application characteristics.

3.3.4 REG

General Description

It is interesting to ask whether a registration-based scheme like GARP can be used to track mobile users in a given network. Unlike in GARP however. registering all the users with all the switches in the active topology may not be desirable. since it can result in large tracking databases at the switches and registrations that take a long time to complete. Instead, only a subset of the switches may be used to track a given user, which we refer to as the write set of that user. Since tracking of a user U takes place only at the switches in the write set of U, a mechanism for routing must be designed to specify how routing is to take place at switches outside this write set. As before, we choose to use packet flooding across the links of some tree spanning the switches of the network.

For REG to work properly, the write set of users must be selected carefully. Clearly, a write set that is small may result in a large number of data broadcasts. A write set that is too large may result in tracking database sizes that are too large. An optimal size of the write set has to be identified that balances these two factors. The composition of the write set can also affect the results. The write set may have a static or a dynamic composition. A static write set for a given user can be some VLAN to which the user permanently belongs to or some home database to which that user is assigned (like a home agent in Mobile IP). A dynamic set for a given user can be some group of switches between a user and its current communicating hosts, some group of switches in the neighborhood of the user or even some databases remote from the user that change dynamically with the location of the user (such as the foreign agents in Mobile IP). Evidently, a static write set may be too far away from the user, making location updates too slow.

Like in the SNOOP_PU protocol, a question arises concerning when the registration messages are to be sent. One interesting area of investigation concerns the manner in which entries are invalidated at the switches when they are no longer needed. In GARP, explicit deregistration messages, known as LEAVEs are used for this purpose, in conjunction with the time-out technique used in snooping. If timers are to be used in REG, at what values should they be set? If deregistrations are used, when are they to be sent?

When both registrations and deregistrations are used, an interesting area of investigation concerns the order and manner in which they are to be sent. Assume that as a user moves, the composition of its write set changes from W1 to W2. When a user moves, it can choose to either send a deregistration message to the W1 ahead of sending a registration message to W2 or conversely, register with W2 ahead of sending a deregistration message to W1. In the latter choice, more packets may be mis-routed while in the first case more packets may need to be broadcast. Now, when a switch S belongs to both W1 and W2, should S see any registration or deregistration messages? To minimize the control overhead in the network, one may choose not to send any control message to S. However, in a completely distributed network, this solution may not be possible to implement. Instead, one may allow S to see both a deregistration and a registration message. In that case, a subtle issue needs to be resolved concerning the race conditions that may occur at S. If the deregistration and registration operations are allowed to take place asynchronously at W1 and W2, S may end up with an invalid entry when these operations are completed. The reason for this is that the registration message may arrive at S ahead of the deregistration message. To avoid race conditions, acknowledgment messages may be used to signal the completion of the deregistration at the switches in W1, upon the receipt of which a registration message can be sent to W2.[3] Of course, the use of acknowledgment messages can increase the time it takes to complete a location update.

[3] Sequence numbers may also be used for ordering the registration and deregistration messages at S.

Evidently, a registration-based scheme requires additional functional components at the entities in the network such as the switches or the mobile users for the purpose of tracking. This can increase the complexity and cost of the entities in the network. The burden for this scheme must be shared between the switches and mobile users in the network according to their respective capabilities. Usually however, most of the functionality would be placed at the switches in the network. This allows the mobile devices to be simple, low power, low cost, portable and with a long battery life.

Our Implementation

In our implementation, we consider the write set of a user to be the subset of switches in the neighborhood of the user that are located at a distance d from the user, such that d is smaller than some radius R. By choosing a write set in the neighborhood of the user, we avoid sending the control messages across the network, which can be expensive. When a user turns on or when it arrives at some switch S, it registers with S and all the switches in the network that are separated from S by no more than R hops. When a user moves from some switch S1 to another switch S2, it first deregisters itself from the write set at S1. An acknowledgment for the deregistration is sent in the opposite direction of the deregistration message. Upon the receipt of the deregistration message, the user registers itself with the write set at S2. To simplify the implementation, we opt not to use timers to invalidate entries at the tracking databases.

To maximize the number of switches that receive a given registration, the control messages are sent across all the possible links in the network. To achieve this, multiple spanning trees may be defined, rooted at every individual switch in the network. If this is too expensive to implement, one can propagate the control messages across all the links without defining a spanning tree for distribution. To avoid broadcast storms, the control messages need to be filtered at switches where the message did not result in a database change[4][5].

Evidently, the explicit registration and deregistration messages add control overhead in the network which increases with R and with the user speed. The number of data broadcasts decreases with R while the size of the tracking databases and the time to complete a location update increases with R. To identify the optimal value of R that balances all these factors, we simulate REG for various application and mobility characteristics.

3.3.5 REG_SNOOP

Since they are not mutually exclusive, it becomes interesting to ask whether the REG and SNOOP schemes can be used in combination to create an optimal scheme for tracking and routing to mobile users. We notice that REG can result in a large number of data broadcasts when the size of the write set is small. By using snooping outside the write set of a user, we can limit the number of data broadcasts that take place outside the write set of a user when routing packets to that user. Furthermore, if snooping is done far away from the user, then the timers used in snooping can be set to relatively high values that account only for topological network changes or changes in user state other than location. Hence, this scheme can reduce the number of data broadcasts over both the REG and SNOOP schemes. Of course, the down-side is

[4] This is the same technique as used in OSPF to limit broadcast storms.

[5] Notice that when this technique is used to update the write set of a user, links other than those in the spanning tree of the network will be enabled for routing data packets. This is a departure from the routing mechanism used in GARP.

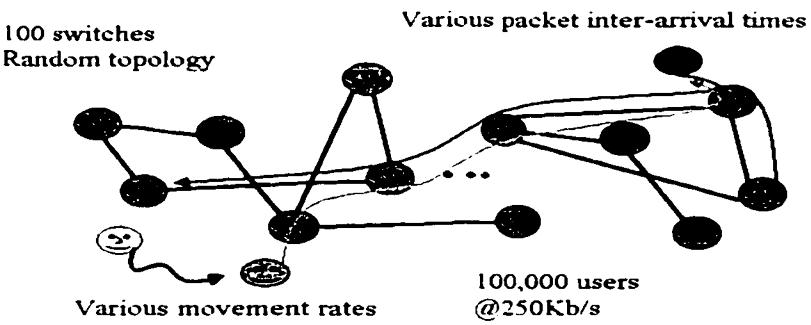


Figure 3.1: The simulation model.

that, by tracking users both inside the write set and at switches outside the write set but along the path to some communicating host, we may increase the average number of entries in the tracking databases at the switches in the network. Furthermore, by using snooping, we may increase the packet loss in the network over the REG scheme if the lifetime values are higher than some threshold S. Evidently, S will depend on the write set, and the mobility and application characteristics of the user.

In our implementation, like in REG, a user is tracked via an explicit registration/deregistration scheme in some spherical write set around the location of the user of some radius R. Outside this write set, a user is tracked at the switches along the paths between the user and its communicating partners, using the snooping scheme with timers set at some maximum value S. We identify the optimal S and R via simulation, for various application and mobility characteristics.

3.4 Experiments

As in an earlier chapter, our simulation environment is based on the Scalable Simulation Framework (SSF). The simulation model is depicted in Figure 3.1.

For generality purposes, we use a random network topology, as generated by the

Waxman scheme [39]. This is an established scheme for generating random network topologies. The scheme works by randomly placing nodes at locations in a plane, then connecting nodes with edges according to some probability function known as the Waxman probability function. We use a software package freely available on the web [13], which allows the user of that software to enforce total connectivity among all the nodes and to vary the size of the network generated.

Using SSF abstractions, we implement a switch model that performs the tracking and routing schemes described earlier. A switch has 10 input output ports, each with storage queues that can hold up to 8000 packets. The packet service time at each queue is fixed at 10 microseconds. Registrations and deregistrations are assumed to consume 1% of the bandwidth consumed by a data packet. This is a reasonable assumption, since a control packet may occupy only about 10 bytes while an average data packet size is frequently about 1000 bytes. The computation of the minimum spanning tree of the network is performed using Kruskal's scheme. The code is provided by researchers at the Washington University, and is freely available on the Internet [11].

Along with ports, a switch is configured to have one forwarding table that consist of entries with the following format: <user address, output port for that address, lifetime value>. Periodically (every fraction of a second) and when snooping is used, the lifetime value of every address is decremented by that time period. Once the lifetime becomes 0, the entry become invalid. When a packet arrives at the input port of a switch, it is placed in the queue for that port, provided there is space left. The forwarding tables are consulted for the address corresponding to the destination of that packet. If an entry is found and valid, the packet is placed in the output port specified in the table. In this case, the packet is either consumed locally or sent to some remote switch. Otherwise, the packet is flooded across all the ports that belong to the spanning tree (except for the incoming port). If snooping is enabled, the switch that receives a packet learns that the source node is reachable via the port where that packet arrived and records this information in its forwarding table while also setting the lifetime to the maximum lifetime value. When the registration scheme is enabled, forwarding tables are updated upon the receipt of a registration or deregistration message. The switches propagate these messages according to the various schemes presented in Section 3.

Using SSF abstraction, we also implement a mobile user model, with various mobility and application usage characteristics. At the beginning of the simulation, users are uniformly distributed across the network. We allocate 8 users per switch, for a total of 800 users across the entire network. Evidently, as users move around, the distribution of users across the network changes. We use a mobility model inspired by existing literature [1] that works as follows: A user is stationary at a certain node for some period of time that is randomly drawn from an exponential stochastic process. When this period elapses, the user moves to one of its network neighbors (movement to any neighbor is equally likely) where it again resides for some random period of time, given by the exponential stochastic process. In our simulations, we vary the rate of the exponential process, thus affecting the rate of user movement. We consider three such rates to be of interest: once every second (high), once every 10 seconds (medium) and once every 100 seconds (slow). To understand the choice of these rates, consider a geographical area populated with 802.11 wireless cells which form an 802 extended LAN. The high mobility rate corresponds to the rate at which square cells of size 30mX30m would be crossed by a car moving at about 110 km hr. The medium mobility rate corresponds to the rate of crossing 300mX300m cells or via pedestrians moving at about 11 km hr. Finally, the last rate corresponds for example, to pedestrians crossing 300mX300m cells.

Unicast communication takes places between users in the network located at various locations. User applications are modeled also using stochastic processes. We consider the following abstractions: packets and packets spurts. Packets are communication entities that consist of a header and data payload. The header encloses the address of the source and destination of that packet, a time stamp representing the origination time of the packet and possibly a radius field used by the registration scheme. Spurts represent a sequence of packets exchanged between the same two users in the network. The time between two spurts and two packets of a given session are selected from two exponential random processes with different means. For example, the mean inter-arrival time between two spurt is set to 10 seconds while the mean inter-arrival time between two packets is set to 1 second. spurts can be uniplex or duplex, according to some probability. Duplex spurt consist of packets sent in both directions between the two nodes involved (this may model a TCP application). Uniplex spurts consist of packets sent only in one direction (this may model a UDP application). When a duplex spurt is ongoing and a packet is received at a mobile user, it in turn composes a packet and sends it to the source node some time in the future.

To measure network performance, we use the following metrics: packet amplification, bandwidth consumption, packet loss, and the average size of the databases used for routing at the switches[6]. The packet amplification is defined as the ratio of

[6] The packet latency is also measured, however it is very small for all the schemes studied, hence

the total number of packet transmissions aggregated over all the switches in the network to the total number of packets emitted by the mobile users inside the network. We also include the control overhead, weighted by 0.01 (as explained above) in the packet amplification results. The bandwidth consumption is the average number of bytes switched in one second at a switch in the network. The packet loss is defined as the percentage of packets that do not arrive at their intended destinations relative to the total number of packet emitted. The average database size is the average of the snapshot of the database size at all the switches in the network, taken every 100 ms. By using these metrics and simple scaling techniques, we analyze a network consisting of 100,000 users, each generating 250 Kb s in data traffic.

Three parameters greatly affect the performance of a scheme, namely: the user mobility rate, packet inter-arrival times, switch and link speeds and network size. In this paper, we perform various sensitivity studies of network performance as it relates to user mobility and the packet inter-arrival times. Given the specifics of our scheme, other variables also affect performance namely: the radius of the destination sphere used for registration and the lifetime value used for snooping. These parameters are also varied in our simulations. We run a few thousand simulations, each simulating a total of 8000 seconds of user behavior. A simulation run takes between a few minutes to a few hours to complete.

it is not considered in this chapter.

3.5 Results

3.5.1 Packet Amplification

In Figure 3.2 we plot the packet amplification for the SNOOP and SNOOP_PU schemes. for different maximum lifetime values between 0.1 seconds and 10 seconds. Results are shown for the mobility rate of 1 move sec and for the mean application rates of 1 packet every 10 milliseconds. 1 packet every 100 milliseconds and 1 packet every 1 second. Results for the other mobility rates (1 move every 10 seconds and 1 move every 100 seconds) are not shown due to lack of space. Experiments for different mean packet spurts were performed (between 1 spurt every 100 seconds to 1 spurt every 1 second). however the variation due to this parameter is small compared to other variations. hence it is not shown in the figures. In Figure 3.3 we plot the packet amplification for the REG scheme. for different radii between 0 and 8. Results are shown for all mobility and application rates. Finally. in Figure 3.4 we plot the packet amplification for the REG_SNOOP scheme. for different radii between 1 and 4. and for different maximum lifetime values between 0.1 seconds and 10 seconds. The results shows are for a mean mobility rate of 1 move per seconds and for all mean application rates. The results for the other two mobility rates are not presented due to lack of space.

As expected. the packet amplification decreases as the maximum lifetime value increases for the SNOOP and SNOOP_PU schemes. For these scheme. the packet amplification is considerably lower for the higher application rates. As expected. the SNOOP_PU results in lower packet amplification than the SNOOP scheme. especially for the lower application rate. Note however that. if the maximum lifetime value is

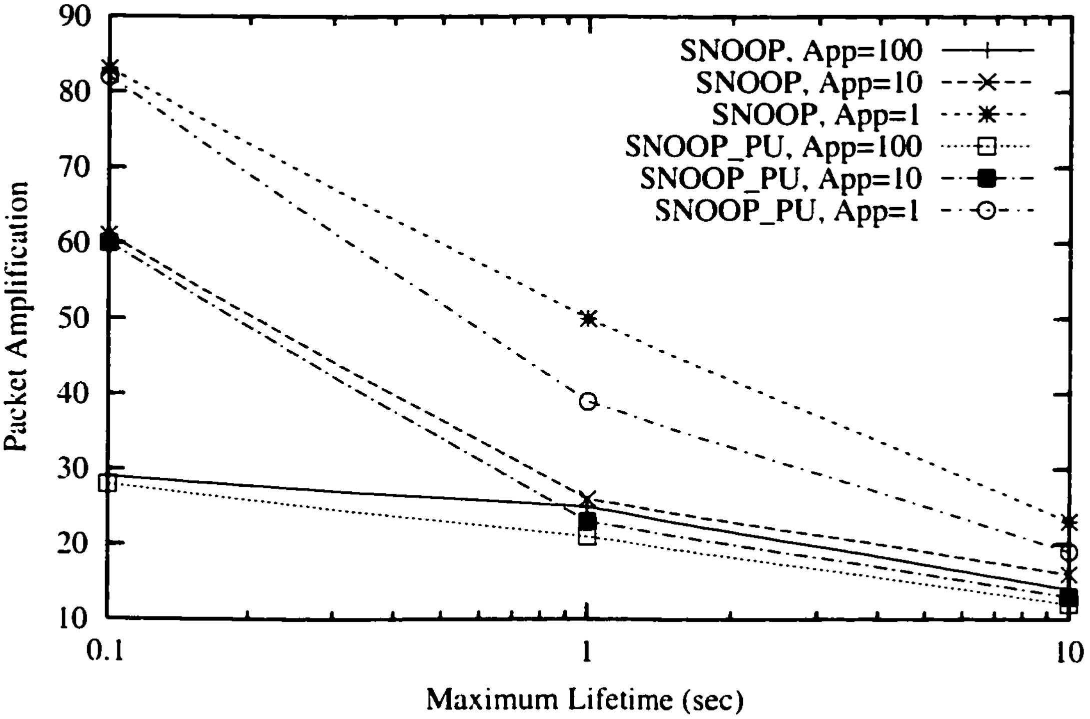


Figure 3.2: Packet amplification for different maximum lifetime values, for the SNOOP and SNOOP_PU schemes. Results are shown for the mobility rate of 1 move sec, for a mean application rate of 100 packets second, 10 packets second and 1 packet second.

very large (10 seconds) or very small (0.1 seconds), then the reduction in packet amplification from the SNOOP_PU scheme is lower since, in the first case, entries time out too infrequently compared to the user mobility rate, and in the later case, the maximum lifetime value is too small to have a drastic impact on the packet amplification.

