Mobile IP Extensions for Multi-Hop Wireless Networks

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Master of Science

by

Wilmer Caripe

Thayer School of Engineering

Dartmouth College

Hanover, New Hampshire

JUNE 1998

Mobile IP Extensions for Multi-Hop Wireless Networks

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Master of Science

by

Wilmer Caripe

Thayer School of Engineering

Dartmouth College

Hanover, New Hampshire

JUNE 1998

Examining Committee:

George V. Cybenko (Chairman)

Edmond S. Cooley

Robert S. Gray

Dean of Graduate Studies

Wilmer Caripe

Behold the welcome, long-awaited spring,

which brings back pleasure

and crimson flowers adorn the fields.

The sun brings peace to all around.

Away with sadness!

Summer returns,

and now departs cruel winter.

From: Carmina Burana, Carl Orff - 1937

A mi familia

Abstract

This thesis presents some extensions to Mobile IP designed to better adapt this protocol to wireless networks. Specifically, the foreign-agent selection process is studied in depth and several selection criteria are proposed and implemented as extensions to an existing Mobile IP software package.

Wireless computer networks bring many advantages to those environments where network connectivity is required subject to constraints such as flexibility, affordability, and scalability. Wireless Local Area Networks (LANs) can be quickly installed in cases where traditional networks are not a feasible option for reasons like deployment time, cost, etc. Mobile IP is a protocol that provides support for *roaming* of computers between different networks. Each Mobile IP-enabled computer (*mobile node*) has a fixed *home agent* on its original network that keeps track of the mobile node's location at all times. The protocol specifies several different mechanisms for the mobile node to keep its home agent updated about its location every time it moves to another network. One of those mechanisms requires the mobile node to select a temporary gateway computer (*foreign agent*) at the visited network; this computer will serve as the mobile node's default router to the rest of the network and will forward location updates to the mobile node's home agent. In a wireless networking environment, mobile nodes will likely encounter situations in which there are several candidates from which to select a foreign agent. However, the Mobile IP protocol does not specify policies to use in such cases. The extensions developed in this thesis approach this problem. Performance measurements of Mobile IP-enabled wireless LANs using these extensions are also presented and analyzed. In terms of throughput for TCP bulk data transfer, compared to the throughput achieved from using the original Mobile IP implementation, these Mobile IP extensions led to a performance improvement of up to 20% under certain scenarios.

Acknowledgements

I would like to thank Professor George V. Cybenko for his outstanding support from the very first time we met, and for creating such an encouraging atmosphere within our research group. To Professor Edmond S. Cooley, thanks for his collaboration and wise advise in shaping up my program at Thayer School and for graciously accepting to be part of my thesis committee. To Bob Gray, for being so unconditionally helpful, always having the answers to my questions. To Brian Brewington, who has been a great source of knowledge about the American culture through thought-provoking conversations, puns and proverbs.

To Katsuhiro Moizumi, Melanie Blanchard and Vladimir Ristanovic, this space is just not enough to express my gratitude for having made such wonderful friends, who undertook the burden of bearing with me these past two years, and made the harsh winters in Hanover feel a lot warmer to me.

I would also like to thank the Fulbright Foundation for giving me the opportunity to attend graduate school at Dartmouth: it has proved to be an amazing and horizon-broadening experience, a turning point in my life.

To all those people I have met at Dartmouth, thanks for providing me with wonderful memories of my stay here. To my good old friends, thanks for always being with me regardless of time and distance, I love you all.

Table of Contents

List of Tables

List of Illustrations

" We are what we think. All that we are arises with our thoughts. With our thoughts, we make the world" Buddha

1. Introduction

The Mobile IP protocol specifies enhancements to the IP protocol that allow transparent routing of IP datagrams to mobile nodes in a computer internetwork. Each mobile node is identified by its home address, regardless of its point of attachment to the internetwork. Whenever a mobile computer is disconnected from its original network (*home network*) and is reconnected later to a different network, it notifies its *home agent* - located in its home network - about its new location (or *care-of address*). The home agent intercepts all packets that are addressed to the mobile node and forwards them to the mobile node's current care-of address, allowing ongoing network connections on the mobile node to remain operative regardless of changes in the mobile node's link layer connectivity. This notification process is called *registration* in the Mobile IP terminology and can be performed either directly or indirectly. In the direct approach, the mobile computer acquires its own temporary IP address (or *colocated care-of address*) at the visited network using mechanisms such as the Dynamic Host Configuration Protocol (DHCP). In the indirect approach, the mobile node registers its new location with its home agent through a *foreign agent*. Foreign agents are computers on the visited network that provide IP mobility services to visiting mobile nodes, relaying registration requests and replies to the corresponding home agents. If it is unable to acquire a colocated address, the mobile node might be able to use one of the foreign agent's IP addresses as its new care-of address, in which case the home agent will forward packets intercepted at the home network to the foreign agent using encapsulation methods. The foreign agent will then deliver packets to the mobile node using its link-layer address, since it does not have an address of its own while visiting that network [15].

Figure 1 shows an example of a wireless network where Mobile IP is deployed. In the figure, the mobile computer - called *mobile node* in the Mobile IP terminology - has moved out of range from its original network (maybe because it is within a car or some other moving vehicle). When the mobile node starts receiving beacons (*agent advertisements*) from Mobile IP-enabled routers (*foreign agents*) and stops receiving advertisements from its home agent - located in its home network - it realizes that it has moved to another subnetwork (a foreign network). At that point it chooses one of the detected foreign agents and proceeds with the registration process to notify its home agent about its current location. Once the registration process is completed successfully, any network connections that the mobile node had already established before it moved away from its home network are transparently resumed.

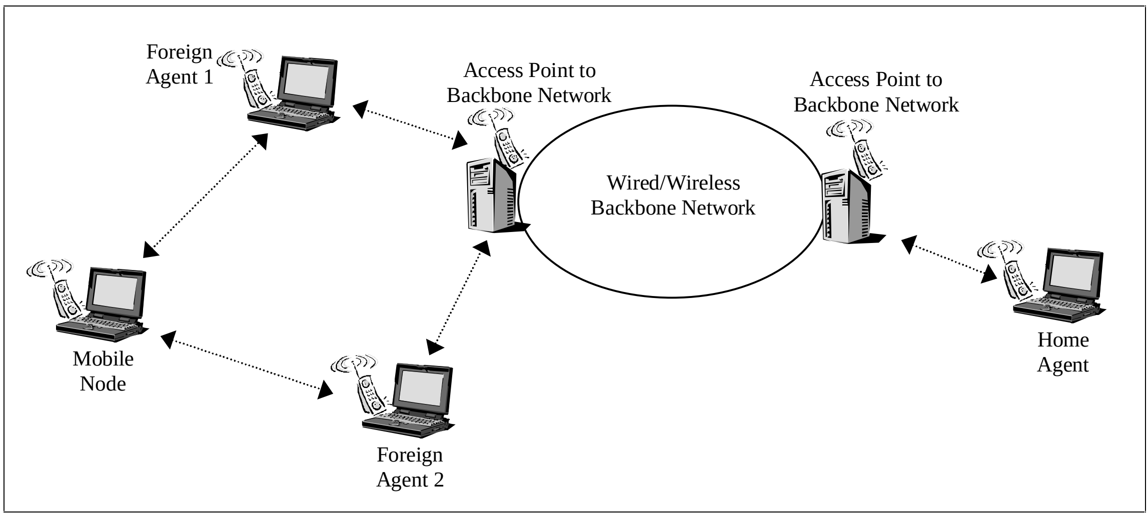

Figure 1 Mobile IP on multi-hop hybrid networks

This thesis focuses specifically on the foreign-agent selection problem on Mobile IP-enabled, multi-hop wireless networks, like the one depicted in Figure 1. In this kind of scenarios, mobile nodes visiting a foreign network need to use care-of addresses provided by foreign agents.

The Mobile IP base protocol does not specify any policies or mechanisms for those cases when a mobile node away from its home network detects multiple foreign agents, like in the example discussed above. However, there are several scenarios where this choice might affect the performance of distributed applications executing on the mobile node. For example, in Figure 1, *Foreign Agent 1* might be located in an area with a very noisy transmission channel, which causes incoming and outgoing packets to be damaged in the transmission channel. This will cause link layer, and probably TCP layer retransmissions, which degrade the network performance considerably. If, on the other hand, *Foreign Agent 2* is in an area with a clear, strong signal and a low background noise level, this agent should be the best option for the mobile node to choose. This example highlights the main motivation for this thesis. By being aware of some characteristics of the foreign agents and their environments, the mobile node should be able to intelligently choose which foreign agent should be its mobility and routing service provider while visiting the foreign network.

Although Mobile IP was not designed for highly dynamic environments, the mobility support it provides is a good starting point for implementing some ideas related that might then be applied to new protocols or frameworks in wireless computer networking such as *Ad-hoc* networking [12].

Several mechanisms that have been used in the design of server selection systems are studied (section 3.2) and some of them are applied in the design and implementation of foreign-agent selection extensions to the Mobile IP protocol. Similarly, cell-switching strategies in Mobile IP networks presented in related work are studied (section 3.3) and some of the fundamental ideas behind them are also applied in the design of such extensions.

In summary, the main questions addressed in this thesis are:

- What criteria are appropriate to use for foreign-agent selection purposes?

  The following selection criteria were designed (Chapter 4) and implemented as extensions to the Mobile IP protocol (Chapter 5):

  - ✓ Foreign agent's advertisement rate
  - ✓ Number of visiting nodes being attended by each candidate foreign agent
  - ✓ Latency on the link between the mobile node and the candidate foreign agent
  - ✓ Signal strength level at the candidate foreign agent's site
  - ✓ Variation in signal strength at the candidate foreign agent's site
  - ✓ Signal-to-Noise Ratio at the candidate foreign agent's site

- Does a systematic foreign-agent selection process improve the performance of distributed applications running on a mobile node, compared to a random selection?

  In Chapters 6 and 7 some scenarios are presented for which the performance achieved from using a systematic selection process (in terms of throughput) is at least as good as the performance achieved from using an implementation of the original Mobile IP protocol.

- Do different selection criteria affect performance in different ways?

  Performance results presented in Chapter 6 indicate that the selection criterion used affects performance (in terms of throughput) in different ways for different scenarios. For example, using the latency criterion can yield better performance than using Signal-to-Noise Ratio for some scenarios, but in some other cases, using signal strength may yield better performance.

The rest of this report is divided as follows: Chapter 2 briefly introduces the proposed approach to answering these questions. Chapter 3 presents background information and related work that are referenced throughput the report; Chapter 4 describes the design of the proposed extensions to the Mobile IP protocol for foreign-agent selection while Chapter 5 presents the details involved in the implementation of those extensions. Finally, Chapters 6 and 7 present the experiments done for measuring performance of the extended Mobile IP software implementation along with a discussion of the results.

"...God hath made man upright; but they have sought out many inventions" Old Testament

2. Proposed Solution

An experimental approach was used in order to answer the questions mentioned in Chapter 1. The Mobile IP base protocol was extended to add more intelligence to the foreign-agent selection made by mobile nodes. Specifically, the following "selection criteria" for choosing a foreign agent were implemented in the new extensions:

✓ Advertisement rate

The rate at which advertisements from a foreign agent are received by a mobile node reflect several characteristics that might affect the performance of those mobile nodes registering with that foreign agent: workload at the foreign agent, link quality and link latency, for example.

In this scheme, a mobile node bases its choice on the number of advertisements received from the detected foreign agents within a given period of time. That foreign agent from which more advertisements are received is selected.

✓ Number of mobile nodes in the foreign agent's visitor list

This criterion assesses workload generated by Mobile IP at each foreign agent. A foreign agent's *visitor list* contains those mobile nodes for which the agent is providing mobility services while they are visiting the respective foreign network.

The key assumption to justify this criterion is that the longer the foreign agent's visitor list is, the more processing and communication overhead the agent will be suffering, and thus, the lower the performance achieved by the mobile nodes using its services. Accordingly, the mobile node should select the foreign agent attending the smallest number of visiting mobile nodes.

✓ Link latency between foreign agent and mobile node

Several factors affect link latency: physical data transmission, communication protocols, efficiency of software implementations, and network congestion. It is a determining factor in the performance of distributed applications: the higher the latency in a network link, the lower the throughput achieved by distributed applications using that link. Therefore, in this scheme the mobile node selects the foreign agent with the smallest estimated round-trip latency.

✓ Signal strength level in the foreign agent's communication channel

Signal strength at a node location indicates how easy it is for the wireless network transceiver to distinguish a true signal from background noise. If the signal is weak, the receiver might not be able to detect an incoming link-layer packet. Packets would get lost, causing link layer retransmissions at the source computer. This process degrades link performance, and therefore the performance of networked applications running on the computers using that link.

In this scheme, mobile nodes obtain signal strength information from detected foreign agents, and opt for the agent with the strongest signal.

✓ Variation in signal strength in the foreign agent's communication channel

Changes in signal strength give some indication about how the wireless transmission channel behaves over time, from the foreign agent's point of view. Positive changes might indicate that the mobile computer is moving into an area with a stronger signal level. Conversely, negative changes in signal strength might indicate that the computer is moving into an area with high levels of interference (or away from the foreign agent).

In this scheme, a mobile node in a foreign network obtains the variation information from detected foreign agents, and selects the agent with the highest positive variation in signal strength.

✓ Signal-to-Noise Ratio in the foreign agent's communication channel

The signal-to-noise ratio (SNR) is the difference in power at the receiver between a true signal and background noise. The higher the signal-to-noise level the higher the possibility of detecting a valid signal and, put in context, the higher the probability of detecting link-layer packets.

SNR is therefore a good indication of the quality of a foreign agent's communication channel. In this scheme, a mobile node selects the foreign agent that presents the highest SNR level among those recently detected foreign agents.

The last three criteria describe the wireless communication channel at the foreign agent's site. As it will be seen in Chapter 4, they are very similar in what they describe, but each of them provides very useful information. For example, the Signal-to-Noise Ratio cannot be used on its own to infer information about the signal strength level, since it is also related to the background noise in the channel. Variation in SNR could have been studied, since it provides very useful information about scenarios with moderate mobility, as variation in signal strength does. However, it was not included in this thesis for several reasons. The number of selection criteria was kept small in order to focus on the study of performance issues. On the other hand, this thesis established the framework that will allow writing new extensions for other selection criteria in a straightforward manner: the design of an extension that implements the variation in SNR will be very similar to the design of the extension for variation in signal strength.

These criteria were implemented as extensions to the Mobile IP protocol. The performance effect achieved by using these criteria was measured by generating simulated traffic flows and recording the data throughput at mobile nodes running the extended Mobile IP implementation. The design and implementation of these extensions and the performance measurement results are presented in the next chapters.

"Imagination is more important than knowledge. Knowledge is limited.
Imagination encircles the world" Albert Einstein

3. Background Information

3.1. Mobile IP

Mobile IP defines extensions to the original Internet Protocol (IP) that allow computers to transparently move from one point of attachment to the network to another without disrupting established network connections.

3.1.1. Functional entities:

The following entities are defined by the Mobile IP base protocol:

- *Mobile Node* (MN)

  A host or router that changes its point of attachment from one network to another.

- *Home Agent* (HA)

  A router on a MN's home network that forwards datagrams to the MN when it is *away from home*.

- *Foreign Agent* (FA)

  A router on a MN's visited network that provides routing services to the MN while registered on that network.

3.1.2. Protocol Overview

The following outline of the Mobile IP protocol is taken from [16]:

1. Mobility agents (that is, foreign agents and home agents) advertise their presence via agent advertisement messages. A mobile node may optionally solicit an agent advertisement message from any local mobility agents

by using an agent solicitation message.

2. A mobile node receives an agent advertisement and determines whether it is located on its home network or on a foreign network.
3. When the mobile node detects that it is located on its home network, it operates without mobility services. If returning to its home network after being registered elsewhere, the mobile node deregisters with its home agent through a variation of the normal registration procedure.
4. When a mobile node detects that it has moved to a foreign network, it obtains a care-of address on the foreign network. The care-of address can either be a foreign agent's care-of address or a colocated care-of address.
5. The mobile node, operating away from home, registers its new care-of address with its home agent through the exchange of a registration request and registration reply message, possibly by way of a foreign agent.
6. Datagrams sent to the mobile node's home address are intercepted by its home agent, forwarded by the home agent to the mobile node's care-of address (either at a foreign agent or at the mobile node itself), and finally delivered to the mobile node.
7. In the reverse direction, datagrams sent by the mobile node may be delivered to their destination using standard IP routing mechanisms, without necessarily passing through the home agent.

The next subsections briefly describe the most important aspects of the Mobile IP protocol specification, most of them already mentioned in the previous outline.

3.1.3. Agent Discovery Mechanism

The Mobile IP agent discovery mechanism is based on the Internet Control Message Protocol (ICMP) router discovery mechanism described in RFC 1256 [6]. It was devised to allow computers connected to broadcast or multicast networks to discover the IP addresses of local routers. The mechanism uses two ICMP messages: *Router Solicitation* and *Router Advertisement*. Whenever a host starts up, it multicasts a router solicitation to ask for immediate advertisements. Conversely, routers multicast advertisements periodically on their local links.

In the agent discovery mechanism, two messages are defined: *Agent Advertisement* and *Agent Solicitation*.

A mobility agent transmits agent advertisements to announce its services on a link. An agent advertisement is an ICMP router advertisement extended to carry a mobility agent advertisement extension. Both the router advertisement message format and the agent advertisement extension are depicted in Figure 2. As shown in the figure, the agent-advertisement extension contains information about the service provided by the mobility agent. Mobile nodes use these advertisements to determine their current point of attachment to the Internet.

Mobile nodes transmit agent solicitations only in the absence of agent advertisements and when a care-of address has not been determined through a link layer protocol or other means [16]. An agent solicitation is basically an ICMP router solicitation. In theory, a mobile node can continue to send out solicitations until a suitable foreign agent is detected. The node may send three initial solicitations at a maximum rate of one per second and follow a binary exponential back-off mechanism afterwards in order to reduce the solicitation rate and limit the overhead on the local link.

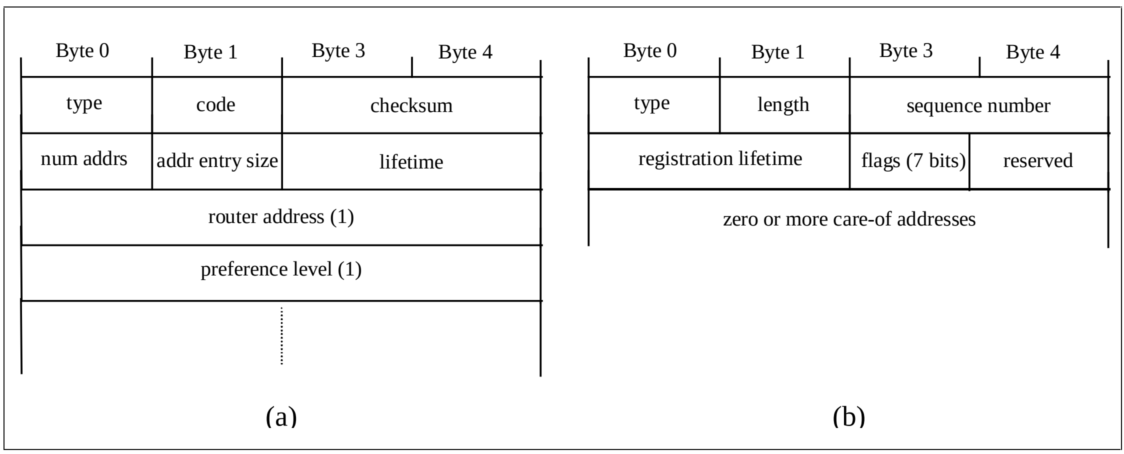

Figure 2 (a) ICMP router advertisement, (b) Mobility agent advertisement extension

A mobile node can distinguish an agent advertisement from other ICMP router advertisement messages by checking the IP *total length* field (in bytes) and the total number of advertised addresses. If the IP length field indicates that the message is actually longer than it should be, the rest of the bytes are considered as one or more extensions [16].

3.1.4. Registration Mechanism

Registration creates or modifies a *mobility binding* at the home agent, associating a mobile node's home address (its fixed address on its home network) with its current care-of address for a certain length of time called the *registration lifetime* [16]. Registration messages can also be used by mobile nodes to renew a mobility binding that is due to expire or to deregister when they return home.

The following overview of the registration process was extracted from [16]:

A mobile node visiting a foreign network can register using two different procedures: one by means of a foreign agent that relays the registration to the home agent and one without any intermediary. There exist certain rules as to which procedure should be used in a given case:

- If a mobile node is using a foreign agent's care-of address, it must register through that foreign agent.
- Mobility agents may specify in their advertisements that mobile nodes must register via a foreign agent.
- When returning to its home network, a mobile node should address the registration directly to its home agent.
- If a mobile agent acquires a colocated care-of address in a foreign network, it may send the registration message directly to its home agent.

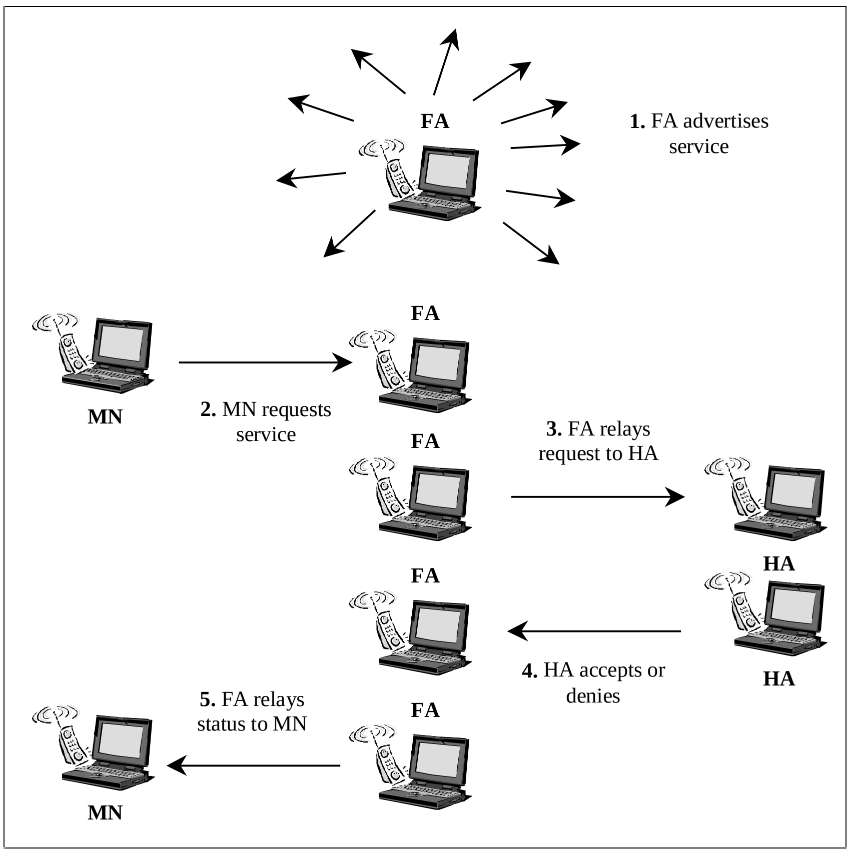

Figure 3 Mobile IP registration overview

This report focuses on the first registration procedure: registering by way of a foreign agent. As illustrated in Figure 3, registration in this case requires four messages:

- The mobile node (MN) sends a registration request to the prospective foreign agent to begin the registration process.