As expected, the packet amplification drops significantly with the increase in the radius for the REG scheme. Evidently, if the radius is 0, then the REG scheme is equivalent to the FLOOD scheme. For radii equal to 6 hops and greater, the packet amplification is at a minimum suggesting that, for this topology, most of the switches

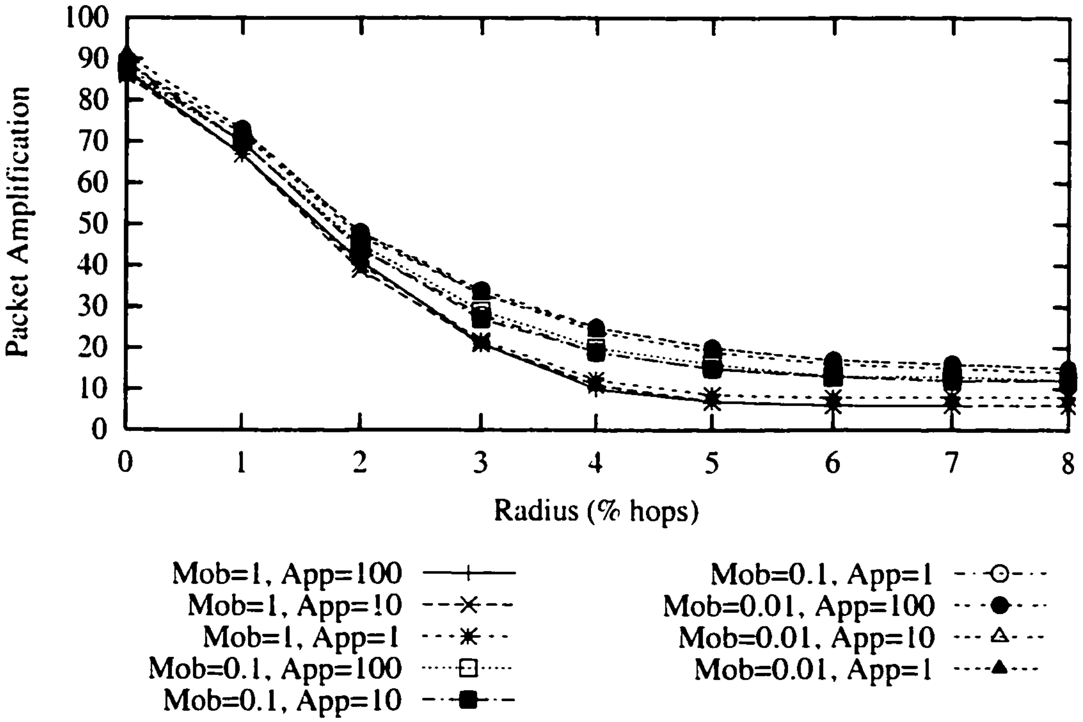


Figure 3.3: Packet amplification, for different radius sizes for the REG scheme. Results are shown for mean mobility rates of 1 move/second, 0.1 move/second and 0.01 moves/second, and for mean application rates of 100 packets/second, 10 packets/second and 1 packet/second.

are at most 6 hops apart. As expected, the control overhead increases with the radius, mobility rate and decreases with the application rate, however it is generally insignificant as a fraction of the total traffic exchanged (less than 1%). The largest contribution occurs when the mobility rate is 1 move per second, the application rate is 1 packet every second and the radius is greater than 6. In this scenario, the control overhead amount to 25% of the total traffic, which increases the packet amplification from 6 to 8.

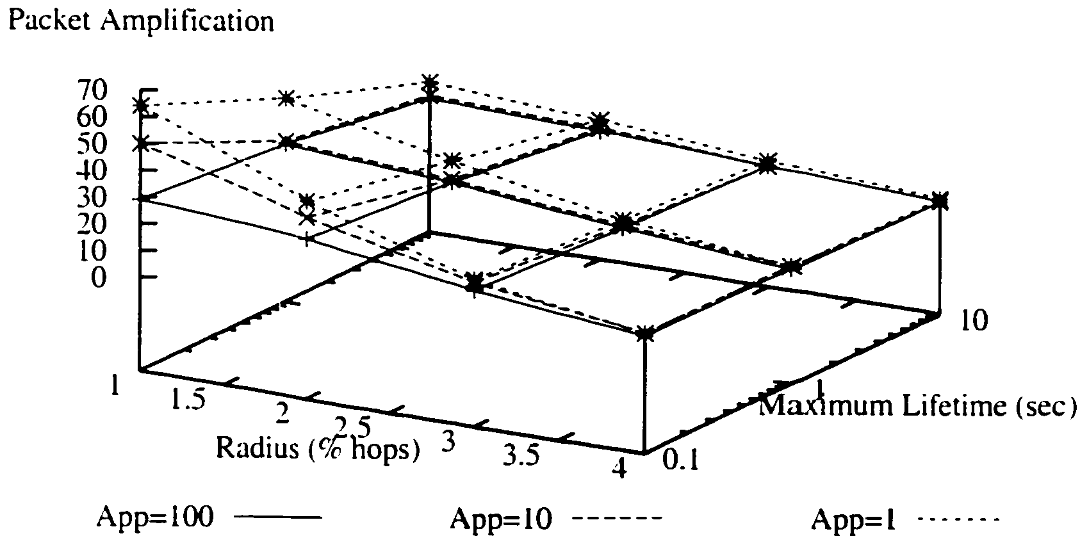


Figure 3.4: Packet amplification, for different radius sizes and different maximum lifetime values for the REG_SNOOP scheme. Results are shown for a mean mobility rate of 1 move/sec, for mean application rates of 100 packet/second, 10 packets/second and 1 packet/second.

3.5.2 Packet Loss

We plot the packet loss for the SNOOP, SNOOP_PU schemes and the REG_SNOOP scheme in Figure 3.5 and Figure 3.6 respectively[7].

As the maximum lifetime value increases, more packet loss is incurred by the SNOOP and SNOOP_PU schemes, as expected. The SNOOP_PU scheme results in drastically lower packet loss than the SNOOP scheme, but this packet loss is not 0%. This is due to the fact that the updates are sent only along the paths corresponding to outstanding sessions, hence any new session does not benefit from the update and may still get mis-routed at the switches. As expected, the packet loss is much lower for the lower mobility rates.

The packet loss for the REG scheme is insignificant (very close to 0%) for all radii values. The reason for this is the fact that deregistrations happen very quickly compared to the rate of packet arrival, hence the window of time when stale entries are maintained at the switches is very small. As the radius increases, there is a slight increase in packet loss, due to the fact that more remote databases hold entries for a given user (and hence invalidates take longer to complete).

The REG_SNOOP scheme results in lower overall packet loss than the SNOOP and SNOOP_PU schemes, but higher packet loss than the REG scheme, especially for a maximum lifetime value of 10 and a radius of 1. This is because the behavior for the REG_SNOOP scheme approaches that of the SNOOP scheme for this scenario.

[7] The results for the REG scheme are not shown due to lack of space.

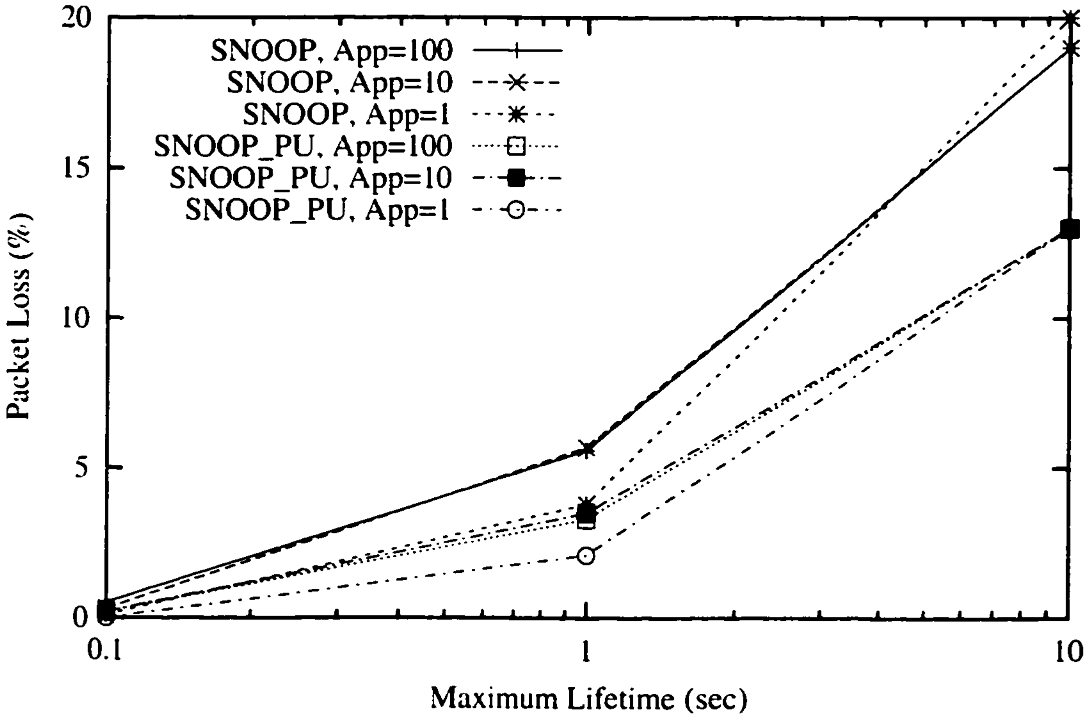


Figure 3.5: Packet loss as a fraction of the traffic emitted, for different maximum lifetime values, for the SNOOP and SNOOP_PU schemes. Results are shown for a mean mobility rate of 1 move/sec, for mean application rates of 100 packets/second, 10 packets/second and 1 packet/second.

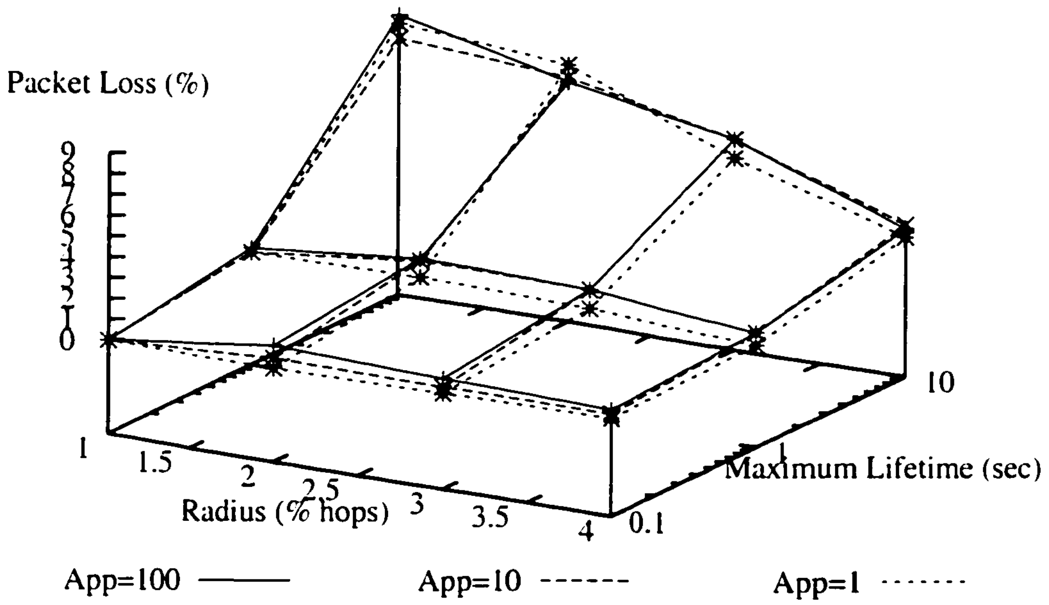


Figure 3.6: Packet loss as a fraction of traffic emitted, for different radius sizes and different maximum lifetime values for the REG_SNOOP scheme. Results are shown for the mean mobility rate of 1 move/second, and for mean application rates of 100 packets/second, 10 packets/second and 1 packet/second.

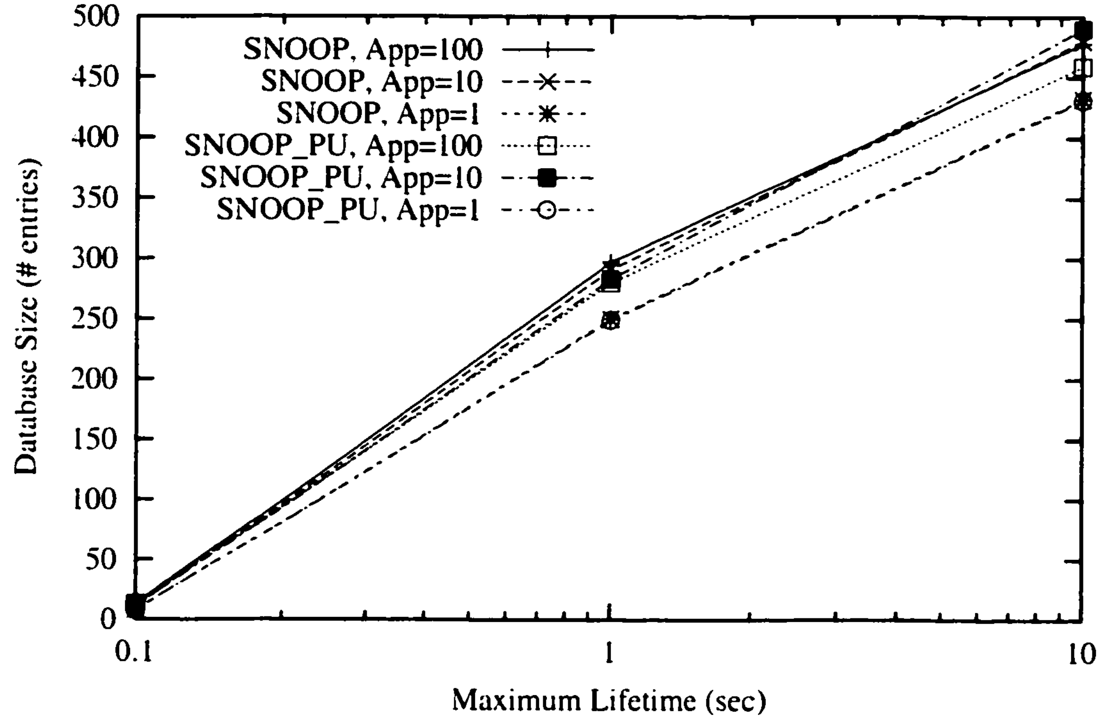


Figure 3.7: Average database size for different maximum lifetime values, for the SNOOP and SNOOP_PU schemes. Results are shown for a mean mobility rate of 1 move sec, for the mean application rates of 100 packets second, 10 packets second and 1 packet second.

3.5.3 Average Database Size

We plot the average database sizes for the SNOOP and SNOOP_PU schemes and for the REG scheme in Figure 3.7 and Figure 3.8 respectively[8].

As expected, the average database size increases with the maximum lifetime value for the SNOOP and SNOOP_PU schemes. In general, the average database size is slightly higher for the higher application rate, consequence of the fact that more snooping can take place in this scenario, leading to more entries being maintained at the switches and hence a higher average database size.

[8] The results for the REG_SNOOP scheme are not shown due to lack of space.

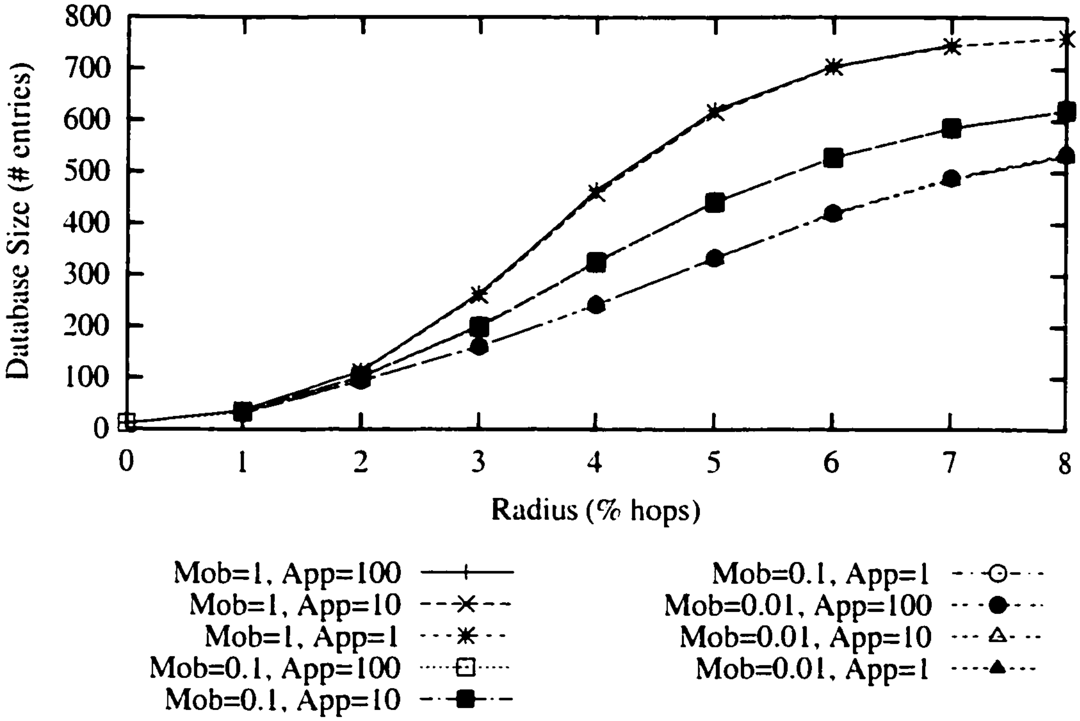


Figure 3.8: Average database size for different radius sizes for the REG scheme. Results are shown for the mean mobility rates of 1 move second, 0.1 move second and 0.01 move second, and for mean application rates of 100 packets second, 10 packets second and 1 packet second.

As expected, the average database size increases quickly with the radius of the write set for the REG scheme, and it approaches 800 (100% of the users in the network) as the radius increases beyond 6, for a mobility rate of 1 move per second. For the lower mobility rate, the average database size converges more slowly toward 100% because there are fewer opportunities to register with the switches in the network.

Generally, as the maximum lifetime increases beyond 0.1 sec, the REG_SNOOP scheme results in higher average databases than the REG scheme, for the radii studied (between 1 and 4). Evidently, this is caused by the additional tracking resulting from the use of the snooping mechanism. The increase is especially significant for the

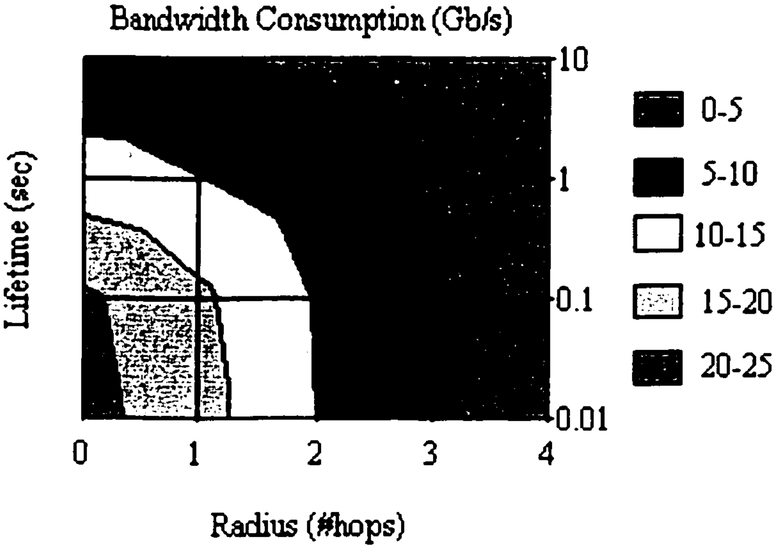


Figure 3.9: The average bandwidth consumption at every switch for different radii and lifetime values.

smallest radius sizes and the maximum lifetime value of 10 seconds.

3.5.4 Comparing the Schemes

No scheme outperforms the other schemes on all counts. In general, the SNOOP scheme results in the highest packet loss rate especially when large maximum lifetime values are used. The SNOOP_PU scheme helps reduce the packet loss, especially when the duration of the user session is large. The REG scheme results in high packet amplification when small radius sizes are used, and high average database sizes when large radii are used. The REG_SNOOP scheme reduces the packet amplification over the REG scheme for small radius sizes, however it leads to an increase in the packet loss.

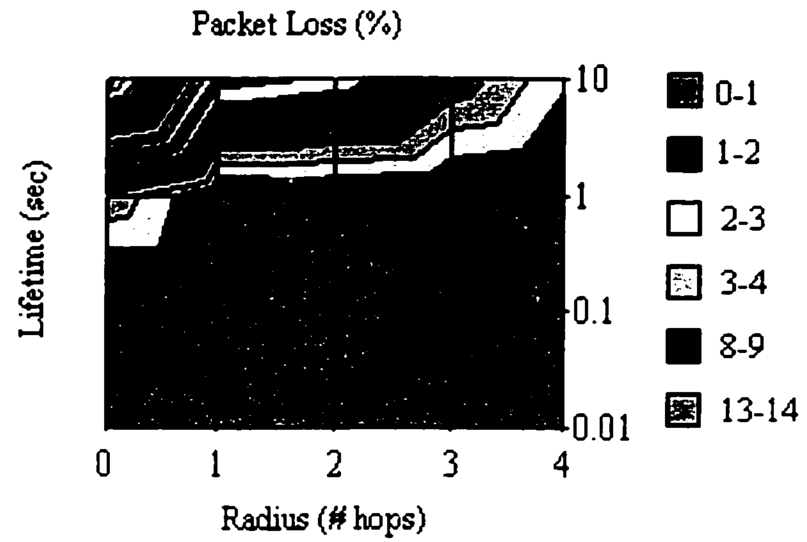


Figure 3.10: The average packet loss for different radii and lifetime values.

3.5.5 Network Tuning

In Figure 3.9 we plot the average per-switch bandwidth consumption at the switches for the SNOOP, REG and REG_SNOOP protocols. The mean mobility rate is 1 moves second and the mean packet inter-arrival time is 1 second. On the y-axis we show the lifetime value and on the x-axis the radius of the registration set. The y and x axis roughly correspond to using the SNOOP and REG protocols respectively. As expected, the largest amount of bandwidth consumption is present at the origin, where every switch sees every packet that a user sends (every packet is basically flooded). As the lifetime value increases, up to five times less traffic is averaged at the switches. No points along the y-axis intersect the region of minimum bandwidth consumption (between 0-5 Gb/s), hence the SNOOP protocol does not attain minimal bandwidth consumption. To achieve this minimum, the REG protocol must be employed either in isolation, or in combination with the SNOOP protocol.

In Figure 3.10 we plot the average packet loss in the network for the SNOOP,

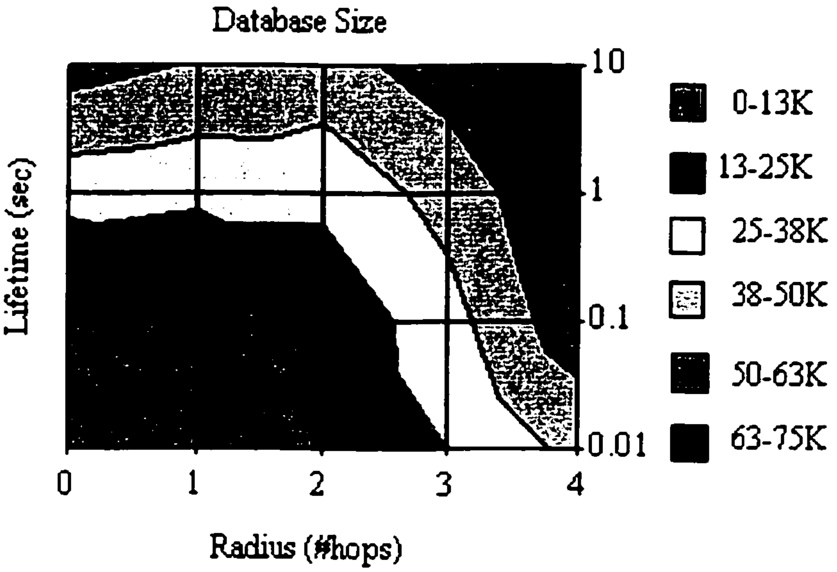


Figure 3.11: The average database size at every switch for different radii and lifetime values.

REG and REG_SNOOP protocols. As before, on the y-axis we show the lifetime value and on the x-axis the radius of the registration set. The y and x axis roughly correspond to using the SNOOP and REG protocols respectively. The mean mobility rate is 1 moves second and the mean packet inter-arrival time is 1 second.

We see that SNOOP can result in packet loss in the double digits for large lifetime values. By exploiting also the REG protocol, packet loss can be reduced for the same lifetime value.

In Figure 3.11 we plot the average per-switch database size at the switches in the network for the SNOOP, REG and REG_SNOOP protocols. We see that as the radius increases, the average database size increases many times. Similarly, as the lifetime value increases, the average database size increases as snooped entries expire more slowly at the switches.

We can see how tradeoffs can be made in the selection of the appropriate lifetime

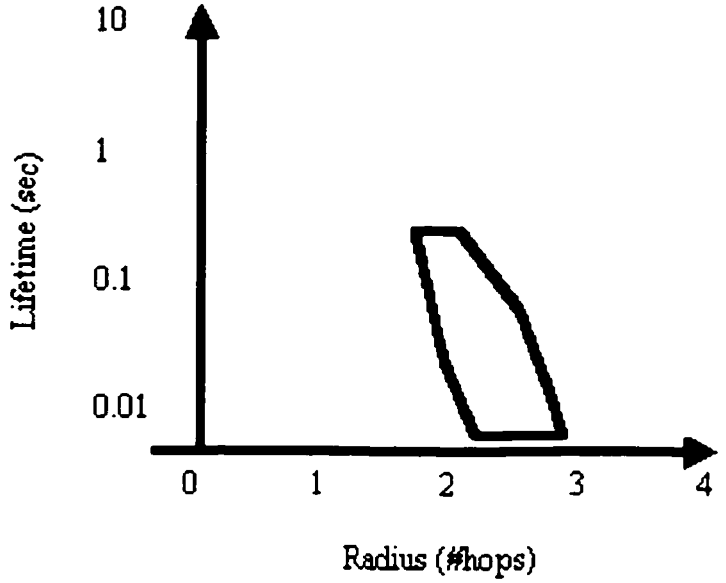


Figure 3.12: The set of datapoints satisfying the design constraints. The mobility rate is 1 move sec.

value and radius size for the operation of the network. A large lifetime value results in less bandwidth consumption but it can lead to increased packet loss and database sizes. A large radius size results in lower bandwidth consumption at the switches and lower packet loss, but it can lead to larger databases. The optimal tuning of a network must take these tradeoffs into consideration. This is discussed at great length in the next section.

In Figure 3.12 we apply a number of constraints on the design of the network, and highlight the set of radius and lifetime values that satisfy these constraints. The constraints on the network design are as follows: the average database size at a switch is limited to 25,000 entries, which is realistic given the state of technology. The average per-switch bandwidth available for switching is limited to 10 Gb/s, also a realistic constraint, and the average packet loss is target at 1%, which is consistent with the needs of most applications. Given these constraints, the set of lifetime and radius

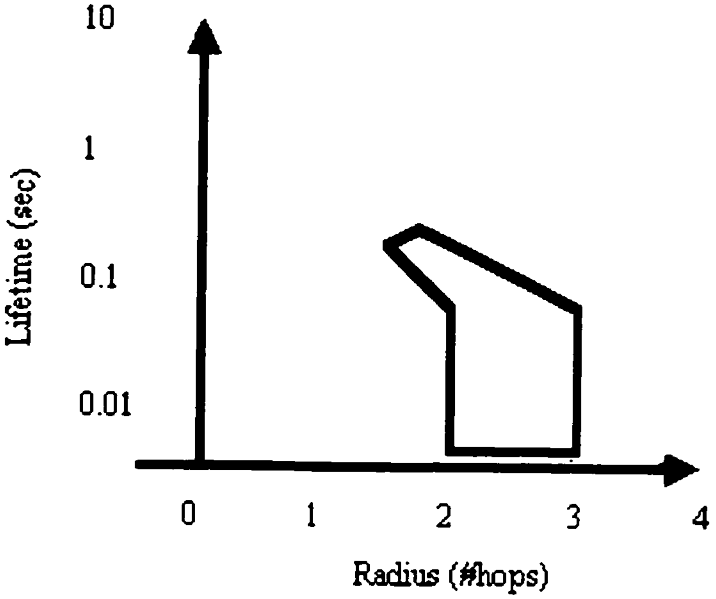


Figure 3.13: The set of datapoints satisfying the design constraints.

values that meet the requirements is reduced to the datapoints inside the unfilled polygon depicted in the graph.

We observe that the solution does not intersect the y-axis, thus the SNOOP protocol can not be used in isolation without violating the design constraints. The REG or REG_SNOOP protocols are needed instead.

Let us look at the impact of the mobility characteristics on the results. In Figure 3.13 we plot the results obtained for an average mobility rate of 0.1 moves/second and an average packet inter-arrival time of 10 ms, shown as before with an unfilled polygon. We notice that the region contains more datapoints for a radius equal to 3 but fewer datapoints for a radius equal to 2. The reason for the earlier observation is that the average database size decreases for the larger radius size, as less user movement is incurred and thus a more uniform user distribution is achieved. The reason for the latter is that the bandwidth consumption increases for the lower radius

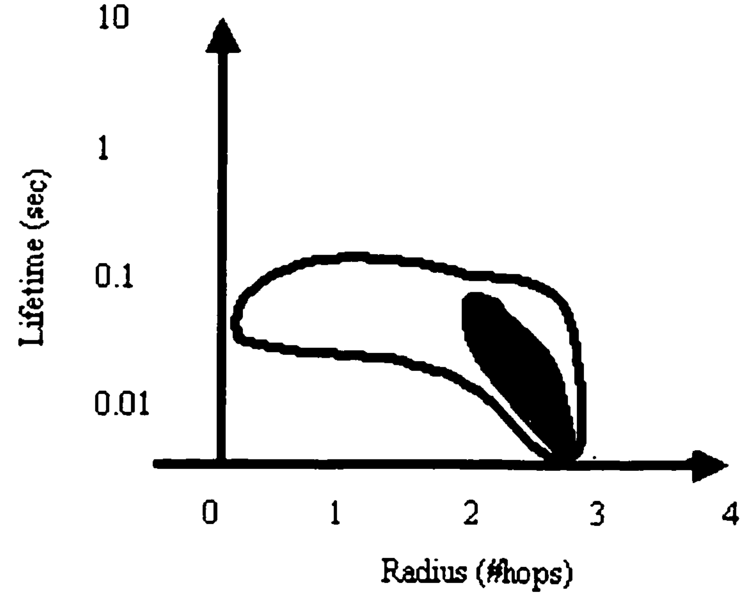


Figure 3.14: The set of datapoints satisfying the design constraints. The mobility rate is 1 move sec.

size, as the REG protocol becomes less efficient at the lower movement rates.

Let us look at the impact of the application characteristics on these results. In Figure 3.14 we plot the results obtained for an average mobility rate of 1 move sec and an average packet inter-arrival time of 10 ms, shown as before with an unfilled polygon. The first observation is that, in the case of this new application, the set of satisfying datapoints extends towards the y-axis, which means that the SNOOP protocol becomes a better choice for meeting our design goals. The reason for this is that, as the time spacing between packets is reduced, switches have the opportunity to listen on a more frequent basis, thus learning can take place in a more effective way. The second observation is that fewer datapoints along the x-axis satisfy the constraints. The reason for this is that, as the packet inter-arrival time decreases, more bandwidth consumption can occur at the switches when the lifetime value is close to zero. Thus, the bandwidth consumption can increase beyond acceptable limits.

The filled polygon in Figure 3.14 represents the intersection of the results achieved using the various mobility rates and packet inter-arrival times. We see that the set of satisfying datapoints decreases considerably when all application and mobility rates are to be considered. Thus, the tuning of the network becomes crucial as less room for error is given.

In summary, we have seen that the SNOOP protocol, currently used in switched LANs, cannot be used efficiently to track fast moving users. Instead, the joint use of the SNOOP and REG protocols is desired. The selection of the radius and lifetime values must be done carefully as to take into account the design constraints, as well as the application and user mobility characteristics.

3.6 Exploiting Mobilane to Optimize Macro-Mobility Support

In an earlier chapter we have seen how IP handoffs can be slow because they involve detecting that a new subnet is reached, acquiring a new address at that subnet and propagating this information to some entities in the network. To mitigate these problems we used anticipation techniques at the lower layers that permit address lookahead and packet n-casting to take place at the higher layers.

Unfortunately, in some networks anticipation cannot be implemented. For example, in wireless LANs, handoffs are hard and forward, which means that connectivity with the existing wireless cell must be discontinued prior to establishing connectivity at the new cell. For these networks, we need to investigate a different method to optimize support for macro-mobility.

3.6.1 Basic Idea

To this end, we investigate the idea that a mobile user maintain a fixed IP address while roaming inside a large region that spans many IP subnets from before. This is equivalent to building a large IP subnet within which swiches track users using either their IP or MAC address. Handoffs simply involve detecting that a new LAN segment is reached and updating tracking information at one or multiple switches in the subnet. Existing published measurements show that such handoffs can complete in a few to a few tens of milliseconds, hence much faster than IP handoffs.

As before, we call this extended IP subnetwork Mobilane, shown in Figure 3.15. By making Mobilane very large, the frequency of IP handoffs is significantly reduced, hence performance is statistically improved.

In some cases it is important that existing Internet infrastructure not be changed, particularly in areas where such infrastructure is already heavily deployed and used. For such cases, we propose that the subnet be deployed as an overlay network, superimposed on existing Internet infrastructure, for the sole purpose of supporting mobile users. This is shown in Figure 3.16.

By exploiting Mobilane, we propose two new schemes to support macro-mobility. The first scheme is MIP_Mobilane, where we use Mobile IP agents to support movement from one Mobilane to another, as depicted in Figure 3.17. Note that multiple agents may be deployed at one Mobilane for load-balancing purposes. The second scheme is E2E_Mobilane, which uses end-to-end techniques to handle movement be-

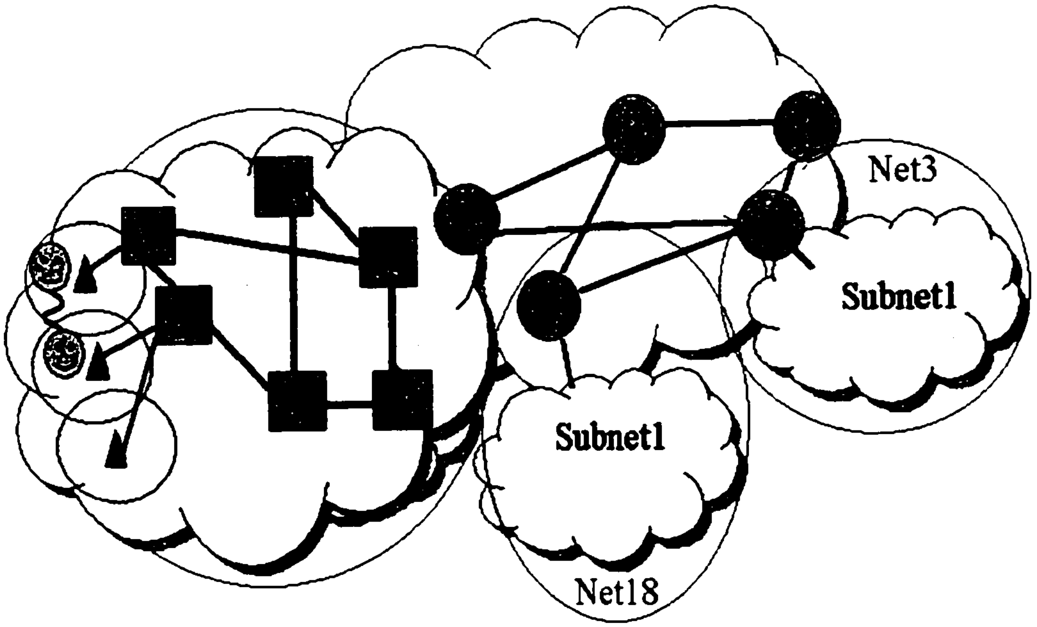


Figure 3.15: The Mobilane network extends over multiple subnets from before.