- The foreign agent (FA) processes the registration request and then relays it to the home agent, whose address is provided by the mobile node in the registration request. The foreign agent might reject the request due to authentication failure or incorrectly formed messages.
- The home agent (HA) sends a registration reply to the foreign agent to grant or deny the request.
- The foreign agent processes the registration reply and then relays it to the mobile node to inform it of the disposition of its request.

Mobile IP registration uses the User Datagram Protocol (UDP) to deliver request and reply messages. Registration messages contain a lifetime field that indicates the amount of time (in seconds) for which the registration information should be considered valid. A zero lifetime indicates that the mobile node has been de-registered.

3.1.5. Related Networking Mechanisms

The following subsections present an overview of some of the mechanisms and protocols supporting the operation of Mobile IP. Most of the contents presented here are taken from [16].

3.1.5.1. Tunneling or Encapsulation

Tunneling alters the normal IP routing from source to destination computers for datagrams by delivering them to an intermediate destination (usually encapsulated within a new IP packet). Once the encapsulated datagram arrives at this intermediate destination, it is de-capsulated, yielding the original IP datagram, which is then delivered to the destination, indicated by the original destination address field. The encapsulator and decapsulator are considered the endpoints of a tunnel.

In the Mobile IP protocol, home agents use tunneling to redirect intercepted packets addressed to a mobile node that is registered on a foreign network. Whenever a mobile node registers a care-of address with its home agent, a tunnel is established either between the home agent and the mobile node (if the latter acquired a colocated care-of address), or between the home agent and the foreign agent relaying the registration on behalf of the mobile node.

3.1.5.2. IP-in-IP Encapsulation

This method allows an IP datagram to be encapsulated within another IP datagram. The outer IP header source and destination addresses identify the endpoints of the tunnel.

Before encapsulating a datagram, the time-to-live (TTL) field in the inner IP header is decreased by one if the tunneling is being done as part of forwarding the datagram; otherwise, the inner header TTL is not changed during encapsulation. If the resulting TTL in the inner IP header is 0, the datagram is discarded. After de-capsulation, the TTL in the inner datagram is not changed. If the inner datagram has a TTL of 0, the datagram is discarded. If the decapsulator forwards the inner datagram through one of its network interfaces, it will decrease the TTL as a result of doing normal IP forwarding.

Mobile IP requires mobility agents to support tunneling datagrams using IP-in-IP encapsulation. Those mobile nodes using a colocated care-of address also must be able to receive datagrams tunneled using this encapsulation mechanism.

3.1.5.3. Minimal Encapsulation

IP-in-IP encapsulation requires the duplication of several fields in the inner IP header. Minimal encapsulation allows enclosing an IP datagram within another IP datagram with less overhead. Mobile IP defines this as an optional encapsulation mechanism.

To encapsulate an IP datagram using minimal encapsulation, a minimal forwarding header is inserted into the datagram. This mechanism cannot be used when the IP datagram to be encapsulated is a fragment of a larger datagram, since there is no room in the minimal forwarding header to store fragmentation information. The encapsulator copies the source and destination addresses from the original IP header into the minimal forwarding header. The destination address field in the IP header is replaced by the IP address of the tunnel endpoint. Finally, the length and checksum fields in the IP header are re-computed accordingly. The de-capsulation process reverses these actions and restores the original IP header for final datagram delivery.

3.1.5.4. ARP, Proxy ARP and Gratuitous ARP

The Address Resolution Protocol (ARP) is devised to resolve a target node’s link-layer address from its IP address [17]. Although that is the main use of the protocol, other uses have been devised over the years. The Mobile IP protocol uses some of these variations:

- A proxy ARP is an ARP reply sent by one node on behalf of another node that is unable to answer its own ARP requests. The sender of the ARP reply usually includes its own link-layer address in the sender link-layer address field. Thus, the node receiving the reply will associate this link-layer address with the IP address of the original target node.

- A gratuitous ARP is an ARP packet sent by a node to update other node's ARP caches. The original ARP protocol requires any node receiving any ARP packet (request or reply) to update its local ARP cache with the sender protocol and link-layer addresses in the ARP packet if the receiving node has an entry for that IP address already in its ARP cache [17].

While a mobile node is registered on a foreign network, its home agent uses proxy ARP to reply to ARP requests it receives on the home network that seek the mobile node's link-layer address. When a mobile node leaves its home network and registers a binding on a foreign network, its home agent uses gratuitous ARP to update the ARP caches of other nodes on the home network. In this way, such nodes associate the link-layer address of the home agent with the mobile node's IP address, causing all packets addressed to the mobile node to be intercepted by the home agent and tunneled to the mobile node's current location.

When a mobile node returns to its home network, all the nodes on this network must once again update their mapping of the mobile node's home IP address to its authentic link-layer address. The mobile node is required to broadcast a gratuitous ARP message on its home network before sending a de-registration request message to its home agent. Once the home agent receives and accepts this de-registration request, it is also required to broadcast a gratuitous ARP in the mobile node's home network, with the correct binding between the mobile node's IP address and its authentic link-layer address. In a wireless environment, the area within transmission range of the mobile node will likely differ from that within range of its home agent. Therefore, a gratuitous ARP has to be transmitted by both the mobile node and its home agent.

3.1.6. Available Mobile IP Implementations

While looking for existing implementations of the Mobile IP protocol, the search was focused on those software packages for which the source code was available; this would allow implementing the new extensions on top of a working software package. The Mobile IP implementation from the State University of New York at Binghamton was selected to implement the proposed extensions; the details concerning the evaluation process are presented in Chapter 5. Following is a list of the software packages that were evaluated.

- Monarch project's Mobile IP for FreeBSD from CMU

This implementation fully conforms to the IETF standard Mobile IP protocol for IPv4, specified in RFC 2002, and includes support for both IP-in-IP and minimal encapsulation. The current version of the software is release 1.1.0 and supports NetBSD and FreeBSD. It requires patches to the OS kernel [4].

- Portland State University's Mobile IP for FreeBSD

  The latest release of this package includes a FreeBSD Mobile-IP system with several security features. It does not support the colocated care-of address mode for mobile nodes. It is mostly implemented in user space with some minor kernel modifications [18].

- National University of Singapore's Mobile IP for Linux

  The latest beta version (v2.0) of this implementation of IETF Mobile IPv4 supports Linux kernel version 2.0.24. It requires patches to the Linux kernel. This version complies with most features defined in RFC 2002 [13].

- State University of New York at Binghamton's Linux Mobile IP

  Version 1.00 is mostly compliant with revision 16 of the IETF Mobile-IP draft. It runs entirely in user space. Consequently, it does not handle the ARP mechanisms properly in some cases [20].

- MosquitoNet Mobile IP implementation for Linux from Stanford University

  This implementation includes patches to the Linux kernel, a home agent implementation and a mobile node implementation. It assumes that a mobile node running the software will be able to acquire a co-located care-of address [22].

- Mobile IP implementation for Linux from the HP Labs at Bristol, UK

  This implementation is mostly compliant with RFC 2002. It requires registrations to be routed through foreign agents. All the implementation runs in user space, which makes mobile nodes unable to answer foreign agents' ARP requests. In order to go around this problem, mobile nodes periodically issue an ARP reply addressed to the FA. This causes the corresponding foreign agent to update its ARP entry associated with that mobile node without issuing any ARP requests [8].

3.2. Server Selection Systems

There is extensive ongoing research on server selection systems for client/server applications. The mechanisms used vary in the degree of transparency they provide and the protocol level at which they are implemented (network level, application level, etc.). Most of the principles on which those systems are based can be applied to the foreign-agent selection problem. Following is a brief description of some server selection systems and the mechanisms they use.

3.2.1. Host Anycasting Service for IPv6

The current version of the Internet Protocol (IP), which supports most of the computers connected to the Internet nowadays, is defined in RFC 791 [19]. This IP version is usually referred to as IPv4. A large group of researchers have been working for years in a new version of the protocol called IPv6 – defined in RFC 1883. The design of this protocol approaches several shortcomings found in IPv4. IPv6 provides bigger address space, reduced administrative overhead, support for address renumbering, improved header processing and a reasonable security infrastructure, among other advantages over IPv4 [16].

Anycast is an IPv6 service designed to provide transparent network support for delivering packets to any computer within a group of hosts sharing the same IP *anycast* address. A host transmits a datagram to an *anycast* address and the internetwork is responsible for providing best effort delivery of the datagram to at least one, and preferably only one, of the servers that accept datagrams for that *anycast* address. The server selection in this case is based on the number of hops: routers forward the *anycast* datagram until it reaches a host supporting the destination *anycast* address [14].

3.2.2. Cisco Distributed Director

This commercial product uses existing routing tables in the network infrastructure to redirect end-user service requests to the closest server as determined by client-to-server topological proximity and/or client-to-server link latency (round-trip times). Director Response Protocol (DRP) servers collect link latency information themselves and consult routing table metrics from neighboring routers supporting either the Border Gateway Protocol (BGP) or the Interior Gateway Protocol (IGP) [5].

3.2.3. The Harvest Information Discovery and Access System

Harvest is a system that provides a scalable, hierarchical architecture for gathering, indexing, caching, replicating and accessing Internet information such as HTTP or FTP objects. The system includes a hierarchical object cache. Every time the cache requires an object, it queries its neighbors and parent in the cache hierarchy, and additionally sends an ICMP echo request packet to the original server containing the object. The cache retrieves the object from whichever host replies first to its request. It can cache Gopher, FTP, HTTP objects and recent DNS name-to-address maps [2].

3.3. Cell Switching Mechanisms in Mobile IP

When a mobile node realizes that it has moved from one subnetwork to another, it should register its new location with its home agent. This registration process, described in previous sections, is also referred to as *Cell Switching* or handoff [3][15][16].

Mobile IP provides three mechanisms for mobile nodes to trigger cell switching (move detection). However, the protocol allows for alternative mechanisms to be used for this purpose. The primary cell switching mechanisms are briefly described next.

3.3.1. Lazy Cell Switching (LCS)

This method of movement detection is based on the lifetime field within the main body of the ICMP Router Advertisement portion of the agent advertisement. If a mobile node fails to receive another agent advertisement from its current foreign agent (or home agent) within the specified lifetime, then the mobile node should assume it has moved out of range from that mobility agent and proceed to register with another agent.

3.3.2. Prefix Matching

A mobile node may use the prefix-length extension attached to agent advertisements to determine whether newly received advertisements come from the subnetwork corresponding to its current care-of address [16]. This prefix length extension contains the network prefix corresponding to the network where the advertising agent is providing mobility services. If the comparison fails, the mobile node may then assume that it has moved to another subnetwork and proceed to register with a new mobility agent.

3.3.3. Eager Cell Switching (ECS)

This mechanism assumes that mobile nodes follow steady trajectories while they move across a wireless network. Given this assumption, it is likely that once a mobile node enters a new wireless cell (defined by a foreign agent's coverage area), it will continue further into this cell and farther from its previous cell. In this scheme, the mobile node switches immediately to the new cell by registering with a newly discovered foreign agent. The protocol makes provisions to avoid switching back and forth among cells when the mobile node is within range of two or more foreign agents and is receiving advertisements from all of them [16].

3.3.4. Other Cell Switching Mechanisms

Recent work has focused on applying cell-switching techniques commonly used in Cellular Telephony to Mobile IP networks. In [3], several cell-switching mechanisms are proposed for Mobile IP networks, and their performance studied. The mechanisms are described next.

- Late Cell Switching (Late)

  This mechanism is similar to the LCS scheme described in the previous sub-section. A mobile node decides to switch cells whenever its communication link with the mobility agent is lost.

- Early Cell Switching (Early)

  This mechanism is similar to the ECS scheme proposed by the Mobile IP protocol. In this case, a mobile node switches to a new cell as soon as it receives an advertisement from a new mobility agent.

- Strong Cell Switching (Strong)

  This mechanism assumes that mobile nodes have the tools to obtain link-layer information (the same assumption applies for the mechanisms presented in the rest of this subsection). When using this scheme, a mobile node switches cells whenever the signal strength of the new cell - measured from packets received from newly discovered mobility agents - becomes stronger than the signal strength of the current cell.

- Late Cell Switching with static hysteresis (Late-SH)

  Using this mechanism, a mobile node waits until the signal strength on the new cell exceeds the signal strength on the old cell by a certain threshold, in order to switch to the new cell. The *hysteresis region* (the region between the two boundaries defined by the signal strength threshold) needs to be large enough to prevent excessive switching due to oscillating movement of the mobile node or fading of the radio signal. However, those boundaries should also be far enough from the outer extent of each cell so that reliable communication is available at each boundary [3]. The mechanism is static in the sense that the hysteresis region is preset, determined by the threshold on the difference in signal strength between the two cells.

- Strong Cell Switching with hysteresis

In this scheme, a mobile node determines when to switch cells by using three signal strength lines in the overlapping area of two wireless cells. The two boundary lines are the same as explained above for Late-SH. The other line, the middle line, represents a line of equal strength. There are two variations to this mechanism, Strong-SH, in which the hysteresis region is preset; and Strong-DH, in which the hysteresis region expands or shrinks dynamically according to the mobile node's movement pattern.

The mechanisms just presented can be divided in two groups: non-hysteresis techniques (Late, Early and Strong) and hysteresis techniques (Late-SH, Strong-DH, Strong-SH). Simulation results presented in [3], which study performance of these techniques for different movement patterns, show that Late and Late-SH have the minimal switching overhead in each category, and that overall, Late has the minimal switching overhead. Results also show that hysteresis techniques are preferable than non-hysteresis techniques when average signal strength is a primary concern. Specifically, Strong-DH is the preferred technique in this case.

"You need chaos in your soul to give birth to a dancing star"
Nietzsche

4. Design of the Solution

As described in Chapter 2, the main goal of this thesis was to add intelligence to the foreign-agent selection process performed by a mobile node in scenarios like that presented in Figure 1. Specifically, the solution aims to provide enough information to the mobile node so that it is able to choose, in a systematic way, the most appropriate foreign agent when it moves into a new foreign network. Different selection criteria were devised and they are explained in detail in the next subsections.

For some of these selection criteria, mobile nodes are able to collect all the necessary data locally without the foreign agent sending additional information. For other criteria, mobile nodes require information from the neighboring foreign agents to perform the selection process. In this case, some protocol extensions have been defined using a mechanism for protocol extensibility defined in the Mobile IP specification [15]. The extension mechanism allows additional information to be included in Mobile IP control messages or in ICMP router discovery messages, using byte packages with the format depicted in Figure 4, which is usually given the name of type-length-value (TLV) format. Agent advertisements follow the TLV format, to extend ICMP router advertisements.

1 byte	1 byte	*n* bytes
type	length (*n*)	data…

Figure 4 Type-Length-Value format for Mobile IP extensibility

Regardless of what criterion is used in the selection of a foreign agent, the process follows the algorithm presented in Figure 5. Since it is unfeasible to receive two or more advertisements exactly at the same time, because network drivers serialize packets delivered to upper protocol layers, the issue of whether to wait for advertisements from two or more foreign agents before invoking elections came up. If this were the case, there might be periods when network connections stall. For example, suppose that a mobile node has just moved to a foreign network and has not registered with any foreign agent yet. Although it receives advertisements from a foreign agent and is ready to register with it, it would be required to wait for advertisements from another agent to start the selection process, while the ongoing network connections on the mobile node are temporarily stalled. Moreover, if there are no other foreign agents on that foreign network, those network connections could be permanently stalled and would eventually time out and fail. Note that if the mobile node is already registered with a foreign agent on that network, this temporary interruption in connectivity will not occur, regardless of how long the selection process lasts, as long as connectivity with the current foreign agent is not lost.

Therefore, it was decided that a mobile node not registered with any foreign agent registers with the first mobility agent from which it receives an agent advertisement. Afterwards, when a mobile node receives advertisements from a new mobility agent (other than its current foreign agent), it will compare the new agent and its current foreign agent to decide which is more suitable to select as its new foreign agent, based on one of the discussed criteria.

```
    When in a foreign network, listen for foreign agent advertisements (done periodically according
to the Mobile IP specification). At the end of each advertisement period, do:
    1. If already registered with a foreign agent
       1.1. If an advertisement from that foreign agent was received in this period → Go to 1
       1.2. else
          1.2.1. Collect data from foreign agents detected (if any)
          1.2.2. If  it is time for performing elections (depends on selection criterion )
                1.2.2.1. Select new foreign agent based on a particular criterion
                1.2.2.2. If a new foreign agent was selected - other that the current one, register with it
                         (Cell Switching).
       1.3. Go to 1
    2. else          /* Not registered yet with any foreign agent */
       2.1. Register with the first foreign agent from which an agent advertisement is received.
            Select randomly if several foreign agents are detected.
       2.2. Register with the selected foreign agent (Cell Switching)
    3. Go to 1
```

Figure 5 Algorithm for foreign-agent selection executed by mobile nodes

For some criteria, the selection process is not invoked every time a new foreign agent is detected. Instead, information is collected at the mobile node until the time the actual selection is made.

4.1. Selection Criteria

The next sub-sections present a detailed explanation of the design of each selection criterion. Some of them required the implementation of Mobile IP extensions, which are attached to agent advertisements by foreign agents so that mobile nodes choosing a new foreign agent decode them and use the transmitted information in the selection process. This scheme provides a clean, transparent solution to the problem of transmitting data from mobility agents to mobile nodes. Interoperability with mobile nodes is guaranteed as long as the extension type values are not reused, and mobile nodes silently discard those extensions with unknown type values. It is also an efficient approach, in the sense that very little network traffic overhead is generated for selection purposes, since information is piggybacked to agent advertisements that must be sent at regular intervals according to the Mobile IP specification [15].

4.1.1. Foreign Agent's Advertisement Rate

Under this criterion, a mobile node selects that foreign agent from which more advertisements are received within a certain period of time. Therefore, each mobile node is responsible for maintaining a list of recently discovered foreign agents (including its current foreign agent), keeping track of the number of advertisements received from each of them.

In this scheme, mobile nodes perform all the bookkeeping tasks. Foreign agents need not send any information to the mobile nodes for their use in the selection process. A mobile node entering a foreign network waits a given period of time, during which it keeps track of the number of advertisements it receives from different foreign agents. At the end of the period, the mobile node selects the foreign agent from which it received the largest number of advertisements during that period. If the number of advertisements received from this foreign agent is greater than the number of advertisements received from the mobile node's current foreign agent (within the same period of time), the new mobility agent is selected and the mobile node proceeds to register with it. Otherwise, the mobile node remains registered with its current foreign agent until the next period of time expires and the process described above is executed again.

An important issue that arises at this point is the period of time to wait for advertisements from mobility agents. There are several ideas that can be applied here:

- ✓ To statically define the waiting period as a given number of seconds

- ✓ To consider the waiting time as a random variable with a certain probability distribution
- ✓ To start the selection process after receiving a set number of advertisements from the mobile node's current foreign agent
- ✓ To start the selection process after receiving a set consecutive number of advertisements from a newly detected foreign agent

There are pros and cons to each of these ideas. Waiting a set number of seconds gives us a deterministic way to determine when to invoke elections, but its effectiveness depends on choosing an appropriate value for the time lapse. If it is too short, too much computation time will be spent in performing the selection process. On the other hand, if the time lapse is too long and the mobile node is going out of range from its current foreign agent, network connections will stall for some time before the mobile node selects a new foreign agent and switches cells. Modeling the time lapse as a random variable raises the question of how to choose the right probability distribution – which will depend on the particular scenario. Also, it would add an additional computational overhead to mobile nodes because of random number generation.

The last two options in the list above have only a small implementation cost. Since mobile nodes already keep track of these numbers, implementing either of these options would be straightforward. Under the third approach, if a mobile node is moving away from its current foreign agent, it might stall waiting for advertisements that it will never receive since it is out of the foreign agent's transmission range. Obviously, the same problem may occur if the fourth option is used. If it is assumed that mobile nodes follow steady trajectories however, the fact that a mobile node starts receiving agent advertisements from a new foreign agent implies that it is moving into a new cell and away from its current foreign agent [16]. This is an acceptable reason for the mobile node to start the selection process when it has received a given number of advertisements from a new foreign agent. An appropriate time-out scheme and an appropriate value for the number of received advertisements to wait for must be defined to guarantee the correct operation of the selection process.

The last alternative was chosen as the most appropriate for the solution presented here. Specifically, a mobile node executes the selection process immediately after receiving two consecutive advertisements from an agent different from its current foreign agent. If the mobile node is still within range of its current foreign agent, it is likely that it will receive advertisements from it during that same period of time, which will prevent the mobile node from selecting another foreign agent according to the algorithm presented in Figure 5. Alternatively, if the mobile node is within range of its current foreign agent, but some advertisements get lost because the communication channel around the foreign agent is highly congested and noisy, it is reasonable to switch to another agent.

4.1.2. Number of Mobile Nodes in the Foreign Agent's Visitor List

This criterion mandates mobile nodes to choose a foreign agent based on the number of mobile nodes for which each of the candidate foreign agents is providing mobility services. The rationale behind this scheme is that there is a certain overhead involved with providing mobility services for a mobile node, such as forwarding registration request, replies and renewals to and from the corresponding home agent. In addition, after the registration process is complete, the foreign agent must act as a default gateway to the rest of the network for the mobile node. Moreover, the foreign agent must de-capsulate and forward packets tunneled by the home agent for the mobile node. These tasks represent an operational overhead that decreases the quality of service that can be provided by a foreign agent that is attending several mobile nodes simultaneously.

The number of mobile nodes that a foreign agent is attending is sent to the mobile nodes in a new Mobile IP extension, which is depicted in Figure 6. This extension is attached to every advertisement sent by the foreign agent. Any extensions defined in the base protocol are included first in the agent advertisement [15], so that the number-of-visitors extension is the last one added to the advertisement message.

1 byte	1 byte	2 bytes
type=**20**	length=**2**	No. of visitors

Figure 6 Format of the number-of-visitors extension

The value for the type field (20) was chosen arbitrarily. It is the value immediately after the type number for the 'Pad Extension', which is the extension with the highest type value defined by the Mobile IP base protocol [15].