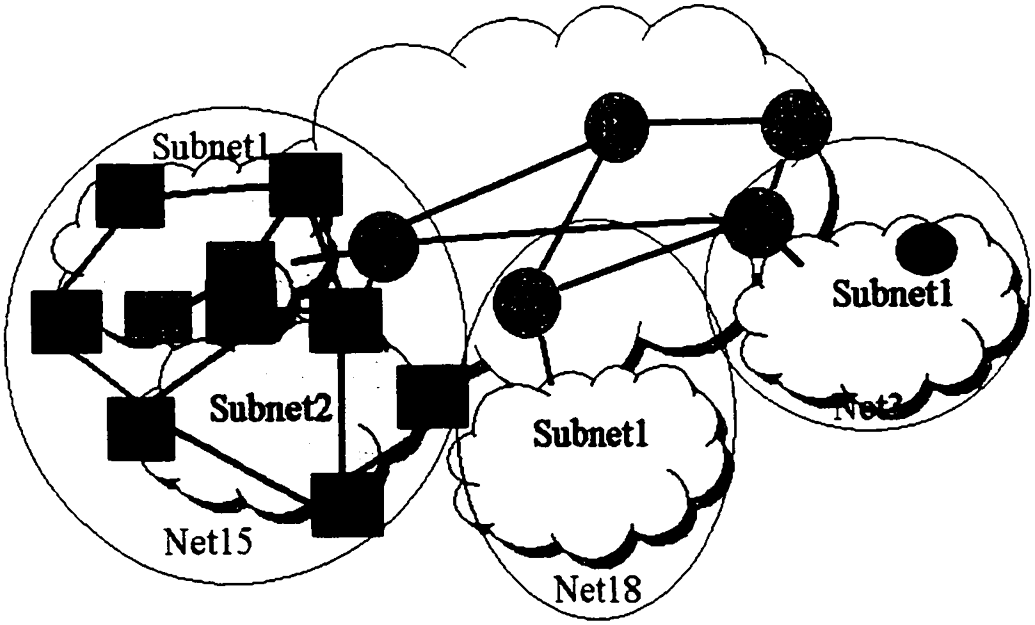


Figure 3.16: The Mobilane network implemented as an overlay network. This network must track and route to/from mobile users in the region.

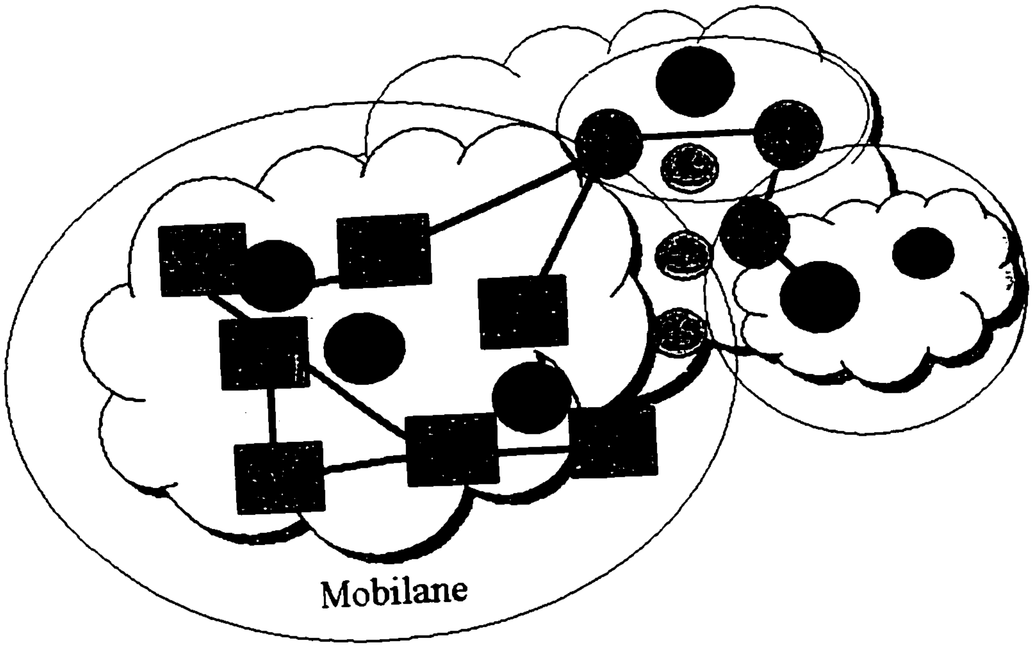


Figure 3.17: MIP_Mobilane exploits Mobilane networks and MIP agents to support user mobility.

tween Mobilanes, as described in Chapter Two. This scheme is depicted in Figure 3.18.

3.6.2 Comparing the Macro-Mobility Schemes

In this section we compare the Mobilane-based schemes to the schemes introduced in Chapter Two. The Mobilane size is equal to that of 100 subnets from the earlier schemes. The target packet loss for the Mobilane design is 1%.

In Figure 3.19 we plot voice call quality. In general, performance is good for the Mobilane-based schemes. Call quality is no longer excellent due to loss inside the Mobilane. There are a small number of calls with bad quality, caused by slow handoffs between two Mobilane networks. This is particularly the case for E2E_Mobilane, where handoffs are slower.

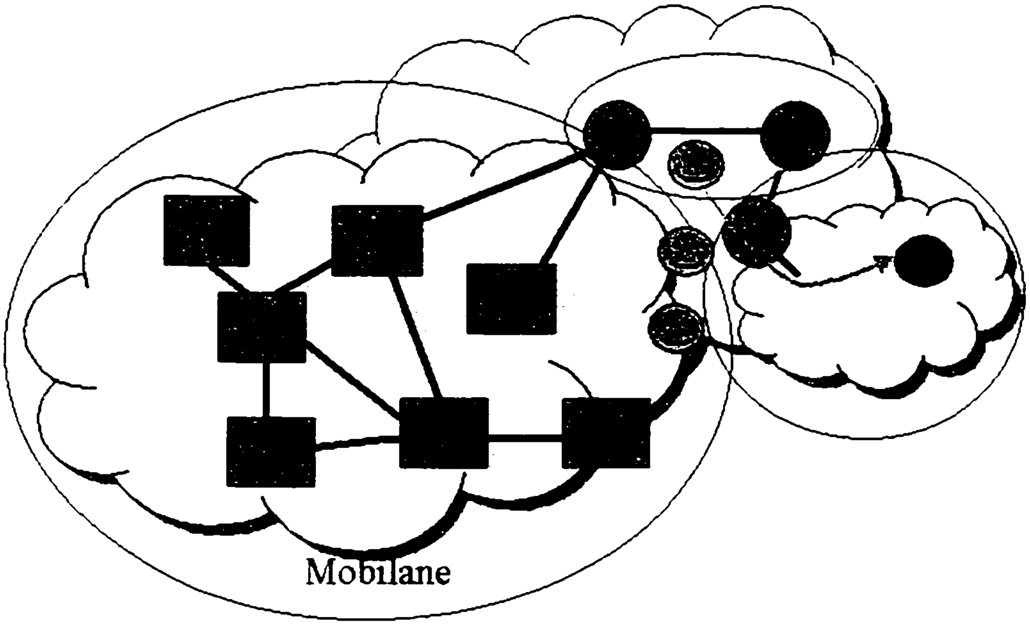


Figure 3.18: The E2E_Mobilane scheme exploits Mobilane networks and end-host software to support user mobility.

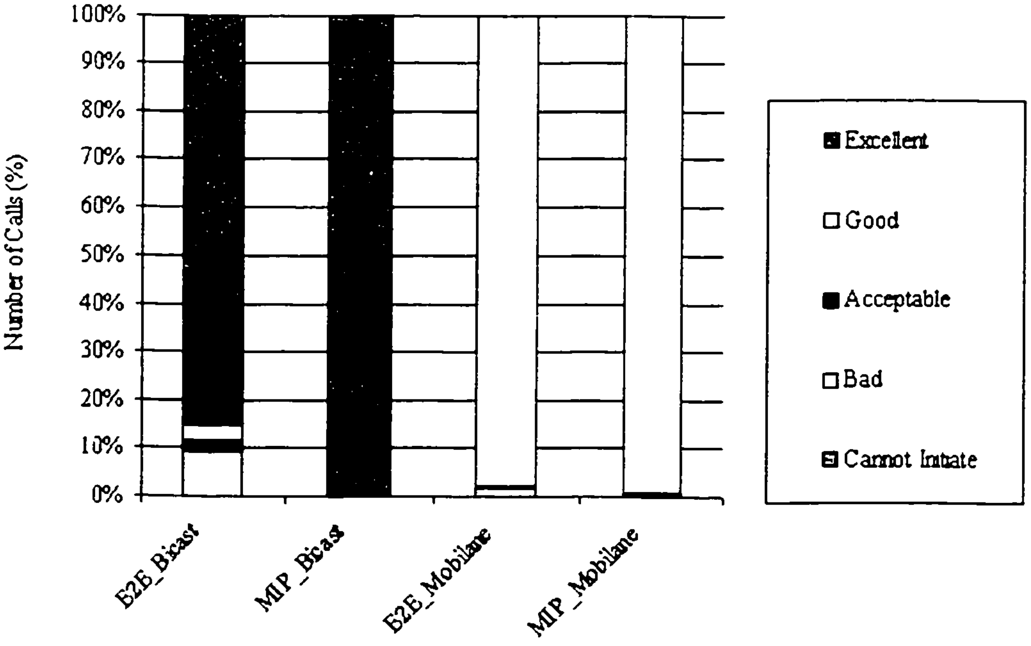
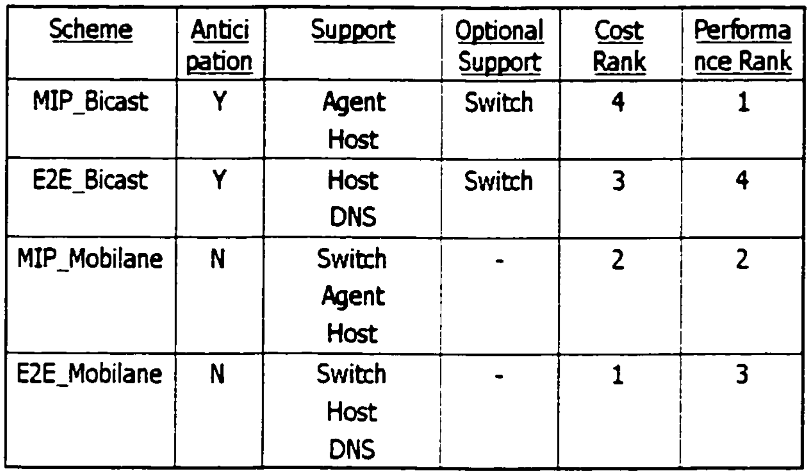


Figure 3.19: Comparing the four schemes to support macro-mobility proposed in this thesis.



Scheme	Anticipation	Support	Optional Support	Cost Rank	Performance Rank
MIP_Bicast	Y	Agent Host	Switch	4	1
E2E_Bicast	Y	Host DNS	Switch	3	4
MIP_Mobilane	N	Switch Agent Host	-	2	2
E2E_Mobilane	N	Switch Host DNS	-	1	3

Figure 3.20: Comparing the four schemes to support macro-mobility proposed in this thesis.

In Table 3.20 we summarize the four schemes for supporting macro-mobility proposed in this thesis. In the five columns we specify the scheme used, whether its design relies on anticipation techniques, the entities where intelligence is placed for implementing the scheme, the entities where optional support may be needed, the cost of the scheme as arbitrarily assigned by the authors and its performance rank as described previously.

The first two schemes we designed MIP_Bicast and E2E_Bicast require anticipation support from the lower layers. In addition, they rely on Mobile IP agents, software support at the end hosts and, in the E2E_Bicast scheme, DNS infrastructure support. Because it depends on both agents and software support at the end hosts, MIP_Bicast is arbitrarily considered to be the more expensive of the two schemes. However, it is also the best performing as shown in the earlier section. E2E_Bicast offers the highest amount of calls with quality less than acceptable, hence arguably the lowest performance. Because of its dependence on anticipation techniques, it is arguably the second most expensive scheme to implement.

The last two schemes we proposed in this thesis are MIP_Mobilane and E2E_Mobilane. They do no rely on anticipation at the lower layers, but instead rely on a low-cost switch infrastructure to support micro-mobility. Hence they arguably represent the two least expensive schemes. In terms of performance, they rank second and third as they result in the second and third fewest calls with less than acceptable quality. If one is to select the optimal scheme to implement macro-mobility support, MIP_Mobilane is a good choice as it ranks second best in terms of both cost and performance.

3.7 Conclusions

With the recent explosion in wireless mobile communication, a great deal of work is taking place to solve the problem of handling mobile users in the Internet. Moreover, large switched LANs have become widely deployed due to recent technological improvements. While LAN technology favors mobile users due to its flat address space and transparent learning capabilities, it can result in packet loss and high bandwidth consumption in a large switched network that is not properly tuned for user tracking.

When tuning the network, one needs to carefully prioritize the goals of the design, in terms of the database size and bandwidth consumption at the switches in the network, and understand the needs of the application in terms of the average packet loss that can be incurred by that application without a significant drop in the quality of service. Furthermore, the application and mobility rates also affect the selection of the optimal scheme and its tuning for best performance. Therefore, whenever possible, one should also account for the mean mobility rate and application rates expected in the network, for an optimal selection and tuning of the tracking and routing scheme.

We compared various schemes to support macro-mobility and found that MIP_Mobilane represents the best choice in terms of cost-effectiveness.

Chapter 4

A Scalable Network Infrastructure

4.1 Introduction

We have seen that the original design of the Internet and its underlying protocols did not anticipate users to be mobile. With the growing interest in supporting mobile users and mobile computing, a great deal of work is taking place to solve this problem. For a solution to be practical, it has to integrate easily with the existing Internet infrastructure and protocols, and offer an adequate migration path toward what might represent the ultimate solution.

We have seen that macro-mobility support can be optimized by deploying a special-purpose network infrastructure to track and route to mobile users inside a large geographical area. An important requirement of this infrastructure is scalability to a large number of users and wide geographical scopes.

In this chapter we propose a scalable infrastructure to support mobility in metropolitan areas. In this respect, we exploit LAN technologies to create the network infrastructure necessary to offer connectivity to mobile users across any geographical area (building, campus, and metropolis). The intrinsic properties of LAN technologies and their underlying protocols, namely flat address space, transparent learning and low complexity render this solution particularly cost effective for supporting user mobility. In particular, we propose a network topology and a set of protocols that render the infrastructure scalable to a large geographical area and many users.

4.2 Related Work

Other researchers have explored the idea of optimizing macro-mobility support by enhancing user tracking within a geographical area of a large size. Such solutions are typically known as micro-mobility solutions.

4.2.1 HFA

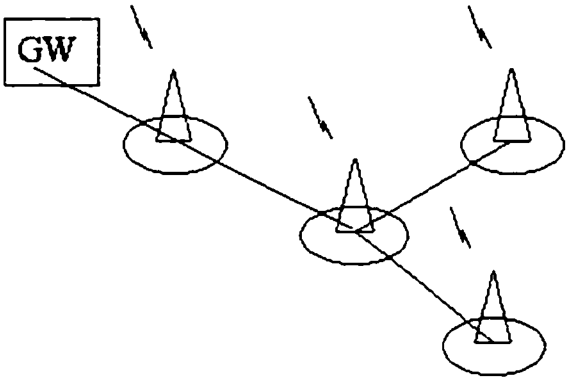
In [30] hierarchical foreign agents (HFAs) are introduced in the context of Mobile IP to smooth out the handoff process when a mobile host MH transitions between subnets. This optimization is accomplished via hierarchical tracking of mobile hosts (MHs) by the foreign agents (FAs) and via packet buffering at FAs.

The FAs of a domain are organized into a tree structure that handles all the handoffs in that domain. The tree organization is unspecified and left up to the network administrator of that domain. One popular configuration is to have a foreign agent associated with the firewall to that domain be the root of the tree (also known as a gateway foreign agent or GFA) and all the other foreign agents provide the second level of the hierarchy.

An FA sends advertisements called Agent Advertisements in order to signal its presence to the MHs. An Agent Advertisement includes a vector of care-of addresses. which are the IP addresses of all its ancestors as well as the IP address of that FA. When an MH arrives at an FA. it registers the FA and all its ancestors with its home agent HA. The registration is seen and processed by the FA. all its ancestors and the HA.

When a packet for the MH arrives at its home network. the HA tunnels it to the GFA. The GFA re-tunnels it to the lower-level FA. which in turn re-tunnels it to the next lower level FA. Finally. the lowest-level FA delivers it to the MH. Therefore. an FA processing a registration should record the next lower-level FA as the other end of the forwarding tunnel.

In Mobile IP with route optimization. binding caches at FAs and binding update messages between FAs can be used to provide smooth handoff. However. tunneled packets that arrive at an FA before the arrival of the required binding message are still lost. Such data loss may be aggravated if the MH loses contact with any FAs for a relatively long period of time. HFA includes an additional buffering mechanism at the FA. Besides decapsulating tunneled packets and delivering them directly to an MH, the FA also buffers these packets. When it receives a previous FA notification. it re-tunnels the buffered packets along with any future packets tunneled to it. Clearly. how much packet loss can be avoided depends on how quickly an MH finds a new FA. and how many packets are buffered at the previous FA. This in turn depends on how frequently FAs send out beacons or agent advertisements. and how long the MH stays out of range of any FA. To reduce duplicates, the MH buffers the identification and source address fields in the IP headers of the packets it receives and includes them in the buffer handoff request so that the previous FA does not need to retransmit those



A Wireless Access Network in Cellular IP. Base stations in an access network are interconnected by wired links. One gateway controls each access network.

Figure 4.1: Cellular IP.

packets that the MH has already received.

While HFA helps reduce the overhead of handoff by handling handoff closer to the MH, it introduces additional complexity in constructing and maintaining the FA hierarchy, and it adds latency due to the need for packet encapsulation and decapsulation at every FA in the FA tree along the path from the CH to the MH. Moreover, scalability issues arise at the root FA and the FAs close to the root of the FA tree because of their involvement in packet tunneling for all the MHs of that domain. Finally, packet buffering results in latency overhead, while encapsulation still generates bandwidth overhead.

4.2.2 Cellular IP

Cellular IP access networks. depicted in Figure 4.1. are connected to the Internet via gateway routers [2]. Cellular IP uses base stations for wireless access connectivity. for IP packet routing and for mobility support inside an access network. Base stations (BSs) are built on regular IP forwarding engines except that IP routing is replaced by Cellular IP routing. MHs attached to an access network use the IP address of the gateway as their Mobile IP care-of-address. The gateway de-tunnels packets and forwards them toward a BS. Inside a Cellular IP network. MHs are identified by their permanent home address and data packets routed without tunneling or address conversion. The Cellular IP routing protocol ensures that packets are delivered to the actual location of the host. Vice-versa. packets sent by the MH are directed to the gateway. and from there. to the Internet.

Periodically. the gateway sends out beacons that are broadcasted across the access network. Through this procedure. BSs learn about neighboring BSs on the path towards the gateway. They use this information when forwarding packets to the gateway. Moreover, when forwarding data packets from users to the gateway. BSs learn about the location of a user. and use that information to deliver packets sent for that user.

If a packet is received at a BS for a user that is unknown to that BS. a paging request is initiated by the BS. The paging request is broadcasted across a limited area in the access network called a paging area. The MH responds to the paging request and its route to the paging BS gets established. Each MH needs to register with a paging area when it first enters that area. regardless of whether it is engaged in communication or idle. Clearly, how fast paging occurs depends on the size of the paging area and on the efficiency of spanning tree traversal. A small paging area can help reduce the latency of paging, however it increases the number of paging area required to cover a given area, which in turn increases the signaling overhead imposed on MHs.

We observe that the paging techniques in Cellular IP are similar to those existent in the Groupe Speciale Mobile system (GSM) [32]. Mobile users are located in system-defined areas called cells that are grouped in paging areas. Every user connects with the base station in his cell through the wireless medium. Base stations in a given paging area are connected by a fixed wired network to a switching center, and exchange data to perform call setups and deliver calls between different cells. When a call arrives at the switching center for a given user, a paging request for that user is initiated across all the cells in that paging area. If the user answers, a security check on the user is performed, and if the test passes, the switching center sets up a connection for that user.

Cellular IP supports two types of handoff: hard handoff and semisoft handoff. MHs listen to beacons transmitted by BSs and initiate handoff based on signal strength measurements. To perform a handoff, the MH tunes its radio to the new BS and sends a registration message that is used to create routing entries along the path to the gateway. Packets that are received at a BS prior to the location update are lost. Just like in Mobile IP, packet loss can be reduced by notifying the old BS of the pending handoff, and requesting that the old BS forward those packets to the new BS. Another possibility is to allow for the old route to remain valid until the handoff is established. This is known as semisoft handoff and is initiated by the MH sending a semisoft handoff packet to the new BS while still listening to the old BS.

After a semisoft delay, the MH sends a regular handoff packet. The purpose of the semisoft packet is to establish parts of the new route (to some uplink BS). During the semisoft delay time, the MH may be receiving packets from both BSs. The success of this scheme in minimizing packet loss depends on both the network topology and the value of the semisoft delay. While a large value can eliminate packet loss, it however adds burden on the wireless network by consuming precious bandwidth.

Cellular IP specifies an algorithm to build a single spanning tree rooted at the gateway to the access network as we described above. A spanning tree is necessary for the broadcasting of packets, to avoid packets from propagating to infinity if the topology of the access network has any loops. However, because it uses only a subset of the links inside the access network, a single spanning tree can result in link overload if traffic in the access network is high. This can be a significant drawback of Cellular IP as high-density access networks supporting many Tb/s of traffic become possible to deploy. Moreover, a single spanning tree can be prone to long periods connectivity loss. Connectivity loss would make this technology unacceptable as a replacement to wired, circuit-switched technology for telephone communications. Finally, Cellular IP specifies an interconnect between base stations that has a flat hierarchy. As access networks cover more area and exhibit higher pico-cell densities, a flat hierarchy would result in latencies of packet traversal across the access network that are unacceptable.

The description of Cellular IP assumes that originally, each wireless cell (or even pico-cell) constitutes an IP subnet. Consequently, they propose that multiple wireless cells be grouped into one subnet to improve roaming between the cells of one subnet. However, this concept is not new. For example, the 802.11 standard uses Extended Service Sets (ESS) to interconnect multiple 802.11 cells within a single

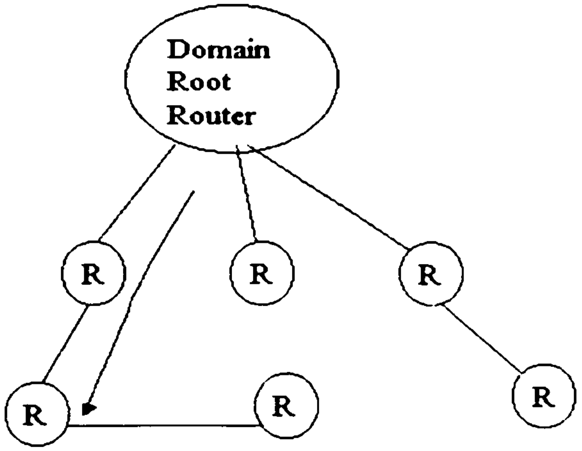


Diagram of a Domain in the Hawaii Architecture. A domain root router acts as the gateway to each domain. Paths are established between the routers of a domain.

Figure 4.2: HAWAII.

subnet. Cellular IP also proposes two protocols for configuration and routing in IP subnets, however, LAN protocols already exist to accomplish these goals. For example, the algorithms for building a spanning tree and for learning as defined by the 802 standards are widely deployed and well known.

Nonetheless, it is clear that deploying wireless access networks as single subnets, like in Cellular IP is important for mobility. In this light, it becomes important to increase the size of IP subnets to the largest size possible in order to maximize their effectiveness in supporting IP mobility.

4.2.3 Hawaii

HAWAII segregates the network into a hierarchy of domains, loosely modeled on the autonomous system hierarchy used in the Internet [31]. The gateway into each domain is called the domain root router. When moving inside a foreign domain, an MH retains its COA unchanged and connectivity is made possible via dynamically established paths, as shown in Figure 4.2. Path-setup update messages are used to establish and update host-based routing entries for the mobile hosts in selective routers in the domain, so that packets arriving at the domain root router can reach the mobile host. The choice of when, how and which routers are updated constitutes a particular setup scheme. HAWAII describes four such setup schemes, which trade-off efficiency of packet delivery and packet loss during handoff. The MH sends a path setup message, which establishes host specific routes for that MH at the domain root router and any intermediary routers on the path towards the mobile host. Other routers in the domain have no knowledge of the IP address of that MH. Moreover, the home agent and communicating host are unaware of intra-domain mobility. The state maintained at the routers is soft: the MH infrequently sends periodic refresh messages to the local BS. In turn, the BS and intermediary routers send periodic aggregate hop-by-hop refresh messages toward the domain root router. Furthermore, reliability is achieved through maintaining soft-state forwarding entries for the mobile hosts and leveraging fault detection mechanisms built in existing intra-domain routing protocols.

HAWAII exploits host-specific routing to deliver micro-mobility. By design, routers perform prefix routing to allow for a large number of hosts to be supported in the Internet. While routing based on host-specific addresses can also be performed at a router, it is normally discouraged, because it violates the principle of prefix routing. Furthermore, host-specific routing is limited by the small number of host-specific entries that can be supported in a given router. However, this concern can be addressed by appropriate sizing of the domain and by carefully choosing the routers that are updated when a mobile is handed off. One of the problems with the implementation of HAWAII is that a single domain root router is used. This router, as well as its neighbors inside the routing tree can become bottlenecks routers for the domain for two reasons: First, they hold routing entries for all the users inside the domain. Second, they participate in the handling of all control and data packets for that domain. Another disadvantage of HAWAII comes from its use routers as a foundation for micro-mobility support. With cells becoming smaller, it is possible that a larger number of routers would be needed for user tracking and routing in a given area; however, this can become prohibitively expensive.

4.2.4 Multicast-based Mobility

Numerous multicast-based mobility solutions have been proposed [6, 26, 35]. In [6, 26], each mobile host is assigned a unique multicast address. Routers in the neighborhood of the user join this multicast address, and thus form a multicast tree for that address. Packets sent to the mobile host are destined to that multicast address and flow down the multicast distribution tree to the mobile host. In [35], packets are tunneled from the home agent using pre-arranged multicast group address, to which a set of neighboring base stations in the vicinity of the mobile host adhere. The most significant drawback of these solutions is that they require routers to be multicast capable; this capability does not exist in the Internet routers of today and would need to be added. In essence, this solution requires that routers learn multicast addresses, in the same way that routers learn unicast addresses in the other schemes for micro-mobility that we discussed. Unlike LAN switches, routers are not designed to learn host addresses, and therefore they would need to be modified for this purpose. Other drawbacks of mobility schemes based on multicast routing are that they require unique multicast addresses to be used, which creates address management complexity and limits the addressing space.

4.2.5 LAN Switching versus IP Routing

In all the solutions for micro-mobility that we presented, fixed IP addresses are used to track mobile users inside some region. This is done via learning at base stations, routers or agents. Despite the use of IP addresses, which are hierarchical, the addressing structure within a micro-mobility region becomes non-hierarchical, just like in a LAN. Consequently, these addresses are tracked in the same fashion as layer-2 addresses in LANs. We further make the observation that, in fact, these addresses are tracked in the same way as virtual channel identifiers in virtual circuit-switched solutions such as ATM (e.g. employed in UMTS for the tracking of users by foreign gateways). In their original design, routers were not intended to perform tracking of individual host addresses, and consequently do not perform host-specific routing in an efficient way. It is unlikely that in the near future, routers will be replaced by routers that perform this function. · instead, layer-2 infrastructure · By design, layer-2 switches track host addresses, hence represent a more suitable solution for mobile tracking inside a micro-mobility region.

4.3 A Network Infrastructure for Mobility Support Using Layer-2 Technology

Over the past decade, we have witnessed tremendous developments in LAN technologies, such as increases in switch processing by a few orders of magnitude, and increases in link bandwidth and distances (owing to the fiber optics technology). These advances resulted in an increase in the size of LANs, and more recently, the deployment of such technologies in metropolitan areas. As noted previously, mobile users can roam inside a LAN without having to update their addresses. Furthermore, layer-2 technology is cost-effective by virtue of its simple protocols and large-scale deployment. For these reasons, layer-2 technology is at the foundation of our network design for mobility support.

This network has a number of design goals, namely: fast tracking of users, a simple, low cost design for easy deployment, incremental scalability from small cities to large cities with many users each consuming high data bandwidth; finally, support for Internet connectivity at high speeds.

As shown in Figure 4.3, the Mobilane is implemented as a grid topology (e.g. the Manhattan Street Network). This is because the grid matches the topology of cities themselves - with the streets being rows and columns -, but also because the grid is scalable by virtue of its distributed nature. Wireless cells are connected to LAN switches in a hierarchical fashion. The hierarchy reduces the number of hops to be traversed when communicating between two access points in the grid, and therefore reduces latency.

The Mobilane topology may not be a perfect grid for various reasons: the Mobilane

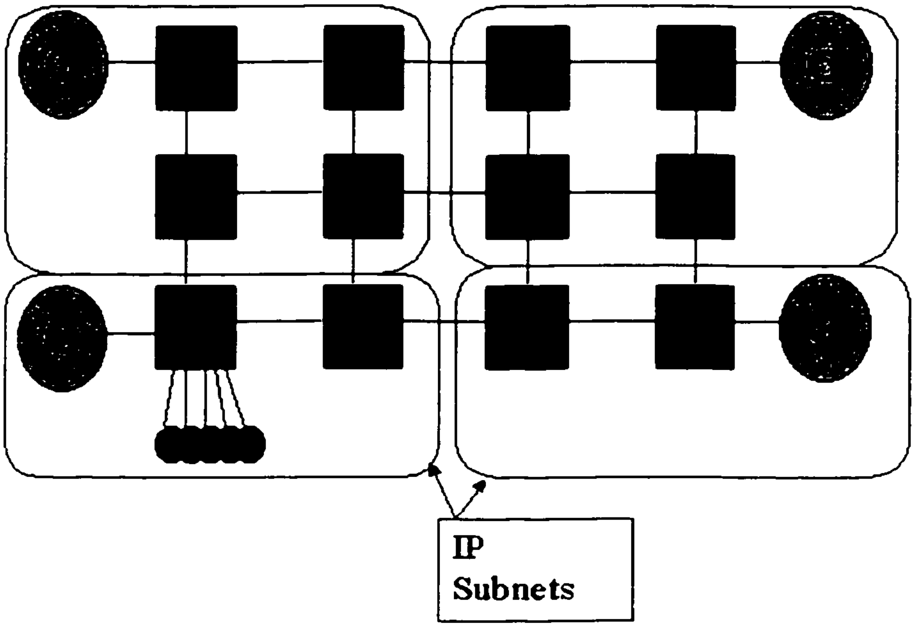


Figure 4.3: One possible subnet partitioning in Mobilane.

can not be deployed as a regular structure due to terrain or administrative restrictions: some switches or links may fail, resulting in rows and columns of varying length. To cope with this situation, we developed the following algorithm to render a grid structure over an irregular topology: We define a virtual row/column to be a set of switches and all the edges between them, such that for any pair of switches in that row/column < a,b >, the path between a and b is entirely contained in that row/column. The following properties are true:

1. The Mobilane is represented by the sum of all virtual rows and by the sum of all the virtual columns.

2. Each virtual row/column must intersect all virtual columns/rows. We partition Mobilane into virtual rows and columns to create the desired grid structure.

Let us take the example of a missing switch, or a "hole". Intuitively, work around "holes" in the topology by including the switches that neighbor the hole into virtual rows/columns as follows: For a hole at location <I,J>, where I is the row number and J the column number:

1. Add switches <I-1,J-1>, <I,J-1>, <I+1,J-1> to virtual column J (assuming these switches are not holes).

2. Add switches <I-1,J-1>, <I-1,J>, <I-1,J+1> to virtual row I (assuming these switches are not holes).

To connect Mobilane to the Internet backbone, a scalable and distributed gateway router is necessary to support the aggregate traffic between the Internet and all the cells in the Mobilane. Indeed, if the number of cells and users is large, this bandwidth becomes also large. For example, for a LAN supporting two million users, consuming 2 Mb/s each, the routing bandwidth is 4 Tb/s. Furthermore, the router must be physically distributed across many smaller routers to allow for load balancing at the links connecting the LAN switches to the subnet router. To address this, we superimpose a number of IP subnets onto one Mobilane (at the logical level), each controlled by one IP gateway, placed at carefully selected locations.

When a user enters a Mobilane, it is given an IP address by the gateway controlling one of the subnets (this gateway is, say, the gateway closest to the user). Evidently, the user IP address does not change while the user is still in the Mobilane. It is the responsibility of the switches inside the Mobilane to appropriately route packets to and from the user regardless of the location of that user. Switches track a user by learning either the MAC or the IP address of that user. The advantage of MAC addresses is their use in LAN switching today. One disadvantage of using MAC addresses is that not all devices come with a built-in MAC address. Another disadvantage is that some form of address resolution needs to take place at the Mobilane in order to map the

IP address of the user to its MAC address (e.g. ARP). If the Mobilane is large, this operation may take a long time to complete. To cope with this challenge, a database may be employed at each gateway to map the IP addresses of users to their MAC addresses. This database holds an entry for every user belonging to the IP subnet for which that gateway is responsible. The database of a gateway is updated at the same time that the user enters the Mobilane and receives an IP address from that gateway.