Regarding when mobile nodes execute the selection process, it was decided to use the same scheme selected for the previous criterion, that is, a mobile node will execute the selection process after receiving two consecutive advertisements from a foreign agent different from its current one. At this point, the mobile node selects the foreign agent with the minimal number of visiting nodes. If this number is smaller than the number of mobile nodes being attended by the current foreign agent by at least a certain threshold, the mobile node proceeds to switch cells. The value of this threshold should be at least 2 to avoid cases where the mobile node switches to a foreign agent with only one less visiting node than the node's current foreign agent. In this case, if the mobile node switched to that foreign agent, it would end up in a symmetric situation, and thus it would suffer cell-switching overhead without any performance benefits.

4.1.3. Link Latency between Mobile Node and Foreign Agent

This scheme involves selecting the detected foreign agent with minimal round-trip latency to the mobile node. Latency refers to network layer latency, that is, the time it takes for a byte to be received by the target machines' network layer from the time it is sent by the source machine's network layer entity. Round-trip latency adds to the latency described above the time it takes for a reply byte from the destination machine to reach the source machine's network layer entity. This measurement might reflect other factors that affect the total network throughput:

- ✓ Link congestion or noise. If a link is congested or noisy more transmitted packets get damaged on the communication channel and therefore more link-layer retransmissions are required, increasing the link latency.
- ✓ Workload at the foreign agent. If a foreign agent is attending many mobile nodes, it has to take care of decapsulating packets addressed to mobile nodes, relaying registration requests and replies, etc., which translates to more incoming traffic into its network interface(s) and therefore longer delays for the network layer to deliver packets.

Regarding the mechanism used to collect latency information, two ideas were studied: off-line polling, in which mobile nodes keep a list of foreign agents for which to calculate roundtrip latency at regular intervals. The other alternative studied was on-line polling, in this case, mobile nodes wait until they detect foreign agents and then calculate roundtrip latency for those specific agents.

The first method requires mobile nodes to know *a-priori* all the possible foreign agents in the network. While this mechanism allows the mobile node to have round-trip latency information readily available for use in the selection process, it is not scalable and restricts the Mobile IP network topology from being changed dynamically by adding new foreign agents. The second mechanism requires mobile nodes to wait for latency information to be available for the selection process, since polling is done *on-line*. The advantage is that less traffic overhead is generated because only those foreign agents that are detected are polled. Given that network bandwidth limitations and dynamic topology changes are common in the wireless scenarios where these new extensions will be used, the second alternative was selected.

In order to calculate round-trip latency with minimum traffic and processing overhead, a mechanism similar to that presented in section 3.2.3 was used: an ICMP ECHO REQUEST message is sent to the target foreign agent(s). It suffices to choose that foreign agent from which an ECHO REPLY is received first, since it is the agent with the minimal roundtrip latency to the mobile node. After sending the ECHO messages, the mobile node enters a *waiting state* until it receives an ECHO reply message from the polled computer, or until the maximum waiting period is reached, in which case the mobile node goes back to normal operation, following the algorithm depicted in Figure 5. Using this mechanism, there is no need to store and calculate roundtrip latencies for all the detected foreign agents, eliminating the need for keeping data structures and historical latency information.

The selection process is executed whenever a mobile agent realizes that it has lost connectivity with its current foreign agent. Since the *on-line* polling mechanism is being used, the amount of time the mobile node has to wait to obtain the latency information must be minimized. The following ideas were considered regarding when exactly a mobile node should start collecting latency information from detected foreign agents:

- ✓ Start polling immediately after a new foreign agent is detected, regardless of whether the mobile node is still within range of its current foreign agent.
- ✓ Keep track of detected foreign agents and start polling them whenever the mobile agent *realizes* that it has lost connectivity to its current foreign agent.
- ✓ Keep track of detected foreign agents and start polling them whenever the mobile agent *suspects* that it might have lost connectivity to its current foreign agent.

The first option might be the most appropriate from the point of view of minimizing the time the mobile node needs to wait for latency information before selecting a new foreign agent. However, if the mobile node spends a lot of time in an overlapping area without actually going out of range from its current foreign agent (e.g. if the density of foreign agents is high), this scheme would generate a large amount of traffic overhead. Here is why: since the mobile node is in an overlapping area, it will receive advertisements from several foreign agents simultaneously. According to this scheme, it will poll each of them immediately to obtain latency information. On the other hand, the mobile node will also receive advertisements from its current foreign agent, which will cause the mobile node to exit the waiting state and continue normal operation, following the algorithm in Figure 5, ignoring any ECHO replies it receives afterwards from previously polled foreign agents. This process will repeat for as long as the mobile node remains in the overlapping area.

The second option avoids the traffic overhead mentioned above but puts off the polling process until the moment when the mobile node must choose a new foreign agent (e.g. because its current registration expired), thus delaying the selection process. The third option was used instead, since it keeps traffic overhead low by initiating the polling process only when the mobile node *suspects* that it might be getting out of range from its current foreign agent.

Whenever an advertisement from the current foreign agent is missing, the mobile node starts polling those foreign agents it has recently discovered (including its current foreign agent) and enters a waiting state for replies. If the mobile node is already in the waiting state when the advertisement was missing, it assumes that the polling process is already taking place and ignores the event. If an advertisement from the current foreign agent is received while the mobile node is in waiting state, it goes back to normal operation, ignoring ECHO replies received from foreign agents previously polled. According to the Mobile IP base protocol, if the mobile node misses three consecutive advertisements from its current foreign agent, it should delete it from its list of valid foreign agents. In this agent is the current foreign agent for the mobile node, it should trigger a handoff or cell switch to another agent. If other foreign agents have been detected meanwhile, one of them is selected in an unspecified way, otherwise, the agent discovery process is initiated [15]. In the scheme presented here, missing three consecutive advertisements from the current foreign agent will indicate the mobile node to switch to whichever foreign agent replied first to the polling process previously initiated – if any. If no other foreign agents have been detected, the mobile node will proceed with the agent discovery process as specified by the Mobile IP protocol.

Note that if the current foreign agent replies to the polling first - in which case no cell switching is necessary - it is likely that the mobile node had previously received an advertisement from its foreign agent too. If by the time the mobile node starts selecting, no ECHO replies are received, it will remain in waiting state (but performing its tasks regularly) until one of the following events occurs:

(1) It receives an advertisement from its current foreign agent,

(2) It receives an ECHO REPLY from a foreign agent previously polled, or

(3) Its current registration expires and it starts executing the agent discovery mechanism as described in the Mobile IP protocol [15].

Another issue that came up during the design of this selection criterion was whether one roundtrip measurement provides accurate information about the latency between the mobile node and the prospective foreign agent. Several round-trip time measurement tests within a wireless LAN using the TCP/IP tool *ping* showed that the first measurement usually produces a higher round-trip time than subsequent measurements. The reason for this is an extra ARP request that the polling computer has to send over the wireless link to learn the peer computer's hardware address, in order to send the ICMP ECHO request packet directly over the local network link. The ARP request is sent after the polling computer looks up its routing table and realizes that the peer machine is located on the same subnetwork. In the context of a mobile node polling a foreign agent, this extra delay will not occur for the following reason: the mobile node polls those foreign agents from which it has recently received agent advertisements. When those advertisements are received, the wireless network card automatically puts the sender's hardware address in the local ARP table. Therefore, when a mobile node prepares to send an ICMP ECHO request to a recently detected foreign agent, it will find a matching entry in the ARP table that avoids sending an ARP request over the wireless link and therefore eliminates the additional delay.

4.1.4. Wireless Link Quality at the foreign agent

In radio communications, most of the noise is added to the transmitted signal in the radio channel and at the receiver. The noise that occurs at the transmitter hardly deteriorates the signal transmission quality, because the signal level is sufficiently high [1]. There are several characteristics of the wireless environment surrounding a foreign agent that might affect performance of mobile nodes for which this agent is providing mobility services. In an area with relatively constant background noise level, the stronger the signal, the higher the signal-to-ratio level, increasing the probability of the receiver detecting a true signal. This means that when a packet is transmitted on the wireless channel, it will actually be detected by the receiver as a data packet and not as background noise in the channel. Otherwise, the packet would be lost, and link-layer (and probably TCP-layer) retransmissions by the sender would be necessary.

Three different characteristics of the wireless channel surrounding the mobility agent were selected and implemented in new extensions. They are further explained in the next subsections.

4.1.4.1. Signal Strength

Signal strength indicates how powerful the signal detected at the foreign agent is. This value is usually calculated by the network card driver for every link-layer packet received. A stronger signal increases the probability of detecting a true data signal in the channel: link-layer packets will be distinguished from background noise by a network interface, and thus forwarded to the network layer entity at the receiving computer.

Although signal strength is a good metric for determining which foreign agent should be selected by a mobile node, its value varies rapidly and constantly over time (it might change for every received link-layer packet). Therefore, an appropriate way to compare foreign agents based on this metric had to be devised to avoid recurrently switching back and forth between foreign agents that have similar signal strength levels. This problem is known as *oscillating regions*, and is usually approached by using hysteresis techniques [3]. A similar approach was followed in this case, as explained next.

1 byte	1 byte	1 byte	1 byte	1 byte
type=**21**	length=**3**	signal quality	signal level	noise level

Figure 7 Format of the signal-strength extension

A mobile node not registered with any foreign agent will register with the first agent it detects. After registered with a foreign agent, the mobile node will stay registered with that agent, even when it detects other foreign agents. It will keep a list of those foreign agents detected along with the corresponding signal strength value. Foreign agents transmit this value in a signal-strength extension appended to agent advertisements. The format of this extension is depicted in Figure 7.

If a mobile node misses three consecutive advertisements from its current foreign agent, it executes the selection procedure. During this process, the mobile node will choose the foreign agent whose signal strength level exceeds its current foreign agent’s last recorded signal strength value by a certain threshold. This threshold can be defined in several ways:

- Statically: the threshold is set to a certain predetermined value. This value can be obtained from empirical results, or by using a trial-and-error approach to fine-tune it.
- Dynamically: in this case, the maximal permissible difference in signal strength will be a dynamic parameter that changes according to the way the system behaves. If the cell-switching rate is relatively high, this value can be increased to reduce the number of handoffs. If, on the other hand, handoffs rarely occur, even in the presence of other foreign agents in the subnetwork, the threshold value can be slightly decreased to get better performance.

Both schemes have been studied in related work, as mentioned earlier in this report. In the design presented here, the static threshold option was selected, since it is straightforward to implement, and provides a starting point to compare the hysteresis technique against other techniques (from the performance point of view).

4.1.4.2. Variation in Signal Strength

In the same way signal strength is a good indicator of the quality of the wireless channel, variation in signal strength also gives some indication of how the wireless channel behaves over time. If the variation in signal strength for a foreign agent is consistently negative, that might indicate that the transmission channel at the agent's site is becoming noisier and with higher levels of interference. If the mobility agent is allowed to move around, the variation could also mean that the agent is moving to a location with more interference and more background noise. Conversely, if the variation is consistently positive, that might imply that the environment surrounding the foreign agent is becoming clear of interference and noise in the transmission channel. Under the assumption that foreign agents can move around, a consistently positive variation could also mean that the agent is moving into an area where the signal is clear and the interference level in the channel is low.

For this criterion to be used by mobile nodes in the selection process, foreign agents are required to keep track of the variation in signal strength locally, and transmit this value to mobile nodes. For doing so, a new protocol extension was defined, as described in Figure 8. This extension should be the last one to be added to an advertisement by a foreign agent – added after all the extensions defined in the base protocol, and in the previous sections of this thesis.

1 byte	1 byte	1 byte	1 byte	1 byte
type=**22**	length=**3**	delta quality	delta level	delta noise

Figure 8 Format of the signal-strength-variation extension

As with the previous criterion, when a mobile node first moves into a new subnetwork, it registers with the first foreign agent it detects. After registering for the first time, the mobile node keeps track of those newly detected foreign agents and records the corresponding information on signal strength variation, but will not switch to a new foreign agent as long as it receives advertisements from its current agent. If the mobile node misses three consecutive advertisements from its current agent, it will proceed to select from the list of recently detected agents, the one with the highest positive variation in signal strength.

4.1.4.3. Signal-to-Noise Ratio

The signal-to-noise ratio (SNR) is the difference in power at the receiver between a true signal and background noise. The higher the signal-to-noise level, the higher the possibility of detecting a valid signal.

Since mobile nodes rely on foreign agents to forward registration requests and replies to their home agents, and in some cases use them as default gateways to the rest of the network, the SNR is an important metric to consider when selecting an appropriate foreign agent. If a foreign agent with a low SNR is selected by a mobile node, many of the packets being sent to the mobile node through that agent might not be received because of the poor signal received. This would originate link-layer (and possibly TCP-layer) retransmissions, and degrade considerably the performance of those networked applications running on mobile nodes using the mobility services provided by this foreign agent.

A mobile node can use the information contained in the signal-strength extension depicted in Figure 7 to calculate the SNR value corresponding to the advertising foreign agent, by using the following formula:

SNR = signal level (**dBm**) **-** noise level (**dBm**),

where **dBm** units represent the power level in dB with respect to 1 mwatt [11].

After having registered with a foreign agent for the first time since entering a foreign network, the mobile node will not switch to any newly detected mobility agent as long as it keeps receiving advertisements from its current agent. The mobile node must keep track of the information provided by those detected agents in the signal-strength extension part of the corresponding agent advertisements. In the case the mobile node misses three consecutive advertisements from its current mobility agent, it calculates the SNR value for those foreign agents recently detected. If there is one foreign agent for which the SNR is higher that the corresponding value for the mobile node's current foreign agent, a cell switching occurs.

"There comes a time when the mind takes a higher plane of knowledge but can never prove how it got there" Albert Einstein

5. Implementation Details

5.1. Selection of a Mobile IP Implementation

5.1.1. Evaluation of existing Mobile IP software implementations

In order to implement the protocol extensions described in the previous chapter, an existing Mobile IP package was selected, and the new extensions were built on top of it. This reduced the development and debugging time drastically, and allowed us to focus on measuring the efficiency of the new extensions.

Those Mobile IP implementations listed in section 3.1.6 were evaluated based on the following criteria:

- Compliance with the Mobile IP specification - RFC 2002

  The implementation chosen had to comply with RFC 2002, or at least with Revision 16 of the draft standard, the last draft version before the protocol was accepted by the IETF as a proposed standard.

- Availability of source code adapted for the Linux operating system

  Since the extensions presented previously were to be built on top of an existing Mobile IP implementation, the source code for the selected package had to be publicly available to allow extensions and modifications.

  The mobile computing infrastructure available at Thayer School for testing of the extensions is entirely Linux-based. Therefore, the selected Mobile IP package had to work on Linux machines. Otherwise, the package would have to be ported, which would require additional developing and testing efforts.

- Ease of installation

  If the Mobile IP software package is easy to install, users will be more likely to try it. In addition, it will be easier to package up the new extensions and make them publicly available for general use.

This criterion led to a trade-off. Certain features specified by the Mobile IP protocol - like the proxy and gratuitous ARP mechanisms - need to be implemented as part of the operating system kernel. Therefore, some implementations require *patches* to the Linux kernel to comply with these features. This makes the installation process more complex, since the operating system needs to be tweaked – a task that requires deep technical understanding. Given this issue, special preference was given to those implementations that run entirely in user space, which means that no kernel modifications are necessary for installation purposes.

- Software Complexity

Since the proposed extensions were to be built on top of the chosen Mobile IP implementation, the software code had to be clear enough to allow a complete understanding of its internals. This is a required condition to guarantee that additions are made in a clean and safe way, without altering the original functionality and without adding bugs to the software.

Criteria\Version	Monarch's	PSU's	NSU's	SUNY's	MosquitoNet's	HP-Labs'
RFC 2002-Compliant	√	√	√	√	√	√
Source Code Availability for Linux	X	X	√	√	√	√
Ease of Installation	-	-	X	√	X	√
Software Complexity	-	-	-	√	-	√

Table 1 Evaluation of existing Mobile IP implementations

All the Mobile IP packages evaluated were compliant with either RFC2002 or Revision 16 of the draft standard. Regarding availability of source code for Linux, the Monarch and PSU packages were discarded, since those implementations are customized for FreeBSD. The evaluation was focused now on the three remaining packages. Only two of the three remaining packages are entirely implemented in user space, State University of New York at Binghamton (SUNY)'s and HP-Labs' versions. Although these two packages have limitations in the way they handle the gratuitous and proxy ARP mechanisms, they implement approaches that overcome this detail in a fairly clean fashion. Their source code is relatively simple and modular, allowing the straightforward implementation of extensions. A summary of this comparison is presented in Table 1.

The two Mobile IP packages that obtained the best ranking in the evaluation - namely SUNY's and HP Labs' versions - were installed and tested. The installation was straightforward and the software packages worked fairly well. A performance test was devised and executed to evaluate how the two implementations behaved in the presence of bulk-data traffic. The following performance measurement tools were used: TTCP, NetPerf, NetTest and TReno. The testbed configuration and specific parameters used for each of the performance measurement tools are presented in detail in Appendix A. In these tests, the performance metric used was throughput (in *kbps*) for bulk-data transfer of a certain amount of data from a mobile node in a foreign network to its home agent, using a foreign agent as default gateway. The results of this testing are summarized in Table 2.

Version\Package	TTCP	NetPerf	NetTest	Treno
SUNY's	187.74	240	210	116.74
HP Labs'	169.8	210	130	90.72

Table 2 Measured performance (in kbps) for SUNY's and HP Lab's Mobile IP implementations

The implementation from the State University of New York clearly outperforms the one from HP Labs at Bristol-UK. Based on these results, the Mobile IP software package from SUNY was used as a starting point for developing the protocol extensions described in earlier sections.

5.1.2. An overview of SUNY-Binghamton's Mobile IP Implementation

This section presents an overview of the software package from a developer's point of view, explaining in general terms how the programs work. Detailed information about the software release and how to obtain the source code, which is the only complete developer's guide available, can be found in [7] and [20].

The software package includes source files to build two executable programs: *mh* and *agent*, which provide the functionality for a mobile node and a mobility agent respectively. The software requires the Linux kernel to be configured with support for the **/proc** file system, IP multicasting, IP-in-IP encapsulation, IP forwarding and appropriate network cards. The next two subsections briefly present the design of the two software entities *mh* and *agent*.

5.1.2.1. Implementation of the mobile node software entity – *mh*

The program that implements a mobile node is rather straightforward. The mobile node is in one of four states at any time: INIT, ATHOME, ATFOREIGN, and ASPOPUP[1]. Initially, it starts the agent discovery process by

[1] The mobile node enters this state after it acquires a colocated care-of address and sends a registration request directly to its home agent.

multicasting an agent solicitation message – called a *whereami* message. Afterwards, the program periodically checks for incoming agent advertisements - through the function *dsn_maker* (decision maker) - and determines what action to take based on what message is received from the network and the current state. The function *doer* then executes this action accordingly:

- If an advertisement from the home agent is received, and the mobile node was registered with a foreign agent, *doer* sends a de-registration request to the home agent.
- If an advertisement from a new foreign agent is received, send a registration request to the foreign agent[2]. If the advertisement comes from the current foreign agent, but a rollover in the sequence number indicates that the foreign agent has re-booted, then register again with that agent. Update the routing table accordingly.
- If a new IP address has been acquired via PPP or DHCP, send a registration request to the home agent to notify it about the change.
- Check for incoming registration replies

The program also takes care on a regular basis of registration renewals and request retransmissions, as specified by the Mobile IP specification.

5.1.2.2. Implementation of the mobility agent software entity – *agent*

The agent software entity implements the functionality of both home and foreign agents. An agent can act as both foreign and home agent simultaneously. The program keeps a list of mobile nodes for which it is allowed to act as a home (foreign) agent. There is no explicit security association in the communication between agents, but there is between mobile nodes and home agents. The program is simply an infinite loop in which incoming messages are processed as follows:

- If an agent solicitation is received, the appropriate security checks are made for the requesting mobile node. If the solicitation is valid, an advertisement is sent as a reply.
- If a registration request is received, proceed according to the Mobile IP protocol specification, in case the requesting mobile node needs home or foreign agent services.

---

[2] This is the way the original SUNY Mobile IP implementation handles the detection of several foreign agents simultaneously. This part of the code was modified afterwards in the implementation of the new extensions.

- If a registration reply is received from a mobility agent, it might be a response to a registration request previously forwarded. Proceed with the appropriate security checks, and then forward the reply to the original requesting mobile node.

The program also takes care of sending agent advertisements on a regular basis, with the frequency specified by the protocol, which is defined in the program by the macro "ALARMINTERVAL". Every second, the agent decreases the lifetime in each mobility binding it keeps track of. In case a mobility binding expires, the program deletes the corresponding entries from the host's routing tables and tear down any tunnel that had been established when that mobility binding was setup.

5.1.2.3. Bug Report

During the study of the SUNY Mobile IP implementation, I found and fixed a couple of minor bugs before proceeding to extend the software package with the new selection process. Appendix B contains a detailed report on the bugs found and how they were fixed.

5.2. Description of the software implementation

Each selection criterion was implemented as a separate Mobile IP software version. That facilitated the debugging process and the performance testing procedures. Since the SUNY's Mobile IP implementation is written in C, the extensions were also written in C.

In the near future, these extensions will be made publicly available at Dartmouth as a unified software release that will include the modified (and fixed) SUNY Mobile IP package and all the new modules, which are explained in this subsection.

The premise was to minimize the modifications to the selected Mobile IP package, to keep the implementation clean and to simplify porting of the software extensions to other Mobile IP packages. The implementation of each criterion introduced new algorithms and data structures, which are explained in detail in the following subsections.

New functions introduced by each version were placed in a file called *extension.c*, which was included in the original *Makefile*. In those cases where changes to the original source files were necessary, those changes were isolated by using conditional compilation, controlled by the definition of a macro with name "_EXTENSION_". Each source file is documented with a list of modifications sorted by date, which can be used as reference for developers.

5.2.1. Foreign Agent's Advertisement Rate

For the implementation of this criterion, only the program that implements the mobile node had to be extended to collect statistics about advertisement rate for every foreign agent detected.

5.2.1.1. Mobile Node Entity

- Files Modified: *mh.c*
- Data Structures

```
typedef long metric;
metric metrics[MAXOVERLAP];
metric fa_metric;
dm_info fagents[MAXOVERLAP];
long num_fagents;
long fa_index;
```

Table 3 Data structures used for the advertisement-rate selection criterion

Table 3 lists the main data structures defined in the implementation of the advertisement rate selection criteria. The type *metric* represents the data type for the metric used during the selection process. In this case, it is a long integer that represents the number of advertisements received from a foreign agent during a given period of time. *fagents* is the data structure used by mobile nodes to keep track of foreign agents recently discovered. It is an array of structures of the type *dm_info*, which is described in Table 4. *dm_info* is defined in the original source code, specifically in the file *mh.h* and is used in the file *mh.c* to keep information about the most recently discovered foreign agent. The *fagents* array is manipulated as a circular stack. The program uses the variables *num_fagents* and *fa_index* to keep track of the contents of the array. The macro "MAXOVERLAP" is defined in the original source code, and represents the maximal number of overlapping "cells" the mobile node supports, that is, the maximal number of foreign agents it is able to detect simultaneously.

The data structure *metrics* is used to keep track of the values of the metric associated with the selection criterion used – which in this case is advertisement rate. Each entry – if valid – contains the metric value associated with the foreign agent that is described in the structure *fagents*, in the entry with the same array index. The variable *fa_metric* is used to keep the value of the metric for the current foreign agent – if any. In case the mobile node is in its home network, these data structures are blank, and are ignored, until the mobile node moves to another subnetwork.

```
typedef struct dmi
{
  unsigned char action;
  unsigned long coaddr;
  unsigned long closeagentaddr;
  char ifname[10];
  unsigned short flags;
  unsigned short seqno;
  unsigned short lifetime;
  unsigned long gw;
  struct sockaddr_in from;
  char rtdismsg[256];
} dm_info;
```

Table 4 Data type *dm_info*

- Functions modified

The following functions were implemented in the file *extension.c* (and used in the modified file *mh.c*) to make the mobile node select a foreign agent based on its advertisement rate:

- `collect_data`

Mobile nodes invoke this function every time an agent advertisement is received.

Arguments:

⇒ IP Address of the detected foreign agent

⇒ Default gateway on that foreign network

⇒ Name of the network interface on which the advertisement was received

⇒ A data structure representing the contents of the agent advertisement (See Figure 2(b))

Returned Value:

⇒ 0, if the foreign agent has been detected before

⇒ 1, if it is the first time the foreign agent is detected

Functions where it is used:

⇒ `dsn_maker`

- `elect`

A mobile node invokes this function whenever three conditions are met: (1) it is away from its home network, (2) it receives at least two consecutive advertisements from a foreign agent, and (3) that foreign agent is not its current foreign agent.

When invoked, this routine scans the data structure *fagents* looking for the foreign agent with the highest number of advertisements. Then it compares this to the number of advertisements received from its current foreign agent. The agent with the highest number of advertisements is selected.

Arguments: None

Returned Value:

⇒ 0, if current foreign agent is preferred over those recently detected

⇒ 1, if a new foreign agent is selected

Functions where it is used:

⇒ `dsn_maker`

- `clear_metrics`

This function initializes all the data structures listed in Table 3. It is invoked every time the mobile node switches cells.

Arguments: None

Returned Value: None

Functions where it is used:

⇒ `doer`

⇒ `elect`

5.2.2. Number of Mobile Nodes in the Foreign Agent's Visitor List

For implementing this criterion, both programs *agent* and *mh* had to be extended. The agent entity had to be modified to append a number-of-visitors extension to agent advertisements sent to the network. The mobile node entity was modified to process this extension properly, and use the information collected for selection purposes.

5.2.2.1. Mobile Node Entity

- Files Modified: *mh.c*, *messages.h*
- Data Structures

In addition to those data structures listed in Table 3, the structure listed in Table 5 is defined in *messages.h* to implement the number-of-visitors extension described in Figure 6. The extension was assigned the type value 20, as the value immediately after the last type value used for extensions defined in the Mobile IP specification.

```
struct visitor_info {
   unsigned char type;
   unsigned char length;
   unsigned short num_visitors;
};
#define VINFOTYPE 20
```

Table 5 Data structure for the number-of-visitors Mobile IP extension

- Functions modified

The same functions created for the previous criterion are used in the implementation of the selection based on number of visitors. The function prototypes, functionality and returned values are the same.

An extra change was made to the file *mh.c*: the function `cannotignore`, which performs validity checks on incoming advertisements, was modified to take into account that agent advertisements are longer, with the addition of the new number-of-visitors extension.

5.2.2.2. Agent Entity

- Files Modified: *agent.c*, *messages.h*
- Data Structures

A variable, *num_visitors*, was added in *agent.c* to keep track of how many mobile nodes are currently registered with that agent at any time.

- Functions modified

The following functions were modified:

- `sendURhere`

In this function, the structure *urhmsg*, which represents an advertisement message plus extensions, was modified to include the number-of-visitors extension. Before sending the advertisement, the structure is appropriately initialized inside this function.

- `update_lifetimes`

  This function was modified to update the variable *num_visitors* every time a mobility binding is deleted for a visiting mobile node, due to expiration of the correspondig registration.

- `processRegisterMe`

  This function was modified to increment the variable *num_visitors* every time a new visiting mobile node requests the agent's services. Note that the variable is incremented without even knowing whether the registration request will be accepted by the home agent.

- `ProcessRegReply`

  In this function the variable *num_visitors* is decreased every time the agent receives a rejection to a registration request it has forwarded before to a mobile node's home agent.

5.2.3. Link Latency between Mobile Node and Foreign Agent

For implementing this criterion, only the mobile node entity had to be modified. No protocol extensions were added. Mobile nodes gather information regarding latency without requiring any information to be sent by foreign agents.

5.2.3.1. Mobile Node Entity

- Files Modified: *mh.c*
- Data Structures

  All those data structures listed in Table 3 are also used in *extension.h* to implement this selection criterion. In addition, the following variables were introduced:

  - *awaiting_echoreply_flag* is true if the mobile node has sent an ICMP ECHO message to a detected foreign agent and is waiting to receive an ECHO reply; its value is false otherwise.
  - *awaiting_echoreply_ticks* represents how many seconds have passed since an ECHO request was sent to a foreign agent.
  - *fawait_ticks* indicates how many seconds have passed since the most recent advertisement was received

from the current foreign agent.

- *echoreply_src* contains the IP address of the most recent foreign agent to reply to an ECHO REQUEST message sent by the mobile node.
- Two macros were defined: “MAXWAIT”, which is the maximal number of seconds to wait for echo replies from foreign agents; and “MAXFAWAIT”, which represents the number of seconds the mobile node has to wait for advertisements from its current foreign agent before realizing that it has lost contact with it.

❑ Functions added/modified

- `collect_data`

  This function works as for other criteria. The only difference is that in this case there is no return value.

- `collect_metric`

  Mobile nodes invoke this function every time the two following conditions are met: (1) an agent advertisement is received and (2) the flag *awaiting_echoreply_flag* is not true. This function sends an ICMP ECHO request to the detected foreign agent and changes the value of the variable *awaiting_echoreply_flag* to true.

  Arguments: None

  Returned Value: None

  Functions where it is used:

  ⇒ `dsn_maker`

- `switch_cells`

  This function is invoked by the mobile node whenever three conditions are met: (1) the lapse of time since the last received advertisement from its current foreign agent exceeds the maximum allowed time (MAXFAWAIT), (2) it received an ECHO reply from a previously polled foreign agent and (3) the reply received is not from its current foreign agent. The function retrieves the information related to the new foreign agent from the data structure *fagents*, and copies it to the global variable *Dminfo*.

  Arguments:

  ⇒ The IP address of the replying foreign agent

  Returned Value:

⇒ 0, if the replying foreign agent is not in *fagents* anymore

⇒ 1, if it was found and its information retrieved

Functions where it is used:

⇒ `dsn_maker`

- `process_echoreply`

Mobile nodes invoke this function every time an ICMP packet is received. The function checks whether the packet is an ICMP echo reply and if so, it also checks whether it corresponds to an ECHO REQUEST previously sent by the program. In case of a positive outcome, the variables *echoreply_src*, *awaiting_echoreply_flag* and *awaiting_echoreply_ticks* are updated accordingly.

Arguments:

⇒ The received ICMP message

⇒ The IP address of the sender computer

Returned Value:

⇒ 0, if the message fails the test

⇒ 1, if the received packet is an ICMP echo reply to a request sent by the mobile node, and it comes from a foreign agent that is in *fagents*.

Functions where it is used:

⇒ `cannotignore`

5.2.4. Signal Strength at the foreign agent

This version required the modification of both entities: the agent program and the mobile node program. The agent program had to be extended to append the signal-strength extension - as described in Figure 7 -to agent advertisements. The mobile node entity was extended to handle this extension properly and use the information provided in the selection process.

5.2.4.1. Mobile Node Entity

- Files Modified: *mh.c*, *messages.h*
- Data Structures

```
struct iw_quality {
   unsigned char qual;        /* Link quality (Max. 15) */
   unsigned char level;       /* Signal level (Max. 63) */
   unsigned char noise;       /* Noise level (Max. 63)  */
};
struct iw_info {
   unsigned char type;
   unsigned char length;
   struct iw_quality iw_stats;
};
#define IWINFOTYPE 21
typedef struct iw_quality metric;
```

Table 6 Data structures defined for the mobile node entity using the signal strength criterion

Table 6 shows the new data structures introduced in the implementation of this selection criterion in the mobile node entity. The structure *iw_quality* contains the values that are sent by mobility agents in the signal-strength extension. As shown in the table, this data structure corresponds to the new *metric* data type. The structure *iw_info* implements the signal-strength extension. The type value assigned to this extension is 21, which is the value immediately after that selected for the number-of-visitors extension.

The data structures *fagents*, *metrics*, *fa_metric*, *num_fagents*, *fa_index* and *fawait_ticks* are also used here, exactly as described for other criteria in previous sections.

- Functions modified
  - `collect_metric`

    Mobile nodes invoke this function whenever an agent advertisement is received. Its interface is exactly as described for the implementation of the latency criterion. Internally it works differently, since the information collected in this case is extracted from the signal-strength extension, included by agents in their advertisements.

  - `switch_cells`

    In this version, the function takes no arguments. The return values are defined identically as for other criteria. In this case, the function is invoked when the mobile node realizes that the time elapsed since the most recent advertisement from its current foreign agent was received has exceeded the maximal allowed time. The function scans the data structure *fagents* looking for the detected foreign agent with the highest value for signal strength. The mobile node will switch to the new foreign agent if its signal strength is higher than the last signal strength value received from its current foreign agent.

5.2.4.2. Agent Entity

- Files Modified: *agent.c*, *stats.c* (new file)
- Data Structures

The agent entity uses those data structures listed in Table 6 to keep track of its signal strength values and include this information in the agent advertisements it sends over the network.

- Functions added

  - `get_wireless_stats`

The agent program invokes this function before sending agent advertisements. This function is strongly dependant on the Linux implementation of the Wavelan card driver, since it reads the file "**/proc/net/wireless**" to obtain the wireless parameter values, which are periodically updated by this driver.

Arguments:

⇒ the name of the wireless network interface

⇒ A pointer to a structure of type *iw_quality*, where the values retrieved by the functions are returned

Returned Value:

⇒ 1, if the wireless statistics for the specified interface name were successfully retrieved

⇒ 0, otherwise

Functions where it is used:

⇒ sendURhere

5.2.5. Signal Strength Variation at the foreign agent

To implement this criterion, both software entities *agent* and *mh* were modified. The changes were mostly the same as those made in the implementation of the signal-strength criterion, with the exception of some data structures, which are presented next.

5.2.5.1. Mobile Node Entity

- Files Modified: *mh.c*, *messages.h*
- Data Structures

The data structures presented in Table 7 were introduced for the implementation of this criterion. The structure *iw_info* represents the signal-strength-variation extension described in Figure 8. The type *metric* is now represented as a structure that contains the information the mobile node retrieves from the signal-strength and the signal-strength-variation extensions. All the other data structures used in the implementation of the signal-strength criterion are used in this version too.

- Functions modified
  - `collect_metric`

    This function works almost exactly as it was implemented for the signal-strength criterion. It now assumes that agent advertisements contain two extensions: signal-strength and signal-strength-variation.

```
#ifdef _LINK_DELTA_
struct iw_quality_delta {
   short qual;          /* variation in link quality */
   short level;         /* variation in Signal level */
   short noise;         /* variation in Noise level  */
};
#endif
struct iw_info {
   unsigned char type;
   unsigned char length;
   struct iw_quality_delta iw_deltas;
};
#define IWDELTATYPE 22
typedef struct {
      struct iw_quality m_qual;
      struct iw_quality_delta m_delta;
   } metric;
```

Table 7 Data structures defined for the implementation of the signal strength variation criterion

5.2.5.2. Agent Entity

- Files Modified: *agent.c*, *stats.c*
- Data Structures

  The agent entity uses those data structures listed in Table 7 to keep track of the variation in signal strength in its transmission channel. It includes the signal-strength and the signal-strength-variation extensions in every agent advertisement it multicasts over the network.

- Functions modified

  - `sendURhere`

    This function was modified to add a new variable for keeping track of previous values for signal strength. When the agent advertisement is to be sent over the network, the variation in signal strength since the last advertisement was sent is computed and included in the advertisement as a signal-strength-variation extension.

5.2.6. Signal-to-Noise Ratio at the foreign agent

For implementing this criterion, the agent entity needs to be extended in the same way it was in the implementation of the signal-strength criterion. Therefore, all the details explained in section 5.2.4.2 apply in this case, and no additional details are necessary.

The mobile node entity required modifications similar to those explained for the signal-strength implementation. The only difference with respect to that implementation is how the function `switch_cells` selects the best foreign agent among those that have been recently detected. In the implementation of this criterion, the function scans *fagents* for the foreign agent with the highest signal-to-noise ratio – as defined in section 4.1.4.3. The mobile node switches to the new foreign agent only if its SNR value is higher than the SNR value calculated from the most recent statistics received from its current foreign agent.

"Art is long, life short; judgment difficult, opportunity transient"

Goethe

6. Experimental Setup

After the Mobile IP extensions were implemented and tested, the next step was to measure the actual performance of the software implementation on real scenarios. A first series of experiments was based on the NetSpec performance measurement tool. New scenarios were devised and tested during a second experimental series using another measurement tool: NetPerf. This section describes the main features of the infrastructure used in both experiment phases.

6.1. Equipment Used

The networking infrastructure was comprised of seven laptop computers:

- 6 Toshiba *Tecra* 500CS with the following configuration:
  - ⇒ Processor: Pentium 133MHz
  - ⇒ RAM size: 16MB
  - ⇒ Linux Partition size: 600MB
  - ⇒ Swap partition size: 65MB
  - ⇒ Linux kernel version: 2.0.30
  - ⇒ Card Manager version: 2.9.11
  - ⇒ 1 Digital RoamAbout PC Card 2.4GHz, Direct Sequence
- 1 Gateway *Solo* 2300 with the following configuration:
  - ⇒ Processor: Pentium II 200MHz
  - ⇒ RAM size: 32MB

⇒ Linux Partition size: 1.7GB

⇒ Swap partition size: 133MB

⇒ Linux kernel version: 2.0.30

⇒ Card Manager version: 2.9.11

⇒ 2 Digital RoamAbout PC Cards 2.4GHz, Direct Sequence

6.2. Experiments based on NetSpec

6.2.1. NetSpec Overview

Several performance measurement tools were evaluated as prospective candidates to be used in the testing infrastructure. NetSpec was selected because of (1) its scripting capabilities that allow the construction of relatively complex testing scenarios, and (2) the source models that are built into the system and can emulate several different types of traffic like FTP, Telnet, and WWW traffic.

NetSpec is a network level end-to-end performance evaluation tool that the University of Kansas' Information and Telecommunication Technology Center developed to help collect the results of performance experiments on ATM networks [23]. This software tool provides a simple, block-structured language for specifying experimental parameters. It also provides support for controlling, from a centralized computer, performance experiments containing an arbitrary number of connections across LANs or WANs.

NetSpec exhibits many features that are not supported by other performance tools like ttcp and Netperf: parallel and serial multiple connections, a range of emulated traffic types (FTP, HTTP, MPEG, etc.), three different traffic modes, scalability, etc. The tool outputs a complete report specifying, for each receiver-transmitter pair, information such as: number of bytes transmitted/received during the script execution, number of packets transmitted/received, packet size statistics, etc.

6.2.2. NetSpec Traffic Source Models

Version 3.0 of NetSpec includes several built-in source models for emulating different types of traffic: telnet, FTP, WWW, MPEG, videoconference and Voice Traffic. These analytic models are empirically derived from traces of real network traffic using known probability distributions [10]. The next subsections present a brief description of these source models that were used to evaluate the performance of the newly implemented Mobile IP extensions;

namely, FTP and WWW traffic.

6.2.2.1. Source model for FTP traffic

Figure 9 shows an example NetSpec script that models FTP traffic between two computers: *mach1* and *mach2*. FTP traffic is modeled by the following parameters [10]:

⇒ **Session interarrival time**. This is the time elapsed between the start of two consecutive FTP sessions. It is exponentially distributed (*ftpSessionInterarrival*) with parameter *lambda*, which represents the average session arrival rate in sessions/microsecond units.

⇒ **Item size**. This parameter specifies the number of bytes transferred during each FTP request. It is modeled using a log-normal distribution; *ftpItemSize* is the built-in model for this parameter.

⇒ **Number of items**. This parameter represents the number of items transferred during a single FTP session. It is modeled using a log-normal distribution, represented by the *ftpNOfItems* built-in model.

```
cluster {
  test mach1 {
    type = burstq (blocksize=ftpItemSize,
                 repeats = ftpNOfItems,
                 period = ftpSessionInterarrival(lamba=0.0001),
                 buffer = 262144,
                 duration = 1800);
    protocol = tcp (window=262144);
    own = mach1:42000;
    peer = mach2:42000;
  }

  test mach2 {
    type = sink(blocksize=262144, duration=1800);
    protocol = tcp(window=262144);
    own = mach2:42000;
    peer = mach1:42000;
  }
}
```

Figure 9 Sample NetSpec script for FTP traffic

6.2.2.2. Source model for WWW traffic

Figure 10 shows an example NetSpec script that emulates WWW traffic. The figure shows some parameters that are used for generating the WWW traffic:

⇒ **Request interarrival time**. It indicates the time elapsed between the arrival of two consecutive data requests at a WWW server. This parameter is modeled using an exponential distribution (*WWWRequestInterarrival*) with parameter *lambda*, which represents the average request arrival rate in requests/microsecond units.

⇒ **Document size**. It represents the size of the requested documents. It is modeled using a Pareto distribution, through the built-in model *WWWItemSize*.

```
cluster {
  test mach1 {
    type = burstq (blocksize=WWWItemSize,
             period = WWWRequestInterarrival(lamba=0.00002),
             buffer = 262144,
             duration = 1200);
    protocol = tcp(window=262144);
    own = mach1:42000;
    peer = mach2:42000;
  }

  test mach2 {
    type = sink(blocksize=262144, duration=1200);
    protocol = tcp(window=262144, rcvlowat=8);
    own = mach2:42000;
    peer = mach1:42000;
  }
}
```

Figure 10 Sample NetSpec script for WWW traffic

6.2.3. Testing Scenarios

6.2.3.1. Physical Layout

The layout shown in Figure 11 was used during the performance measurement experiments performed for each of the selection criterion. Two subnetworks were defined with subnet numbers 10.0.0.0 (foreign network) and 10.0.1.0 (home network). A laptop computer was placed in the home network to play the role of *traffic sink* for the NetSpec scripts, serving as an FTP/WWW client to which files would be transferred from the mobile nodes visiting the foreign network. The Gateway *Solo* machine was used as a router that would relay the traffic to and from the home network, transparently to the Mobile IP configuration. Two foreign agents advertise their services in the foreign network. The mobile nodes are placed in an overlapping area to test the newly implemented selection mechanisms.

One set of experiments was run for each selection criterion, except for the variation in signal strength. The variation in signal strength is an effective metric when used in networks where either mobile nodes or mobility agents (or both) are free to move back and forth between clear transmission areas and noisy areas with high interference levels. In a static experimental layout like the one presented in Figure 11, no performance improvement was expected to occur from using this criterion in the selection process. As a recommendation for future work, scenarios where the mobility issue is studied should be devised and tested using this criterion, and all the others presented in this thesis.

Two mobile nodes were used during each experiment run, one executing the original SUNY Mobile IP implementation, and the other one executing the extended version corresponding to the selection criterion being tested. By having the two versions running simultaneously under the same conditions and generating similar traffic flows, it is possible to compare how the two versions perform under exactly the same operating characteristics.

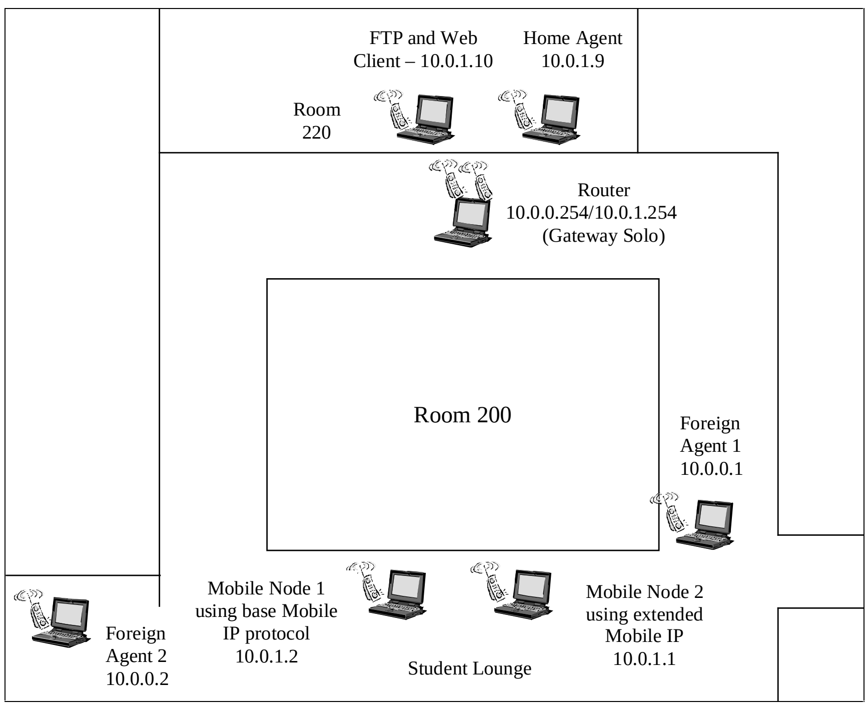

Figure 11 Physical layout for experiments based on NetSpec

For the number-of-visitors extension, one additional Toshiba Tecra was used, placed as described in Figure 12. This mobile node was running the modified Mobile IP version, implementing the number-of-visitors criterion, also modified to restrict the mobile node to register only with *foreign agent 2*. In this way, one of the foreign agents will always have at least an additional visitor than the other. Therefore, *mobile node 2* will have an obvious choice when selecting a foreign agent based on this criterion. The additional mobile node generated an average FTP traffic rate of 20kbps, for each experimental phase (WWW and FTP) in which the performance of the other two mobile nodes was being measured.