A protocol is designed for the Mobilane that takes advantage of multiple paths in the network and that balances the traffic load across all the links and switches in the Mobilane. The protocol works by partitioning switches into control and data partitions. Each control partition must have one or more switches in common with every data partition. Similarly, each data partition must have one or more switches in common with every control partition. For example, each row in the grid could be a control partition and each column, a data partition. A protocol similar to the Generic Attribute Registration Protocol (GARP) [15] is used to track users inside a given control partition according to the user location. Data packets for a given user are propagated along a given data partition (as given by the location where the data packet was first injected into the network) until the control partition for that user is reached and the packet delivered to the user. An example of the workings of this protocol is depicted in Figure 4.4. More details are given in the following subsections.

Registration Protocol

As mentioned earlier, the propagation of tracking information in the partition is by means of a dynamic registration protocol similar to the Generic Attribute Registration Protocol (GARP) [8]. The protocol consists of a number of control messages:

1) The JOIN message informs switches of the arrival of one or more users. 2) The LEAVE message informs switches of the user(s)'s departure from the switch's jurisdiction. 3) An ACK message confirms that the registration updates have been received.

A switch of a partition is a local switch if it is directly connected to the access point where a user resides. The other switches in the partition are known as remote switches. A local switch propagates control information to one or more remote switches in the partition. If rings are used to connect the switches along rows and columns, then the control messages at the local switch are propagated as follows: 1) either left or right when the control partition is a row or a virtual row 2) either up or down when the control partition is a column or a virtual column 3) either left and down, left and up, right and down or right and up when a control partition is a pair of rows (or virtual rows) and columns (or virtual columns). When the Mobilane is configured, switches are programmed to propagate control messages in a fixed direction selected form the choices outlined above.

When a control message arrives at a remote switch, it is used to configure the routing tables appropriately. It is also remembered and sent out on the port directly opposite to the port where the control message was received (e.g., if the control message is received on the left port, it is propagated on the right port). When a control message arrives to a switch that has already seen it, it is consumed and removed from the ring.

If a ring topology is not used to connect switches in a row or column, the control message at the local switch is multicast in two directions when the control partition is either a row (virtual row) or column (virtual column) (e.g. up and down for column).

and four directions when pairs of rows (virtual rows) and columns (virtual columns) are used (up, down, left and right). Note that if a given control message did not cause the routing tables for a given port to be updated, then the message is not propagated along that port. One such case occurs when the control partition is a row and the user moves from one switch to the next switch in that row. The new local switch observes that the routing tables for the horizontal ports need not be updated, hence a control message is not sent to the remote switches.

Every time a mobile crosses cell boundaries, a JOIN message is sent to the local switch by the wireless cell to which the user is connected. The JOIN is propagated across the new control partition. An ACK reply is propagated in the reverse direction across the same control partition. Upon reception of an ACK message for the JOIN at the local switch, a LEAVE message is sent to the old switch port, which then gets propagated across the old control partition (we assume that communication between the new and old local switches is possible). An ACK reply is then propagated across the old control partition and sent to the new local switch, which completes the registration process.

Acknowledgments are used to ensure that user registration and de-registration with the switches of a partition are atomic. When moving from one partition to another, a user leaves the old partition only after it has joined the new partition and received an acknowledgment from it. Notice that simply receiving a packet from the switches in the new partition is not enough to guarantee atomicity because some switches may not have been properly set. This is because the packet may come from another user within the partition without traversing all the switches in that partition. However note that acknowledgments can be piggybacked to the user packets that may be traversing the partition. in order to reduce latency and bandwidth consumption. This technique assumes that cell overlap is possible and its extent long enough to allow for the new JOIN to complete. and the ACK to be received.

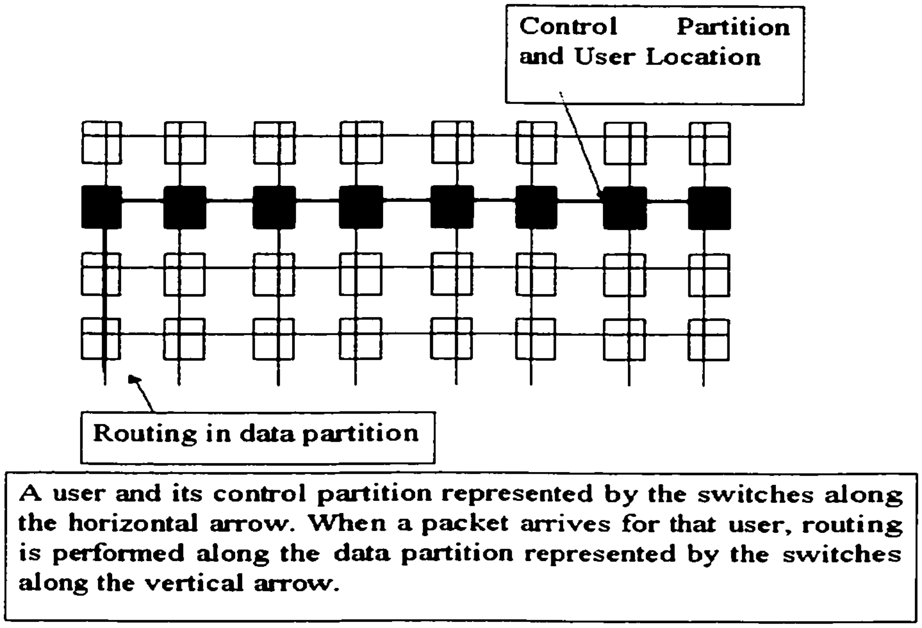
A terminal switch is a switch that determines not to propagate a registration message received at an input port to any output port. Each switch that is not a terminal switch waits for acknowledgments from other switches before submitting an acknowledgment. Terminal switches update their databases and then send acknowledgments without further wait. When single rows or columns are used for control partitions. the local switch waits for a single acknowledgment before proceeding with the LEAVE. When pairs of rows and columns are used as control partitions. then the local switch waits for two acknowledgments. one from the row switches. and one from the column switches. Note that if a switch moves from one partition to another partition that have a common row or column. then a single acknowledgment is needed. corresponding to the column or row that is different.

Routing

In our routing protocol. we capture the mechanism by which routing at the switches is to be performed along data and control partitions. When a switch receives a data packet for some user. the following possibilities arise:

1. if the user is known to the switch (i.e. the switch belongs to the control partition for that user). then the packet is routed according to the information in the table.

2. if the user is not known to the switch (i.e. the switch does not belong to the control partition for that user but belongs to the data partition for that data packet). then the data packet is routed according to some routing algorithm selected when the


Control Partition and User Location
Routing in data partition

A user and its control partition represented by the switches along the horizontal arrow. When a packet arrives for that user, routing is performed along the data partition represented by the switches along the vertical arrow.

Figure 4.4: Control and data partitions in Mobilane.

Mobilane was originally configured.

We identified three such routing algorithms:

1. vertical. If the switch is the first switch in the data partition to see the packet, and the ring topology was not used to connect the switches of a row or a column, then the packet is routed BOTH up and down. If the switch is the first switch in the data partition to see the packet, and the ring topology was used to connect rows and columns, then the data packet is routed either up or down, depending on the switch configuration. This routing scheme is called Vertical-First routing.

2. horizontal. If the switch is the first switch in the data partition to see the packet, and the ring topology was not used to connect the switches of a row or a column, then the packet is routed BOTH left and right. If the switch is the first switch in the data partition to see the packet, and the ring topology was used to connect the switches of a row or a column, then the packet is routed either left or right, depending on the

switch configuration. This routing scheme is called Horizontal-First routing.

3. sometimes horizontal and sometimes vertical. Here we combine the two algorithms above. Some packets are routed using Vertical-First routing while others are routed using Horizontal-First routing.

We describe each algorithm in more detail in the following subsections.

Vertical-First Routing

Vertical-First routing refers to a scheme where a switch in the grid routes vertically a packet that it receives for a user unknown to it. Eventually, the packet arrives at a switch that knows the user (at the intersection of the control partition for that user and the data partition for that data packet), from where the packet is then routed home.

For this routing scheme to work, the tracking protocol requires that the control partition for every user include the row in which that user is located. Vertical-First routing with row control partitions is depicted in Figure 4.4. At first, the packet is routed up the column, where it eventually reaches a switch that "knows" the user. If the column is a ring, the packet is only routed in one direction (either up or down). Otherwise, the packet is multicast up AND down. The first switch that knows the user routes the packet horizontally. If the row is a ring, the packet is only routed in one direction (either left or right). Otherwise, the packet is multicast left AND right. From there on, the packet is routed horizontally until the switch to which the user is attached is found and the packet gets delivered.

Horizontal-First Routing

This works identically to Vertical-First Routing, except that the direction taken at an uninformed switch is horizontal instead of vertical.

Alternating Vertical-First and Horizontal-First Routing

Intuitively, one can see how Vertical-First Routing or Horizontal-First Routing can lead to load unbalance if the users or the input output vertices are concentrated along a particular row or column. For example, if the users are concentrated along some column and Horizontal-First Routing is used, then traffic is routed along the vertical axes by using primarily that column. Also, if the input to the grid is concentrated along one column and Vertical-First Routing is used, then traffic is routed vertically by using primarily that column. To avoid this situation, the routers of the Mobilane must be distributed in a balanced fashion across the Mobilane. However, depending on the region, it may not be possible to control the distribution of routers so to ensure uniform coverage of all rows and columns. Under these circumstances, a routing algorithm that distributes the load fairly across the Mobilane becomes essential. This is particularly important when the link bandwidth is scarce (e.g., microwave radio links or low-cost Mobilane infrastructure).

Consequently, to balance the load, we designed a scheme that alternates Vertical-First Routing with Horizontal-First Routing. The scheme works as follows: instead of always routing horizontally or vertically when the destination user is unknown, switches route vertically sometimes and horizontally other times. By alternating between Vertical-First Routing and Horizontal-First Routing, we ensure that, in the long run, the load is divided fairly across all rows and all columns, regardless of the user and router distribution. In order to guarantee that all packets reach their intended destinations, we must ensure that packets are routed in the same fashion (vertically or horizontally) at all the switches that do not recognize the user for whom a given packet is intended. This is achieved by having the first switch that routes the packet select the routing direction. This switch marks the packet to communicate to the other switches that a routing direction had been selected for that packet. The other switches observe that the packet is marked and are required to follow suit and route in a similar fashion. They route the packet to the port directly opposed to the incoming port (e.g. the left port if the packet was received on the right port). The first switch selects the routing direction by alternating between the vertical and horizontal direction in a round-robin or random fashion.

Clustering

It is expected that users visit cells in the neighborhood of their location more frequently than remote cells. In our schemes, when rows are used as control partitions, and the user moves to the next row, registration with the new row and deregistration with the old row need to take place. When pairs of rows and columns are used as control partitions and the user moves either to the next row or the next column, registration with the new row or new column need to take place. If the user moves frequently, then these updates can be expensive. One solution is to perform user tracking in a hierarchical way that exploits the spatial locality of user movement. Instead of a local switch, we use a cluster of switches, called the local cluster. As before, we use rows (or columns, or both) to track and locate users. However, while before the row (or column, or both) was used to obtain the location of the user exactly, now it is used to obtain the location of the user cluster to which the user belongs. To find the actual the location of the user. we use the tracking at the switches inside the local cluster. Tracking inside a local cluster is performed as follows:

1. if rows are used as control partitions to find the local cluster. then columns (or segments of columns) must be used as control partitions to find the user inside that cluster. By using columns as control partitions inside the local cluster. we ensure that the registration information can be propagated between the local cluster and the row that tracks that cluster.

2. if columns are used as control partitions to find the local cluster. then rows are used as control partitions to find the user inside that cluster.

3. if pairs of rows and columns are used to find the local cluster. then pairs of rows and columns are used to find the user inside that cluster.

When a user moves inside a local cluster. tracking updates need to take place only at switches in that cluster. When a user leaves a local cluster and joins another. we update the tracking databases in the rows and columns of the grid that track the new local cluster. as well as inside the new local cluster. Depending on the size of the cluster. updating tracking databases inside the local cluster can be significantly less expensive than updating an entire row or column of the Mobilane. Clustering can significantly reduce the cost of completing a location update.

Routing is performed in two steps: first. the local cluster is found. via one of the routing algorithms we described before. Second. the actual user location is found. via routing at the switches inside the cluster.

Routing in a Mobilane with an Arbitrary Topology

Partition the Mobilane into virtual rows and columns and apply the same routing algorithms as for the grid Mobilane.

Overall Benefits of the Mobilane Design

This LAN design has a number of advantages. The Mobilane does not rely on a single spanning tree or root switch. This is important for scalability as the Mobilane extends to large geographical scopes. By exploiting control and data partitions, it minimizes the latency of user location updates without affecting the latency of packet routing inside the Mobilane. Finally, its operation relies on existing Mobilane switching techniques and protocols, which makes the solution simple, inexpensive and easy to implement and deploy.

4.4 Simulation Experiments

We use a stochastic mobility model from the University of Waterloo [34], which simulates daily movements of mobile subscribers, incorporating realistic activity patterns. The input data to the model includes an activity transition matrix and an activity duration matrix, derived from the trip survey [37] conducted by the Regional Municipality of Waterloo in 1987. For this survey, a travel diary was completed by each household member over 5 years of age, in which details were recorded on all trips taken during the day of the survey. Included for each recorded trip were the trip start and end times, the trip purpose at the origin and destination, and employment or student status.

There are nine types of activities defined in the model. which are as follows: work. school. shopping. personal business. work-related. serve passenger. social recreation. return home and other. Each activity has an associated time of day. duration and location (at the level of a radio cell). In a given simulation run. activities are selected based on the previous activity and the current time period. The probability of transition from one activity to another uses the activity transition matrix. Once the next activity is selected. its duration is determined using the activity duration matrix. Finally. the location of the activity is selected. based on certain heuristics and the activity type (e.g.. if the activity purpose is "shopping". a cell is randomly selected from a set of cells neighboring the subscriber. within a radius as given by the retail characteristics of the region). Once the location of the next activity is selected. the intermediate route (in terms of cells crossed) and the total distance are determined from a geographical lookup table. Using a user-defined system-wide average speed. the time spent in each intermediate cell is calculated. The subscriber stays in the destination cell for the duration of the activity. and the sequence is repeated. The output from the model is thus a trace of the daily movement of individual cellular subscribers over a period of several days. in terms of cells crossed and time spent in each cell.

The geography of the region simulated. which is shown in Figure 4.5 (drawn by hand from the output of a script which processes the geographical lookup table). covers 312 km2 and is divided into 45 cells. each covering an average of 7 km2. For each subscriber. a home cell. a work cell and a school cell are selected randomly from among the cells in the region. Starting at home. users take daily trips such as going to work and back. going shopping and going to the movies. We simulated 35

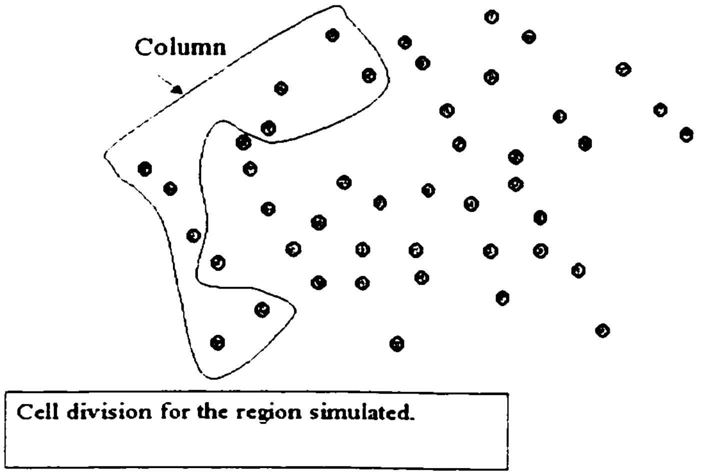


Figure 4.5: Cell presence and sample column composition.

Range	$7\ km^2$
Bandwidth	100 kb s
Total number of cells	45
Example of protocol	Cellular telephony

Table 4.1: Macro-cell scenario.

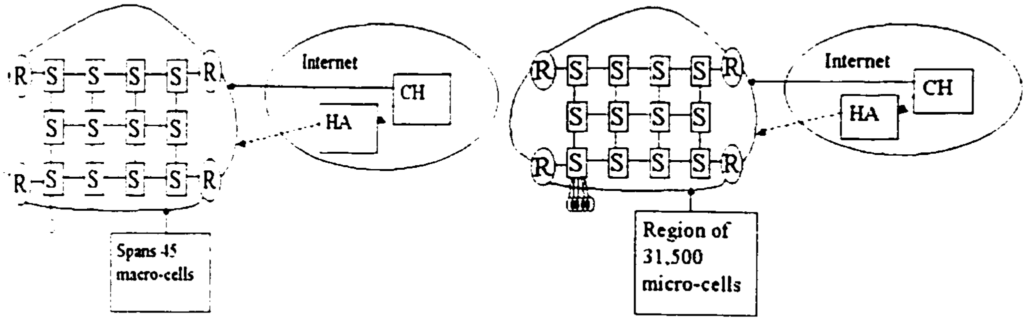
days during which subscribers were tracked on a per minute basis. The simulation is event-driven and has a granularity of one minute. An event that occurs between two minutes is considered to have occurred at the earlier minute. Since they happen during the same minute, the JOIN, LEAVE and ACK corresponding to a cell crossing occur simultaneously in our model.

In our simulations, two wireless cell types are employed: macro-cells and micro-cells. A macro-cell maps directly to a cell as defined in the mobility model discussed above. The characteristics of the wireless macro-cells are summarized in the Table 4.1.