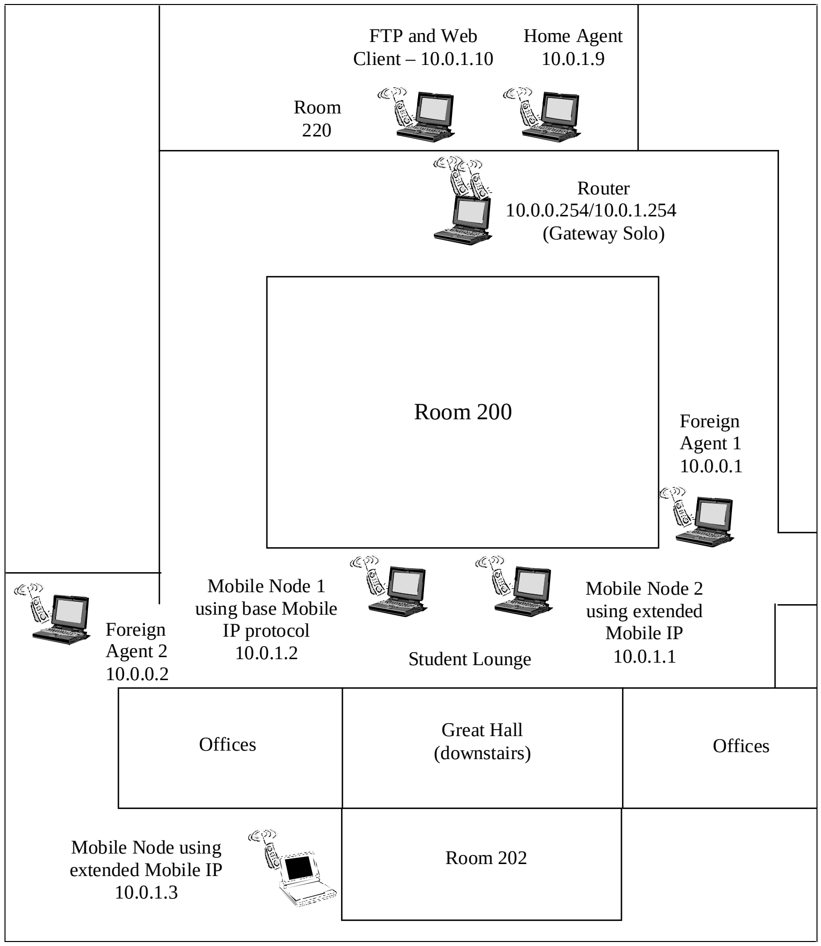

Figure 12 Physical layout in experiments based on NetSpec for the number-of-visitors criterion

6.2.3.2. Emulated Traffic Characterization

The experiments were divided into two phases: one for WWW traffic and another one for FTP traffic. They were run separately to study the performance of the new implementations for each type of traffic independently. The division of the experiments should help understand the behavior of the new Mobile IP extensions for different network traffic types.

Each of the mobile nodes in the student lounge executed a NetSpec script that generated a 5-minute traffic flow from the mobile nodes to the client in the home network. An example script that generates WWW traffic from *mobile node 1* is shown in Figure 13. The next subsections explain the values selected for each parameter.

```
cluster {
  test {
    type = burstq( blocksize=WWWItemSize(min=8, shape=0.40),
                   period = WWWRequestInterarrival(lambda=0.0000001315,
                                                   min=1000),
                   buffer = 65536,
                   duration = 300);
    protocol = tcp(window=65536);
    own = 10.0.1.1:52011;
    peer = sneezy:52011;
  }

  test sneezy {
    type = sink(buffer=65536, duration=300);
    protocol = tcp(window=65536, rcvlowat=8);
    own = sneezy:52011;
    peer = 10.0.1.1:52011;
  }
}
```

Figure 13 Script used in the experiments for generating WWW traffic

6.2.3.2. Parameters for the WWW source model

This subsection explain the values assigned to some of the WWW source model's parameters, as shown in Figure 13.

⇒ *WWWItemSize*

The pareto distributed *WWWItemSize* has one parameter: *shape*, which was set to the minimum value (0.40), in order to maximize the expected transfer size.

⇒ *WWWRequestInterarrival*

This exponentially distributed model has one parameter: *lambda*. Its value was defined so that the average data rate would be 100kbps, following this formula:

Average Data Rate (*bits/sec*) =
Request Arrival Rate (*requests/sec*) * No. of Bytes per Request (*bytes/requests*) * 8

According to the source models' documentation, the average number of bytes transferred per WWW request is 95,000 [9]. Manipulating the expression above, the corresponding Request Arrival Rate (*lambda*) is 1.315e-07 requests/microsecond.

⇒ Buffer size

This parameter was set to the value of the TCP congestion window, 64KB. This value was chosen after some earlier tests showed that it was the maximal value allowed by Linux kernel 2.0.30 (the kernel on the laptops used in the experiments).

6.2.3.2. Parameters for the FTP source model

Figure 14 shows the script used for generating FTP traffic from one of the mobile nodes to the FTP client on the home network. The values used for the parameters in the script are discussed next.

```
cluster {
  test {
   type = burstq(blocksize=ftpItemSize(min=8),
                 repeats = ftpNOfItems(min=1),
                 period = ftpSessionInterarrival(lambda=0.0000000178,
                                                 min=1000),
                 buffer = 65536,
                 duration = 300);
   protocol = tcp (window = 65536);
   own = 10.0.1.1:52111;
   peer = sneezy:52111;
  }

  test sneezy {
   type = sink(buffer=65536, duration=300);
   protocol = tcp(window=65536, rcvlowat=8);
   own = sneezy:52111;
   peer = 10.0.1.1:52111;
  }
}
```

Figure 14 Script used in the experiments for generating FTP traffic

⇒ *ftpItemSize, ftpNOfItems*

These two models are built into the system, the values set in the script are the minimum allowed size of the items. Therefore, each session will have at least one item transferred, which is at least 8-bytes long.

⇒ *ftpSessionInterarrival*

This exponentially distributed model has a parameter: *lambda*, which represents the session arrival rate. This value was calculated to keep an average data rate of 50kbps, using the following formula:

Average Data Rate (*bits/sec*) = Session Arrival Rate (*sessions/sec*) * No. of Items per Session (*items/sessions*) * No. of Bytes per Item (*bytes/items*) * 8

According to the source models' documentation, the average number of items transferred in an FTP session is 7, while the average number of bytes per item is 50,000 [9]. Manipulating the expression above, the corresponding Session Arrival Rate (*lambda*) is 1.78e-08 sessions/microsecond.

⇒ Buffer size

As in the WWW source model, this parameter was set to the value of the TCP congestion window, 64KB.

6.3. Experiments based on NetPerf

6.3.1. NetPerf Overview

NetPerf is a LAN-oriented network performance benchmark. It measures throughput, minimal latency, TCP transaction speed (e.g., connection, request, response, disconnection), and CPU utilization during tests. The most common use of netperf is measuring bulk data transfer performance. This is also referred to as "stream" or "unidirectional stream" performance. Essentially, this test measures how fast one computer can send data to another and/or how fast the other computer can receive it. It supports several transport mechanisms, such as TCP and UDP.

The tool provides facilities for specifying several parameters (e.g. buffer and packet size for TCP stream tests). Users may specify a confidence level for the performance results to be obtained, together with the minimal and maximal number of times the tests should be performed with the provided parameters to reach that confidence level. The way NetPerf ‘s features were used in this experiment series is explained in the following subsections.

6.3.2. Testing Scenarios

6.3.2.1. Physical Layout

The layout depicted in Figure 15 was used for all the criteria except number-of-visitors. The physical layout presented here will allow testing each criterion separately. The mobile node shown in the figure ran a different Mobile IP version at each experiment phase (including the original Mobile IP version). During each of these experimental phases, the performance of the corresponding Mobile IP version was measured. In the figure it can be noted that all the mobility agents have now two network interfaces. The two foreign agents have two wireless network adapters and the home agent has one wireless and one Ethernet network adapter. Also, the two foreign agents were placed in different positions now, compared to the layout shown in Figure 11. *Foreign Agent 1* now is farther apart from the student lounge. On the other hand, *Foreign Agent 2* was placed closer to the student lounge. The additional hop represented by a router between the foreign agents and the home network was eliminated and now the home agent plays the role of default gateway to rest of the network for the home network.

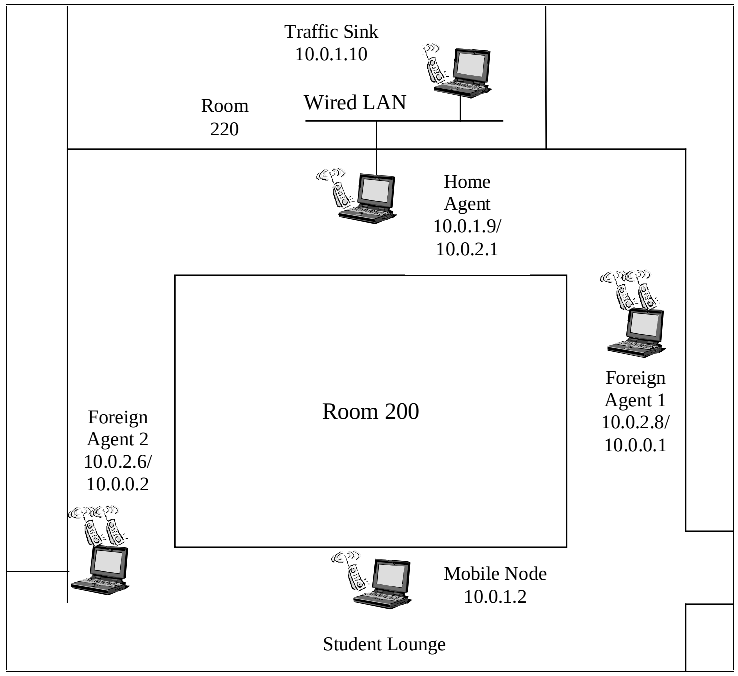

Figure 15 Physical layout for experiments based on NetPerf

In the case of the number-of-visitor criterion, the layout depicted in Figure 16 was used. The foreign agents in this layout were left in the original position they were for the NetSpec-based experiments. An additional Toshiba Tecra was used as a probe mobile node that was running a modified Mobile IP version that would only register with *Foreign Agent 2*. As in Figure 15, the home network was setup as a wired LAN built using an *Asanté 10T* hub. Both the home agent and the *traffic sink* computers were connected to this hub. The goal of this modification was to further reduce the channel contention for the home agent when forwarding packets from the mobile node to the *traffic sink*.

6.3.2.2. Emulated Traffic Characterization

As mentioned before, one experimental phase was performed for each selection criterion and also for the original Mobile IP. During each phase, the mobile node located in the student lounge executed a utility script provided with the NetPerf distribution, which allows to automatically perform bulk-data transfer experiments for several combinations of parameters. The TCP bulk transfer tests were configured to generate traffic addressed to the *traffic sink* computer located at the home network. Following is a description of the parameters used in the execution of the NetPerf tests.

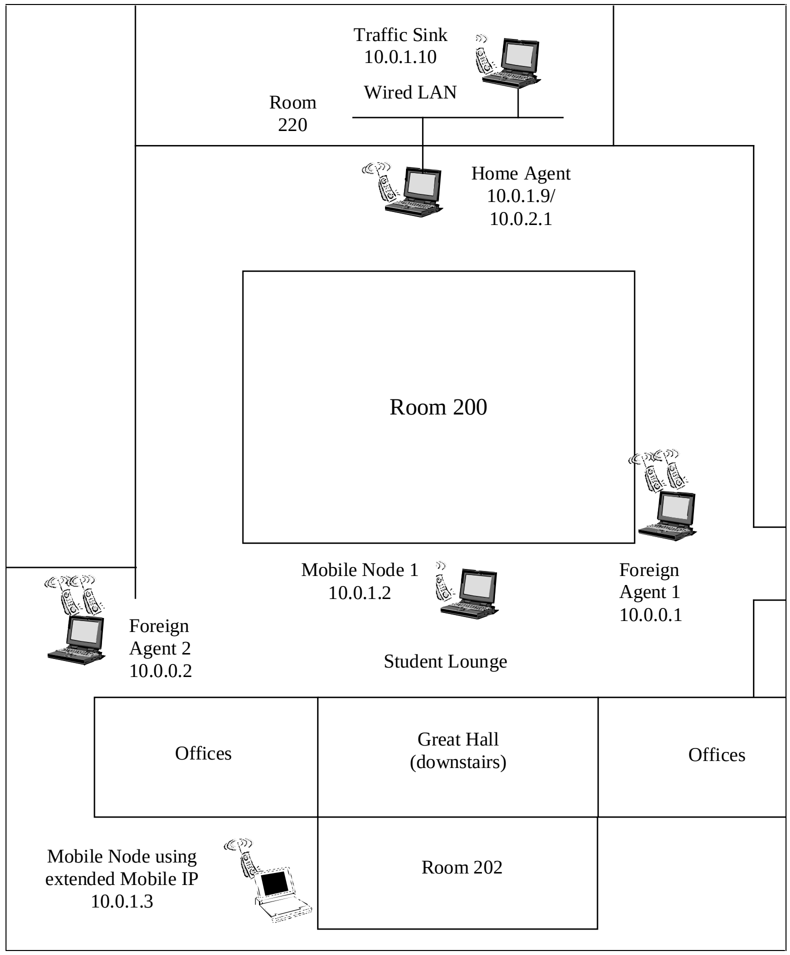

Figure 16 Physical Layout in experiments based on NetPerf for the number-of-visitors criterion

⇒ Buffer Size

This parameter was set to 8192 bytes. It was decided to use only this value, which is found in popular TCP/IP implementations [21]. More emphasis was given to studying the effect of the next parameter on the performance of the Mobile IP extensions.

⇒ Packet Size

The following values were used: 512, 1024, 2048 and 4096 bytes. The Maximum Transmission Unit (MTU) along the path from the mobile node to the *traffic sink*, that is the longest IP packet that can be transmitted over the network without being fragmented because of physical layer limitations, is 1440 (without including IP and TCP headers). Two of the packet sizes used are below this limit, and two of them are above. In a way, this was done intentionally to test how the IP fragmentation mechanism affects the performance of the Mobile IP extensions.

⇒ Test Duration

Each bulk transfer test was 60 seconds long.

⇒ Confidence Level

This tells the program to keep performing tests until it reaches this confidence level regarding the average value of the metric that is being obtained (throughout in this case)

⇒ Maximum/Minimum Number of Iterations

The maximum number of iterations to do trying to reach the given confidence level was set to 30. The minimum value was set to 10.

"It is not enough to have a good mind; the main thing is to use it well"
Rene Descartes

7. Experimental Results

7.1. Analysis

The experiments described in Chapter 6 showed interesting results in terms of throughput achieved by the traffic flows generated using NetSpec and NetPerf. This subsection highlights some observations from a subset of the results obtained. The rest of the graphs and tables listing throughput results can be found in Appendix C.

7.1.1. Results from NetSpec-based Experiments

As mentioned in the previous chapter, for each selection criterion, two experimental phases were executed, one for FTP and another one for WWW traffic, using the experimental layout described in the previous chapter. During each experimental phase, the two mobile nodes were setup to run the corresponding NetSpec script simultaneously, by using the Unix *cron* utility. In the case of the number-of-visitors criterion, an additional mobile node was used during the experiments, as illustrated in Figure 12. This additional laptop was setup to execute a NetSpec script every 10 minutes. The script generated a 20kbps FTP traffic rate for 5 minutes.

During each experimental phase corresponding to a given criterion, the performance results (measured in throughput) from several runs were collected. To assess how good the performance of the mobile node running the extended Mobile IP version was with respect to the one executing the original Mobile IP version, the difference of the obtained throughput measurements for each script run was taken. Therefore, if the extended version outperforms the original version, the corresponding difference point is positive. If the performance of the two versions was approximately the same, the difference point is near zero. On the other hand, if the original Mobile IP version outperforms the extended version, the difference point is negative.

In the experiments in which emulated WWW traffic was used, little improvement in throughput because of using the extended Mobile IP implementation was found, for most of the criteria. Figure 17 shows the plot of the difference in throughput for simultaneous experiment runs for the number-of-visitors criterion. Intuitively, most of the points in this graph should be greater than zero to indicate an improvement in performance. However, the graph shows that most of the points are centered around zero, which represents little difference in performance. In fact, Table 8 indicates that the average value for the difference in throughput in this case is -0.77. This can be clearly seen in the histogram presented in Figure 18, which shows that approximately 60% of the difference points are in the range (-5kbps, 5kbps].

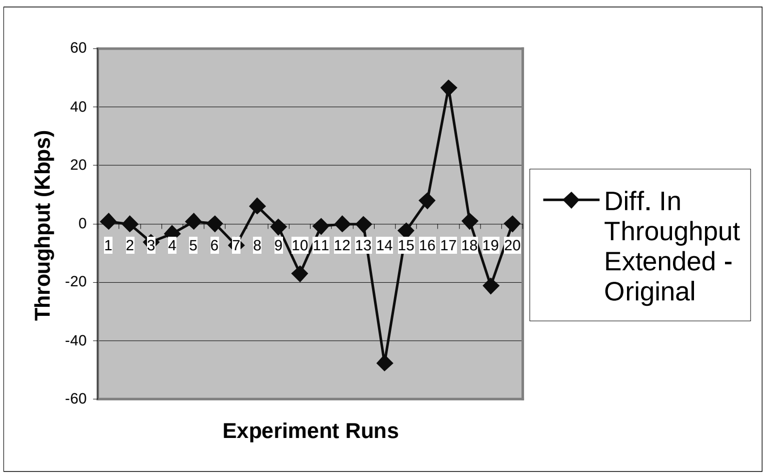

Figure 17 Criterion: No. Of Visitors - Differences in measured throughput for WWW traffic

The same results were found for the criterion Signal-to-Noise Ratio, as the graph in Figure 19 shows. The histogram presented in Figure 20 shows that more than 60% of the difference points lay on the interval (-5kbps, 5kbps]. However, the histogram shows more points on the positive side than in the negative side of the graph. On the other hand, the histogram in Figure 22, corresponding to the number-of-visitor criterion for FTP traffic, shows that the difference points in that case are evenly spread on both sides of the graph, with an average value of -0.92, as indicated in Table 8.

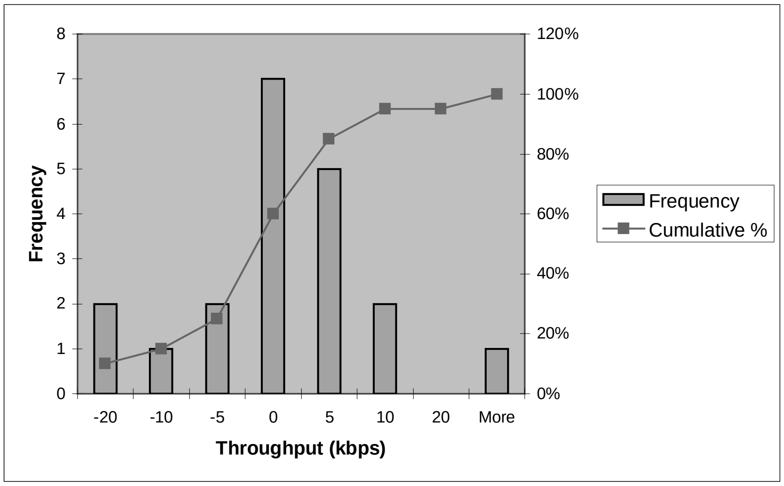

Figure 18 Criterion: No. Of Visitors - Histogram of differences in throughput for WWW traffic

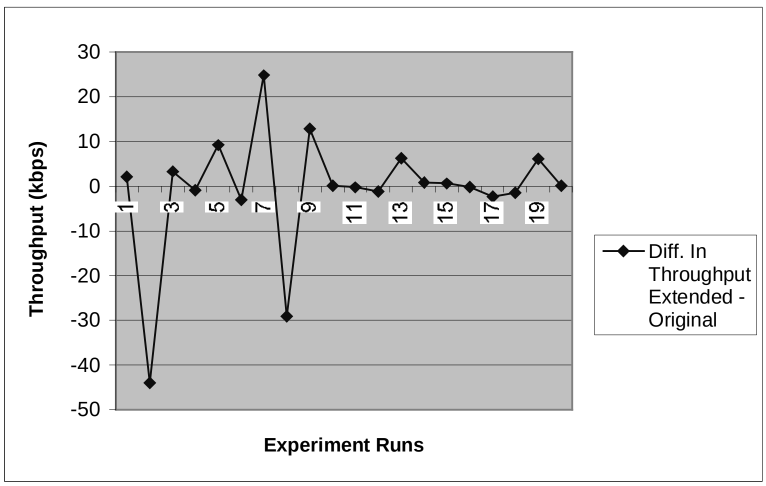

Figure 19 Criterion: SNR - Differences in measured throughput for WWW traffic

Traffic Type	Advertisement Rate	No. of Visitors	Latency	Signal Strength	Signal-to-Noise Ratio
WWW	-0.77	-2.21	-2.40	1.51	-0.82
FTP	-0.92	-0.92	1.12	0.21	-9.72

Table 8 Average difference in throughput for each criterion

For the experiments where FTP traffic was used, more variability in measured throughput was observed. Figure 21 and Figure 23 show the difference in FTP throughput for the two criteria discussed before: number-of-visitors and signal-to-noise ratio. The plots show no apparent effect of using the extended Mobile IP version on the throughput achieved by the generated FTP sessions. This can also be observed in the corresponding histograms that are presented in Figure 22 and Figure 24. However, the average difference in throughput shown in Table 8 for the case of the SNR criterion shows that the original Mobile IP version outperforms the extended version. The results for other criteria follow a similar pattern, and can be found in Appendix C.

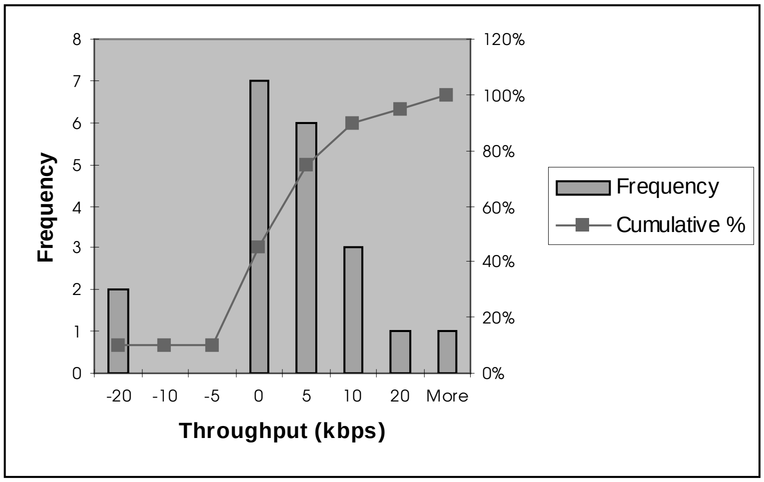

Figure 20 Criterion: SNR - Histogram of differences in throughput for WWW traffic

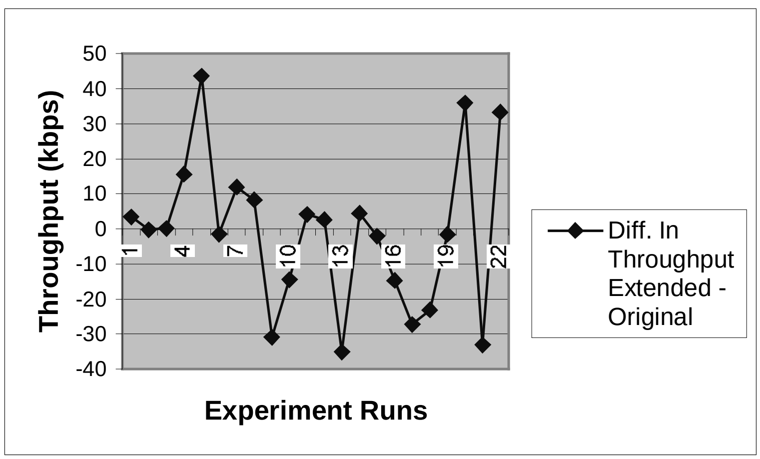

Figure 21 Criterion: No. Of Visitors - Differences in measured throughput for FTP traffic

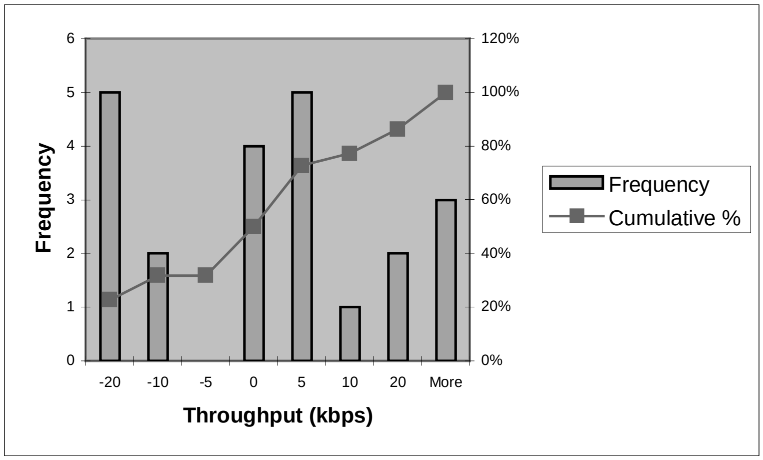

Figure 22 Criterion: No. Of Visitors - Histogram of differences in throughput for FTP traffic

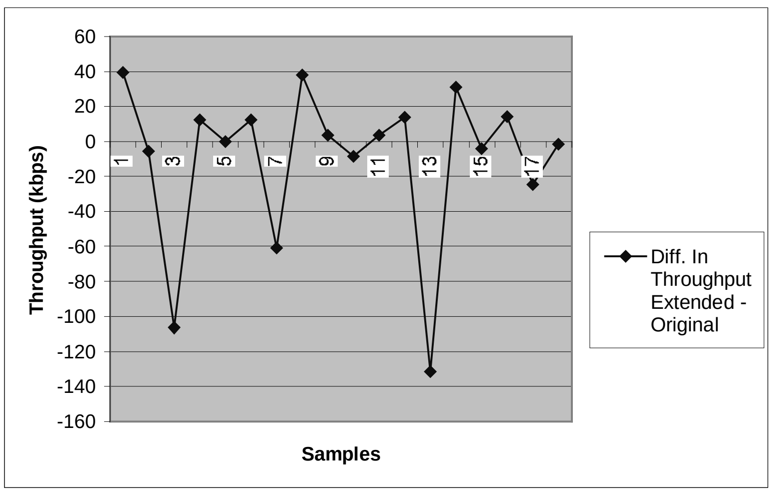

Figure 23 Criterion: SNR - Differences in measured throughput for FTP traffic

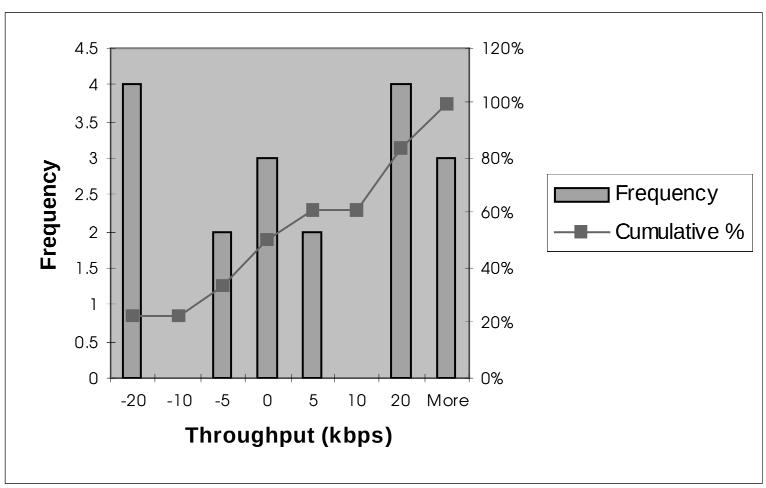

Figure 24 Criterion: SNR - Histogram of differences in throughput for FTP traffic

There are several factors related to the experimentation setup that explain these results. They are discussed below.

- In the layout used during these performance measurement experiments, it can be seen that the two cells defined by the two foreign agents are overlapping to the extent that both agents can detect the network traffic generated by the other. The two foreign agents belong to the same logical subnetwork; moreover, the two mobile nodes are within each other's range of transmission. Therefore, regardless of which foreign agent a mobile node selects, the traffic from the other foreign agent would compete with its network traffic for the wireless channel. This translates into a large number of collisions at the link-layer level when the mobile node is sending packets to its foreign agent, and also when the foreign agent is forwarding these packets to the target machine in the home network. These collisions will lead to link-layer packet retransmissions and probably TCP layer retransmissions too, degrading considerably the performance of the emulated traffic sessions in the experiments. This contention for the wireless channel occurs whenever generated traffic is being transmitted, regardless of the foreign-agent selection criteria used.
- Another important detail is that each mobility agent uses the same network interface to receive and forward packets, which generates additional contention because while a foreign agent is receiving packets from a mobile node, no packets can be forwarded. The network card has to switch back and forth between sending and receiving, which translates into long waiting periods especially when a high volume of network traffic is being processed, as in the FTP experiments whose results are shown above. Again, this problem arises at both foreign agents, regardless of the selection criteria being used by the mobile nodes.

In the NetSpec-based experiments, the average throughput for FTP traffic is nearly 4 times the average throughput for WWW traffic, which indicates that the generated FTP traffic represented a bigger load to the network than the generated WWW traffic. Therefore, the two issues mentioned concerning the shared wireless channel in the experimental layout, clearly relate to the fact that the results show that the new selection mechanisms have no positive effect on the measured throughput for generated FTP traffic.

	Selection Criteria	WWW Traffic		FTP Traffic	
		Extended Mobile IP	Original Mobile IP	Extended Mobile IP	Original Mobile IP
Total no. of elections executed	Adv. Rate	61	N/A	328	N/A
	Latency	1	N/A	16	N/A
	SNR	12	N/A	0	N/A
	S. Strength	3	N/A	35	N/A
	No. of Visitors	141	N/A	362	N/A
Total no. of times the election process picked a new foreign agent	Adv. Rate	0	N/A	0	N/A
	Latency	1	N/A	4	N/A
	SNR	1	N/A	0	N/A
	S. Strength	0	N/A	0	N/A
	No. of Visitors	4	N/A	9	N/A
Total no. of registration requests sent by the mobile node	Adv. Rate	679	561	9888	2727
	Latency	3084	957	14382	2004
	SNR	3835	997	2373	1274
	S. Strength	3251	502	9882	3321
	No. of Visitors	530	597	8099	1740
Total no. of times the mobile node switched cells[3]	Adv. Rate	2	394	39	1084
	Latency	23	330	28	1011
	SNR	12	754	15	916
	S. Strength	3	333	32	1989
	No. of Visitors	6	285	24	1076

Table 9 Mobile IP data collected from the mobile nodes' logs during the experiments

Table 9 shows some statistics that were collected from the *mh* program logs while running the performance measurement experiments. The numbers in this table show, for example, that during the experiments corresponding to the advertisement-rate criterion, even when the selection process was invoked several times, a new foreign agent was not selected in any case. This occurs because the mobile node was placed in an area where it received advertisements from both foreign agents at the same time; therefore, after selecting a foreign agent for the first time, the mobile node running the extended Mobile IP version remained registered with that agent. A similar argument explains why this number is low for the latency criterion. Since both foreign agents were placed relatively close to the mobile nodes during the experiments, there were few opportunities when a mobile node missed more than two advertisements from its foreign agent. In cases where this happened, it might have been caused by network congestion, and the mobile node most probably would have received another advertisement from its foreign agent before any ECHO reply message from a candidate foreign agent. It was observed during the experiments that after a

[3] This includes cell switches due to selections – either arbitrarily (as in the original Mobile IP version) or using the selection process - and those due to registration expiration.

mobile node registered with a foreign agent, it remained in that foreign agent's visitor list most of the time.

In the case of the SNR criterion, a similar rationale applies. During the experiments, it was rare when a mobile node missed three consecutive advertisements from its foreign agent, since both foreign agents where within a relatively short distance from the mobile nodes. This justifies the number of elections executed being so low. The same argument applies to the signal-strength criterion; due to the testing layout, both foreign agents had similar signal-strength levels, which deterred the selection-enabled mobile node from switching to a new foreign agent.

Regarding the number of registrations sent by the selection-enabled mobile node - shown in Table 9, it is relatively high for both cases (WWW and FTP traffic) and for all the selection criteria, compared to the same values for the mobile node running the original Mobile IP version. By looking at the log files, it was determined that this number was so high due to registration renewals being rejected because of id mismatch at the home agent while performing security checks on incoming registration messages. The *nonce* security mechanism used by the SUNY Mobile IP implementation to authenticate registration messages consists of two randomly-generated numbers (*nonces*) included on each control message, one by the mobile node and the other one by the home agent. When a mobile node receives a registration reply from its home agent, it saves the nonce sent by the home agent in the reply. When sending the next registration request, the mobile node includes this number and generates its own *nonce* that is also included in the registration request. The home agent receives the registration request and checks whether the number included by the sender in the home agent nonce field of the request corresponds to the last nonce it sent in a registration reply to this mobile node. It the numbers do not match, a registration denial is sent back to the mobile node, together with a new *nonce* that can be used by the mobile node to re-synchronize and send a new registration request. The home agent also includes in this message the nonce generated and included in the registration request by the mobile node. The mobile node, upon receiving a registration reply, also checks that the *nonce* the home agent included in the mobile node nonce field of the reply corresponds to the last nonce the mobile node generated. If the numbers do not match, the reply is silently discarded [16].

The mobile node running the extended Mobile IP version sent several registration renewals to the home agent, but most of them were rejected because of *nonce* mismatch. Since the communication channel was highly congested due to the traffic flows being generated, several requests and replies might have gotten lost, causing the nonce association between mobile node and home agent to become inconsistent. During the experiments, the selection-enabled mobile node kept sending registration requests using the mechanism described above until it resynchronized its *nonce* association with the home agent. This did not occur with the mobile node running the original Mobile IP implementation. In the scenario used for these tests, the mobile nodes were located in an overlapping region where they could detect both foreign agents simultaneously. Consequently, the selection-oblivious mobile node switched back and forth between the two foreign agents in a non-deterministic way, rarely renewing the registration with its current foreign agent. Therefore, most of the time it avoided the renewal process, that is he process in which most of the id mismatch retransmissions occurred for the other mobile node. This explanation is also based on the fact that it is more likely for a registration renewal to be rejected due to id mismatch than for an initial registration. A registration renewal uses the *nonce* it saved from the reply it received the last time the mobile node registered with its home agent, which is likely to be outdated by the time the renewal is submitted. On the other hand, when issuing an initial registration request, the mobile node uses a *nonce* included in a recently received reply.

The number-of-visitors extension, according to Table 9, had the highest number of invoked elections and also the largest number of turnover selections – that is, those elections in which a new foreign agent is selected. However, these figures do not reflect any positive effect on performance as presented earlier in this chapter, mainly because of the specific testing scenario used.

In the experimentation layout used for this criterion, which is depicted in Figure 12, three mobile nodes are competing for the resources on two foreign agents. One of the mobile nodes was permanently registered with one of the foreign agents. The mobile node running the original Mobile IP implementation switched back and forth between agents in a non-deterministic manner. The selection-enabled mobile node selected that foreign agent attending the minimal number of visiting nodes. The options this mobile node could find are:

- Its current foreign agent is attending only one mobile node. In this case, the new foreign agent must be attending the other two, in which case there is no benefit in switching to that agent. Eventually, the mobile node running the original Mobile IP version will switch to this foreign agent and that would lead to the next case.
- Its current foreign agent is attending two mobile nodes and the new foreign agent is attending just one mobile node. If the selection-enabled mobile node switched to the new foreign agent, then the situation would be symmetric, since that foreign agent would then be attending two mobile nodes. The design of this criterion dictates that, in cases like this, the mobile node stays registered with its current foreign agent, since there is no visible performance benefit in switching cells. An issue recommended for future work and research is how to obtain and standardize information about the load that a mobile node represents for its foreign agent. This

information can be used in the foreign-agent selection process to guarantee a better choice in cases like the one presented here, in which no performance benefit is expected to be obtained from switching cells, based on the information currently available.

- Its current agent is attending three mobile nodes. In this case, the decision of switching to the new foreign agent is obviously the most appropriate. When the mobile node switches, it would be the only mobile node attended by that foreign agent. It might stay that way until the mobile node running the original Mobile IP version eventually switches to that foreign agent too. If this occurs, the selection-enabled mobile node is now in the case previously described.

In a testing infrastructure like the one used for these experiments the selection-enabled mobile node will always get to a point where it needs to sacrifice performance (like in the second case above). This is due to the randomness in the switching pattern of the mobile node that is running the original Mobile IP software. The previous paragraphs, together with the channel contention factor presented earlier in this chapter, explain why, even when it is intuitive that this criterion provides mobile nodes with the opportunity to improve their performance, the results obtained in the experiments performed show no such improvement in throughput.

Table 9 also shows that the extended Mobile IP version reduces significantly the number of times the mobile node switches back and forth between the foreign agents. This is due to the selection process implemented by the extended Mobile IP version which, as opposed to the original randomly selecting a new foreign agent every time a new agent is detected, does not switch to a new foreign agent until it is strictly necessary.

7.1.2. Results from NetPerf-based Experiments

The experimentation layout used for the NetPerf-based experiments approaches some of the issues presented above concerning the layout used for the NetSpec-based experiments. For example, the foreign agent cells overlap to a minimum, as shown in Figure 15. In addition, each foreign agent uses two network interfaces, one network interface for the link in which it provides mobility services, and another one on the link that connects it with the rest of the network, including all the possible home agents. By doing the performance measurement separately for each criterion and the original version, the channel contention is minimized, and controlled as a factor that might affect performance.

The tests performed with NetPerf are completely determined by its parameters, which means that they can be repeated for each Mobile IP version, guaranteeing that the traffic flows generated are identical. With environment characteristics remaining relatively stable, it is valid to compare throughput results for the different Mobile IP versions, as shown in Figure 25. This figure presents, for all the criteria but number-of-visitors, the throughput achieved in bulk transfer tests for different packet sizes. The figure also includes throughput results for the original Mobile IP version. The tests whose results are shown in this figure were performed on the scenario depicted in Figure 15. In this case, as opposed to the NetSpec-based experiments, the improvement in throughput of all the selection criteria with respect to the original version is clear, reaching up to 20% in cases like the signal-strength criterion for a packet size of 2KB.

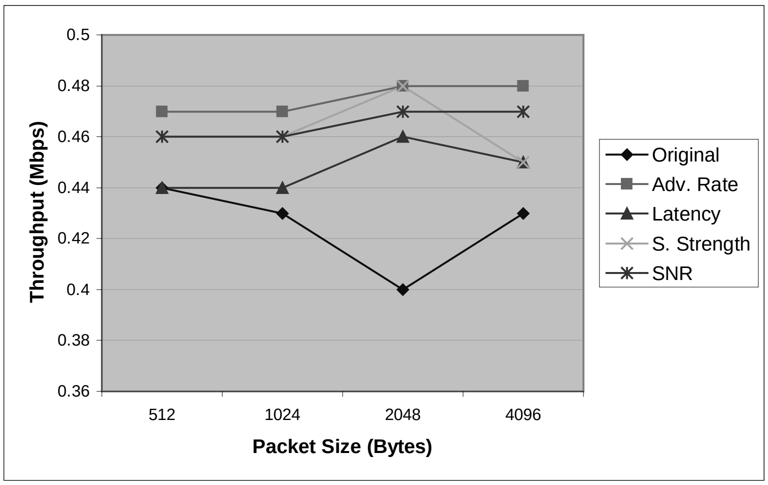

Figure 25 Throughput Comparison for Bulk Data Transfer at 95% confidence level

In the case of the number-of-visitors criterion, Figure 26 shows the performance comparison with the original Mobile IP version tested on the scenario described in Figure 16. The graph shows that the extended version exhibits better performance than the original version as the packet size grows larger. However, the difference in performance is not as marked as it was for the other criteria, as discussed before. This might be due to the following reasons:

- The load generated by the additional mobile node was not representative in comparison with the bulk data generated by the mobile node being tested.
- Both foreign agents had disadvantages that would bring the performance of the visiting mobile nodes down. Referring to Figure 16, *Foreign Agent 2* always had at least one mobile node in its visitor list, and *Foreign Agent 1* was on the side of the lounge that has a small entrance and many more walls. This made the communication channel between the agent and the mobile node very error-prone for both, traffic generated by the mobile node and for control messages that were relayed by the foreign agent to the mobile node, leading to

lost packets that would need to be re-transmitted.

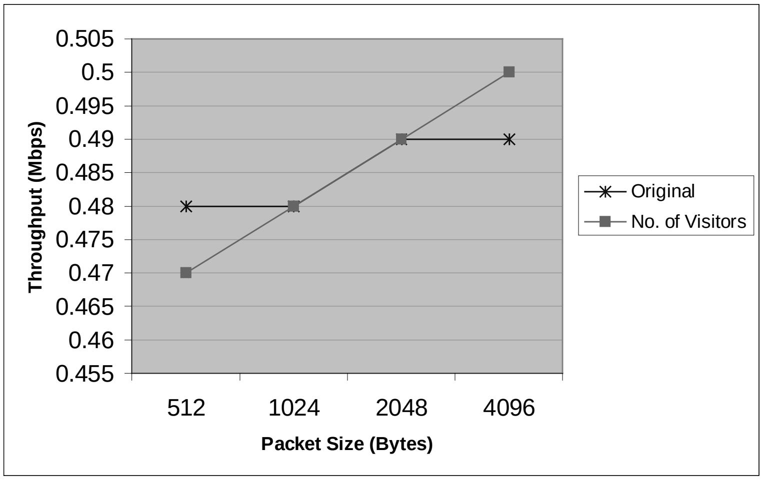

Figure 26 Criterion: No. Of Visitors - Throughput for Bulk Transfer at 95% confidence level

In summary, this experimental phase showed better results, which lead me to conclude that for the particular scenario being tested, the Mobile IP extensions developed to implement the different foreign-agent selection criteria, actually provide a representative improvement in throughput compared to the original Mobile IP version.

7.2. Lessons Learned

While designing the experiments and analyzing the results, many issues that were not considered during the design phase were pointed out. Following is a discussion of these issues for each of the selection criteria.

✓ Advertisement Rate

The design of this criterion makes it vulnerable to oscillation between foreign agents in overlapping areas. The fact that a selection-enabled mobile node invokes the selection procedure whenever it receives two consecutive advertisements from the same foreign agent, without discerning whether it has been registered with that agent recently, makes it unstable in scenarios where mobile nodes are placed in overlapping regions. Although this oscillating behavior was not observed in the layout depicted in the Figure 11, since the two foreign agents had approximately the same advertisement rate during the whole test of this criterion, there are other scenarios in which switching oscillation can occur. For example, consider the case in which the two foreign agents are far apart from each other and the selection-enabled mobile node is moving back and forth between them, without loosing connectivity with either one. In this example, using the current selection trigger, the mobile node may switch to the agent that is closer to it (even when it is not completely out of range from its current foreign agent), generating representative overhead as it moves in an oscillating way.

On the other hand, the experiments showed that this criterion is highly stable to oscillation in overlapping regions in which the mobile node remains relatively immobile and equidistant with respect to the two overlapping foreign agents.

- ✓ Number of Visitors

Even when this criterion uses the same policy as the previous one for triggering the selection procedure, it does not suffer from the discussed oscillating behavior. This is because the selection metric is not based or related to physical location with respect to the foreign agent. It depends on a more general characteristic of the scenario, namely, the number of visiting mobile nodes attended by each foreign agent.

However, fairness is not guaranteed by the design of this criterion, in cases in which several selection-enabled mobile nodes are deciding between two foreign agents. For example, if all the mobile nodes are registered with one of the foreign agents, they all might at the same time switch to the other foreign agent, which is not attending any mobile node, leading to a symmetric situation that may become cyclic. A solution for this issue is to randomly select the number of consecutive advertisements (from a given set of values, e.g. {2,3,4}) that need to be received from a foreign agent before triggering the selection process. This will minimize the probability of the mobile nodes invoking the selection mechanism at the same time. More sophisticated mechanisms to guarantee fairness, which might require distributed knowledge about the networking scenario, should be studied in future research related to the goals and issues approached in this thesis.

- ✓ Link Latency, Signal Strength, Signal-Strength Variation and Signal-to-Noise Ratio

In the design of these criteria, the fact that the mobile node has to wait until three advertisements from its current foreign agent are missing to start the selection process has advantages and disadvantages. It complies with the Mobile IP base specification, which states that a foreign agent should be discarded by a mobile node from its list of valid agents after having missed three consecutive advertisements from that agent. It also prevents the mobile node from oscillating between different foreign agents in an overlapping region.

The flip side to this policy is that it might deter the mobile node from switching to other foreign agents that provide better service. In the experimental scenarios used in this thesis, both foreign agents were placed in locations with similar characteristics in terms of signal quality, in which case the issue mentioned here was not a key factor to consider. However, there are other cases in which this issue might represent a difference in performance. For example, the mobile node may initially select a foreign agent and afterwards it may move into the transmission range of another foreign agent with better signal strength without moving completely out of its current foreign agent's coverage area. This issue requires further study to determine for which scenarios it will be preferable to avoid the waiting period. Another interesting issue is how long the waiting period needs to be for different scenarios in order to maximize the performance of the mobile nodes. These open issues can be approached in future work that focuses specifically on these selection criteria and the ways to improve their effect on the mobile node's performance.