An architecture where cells are further subdivided into micro-cells is also discussed (Table 4.2). Macro-cells are subdivided into micro-cells to increase user bandwidth.

Range	0.01 km^2
Bandwidth	11 Mb s
Total number of cells	31.500
Example of protocol	802.11b

Table 4.2: Micro-cell scenario.



(a) Macro-cell configuration.

(b) Micro-cell configuration.

Figure 4.6: Macro and micro cell configurations.

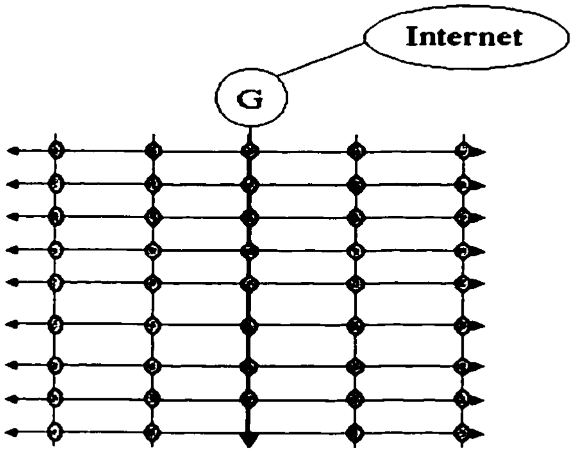
Mainly, one macro-cell is subdivided into 700 smaller cells, allowing short-range wireless protocols (like 802.11b) to be used inside a micro-cell. This results in approximately 7.5 Gb s bandwidth to be available inside a cell, i.e. an increase by 770,000X in bandwidth per cell. Higher user bandwidth is crucial for delivering multimedia to mobiles. A single H.263 video stream consumes 10Kb s, so this would allow for approximately 770,000 users to receive different video streams simultaneously inside a cell. For this configuration, multiple switches are used to balance the load across the region and to reduce the number of ports needed for connecting the micro-cells at a switch. Since the total load across all the micro-cells in the region is about 340 Gb/s, we chose to spread this load across 45 switches, each responsible for 7.5 Gb s sent to/from users in the 700 micro-cells of a cell. The switches are grouped in a grid consisting of 9 rows and 5 columns (9X5).

To compare the performance of our system against the performance of other schemes, we consider three scenarios as follows:

1. The Mobile IP scenario: nine foreign agents populate the region, each responsible for one ninth of space (as given by a row in the grid). For experimental purposes, the HA is considered to be outside the region, with each FA being directly connected to the HA (at equal distances). Location updates are sent to the HA. To reach its destination user, traffic arrives at the HA from where it gets forwarded to one of the foreign agents inside the region. This is shown in Figure 4.6 by the interrupted line between the CH, HA and the region. Notice that Mobile IP allows the CH to receive binding updates with the latest COA of the user. However, in our simulations, each call is generated by a different CH, hence the information in the binding cache is irrelevant.

2. A typical micro-mobility algorithm (such as HFA or Cellular IP), labeled Micro IP. 45 routing devices (such as foreign agents in HFA or switches in Cellular IP) populate the region, and are connected via a hierarchical interconnect as shown in Figure 4.7. A gateway connects the root of the tree to the Internet. Location updates are handled locally by the hierarchy and do not need to reach the HA. However, all traffic from the CH traverses the HA prior to arrival at the region.

3. Smooth IP: The Mobilane is populated by 45 switches interconnected via a regular, 9X5 grid topology. Each switch controls one macro-cell or 700 micro-cells in a cell area. To superimpose the grid structure on the region, the switches are grouped into irregular columns and rows (an example of a column is shown in Figure 4.5). Figure 4.6 depict the Smooth IP implementation for each wireless cell type we considered. In this work, Vertical-First Routing is used, and no clustering. For the



The tree hierarchy in Micro IP. The gateway (G) is connected to the root of the tree, and to the Internet. The region is populated by 45 routing devices.

Figure 4.7: The tree structure for Micro IP.

users in the Mobilane. DNS is updated with the COA of the gateway responsible for the region. Traffic from the CH arrives directly at the gateway from where it is forwarded to the switches. This is shown in the figure by the uninterrupted line between the CH and the region. Location updates are handled by the switch local to the moving user and do not need to be forwarded to the HA.

A stochastic model for user voice traffic is assumed in order to test the impact of the mobility architecture on user traffic performance. Traffic is originated from a CH, located outside the area, and is addressed to a user inside the area. Users receive phone calls at random times throughout the day. The number of phone calls per day and the duration of each phone call are entered statically as parameters to the simulation.

We identified a number of important metrics for evaluating our architecture: the peak per minute control load, including the total number of JOIN's, LEAVE's and

ACK's at a switch BW_CTL: the peak per minute data load at a switch BW_DTA: the time to complete a location update. in terms of the number of switches traversed LAT_CTL. Note that. to complete an update. multiple switches may be traversed in parallel (e.g.. across the row and column of a control partition). In this case. we consider update latency to be the longest of the parallel tracks traversed to complete the update. In the case of LAT_CTL. we consider the time to setup paths to users entering the region (Init). as well as the peak (Peak) and average (Avg) per minute values across all users. We also consider the time for a packet to travel from source to destination. which we call routing latency or LAT_DTA. We measure routing latency in terms of the average number of switches traversed per packet across all users.

To compute BW_CTL. we count the number of control messages per minute. multiply it by the length of the control message and divide it by 60. The length of the control message is assumed to be 16 Bytes. We think 16 Bytes is a good estimation since it includes 12 Bytes for the source and destination addresses. 2 Bytes for the checksum and 2 bytes for various attributes and options.

To compute LAT_CTL. we measure the number of cell crossings that occur during a simulation. and multiply this number by the number of switches involved in a location update for each algorithm. The duration of the location update depends on the scenario as follows: For the Mobile IP scenario. it represents the time to update the database at the local FA. to transfer the update message between the FA and the HA and the time to update the database at the HA: For the Micro IP scenario. it represents the time to update the appropriate routing devices in the tree: Finally. for Smooth IP. a location update includes the time to update the appropriate switches in the Mobilane.

To compute LAT_DTA, we measure the number of switches involved in transferring each packet across the extended LAN and multiply it by the time to transfer the packet at the switch. We add the time to transfer the packet between the CH and the gateway in charge of the mobile. For the Mobile IP and Micro IP scenarios, we add to this number the time to transfer the packet between the HA and the region. We show the sensitivity of LAT_DTA to transfer time between the HA and the region. We assume that the distances between the CH and the HA, between the HA and the region and between the CH and the region are the same.

4.4.1 Control Load

In this section, we investigate the scaling of control load with the size of population in the region, in the case of Smooth IP with macro-cells and Smooth IP with micro-cells. It is important that the control load amount to a small number (relative to the capacity of the switches of today) for Smooth IP to be a feasible solution, for a variety of population sizes.

Figure 4.8 shows BW_CTL for Smooth IP with macro-cells and Smooth IP with micro-cells, and for various population sizes between 1000 and 5,500 people. Prohibitively long simulation times limited the population size studied to this range.

The dependency of BW_CTL on population size is sub-linear. When scaling from 1000 to 5500 people, we noticed sub-linear effects: less than 3.3X and 2.5X increase for Smooth IP with macro-cells and Smooth IP with micro-cells, respectively. This is explained by the fact that, for this population range, an increase in population size contributes primarily towards the spreading of traffic across more cells and more time slots and only secondarily towards increasing peak traffic in a given time slot and a

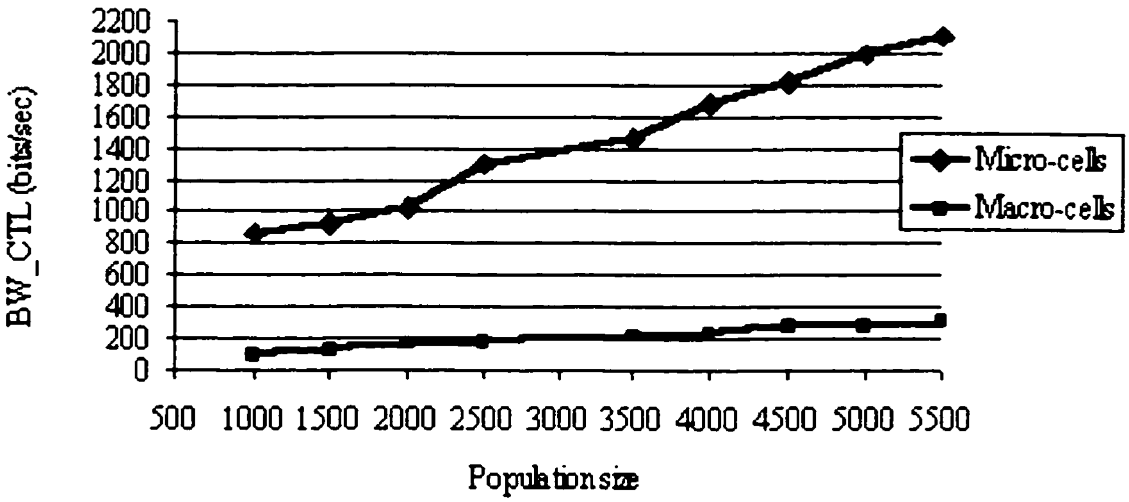


Figure 4.8: Control load.

given cell. More substantial increases in BW_CTL are expected for high population ranges due to higher user clustering across time slots.

As expected, cell subdivision results in a significant increase in BW_CTL because of the increase in the rate of cell crossings. The subdivision of each macro-cell into 700 micro-cells results in about 7X increase in BW_CTL. Assuming linear increase for high population densities, a population of 500,000 people (which is large for a region of 312 km2), would generate about 200 Kb s in control traffic in the case of Smooth IP with micro-cells. Assuming a registration latency of 1 microsecond, (or the equivalent of ~10 SDRAM lookups), a single switch can support 1,000,000 registrations per second, or an equivalent of 122 Mb s. Since the maximum value achieved by BW_CTL is ~200 Kb/sec, there is more than enough bandwidth to support the registration bandwidth generated by this region fully populated (at 500,000 people).

In this section we demonstrated the feasibility of Smooth IP with macro-cells and Smooth IP with micro-cells, with regard to supporting registration traffic for tracking user movement. In the rest of the chapter, we present results only for Smooth IP with

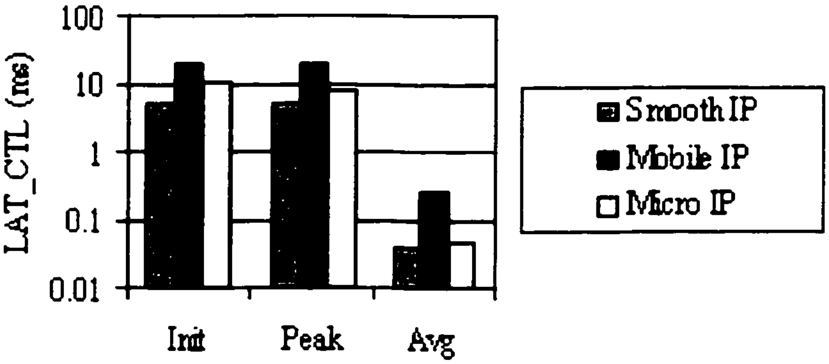


Figure 4.9: Control latency.

micro-cells, because it represents a more interesting experimental case, and it delivers higher data bandwidth to users.

4.4.2 Control Latency

Figure 4.10 shows the behavior of the average LAT_CTL as a function of distance to the HA for Mobile IP. LAT_CTL for Smooth IP with one gateway is plotted for reference. A 1 ms processing delay at the FA and every switch in the Mobilane is assumed. Processing time at the HA in considered part of the distance to the HA. As we can see from the graph, Mobile IP does generally worse than Smooth IP because of the overhead of communicating with the HA upon every location change. However, if the HA is nearby to where the user is located, HA updates may take less time than local updates like in Smooth IP. This is because, for Smooth IP, a larger number of electronic switches need to be involved in one location update than for Mobile IP.

Figure 4.9 shows Init, Peak and Avg. LAT_CTL for Smooth IP, Micro IP and Mobile IP. The distance to HA is arbitrarily set to 20 ms. We observe that Smooth IP perform better than both Mobile IP and Micro IP in all three cases: the time to setup connectivity to subscribers (Init), and the peak and average LAT_CTL. The reason

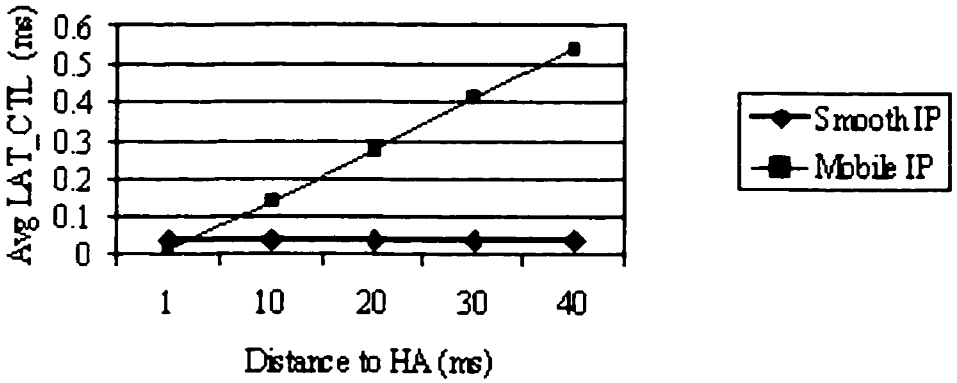


Figure 4.10: Average control latency.

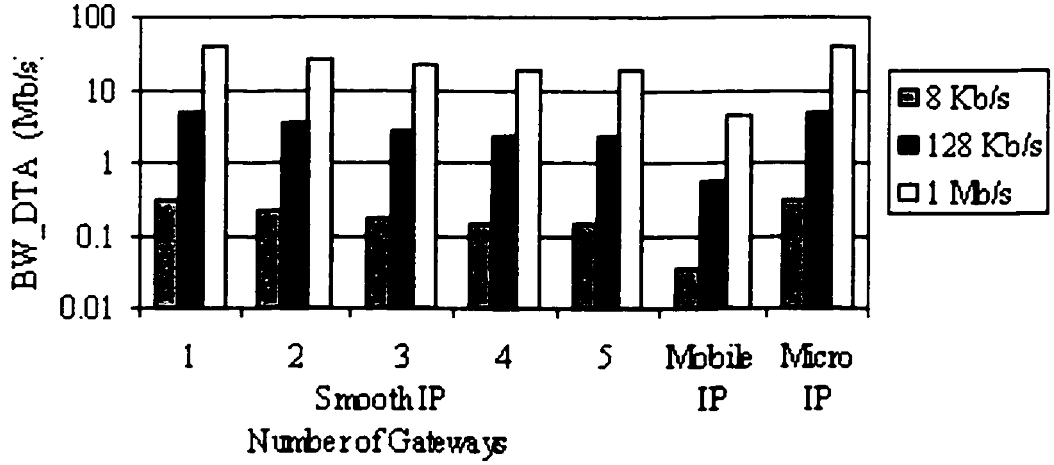


Figure 4.11: Data load.

Smooth IP performs better than Mobile IP is because no HA overhead is incurred. The reason Smooth IP performs better than Micro IP is because fewer switches need to be informed of the location of users, due to the selective multicast nature of the routing algorithm in Smooth IP.

4.4.3 Data Load

Figure 4.11 plots BW_DTA for different numbers of gateways, different per-user application bandwidth (8Kb/s, 128 Kb/s and 1 Mb/s) and for three different schemes: Mobile IP, Micro IP and Smooth IP (with five different numbers of gateways). We notice that the best results are achieved for Mobile IP. This is because, in this implementation of Mobile IP, foreign agents only track and route to/from users in their portion of the region and do not transfer traffic sent for another foreign agent in that region. When multiple gateways are used, BW_DTA decreases considerably from the case of a single gateway. When four gateways are used, less than half as much peak traffic is born by the switches than when one gateway is used. This means that, when multiple gateways are used, smaller, cheaper switches can be deployed in the Mobilane.

4.4.4 Data Latency

Impact of Dynamic DNS

Figure 4.12 displays LAT_DTA for various distances between the MH and the HA and the CH and for three different scenarios: Smooth IP with one gateway, Micro IP and Mobile IP. LAT_DTA increases monotonically with distance: a slower rate of growth is observed for Smooth IP (owing to the use of dynamic DNS which circumvents the need for tunneling through the HA). For a distance value of 40 ms, LAT_DTA is almost twice smaller if dynamic DNS is used. Notice that for small distance values (a few ms), LAT_DTA is actually larger in the case of Smooth IP and Micro IP than for Mobile IP. This is because of the inherent overhead of traversing the switches

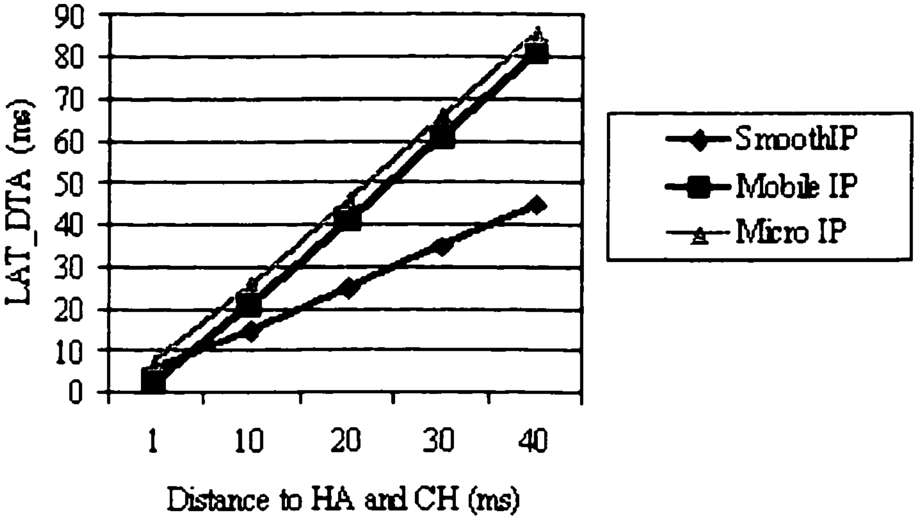


Figure 4.12: Data latency.

populating the region. In the case of Mobile IP, fewer electronic components are traversed as packets are routed between the CH, HA and the MH.