"Our lives begin to end the day we become silent about things that matter"
Dr. Martin Luther King Jr.

8. Conclusions

Mobile IP defines extensions to the original Internet Protocol (IP) that allow computers to transparently move from one point of connection to the network to another without disrupting established network connections. This idea is suitable for multi-hop wireless networks, where mobile computers are free to move from one cell to another one, but still require connectivity to the rest of the network. In a Mobile IP-enabled multi-hop wireless network, however, mobile nodes are likely to be located in overlapping areas where more than one foreign agent can be simultaneously detected. The Mobile IP base protocol does not define policies nor mechanisms to select a foreign agent among several candidates. This thesis studied the improvement in performance that can be achieved if a mobile node is able to select the most appropriate foreign agent. In order to add intelligence to the foreign-agent selection process, the following criteria were defined:

- Foreign agent's advertisement rate
- Number of visiting nodes being attended by each candidate foreign agent
- Latency on the link between the mobile node and the candidate foreign agent
- Signal strength level at the candidate foreign agent's site
- Variation in signal strength at the candidate foreign agent's site
- Signal-to-Noise Ratio at the candidate foreign agent's site

These criteria were implemented as extensions to the Mobile IP protocol, on top of an existing Mobile IP software package, and the performance of each of them was measured in two different sets of experiments involving different scenarios (characterized by nature of traffic flow, physical layout and performance measurement tools).

The performance results show that, for one set of experiments, little improvement in the measured throughput was achieved by using the new selection process. The analysis of the results showed that the apparent lack of effect is due to the experimental setup used. This scenario presented high channel contention for the traffic flows generated, and also several bottlenecks, among them the mobility agents with just one wireless network interface used for both receiving and forwarding packets from the mobile nodes to the home agent. Results for the NetPerf-based experiments, which corrects the drawbacks mentioned for layout of the previous experimental phase, show that there is an improvement in the throughput of bulk-data transfers for each of the Mobile IP versions implementing the new selection criteria, relative to the original Mobile IP version.

This thesis focused on throughput as the performance metric for all the experiments executed. However, other metrics can also be used to determine the performance improvement achieved from using the foreign-agent selection process. Prospective metrics might be: handoff frequency between different foreign agents, latency between the mobile node and its home agent, average number of hops from the mobile node to its home agent, etc.

In summary, six different extensions to the Mobile IP protocol have been defined, designed and implemented into working software packages. It was shown that the extensions actually improve the throughput in bulk-data transfers originated at mobile nodes using the selection extensions in certain scenarios. Further performance measurement scenarios and refinements are proposed in the next chapter.

"The important thing is not to stop questioning"
Albert Einstein

9. Future Work

Improvements to the Mobile IP extensions

There is a lot of room for enhancement of the selection criteria presented in this report. Some ideas are presented here, and their development as a continuation of this thesis is highly encouraged.

- ✓ Advertisement rate, Number of visiting nodes

  The design of these two criteria instructs a mobile node to execute the selection process after receiving two consecutive advertisements from a foreign agent different from its current agent. This is a somewhat arbitrarily selected value that just indicates that the mobile node has started to move into the coverage area of a new foreign agent. But it does not guarantee that after the completion of the interval, the mobile node will have a fair sample of the foreign agents present on the link. An important issue to study is whether the value of this parameter affects performance of the mobile node using any of these criteria during the selection process.

- ✓ Signal strength, Signal-to-Noise Ratio

  Although stipulated in the design of this criterion, in some cases it might not make sense to compare the signal strength level of those recently discovered foreign agents with the strength value for the current foreign agent. According to the Mobile IP base protocol, the mobile node can assume that it has lost contact with its current foreign agent, since it has already missed three advertisements from it (which triggered the selection process). This detail might be reviewed in future versions of the extensions, and its performance effect assessed. The same issue applies to the signal-to-noise ratio criterion, and can be approached in the same way for that criterion too.

The current version uses a statically set threshold for the difference in signal strength between the current foreign agent and candidate foreign agents. An interesting issue to study is how the value assigned to this threshold affects the mobile node's performance on a given scenario. If there is a performance effect, the idea of a dynamic threshold using hysteresis techniques might be a powerful way to boost the performance of the mobile node under certain scenarios where the signal strength value is too dynamic for a statically assigned threshold.

New Testing Scenarios

Two main issues should be approached in further tests of the selection process developed in this thesis:

- Mobility

New testing scenarios where mobile nodes and mobility agents are allowed to move, changing positions with respect to one another, should be designed in order to test robustness and performance behavior of the new extensions on those kinds of environments. This is the case for the variation-in-signal-strength criterion, which was not included in the experiments done in this thesis because no effect in performance was likely to be found, due to the static nature of the scenarios tested.

- Multi-Hop topologies

The selection process was tested on 2-hop networks, that is, networks in which mobile nodes need to use exactly one relay to the home network. In scenarios with a higher number of hops, foreign agents may act at the same time as mobile nodes, trying to select the best relay to the rest of the network. The new selection process should be tested in this kind of scenarios in order to study issues not approached in this thesis like those mentioned below.

⇒ *Scalability*. An interesting issue to explore is how the selection process affects the performance of the overall network. For example, there might be a threshold in the number of hops that marks the limit until which the selection process gives an improvement in performance. After going beyond that threshold, factors like hierarchical tunneling or the overhead generated by the selection process might outweigh any benefits obtained from every mobile node using the new selection process.

⇒ *Need for more elaborated selection schemes*. Several chained mobile node-foreign agent associations are likely to be found in Mobile IP-enabled, multi-hop wireless networks. In some of those cases, using solely a certain criterion along the chain of associations (each mobile node using the same criterion) will not provide as good performance as using a combination of criteria, depending on the particular scenario in which each mobile node is executing. This issue is discussed in more detail in the next sub-section.

General Metric Model

The experimental results discussed in chapter 7 usher in some issues for further study. The most important ones, which should be considered as a direct extension to the work developed in this thesis, are:

- Finding out those scenarios where it is most appropriate to use a given selection criterion
- Given a scenario, how can the different criteria be combined to obtain maximum performance?

The two questions above embrace a more general idea. After empirical observations are made to independently determine in which type of scenarios each criterion guarantees an improvement in performance, a model like the one shown in Equation 1 can be devised.

$$M = \sum_{i=1}^{n} w_i * m_i$$

Equation 1 General model for a combined metric

Each m_i represents the measured value of the *i-th* metric (e.g. number of visitors, signal strength level, etc.), and w_i is a weight factor obtained from a knowledge base that is derived from the empirical observations mentioned above. The weighted average ***M*** will give an overall *grade* for the foreign agent to which the collected metrics correspond, a value that can be used in the selection process. In order for such a model to be built, scenarios have to be characterized according to certain parameters, and then the task would be reduced to finding the best w_i for each combination of the parameters that define a specific scenario. Once again, answering the questions above and building this model will require the execution of a representative number of experiments in several different scenarios, which *per se* constitutes a large project. Another way to build this knowledge base is to collect information from production networks using the Mobile IP extensions developed as a part of this thesis. Both alternatives are feasible to accomplish, leading to a model that can be of extreme utility for general Mobile IP wireless networks.

References

[1] **Y. Akaiwa**. *Introduction to Digital Mobile Communication,* John Wiley & Sons, New York, NY, 1997.

[2] **C. M. Bowman, P. B. Danzig, D. R. Hardy, U. Manber, M. F. Schwartz**. *The Harvest Information Discovery and Access System*, Proceedings of the Second International World Wide Web Conference, pp. 763-771, October 1994.

[3] **T. Camp, J. Lusth, J. Matocha, C. Perkins, and H. Stern**. *Reduced Cell Switching in a Mobile Computing Environment*, Submitted to Computer Networks and ISDN Systems, Oct. 1997.

[4] **Carnegie Mellon University**. *CMU Monarch Project IETF Mobile Ipv4*, http://www.monarch.cs.cmu.edu/mobile_ipv4.html

[5] **Cisco Systems Inc**. Cisco DistributedDirector Web Page. http://www.cisco.com/warp/public/751/distdir/dd_wp.htm

[6] **S. Deering, ed**. *ICMP Router Discovery Messages*, RFC 1256, September 1991.

[7] **A. Dixit, V. Gupta**. *Mobile-IP for Linux (ver 1.00)*, Dept. of Computer Science, State University of New York, Binghamton, NY, 1996.

[8] **Hewlett-Packard Labs at Bristol, UK**. *An implementation of Mobile IP under Linux,* http://www-uk.hpl.hp.com/people/jt/MobileIP/index.html

[9] **B. O. Lee.** *Wide Area ATM Network Experiments using Emulated Traffic Sources*, University of Kansas, Lawrence, KS, 1995.

[10] **B. O. Lee, V. S. Frost, R. Jonkman.** *NetsPec 3.0 Source Models for telnet, ftp, Voice, Video and WWW traffic*, Information & Telecommunication Technology Center, University of Kansas, Lawrence, KS, January 1997.

[11] **W. C. Y. Lee**. *Mobile Communications Design and Fundamentals,* John Wiley & Sons, New York NY, 2nd

edition, 1993.

[12] **H. Lei, C. Perkins**. *Ad Hoc Networking with Mobile IP,* Proceedings of 2nd European Personal Mobile Communication Conference, September 1997.

[13] **National University of Singapore**. *Mobile IP at NUS*, http://mip.ee.nus.sg/

[14] **C. Partridge, T. Mendez, W. Milliken**. *Host Anycasting Service,* RFC 1546, November 1993.

[15] **C. Perkins**. *IPv4 Mobility Support*, RFC 2002, October 1996.

[16] **C. Perkins**. *Mobile IP Design Principles and Practices,* Addison-Wesley Longman, Reading, Mass. 1998.

[17] **D. Plummer**. *An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48-bit Ethernet Addresses for Transmission on Ethernet Hardware*, RFC 826, November 1982.

[18] **Portland State University**. *Secure Mobile Networking Project Home Page*, http://www.cs.pdx.edu/research/SMN/

[19] **J. B. Postel, ed.** *Internet Protocol,* RFC 791, September 1981.

[20] **State University of New York at Binghamton**. *Linux Mobile-IP*, http://anchor.cs.binghamton.edu/~mobileip/

[21] **W. R. Stevens.** TCP/IP Illustrated Volume 1, Addison-Wesley, Reading, Mass. 1994.

[22] **Stanford University**. *MosquitoNet Mobile IP*, http://mosquitonet.stanford.edu/software/mip.html

[23] **University of Kansas' Information and Telecommunication Technology Center.** *The official NetSpec Home Page,* http://www.ittc.ukans.edu/netspec/

Appendix A

Performance Comparison between Mobile IP implementations

Two Mobile IP implementations were compared based on the throughput of bulk data transfer from the mobile node to its home agent. For performing this test, the physical layout shown in Figure 27 was used. In the experiment, three Toshiba *Tecra* computers were used. These machines have the same configuration as those described in section 6.1. The foreign agent in this case has two network interfaces, one on the home network and the other one on the foreign network, where it advertises its services.

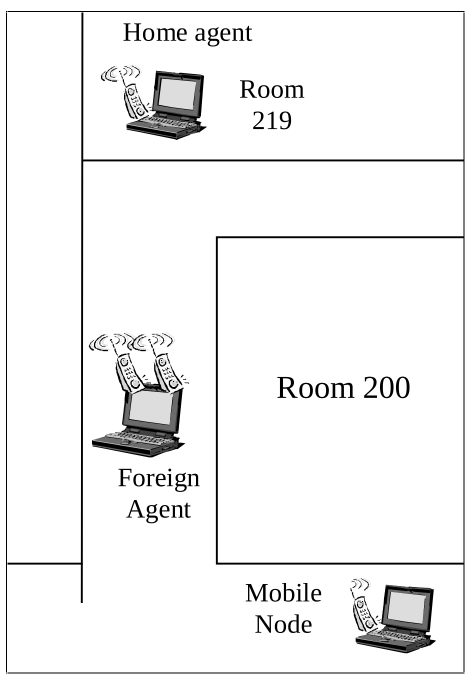

Figure 27 Layout used for performance comparison of Mobile IP implementations

The two Mobile IP implementations compared were those from State University of New York and HP-Labs at

Bristol. For doing the performance measurement, four different tools were used: TTCP, NetPerf, Nettest, and TReno. One experiment was conducted with each of these tools. The measurements in each case, and the results obtained are explained next.

1. Testing with TTCP

   TTCP times the transmission and reception of data between two systems using the UDP or TCP protocols. Additional information about this tool can be found at ftp://ftp.arl.mil/pub/ttcp/.

   The following command lines where used to activate TTCP:

⇒ At the mobile node: `ttcp –t –s  home-agent`

⇒ At the home agent: `ttcp –r –s`

   These commands activated the TTCP program using the default value of 1024 transmitted messages, each of which is 1 KB in size.

2. Testing with NetPerf

   NetPerf is a LAN-oriented network performance benchmark. It measures throughput, minimal latency, TCP transaction speed (e.g., connection, request, response, disconnection), and CPU utilization during test. It supports many transport mechanisms. More information about this tool can be found at: http://www.netperf.org/netperf/training/Netperf.html.

   This test used NetPerf's default parameters by issuing the following command line at the mobile node:

```
netperf –H home-agent –p <port-number>
```

   This test causes a TCP connection to be established between the mobile node and the home agent, through which data is transmitted as fast as possible for 10 seconds.

3. Testing with Nettest

   For performing this test, the following command line was issued at the mobile node:

```
nettest home-agent – 10240 <port-number>
```

   This command caused 100 buffers of size 10240 bytes to be transferred from the mobile node to the home agent.

4. Testing with Treno

   TReno is a TCP internet-throughput measurement tool based on a user-level implementation of a TCP-like protocol. This allows it to measure throughput independently of the TCP implementation of end hosts and to serve as a useful platform for prototyping TCP changes. Treno uses the same technique as traceroute to probe the network. By sending out UDP packets with low TTL, hosts and routers along the path to the final destination will send back “ICMP TTL Exceeded” messages which have similar characteristics to TCP ACK packets.

More information about this tool can be found at http://www.psc.edu/networking/treno_info.html. The testing procedure was initiated at the mobile node by using the following command:

```
treno –v –n home-agent
```

This makes the mobile node send UDP packets as fast as possible to the home agent for a period of 10 seconds.

Both Mobile IP implementations were tested using each of the tools mentioned above. The performance results (transfer throughput) are summarized in Table 2, which is presented in section 5.1.1.

Appendix B

Report of Bugs Fixed in the SUNY Mobile IP implementation

This appendix is intended to give a description of the bugs found in the original SUNY Mobile IP implementation, and how they were fixed. Two files were modified to fix bugs: *agent.c* and *mhlow.c*

Fixes made on file **agent.c**

⇒ Function `newtunnel`

This function, as shown in Figure 28, scans the variable *tunnelbitvec*, which is a bit array indicating which tunnel numbers are already in use. The function returns the first available tunnel number it finds, or 0 if there are none available. The tunnel number 0 is not used, since this value is used to indicate failure in the search. However, the value 0 is actually used by the operating system to reference the first tunnel interface. This means that one tunnel interface in the operating system will never be used by this implementation. This is especially important in the cases when the number of tunnel interfaces is small, 2 for example. This problem was fixed as shown in Figure 29.

The appropriate changes were made to the function `processRegisterMe`, also in the file *agent.c*, to adapt it to the change in returned value type for the function `newtunnel` from `char` to `short`.

This change and all the others documented in this appendix were included in the original code by conditional preprocessor directives, which are activated by defining a macro with name “_CORRECTION_” at compilation time.

```
char
newtunnel() {
   char i;

   for (i = 1; i < MAXTUNNELS; i++)  {
      if ((tunnelbitvec & (1 << i)) == 0)
            return(i);
   }
   return(0);
}
```

Figure 28 Original function `newtunnel`

```
short
newtunnel() {
   char i;

   for (i = 0; i < MAXTUNNELS; i++)  {
      if ((tunnelbitvec & (1 << i)) == 0)
            return(i);
   }
   return(-1);
}
```

Figure 29 Modified function `newtunnel`

Fixes on the file **mhlow.c**

The modifications included in this file fixed the problem of not closing open sockets when they are not used anymore, which would cause system calls to fail in the long run because of "too many open files".

⇒ Function `deleteARP`

Figure 30 shows the portion of this function that was modified in order to close the socket descriptor after the failed `ioctl` operation.

⇒ Function `lowarpsend`

This function was modified by adding the statement line `close(sockid)` just before the end of the function, so that the socket opened at the beginning of the function to perform the ARP operation is closed.

```
s = socket(AF_INET, SOCK_DGRAM, 0);
if (s < 0) {
      perror("arp: socket");
      return(-1);
}
if (ioctl(s, SIOCDARP, (caddr_t)&ar) < 0) {
      if (errno == ENXIO && debug >2)
            fprintf(stderr,"%s (%s) -- no entry\n",
                host, inet_ntoa(sin->sin_addr));
      else
            perror("SIOCDARP");
#ifdef _CORRECTION_
      close(s);
#endif
      return(-1);
}
```

Figure 30 Excerpt of modified function `deleteARP`

Appendix C

Detailed NetSpec-Based Experiment Results

Advertisement Rate		No. of Visitors		Latency		Signal Strength		Signal-to-Noise Ratio	
Extended Mobile IP	Original Mobile IP	Extended Mobile IP	Original Mobile IP	Extended Mobile IP	Original Mobile IP	Extended Mobile IP	Original Mobile IP	Extended Mobile IP	Original Mobile IP
1.727	1.973	2.689	1.833	1.987	2.315	2.458	17.279	3.961	1.832
1.825	6.148	2.138	2.179	2.069	2.770	1.806	2.527	7.808	51.780
1.860	27.249	2.087	8.392	1.945	1.902	12.951	2.935	3.679	0.369
1.764	6.194	2.427	5.731	2.009	1.832	12.720	1.946	1.833	2.781
7.381	8.901	3.221	2.443	4.760	2.886	2.475	2.151	9.755	0.563
2.108	4.156	1.947	1.945	1.802	1.753	2.268	2.397	1.721	4.776
44.436	1.870	1.733	9.032	1.783	3.362	3.346	1.312	35.919	11.173
13.448	2.144	9.431	3.358	4.607	5.175	2.613	36.003	3.407	32.471
15.982	2.217	2.003	3.078	2.180	11.569	1.799	2.353	14.762	1.893
14.458	1.947	15.394	32.430	2.050	3.326	2.982	7.788	1.966	1.895
1.836	1.778	2.010	2.874	1.789	30.841	58.035	1.817	1.723	1.989
9.898	4.801	1.797	1.842	37.234	1.832	1.810	1.780	1.934	3.123
2.301	2.378	2.395	2.560	2.389	1.738	2.515	2.397	6.447	0.201
1.872	2.612	2.010	49.634	2.573	4.748	1.721	2.157	2.664	1.879
2.072	3.636	1.717	4.098	1.848	2.761	4.277	29.231	2.503	1.913
2.008	4.741	9.784	1.855	1.832	24.312	5.502	14.650	1.848	2.006
7.288	31.232	49.514	2.961	2.472	2.183	1.818	2.353	2.069	4.429
2.577	1.882	3.102	2.129	2.038	18.600	1.879	3.188	1.865	3.366
5.337	4.570	1.839	23.091	2.878	2.026	2.212	1.871	6.115	0.073
1.730	4.657	1.775	1.757			2.424	1.760	2.022	1.924
1.823	1.821					35.525	2.344		
7.676	2.031					2.018	1.909		
7.771	1.762					15.569	1.895		
0.275	46.115					2.282	2.717		
1.910	3.888								

Table 10 Throughput measurements (in kbps) for experiments using simulated WWW traffic

Advertisement Rate		No. of Visitors		Latency		Signal Strength		Signal-to-Noise Ratio	
Extended Mobile IP	Original Mobile IP	Extended Mobile IP	Original Mobile IP	Extended Mobile IP	Original Mobile IP	Extended Mobile IP	Original Mobile IP	Extended Mobile IP	Original Mobile IP
24.962	18.785	18.620	15.148	60.407	6.225	15.790	20.155	42.067	2.595
32.090	25.660	8.337	8.556	26.296	33.451	7.794	58.874	34.643	40.210
97.286	20.169	3.353	3.194	35.187	28.463	48.485	13.378	1.724	108.162
11.175	91.986	17.717	2.090	29.552	15.310	64.321	16.759	27.341	14.848
22.139	25.008	45.325	1.720	32.891	38.644	8.824	35.987	1.754	1.832
8.206	38.654	3.921	5.398	21.559	21.529	27.868	23.951	14.487	2.129
43.929	22.466	16.569	4.643	41.322	4.599	32.012	49.887	2.934	63.750
36.485	27.549	10.092	1.771	28.951	35.891	14.292	31.610	45.318	7.481
2.373	16.861	2.378	33.215	11.848	72.793	9.248	8.659	23.162	19.480
32.020	40.194	10.734	25.143	15.459	14.134	61.449	32.200	8.811	17.343
90.625	1.833	6.630	2.428	11.455	26.737	41.862	27.212	5.215	1.733
33.868	49.595	9.584	6.927	16.638	6.046	15.487	2.380	15.748	1.992
13.267	54.568	10.303	45.386	17.366	5.637	49.548	12.617	1.747	133.301
17.612	26.531	6.254	1.783	1.846	43.418	29.375	6.410	33.925	3.030
23.277	53.149	15.335	17.350	14.376	31.129	18.075	47.668	2.401	6.546
2.287	33.814	9.145	23.896	33.523	13.358	30.543	7.645	17.014	2.777
14.153	16.579	12.700	39.914	31.502	7.684	10.070	6.330	13.783	38.326
55.207	16.499	10.644	33.784	26.022	8.670	15.954	37.625	10.428	11.998
31.910	14.832	18.806	20.362	12.860	47.016	2.690	15.877		
20.029	55.063	48.683	12.748	26.277	12.142	10.889	55.071		
23.202	73.803	4.794	37.879						
35.228	41.960	41.060	7.795						

Table 11 Throughput measurements (in kbps) for experiments using simulated FTP traffic

Difference In Throughput (kbps)	Advertisement Rate		No. of Visitors		Latency		Signal Strength		Signal-to-Noise Ratio	
	Freq.	*Cum. %*	*Freq.*	*Cum. %*	*Freq.*	*Cum. %*	*Freq.*	*Cum. %*	*Freq.*	*Cum. %*
-20	3	12.00%	2	10.00%	2	10.53%	2	8.33%	2	10.00%
-10	0	12.00%	1	15.00%	1	15.79%	1	12.50%	0	10.00%
-5	0	12.00%	2	25.00%	1	21.05%	1	16.67%	0	10.00%
0	11	56.00%	7	60.00%	7	57.89%	8	50.00%	7	45.00%
5	4	72.00%	5	85.00%	7	94.74%	7	79.17%	6	75.00%
10	3	84.00%	2	95.00%	0	94.74%	0	79.17%	3	90.00%
20	3	96.00%	0	95.00%	0	94.74%	3	91.67%	1	95.00%
More	1	100.00%	1	100.00%	1	100.00%	2	100.00%	1	100.00%

Table 12 Histogram of differences in measured throughput for WWW traffic

Difference In Throughput (kbps)	Advertisement Rate		No. of Visitors		Latency		Signal Strength		Signal-to-Noise Ratio	
	Freq.	*Cum. %*	*Freq.*	*Cum. %*	*Freq.*	*Cum. %*	*Freq.*	*Cum. %*	*Freq.*	*Cum. %*
-20	7	31.82%	5	22.73%	3	15.00%	5	25.00%	4	22.22%
-10	2	40.91%	2	31.82%	2	25.00%	3	40.00%	0	22.22%
-5	3	54.55%	0	31.82%	3	40.00%	0	40.00%	2	33.33%
0	2	63.64%	4	50.00%	0	40.00%	1	45.00%	3	50.00%
5	0	63.64%	5	72.73%	2	50.00%	3	60.00%	2	61.11%
10	3	77.27%	1	77.27%	1	55.00%	0	60.00%	0	61.11%
20	1	81.82%	2	86.36%	5	80.00%	2	70.00%	4	83.33%
More	4	100.00%	3	100.00%	4	100.00%	6	100.00%	3	100.00%

Table 13 Histogram of differences in measured throughput for FTP traffic

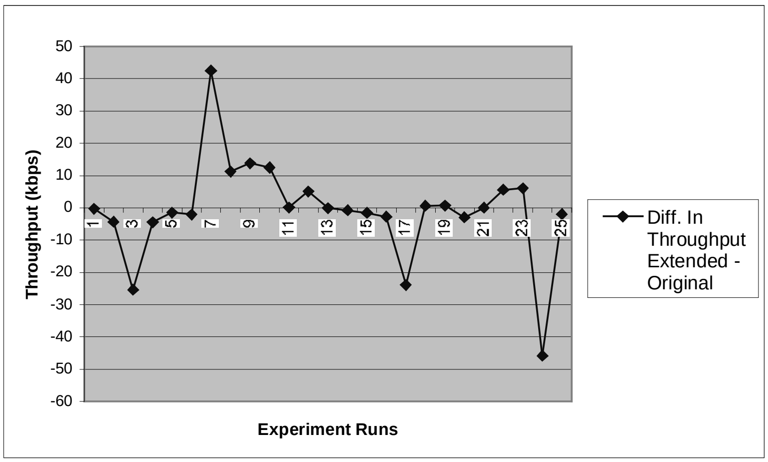

Figure 31 Criterion: Adv. Rate - Differences in measured throughput for WWW traffic

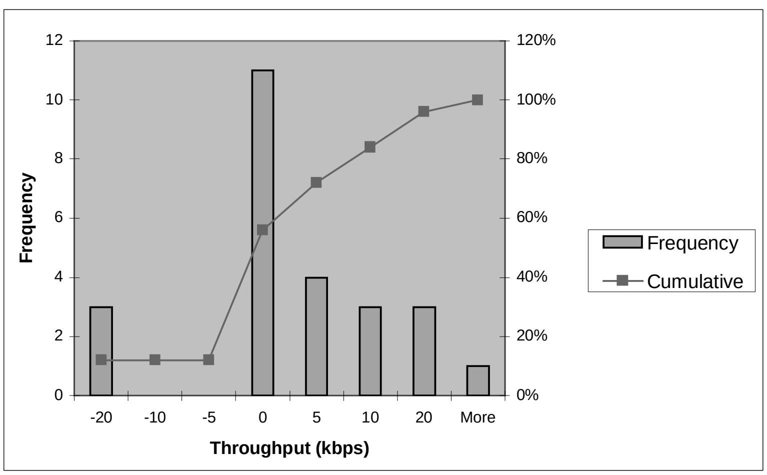

Figure 32 Criterion: Adv. Rate - Histogram of differences in throughput for WWW traffic

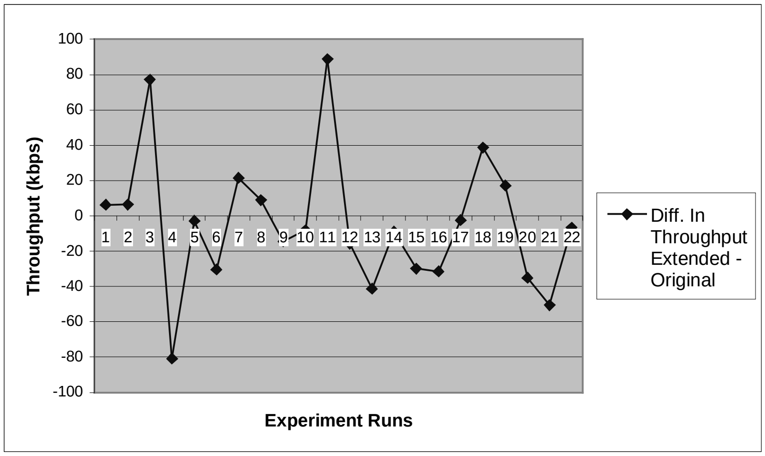

Figure 33 Criterion: Adv. Rate - Differences in measured throughput for FTP traffic

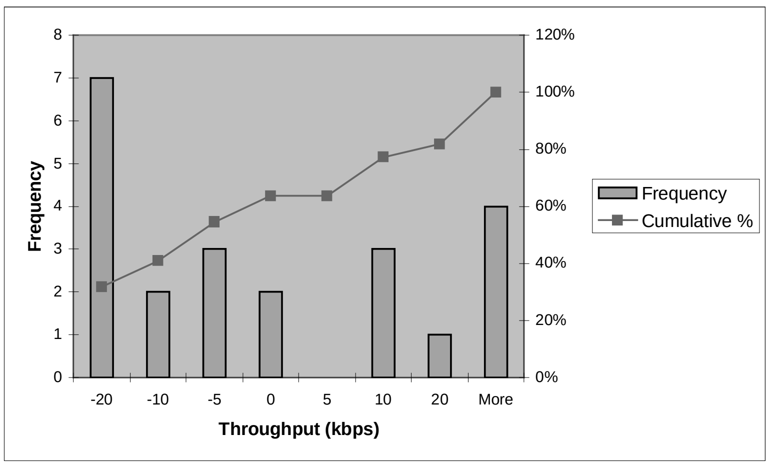

Figure 34 Criterion: Adv. Rate - Histogram of differences in throughput for FTP traffic

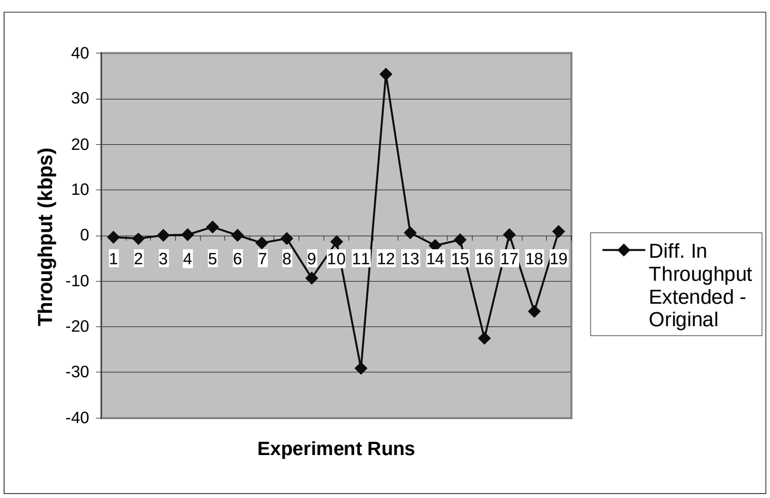

Figure 35 Criterion: Latency - Differences in measured throughput for WWW traffic

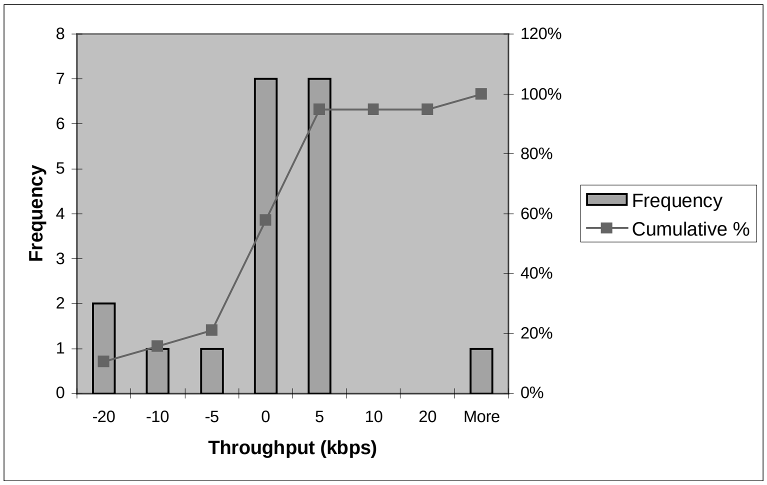

Figure 36 Criterion: Latency - Histogram of differences in throughput for WWW traffic

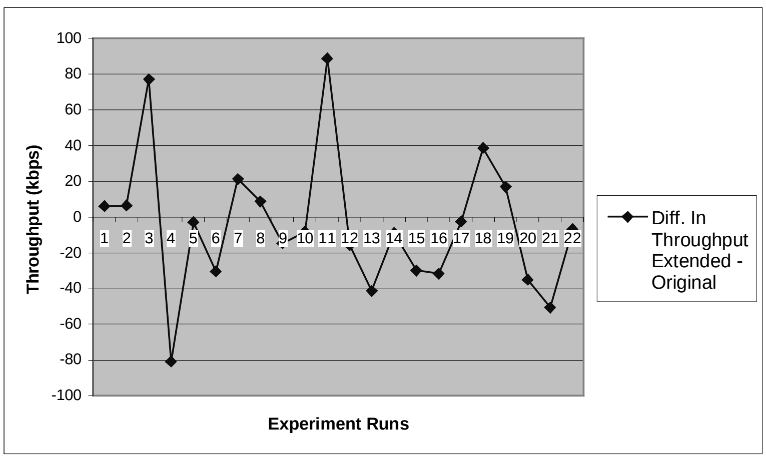

Figure 37 Criterion: Latency - Differences in measured throughput for FTP traffic

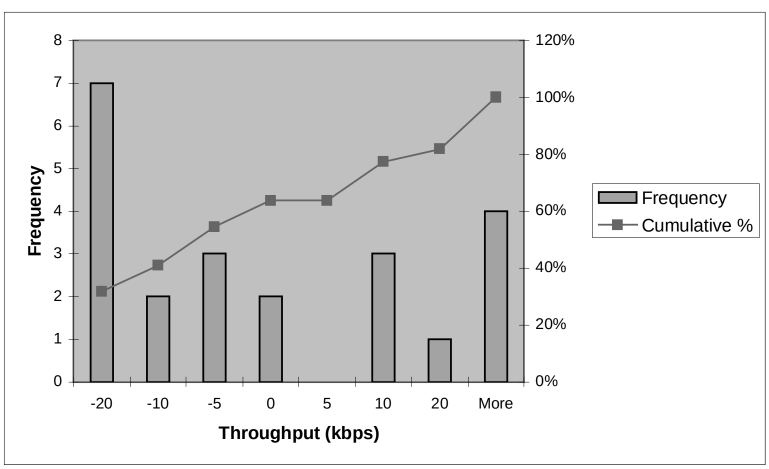

Figure 38 Criterion: Latency - Histogram of differences in throughput for FTP traffic

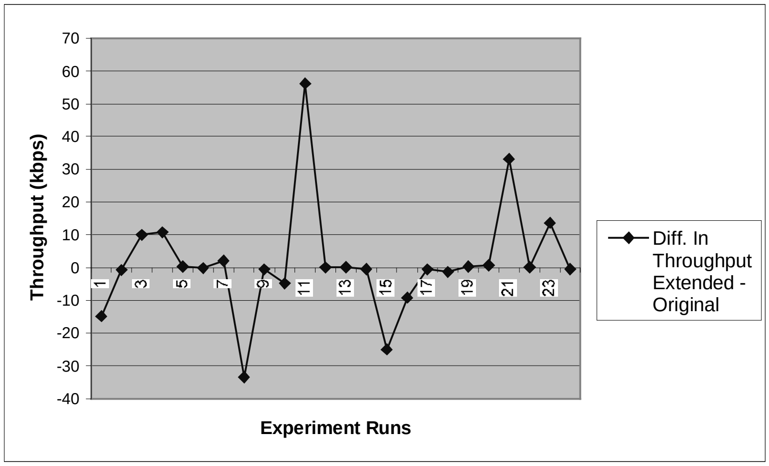

Figure 39 Criterion: Signal Strength - Differences in measured throughput for WWW traffic

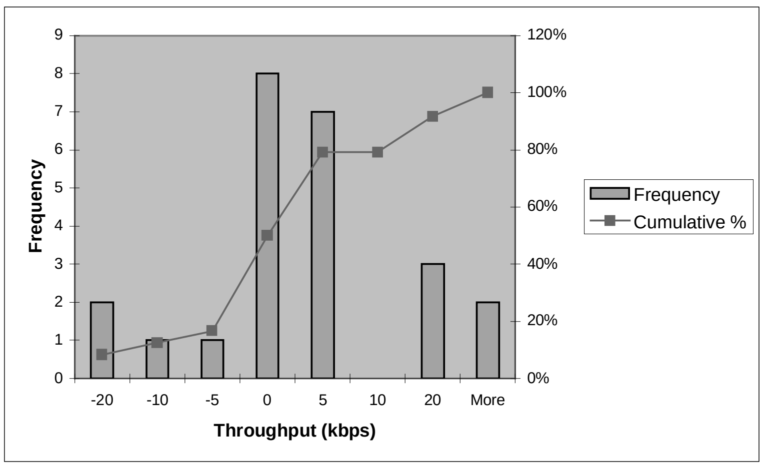

Figure 40 Criterion: Signal Strength - Histogram of differences in throughput for WWW traffic

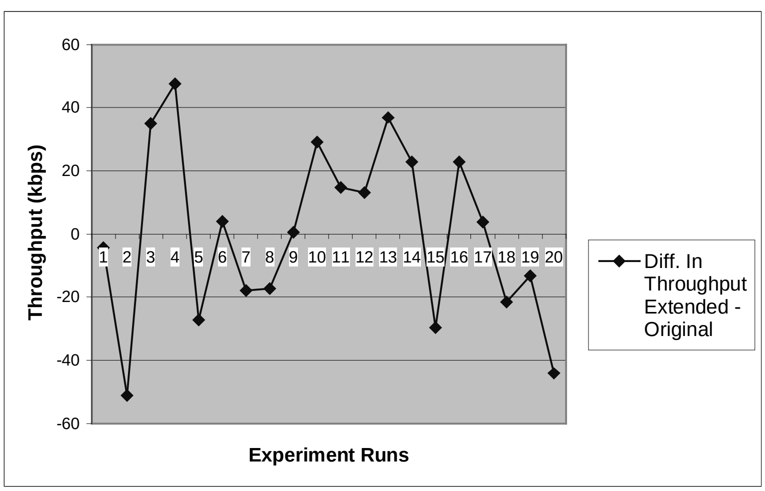

Figure 41 Criterion: Signal Strength - Differences in measured throughput for FTP traffic

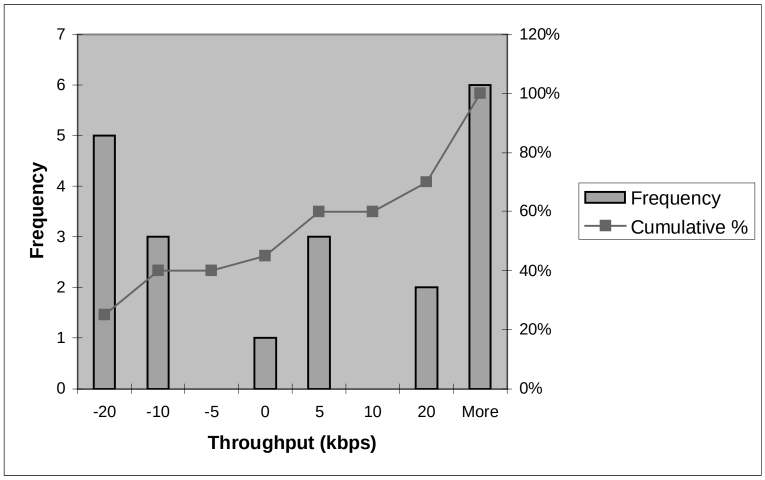

Figure 42 Criterion: Signal Strength - Histogram of differences in throughput for FTP traffic

Detailed NetPerf-Based Experiment Results

Mobile IP Version	Packet Size (Bytes)			
	512	1024	2048	4096
Original	0.44	0.43	0.4	0.43
Adv. Rate	0.47	0.47	0.48	0.48
Latency	0.44	0.44	0.46	0.45
S. Strength	0.46	0.46	0.48	0.45
SNR	0.46	0.46	0.47	0.47

Table 14 Throughput measurements (in mbps) for experiments using NetPerf

Mobile IP Version	Packet Size (Bytes)			
	512	1024	2048	4096
Original	0.48	0.48	0.49	0.49
No. of Visitors	0.47	0.48	0.49	0.5

Table 15 Throughput (in mbps) in NetPerf-based experiments for number-of-visitors criterion

Glossary

Ad-hoc network	a networking infrastructure consisting of mobile routers, and hosts that are free to move around arbitrarily and that are connected via wireless communication.
Advertisement rate	number of advertisements generated by a mobility agent in a given period of time.
Agent advertisement	an ICMP Router Advertisement, modified by attaching an extension specifically defined for advertising the presence of mobility agents.
Agent discovery	the process by which a mobile node detects the presence of any prospective mobility agents.
Agent solicitation	a message sent by the mobile node requesting mobility agents on a sub-network to send agent advertisements.
ARP	Address Resolution Protocol.
Background noise	a random process that affects the signal transmission.
Cache	low-latency storage area used for keeping frequently accessed data.
Care-of address	the end point of a tunnel toward a mobile node for datagrams forwarded to the mobile node while it is away from its home network.
Cell switch	to change the point of connectivity to the network. To register with a foreign agent after being registered elsewhere.
Decapsulation	to receive packets at the endpoint of a tunnel.
DHCP	Dynamic Host Configuration Protocol.
DNS	Domain Name System.
ECS	eager cell switching; accepting service from a new point of attachment to the Internet as

soon as the service is offered.

Encapsulation see Tunneling.

Ethernet A standard for link layer packet transmission; it uses a multiple access method called Carrier Sense Multiple Access with Collision Detection (CSMA/CD), operates at a raw data re of 10Mbps and uses 48-bit addresses.

Exponential backoff mechanism commonly used in link-layer and network-layer protocols for setting up time out periods for packet retransmissions. The time-out period is doubled every time a packet needs to be retransmitted, until a maximum limit.

Foreign agent a router on a mobile node's visited network that provides routing services to mobile nodes while registered; the end point for tunnels established by a home agent to forward packets to a registered mobile node's care-of address; selected as a default router by registered mobile nodes.

Foreign network Any network other than a mobile node's home network.

FTP File Transfer Protocol.

Gratuitous ARP a broadcast, unrequested ARP reply to initiate the resolution of a mobile node's home address, expecting that nodes receiving the broadcast will update their ARP caches.

Handoff see Cell switch.

Home address an IP address that is assigned for an extended period of time to a mobile node; the address remains unchanged regardless of where the node is attached to the Internet.

Home agent a router on a mobile node's home network that tunnels datagrams for delivery to the mobile node when it is away from home and maintains current location information for the mobile node.

Home network a network that has a network prefix matching the mobile node's home address.

HTTP Hyper-Text Transfer Protocol.

Hysteresis region a region, in a wireless scenario with two foreign agents A and B, delimited by two lines. One of the lines represents the boundary where A's signal strength is higher than B's by a certain level (X dB for example). The other line indicates the boundary where B's signal strength is higher than A's by X dB.

Hysteresis techniques	cell switching mechanisms that use a hysteresis region to trigger the switching from one foreign agent to another one.
ICMP	Internet Control Message Protocol.
IETF	Internet Engineering Task Force.
IP	Internet Protocol.
IP broadcasting	to send an IP datagram to all the hosts on a subnetwork.
IP multicasting	to send an IP datagram to only a group of hosts on a network.
IPv4	IP version 4.
IPv6	IP version 6.
LAN	Local Area Network.
LCS	Lazy Cell Switching; waiting as long as possible to change to a new point of attachment.
Link latency	the amount of time it takes for a byte transmitted by a protocol layer on a source machine to reach the same protocol layer on a target machine.
Mobile IP	Mobility extensions to the IP protocol.
Mobile node	a host capable of changing its point of attachment from one network to another.
Mobility agent	either a home agent or a foreign agent.
Mobility binding	the association of a home address with a care-of address, along with the remaining lifetime of that association.
MPEG	Moving Picture Experts Group; family of standards used for coding audio-visual information (e.g., movies, video, music) in a digital compressed format.
MTU	Maximum Transmission Unit; the maximum size of a packet that can be transmitted on a link-layer link without fragmenting.
Multi-hop network	a network infrastructure comprised of at least two different sub-networks linked together by a router.
Nonce	a randomly chosen value, different from previous choices, inserted in a message to defend against replays.
Path MTU	the maximum size of a packet that can be transmitted over the path from one network endpoint to another endpoint without fragmenting. The minimum of all the MTUs for all

links along the path.

Proxy ARP use of ARP by one network node to impersonate some other network node by pretending to own a link layer address that is associated to other's IP address.

Registration the process that occurs when a mobile node is away from its home network and it registers its care-of address with its home agent.

RFC Request For Comments.

Router advertisement an ICMP message issued by routers to offer their routing services on a link.

Signal strength the detectable power of the signal carrying the data bits, as seen by the receiver of the signal.

Signal to noise ratio difference in strength between a true signal and background noise.

TCP Transmission Control Protocol.

Throughput the amount of data from a source to a destination processed by the protocol for which throughput is to be measured.

TLV format Type-Length-Value; a generic idea for organizing data into a collection of functional compartments or protocol structures.

TTL field Time-To-Live; a field in the header of an IP datagram, which sets an upper limit on the number of routers through which the packet can pass.

Tunnel the path followed by a datagram while it is encapsulated.

Tunneling bypassing the normal IP routing of a packet by enclosing (encapsulating) it within another IP packet addressed to an alternate destination computer.

UDP User Datagram Protocol.

Visited network a foreign network.

Visitor list the list of mobile nodes visiting a foreign agent.

WWW World Wide Web.