Impact of Multiple Gateways

Figure 4.13 shows the variation in LAT_DTA for Smooth IP in terms of the number of gateways. We notice that LAT_DTA decreases slightly as more gateways are used. This is for two reasons: when the Mobilane is populated with more gateways, these gateways are placed closer to subscribers than when fewer gateways are used. Secondly, subscriber movement exhibits a small degree of locality: on average, subscribers are closer to their home gateways than to any other gateway in the Mobilane. A distance of 20 ms between the Mobilane and the CH, and a 1 ms processing delay at the gateway are assumed.

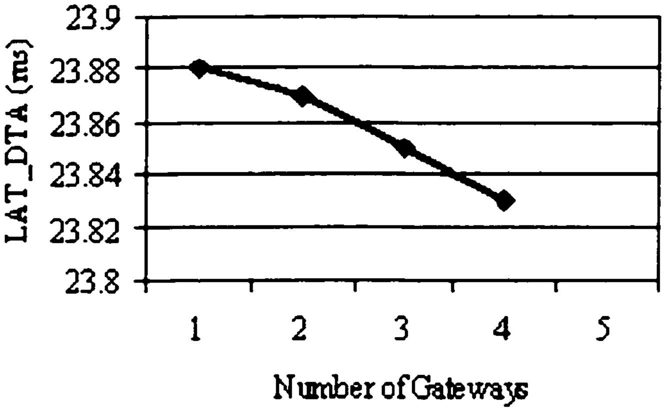


Figure 4.13: Data latency in terms of the number of gateways.

4.5 Contributions

Our simulations demonstrate that Smooth IP is a scalable. feasible solution for supporting IP mobility over a large metropolitan area. By virtue of localized registration and distributed routing algorithms. and the use of dynamic DNS. Smooth IP fairs better in performance than other existing schemes. Reliance on existing. low-cost switching technologies. as well as load balancing techniques and the use of multiple distributed gateways which have the effect of lowering the peak load at switches. make it possible to implement Smooth IP with small and inexpensive technology.

4.6 Migration Path

To transition to the mobile network of tomorrow. it may not be possible to design the supporting network infrastructure from scratch. Instead. support for mobility may need to be built on existing network structures. such as small subnets controlled by LAN switches and interconnected by IP routers with a small number of host-specific entries. Under such circumstances, one possibility is the use of Virtual Private

Networks (VPN) to offer extended LAN connectivity across multiple small subnets. In order to support mobile users in the most effective fashion, the protocol to handle mobile users needs to be flexible enough to operate at different layers in the protocol stack, and versatile enough not to require changes in the implementation of the LAN switches and IP routers of that network. In particular, the protocols running on LAN switches should be based on existing 802 protocols, since they are implemented in hardware and therefore cannot be easily replaced or reprogrammed. The main challenge becomes how to use and optimize existing protocols for the purpose of efficient support for mobility.

4.7 Conclusions

This chapter surveys the state-of-the-art in providing mobility support to mobile users in metropolitan areas. In particular, emphasis is placed on micro-mobility techniques designed to accelerate Mobile IP. One observation is that all micro-mobility solutions work in a similar way by requiring that network devices inside a given geographical area learn about the location of users and keep track of them as they move inside that area. The differences among these techniques are the type of device required to do the learning (it could be an IP router, Mobile IP agent or LAN switch) and the protocols for routing packets using the learning databases.

In this chapter we also propose the deployment of an infrastructure based on simple and cost-effective layer-2 technology to support user mobility in geographical areas of many scopes. One important feature of this infrastructure is its scalable and efficient design, geared at accelerating Mobile IP in large geographical regions. By relying on existing technologies, and by virtue of working with Mobile IP, this solution is global. cost-effective. easily deployable and compatible with the Internet of today.

Chapter 5

Conclusions

In this thesis. we address the problem of supporting macro-mobility and micro-mobility in the Internet. with the goal of improving the user-perceived quality of Internet applications for mobile users. Specifically. we assess. design and optimize various schemes for tracking mobile users that account for the state of technology. and are optimized for a variety of user mobility rates and application characteristics. Finally. we also design a network infrastructure overlay that optimizes mobility support in large metropolitan areas.

We show that existing schemes to support macro-mobility in the Internet can be impaired by various delay overheads arising from the address acquisition process. and propagation delays in the network. We demonstrate that anticipation-based techniques such as address lookahead and packet n-casting can be very effective at improving the user-perceived quality of voice calls and the reliability of TCP-based applications.

Using numerical simulation, we show that a combination of transparent learning and explicit registration techniques are needed to efficiently track mobile users in a

LAN-based MAN. We compare various schemes to support macro-mobility and rank MIP_Mobilane as the best choice in terms of cost-effectiveness.

Finally, we make the observation that all solutions to support micro-mobility work in a similar way by requiring that network devices inside a given geographical area learn about the location of users and keep track of them as they move inside that area. The difference among these techniques is the type of device required to do the learning (it could be an IP router, Mobile IP agent or LAN switch). We propose the deployment of an infrastructure based on simple and cost-effective layer-2 technology to support user mobility in geographical areas of many scopes. This infrastructure is scalable and effective, and compatible with the existing Internet.

5.1 Future Work

In this thesis, we have shown that a great deal of packet loss can be incurred in networks supporting mobile users. We exploited various techniques to accommodate packet loss, such as buffering and tunneling mechanisms at Mobile IP agents, source buffering and retransmission, and special-purpose infrastructure for fast tracking of mobile users. The context for the work was homogeneous IP networks supporting unicast user-to-user communication.

In future work, we plan to extend this investigation to ad-hoc and heterogeneous networks, supporting multipoint-to-multipoint (also referred to as "multipoint" in this writing) communication. Unfortunately, the same mechanisms cannot be easily exploited under the new scenarios. First, ad-hoc and heterogeneous networks cannot make use of special-purpose infrastructure that is deployed on a large scale for supporting mobile users. Multipoint communication makes buffering and retransmission at the source, which accounts for the characteristics of an individual corresponding host more difficult to implement, as scalability becomes a crucial concern. In the rest of this section, we address each of these new contexts, the challenges and research opportunities they bring about.

5.1.1 Mobility Management in Ad-Hoc Networks

Network infrastructure can be expensive to deploy, is typically fragmented over many standards and cannot be easily adapted and modified to incorporate new technologies and standards. For these reasons, some have predicted that investments into network infrastructure will be very limited in the future.

Since network infrastructure becomes a sparse, scarce or even non-existent resource, a good approach for supporting mobility is to rely in part on intelligence at the mobile nodes of a network for tracking and routing to the mobile users in that network.

Under this scenario, packet loss due to user movement may be addressed in one of the following ways:

- Precise tracking of mobiles (by the fellow mobile hosts in the network). There is a wealth of existing literature in this area (e.g. distance vector routing, link state routing) [32]. Unfortunately, current technology significantly limits the number of users that can be tracked by a node built at a reasonable cost. Furthermore, if the topology is very dynamic, the exchange of routing information can be expensive and may not complete fast enough, thus routing information at the mobile nodes may become stale, resulting in further packet loss.
- Limited, selective packet flooding. This is a possibility we would like to explore.

It allows tracking information to be less precise and even inaccurate. However. the disadvantage of packet flooding is the consumption of scarce link bandwidth and bandwidth and processing power at the nodes.

- The combination of the two proposal above. i.e. host-specific routing and limited packet flooding. This is a new approach we would like to propose. Tradeoffs must be investigated as to properly account for bandwidth and battery consumption at the wireless links and nodes. the size of the routing databases at the mobile nodes and the needs of the application in terms of packet loss and packet delay. The investigation of this tradeoff will be at the core of our research in this context.

One final issue in this area pertains to network security and user privacy. In these respects. protocols that exploit multi-hop routing can be compromising to communication due to their distributed design. To overcome these problems. one possibility is to perform application or transport layer encryption. However. such an approach relies on secure end-to-end key exchanges among communicating parties. which can be difficult to provide. Additional lower-layer support may be necessary to enforce network security and user privacy.

5.1.2 Mobility Management for Multipoint Applications

Multipoint applications present further challenges as diverse sender and receiver sets need to be supported efficiently and reliably despite the user movement.

In this context, packet loss can be alleviated through diversity. by relying on the multicast support already present (by virtue of the supported application). either in the network infrastructure (e.g. IP multicast), or at the distribution source (intelligent server).

For example, if IP multicast is used, a moving host can join multicast routers at the endpoints it is visiting, and receive packets at any such endpoint(s) that the mobile can reach at one time.

If intelligence is needed at the source (e.g., counting acknowledgments), challenges arise due to scalability as discussed earlier, but also the need to distinguish multiple individual recipients ("true" multicast) from a single recipient simultaneously registered at multiple endpoints (mobility). These challenges are even harder to overcome when the source becomes mobile and hence less readily reachable. Finally, when multi-point sources are used, the complexity of the problem is heightened even further.

One approach is to develop feedback schemes that allow the source(s) to make appropriate decisions using hints from the network, receiver set and other sources. The schemes can be dynamic and adaptive as to account for user mobility and timing delays in the network, as well as the needs of each individual application.

5.1.3 Mobility Management in Heterogeneous Networks

The challenges here arise from the richness of the problem and the need to support a wide variety of networks (including infrastructure-based and ad-hoc networks) with various characteristics and capabilities. An important challenge involves bridging hosts situated in different network environments through communication sessions that effectively exploit the resources of each environment. Work in this context is divided in two areas, according to the application types supported.

Singlepoint Applications

For heterogeneous networks. the main issue is that no single scheme fits all network designs and characteristics. Thus. one idea is to partition heterogeneous networks into a number of small homogeneous networks. and use intelligent gateways for robust communication across such networks. Each gateway operates at many layers. has knowledge of the network. application and mobility characteristics and semantics of the network it controls. and is capable to interface to the outside world in a effective manner.

An important component of this research comes from understanding the characteristics and demands of many networks that have not been explored before in our research. including terrestrial digital broadcasting and data-casting. satellite and ATM networks. and developing mobility support schemes for such new networks.

Multipoint Applications

Multipoint applications present further challenges as diverse sender and receiver sets need to be supported efficiently and reliably despite the user movement. One possibility is that a hybrid communication model be used to partition the heterogeneous multipoint sender and receiver sets into a number of small homogeneous data groups. and that robust unicast communication protocols be used across the data groups. A network of high-level. intelligent agents uses detailed knowledge of the network. application and mobility characteristics and semantics to adapt to the heterogeneity constraints.

5.1.4 Conclusions

In conclusion, in our future work we propose to explore mobility support in a variety of new contexts, specifically ad-hoc and heterogeneous networks supporting multipoint applications.

The main thrust of our research will involve the modeling and simulation of complex and rich network environments, populated with users moving at a wide-range of speeds, involved in multimedia and data communication that is both singlepoint and multipoint.

We expect to develop schemes for mobility management that are cost-effective, taking into account the state of technology, the network and mobile node characteristics and the needs of realistic applications. We also expect to develop a system architecture and design of multi-layer gateways that support network heterogeneity and multipoint applications. The designs will be specific to industry needs such as support for terrestrial digital broadcasting and ad-hoc networks for industrial automation.

Eventually we hope that the proposed work will result in its standardization as part of a suitable IEEE working group and its general public promotion and acceptance.

Bibliography

[1] I. Akyildiz, Y. Lin, W. Lai, R. Chen, “A New Random Walk Model for PCS Networks”, IEEE JSAC , 18(7): 1254-1260, 2000.

[2] A. Campbell, J. Gomez, S. Kim, A. Valko, C. Wan, “Design, Implementation and Evaluation of Cellular IP”, IEEE Personal Communications (June July, 2000).

[3] S. Cheshire, M. Baker, "Internet Mobility 4X4", Proceedings of the ACM SIGCOMM 1996, Stanford, CA, August 1996.

[4] Y. Cui et al, "Efficient PCS Call Setup Protocols", IEEE Infocom, San Francisco, CA, 1998

[5] M. Gritter, D. Cheriton, "An Architecture for Content Routing Support in the Internet", USENIX Symposium on Internet Technologies and Systems, San Francisco, 2001

[6] A. Helmy, "A Multicast-based Protocol for IP Mobility Support", Second International Workshop on Networked Group Communication, Palo Alto, CA, November 2000.

[7] C. Hristea. F Tobagi. "A Network Infrastructure for IP Mobility Support in Metropolitan Areas". Computer Networks 38 (2002) 181-206.

[8] C. Hristea. F. Tobagi. "Assessing the Impact of Handover Delay on User-Perceived Quality of Internet Applications". in submission.

[9] C. Hristea. F. Tobagi. "IP Routing and Mobility". IWDC. Italy 2001.

[10] C. Hristea. F. Tobagi "User Tracking and Routing in Mobile Packet-Based Networks". in submission.

[11] http://www.arl.wustl.edu/~jst/cs/541/src

[12] http://www.metricom.com

[13] http://www.nrg.cs.uoregon.edu/topology_generation

[14] http://www.ssfnet.org/homePage.html

[15] IEEE 802.1d MAC Layer Bridging Standard.

[16] IEEE 802.1s Multiple Spanning Tree Standard.

[17] IETF Dynamic Host Configuration Protocol. RFC 2131. 2132.

[18] IETF Ethernet Address Resolution Protocol. RFC 826.

[19] IETF "Fast Handovers for Mobile IPv6". < draft-ietf-mobileip-fast-mipv6-03.txt>, work in progress. July 2001

[20] IETF "Low latency handoffs in Mobile IPv4". < draft-ietf-mobileip-lowlatency-handoffs-v4-00.txt>, work in progress. February 2001.

[21] IETF Mobile IP Standard. RFC 2002. 2003. 2004.

[22] IETF "Simultaneous Bindings for Mobile IPv6 Fast Handoffs". draft-elmalki-mobileip-bicasting-v6-02.txt . June 2002.

[23] ITU-T Recommendation G.107. "E-model. a computational model for use in transmission planning". December 1998.

[24] V. Lines et al. "66MHz 2.3M Ternary Dynamic Content Addressable Memory". IEEE 2000.

[25] A. Markopoulou. F. Tobagi. M. Karam. "Assessment of Voice over IP Quality over Internet Backbones" Infocom 2002. New York. June 2002.

[26] J. Mysore. V. Bharghavan. "A New Multicasting-based Architecture for Internet Host Mobility". Mobicom 1997. Budapest. Hungary. September 1997 .

[27] "Naming. Addressing and Identification Issues for UMTS". UMTS Forum. December 2000.

[28] C. Parsa. J. Garcia-Luna-Aceves. "Differentiating Congestion vs. Random Loss: A Method For Improving TCP IP Performance over Wireless Links". http: citeseer.nj.nec.com 504913.html

[29] C. Perkins. "IP Mobility Support". RFC 2002. October 1996.

[30] C. Perkins. K.Y. Wang. "Optimized Smooth Handoffs in Mobile IP". Proceedings of the IEEE Symposium on Computers and Communications. June 1999. Red Sea. Egypt.

[31] R. Ramjee et al. "HAWAII: A Domain-Based Approach for Supporting Mobility in Wide-Area Wireless Networks". Proceedings of the IEEE International Conference on Network Protocols. 1999.

[32] J. Schiller. "Mobile Communications". Pearson Education Limited 2000.

[33] W. Sincoskie. C. Cotton. "Extended Bridge Algorithms for Large Networks". IEEE Network. January 1988-Vol. 2. No. 1.

[34] J. Scourias and T. Kunz. "An Activity-based Mobility Model and Location Management Simulation Framework". MSWiM. Seattle. WA. 1999.

[35] S. Seshan. H. Balakrishnan. R. Katz. "Handoffs in Cellular Wireless Networks: The Daedalus Implementation and Experience". Kluwer Journal on Wireless Networks. 1995.

[36] A. Snoeren. H. Balakrishnan. "An End-to-End Approach to Host Mobility". Mobicom 2000. Boston. MA. August 2000.

[37] Tranplan Associates. "Waterloo Region Travel Survey 1987: An Overview of the Survey Findings". Regional Municipality of Waterloo. Department of Planning and Development. October 1989.

[38] J. Vatn. G. Maquire Jr. "The Effect of Using Co-Located Care-of-Addresses on Macro Handover Latency". In 14th Nordic Teletraffic Seminar. August 1998.

[39] B. Waxman. Routing of Multipoint Connections. IEEE J. Select. Areas Commun.. December 1988.

[40] P. Yalagandula, A. Garg, M. Dahlin, L. Alvisi, H. Vin, "Transparent Mobility with Minimal Infrastructure", http: citeseer.nj.nec.com 462506.html