Routing Support for Mobile Computing

A Thesis Submitted to the College of

Graduate Studies and Research

in Partial Fulfillment of the Requirements

For the Degree of Master of Science

in the Department of Computer Science

University of Saskatchewan

Saskatoon, Saskatchewan

by

Wayne L. Mackrell

September 1998

Permission To Use

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
University of Saskatchewan
Saskatoon, Saskatchewan, Canada
S7N 5A9

Abstract

Mobile computing represents a growing trend in the use of computers worldwide. In addition, network connectivity is now being extended to devices (other than mobile computers) that frequently change location. As a result the Internet has to provide adequate support for mobility. Mobile IP is a protocol that has been developed to provide this support.

Mobile IP is designed to provide mobility support transparently at the network layer. As a result, mobile nodes can keep the same address regardless of what network they are connected to in an internetwork. Mobile IP accomplishes this through the use of forwarding agents that intercept traffic at the mobile node's home network and forward it to the mobile node at its actual location.

Much of the research effort in Mobile IP has been concerned with designing and implementing the protocol. Little experimental evaluation of the performance impacts of Mobile IP has been done. In this thesis, a working implementation of Mobile IP is studied in a test-bed environment. This setup allows some of the performance implications of Mobile IP to be studied in controlled experiments.

Round trip times, delay, and network load are some of the performance issues examined in this thesis. Round trip times are studied by sending data to a correspondent node and then measuring the time for the data to return to the mobile node. The delay measured is the time between data arriving at an agent and the time that data is forwarded from the agent. Delays are measured at both the home agent and foreign agent. Network load is studied by examining packet traces and measuring the amount of Mobile IP traffic.

The thesis also examines how effectively Mobile IP provides transparent mobility support for applications and higher-layer network traffic. FTP, NFS, X-windows, telnet, and ping are used to provide packet traffic to the mobile node. The results of this investigation indicate that Mobile IP is a promising approach for mobility support in an internetwork, but there are costs in terms of performance.

Acknowledgements

I am indebted to many people for helping along the way.

I wish to thank Rick Bunt, my supervisor, for his wisdom, guidance, patience, quick reviewing, valuable comments, and for giving me the chance to pursue this degree. I also thank the members of my commitee: Carey Williamson, Derek Eager, and Bob Gander for their valuable input to this thesis.

I would like to thank Greg Oster and Tim Harrison for helping me with setting up the testbed and answering my silly questions. I thank Kevin Froese for running around for me while I was in Calgary.

I also thank Jean Paul Tremblay and Winfried Grassmann for their support and guidance during my university career.

I appreciate the support I received from TR-Labs, Hewlett Packard, and the University of Saskatchewan. Without support this would not have been possible.

I thank my children Riley and Cassandra for giving me a good reason to succeed and bringing a lot of pleasure into my life. I thank Teresa Mackrell for being an important part of our family and a good friend.

I thank my parents for teaching me the value of hard work, and helping out whenever they could. I also thank my Aunt Margaret and my grandmother who encouraged my university career.

I thank Bart for listening to my rants and acting as a sounding board all these years.

I especially thank my best friend, co-worker, and favorite proof-reader, Christy Kenny. Thanks for all your help, for supporting me when I really needed it, and for having confidence in me.

Table Of Contents

List of Tables

List of Figures

Chapter 1

Thesis Overview

1.1 Problem Introduction

When the Internet and its protocols were originally developed, computers were large machines that were not easily moved. Even the desktop computers that rapidly gained popularity were meant for stationary use, and commonly stayed in the same location for most of their usable existence. As a result, the internetworking protocols were developed to easily support the delivery of information to static physical locations possibly large distances apart. Little thought was given to supporting computers that may roam among different networks and move large distances while trying to still maintain their connections and services.

Since the early days of the Internet, things have changed substantially. Small portable computers appeared soon after the desktop computers appeared, but there was not much concern about connecting these devices to the Internet as internetworking was not popular for personal computing. As more and more people became exposed to the Internet, there was a growing desire to access e-mail and other services not only from desktop computers, but also from *mobile computers*. The development of the world wide web [14] quickly accelerated the interest of private individuals in the Internet, and the desire for Internet services on all types of computing devices has exploded.

At present, a large proportion of the computers sold are portable rather than desktop. In addition, there are plans for expanding Internet connectivity to include all kinds of devices, such as personal assistants, vehicles, and practically any device that may need to exchange information with another device. As a result, the old idea of an internetwork consisting of many statically located physical computers is no longer valid. In addition, however, the Internet already encompasses millions of computers and to change the basic infrastructure to accommodate mobility is not desirable because of the massive changes that would be needed worldwide. As a result, the challenge becomes one of supporting mobility within the existing architecture. A good overview of the challenges in mobility support can be found in [9].

Mobility support is crucial for the Internet, whose population of mobile and portable computers is rapidly growing. Mobile IP [21] is a network layer protocol developed by IETF (the Internet Engineering Task Force) to provide *transparent addressability* for mobile nodes as they roam over potentially large physical distances. The Mobile IP protocol has been accepted as a proposed standard by the IETF and is documented in RFC 2002. RFCs (requests for comments) are used by the IETF to document approved standards that are to be used on the Internet. The protocols TCP (Transmission Control Protocol) and IP (internet Protocol), which are used widely throughout the Internet, are themselves documented in RFC 791 [25] and RFC 793 [26]. Details on Mobile IP are given in Chapter 2.

When a mobile node is away from its home network, it needs assistance to exchange information with other nodes, hereafter called *correspondent nodes* because they correspond with the mobile node. Transparent addressability means that a mobile computer can exchange information with other computers regardless of its location, without requiring a new IP address and without any application-level programs or computers with which the mobile node communicates being aware of the location change. This represents a significant step in mobility support, because the typical protocols used for delivering messages between computers require fixed locations in order to correctly deliver the data.

This thesis examines Mobile IP and its performance on an experimental testbed. Performance is examined in terms of both general operability and performance measurements taken on actual traffic streams. The effect of Mobile IP on higher layer protocols is also examined, as are certain scalability issues. This chapter outlines how this exploration of Mobile IP will proceed.

1.2 Research Goals

1.2.1 Introduction

The introduction of Mobile IP to the internetworking fabric will impact performance in several ways. In order to determine the performance characteristics of Mobile IP, the research is based on specific goals. These goals are presented here to introduce the experiments that were performed and to provide a structure to the presentation. The results presented later, based upon these goals, show the impact of Mobile IP on the end user, the networks and the computers involved in providing Mobile IP connectivity.

1.2.2 Operability of Mobile IP

The first task was to set up an experimental test-bed consisting of multiple computers and two separate networks (see Chapter 3 for details). With this accomplished, the workability of the test-bed and the Mobile IP protocol were assessed. Mobile IP operability was tested with the following criteria: reachability on the mobile node's home network, reachability on a foreign network, correct handling of movement, correct handling of multiple mobile nodes, and the correct operation of various protocols over the Mobile IP link. The goal of the operability testing is to show how well Mobile IP handles the general operation of a mobile node in the test-bed environment.

1.2.3 Movement Transparency

A key component of Mobile IP is its ability to support connections to mobile nodes while they are changing location. A goal of this thesis is to determine the effectiveness of the Mobile IP approach at providing this service. The performance of several applications is monitored to observe any ill effects from the movement of the nodes on which they are running.

1.2.4 Throughput of High Level Protocols

Mobile IP is designed to be transparent to applications and higher layer protocols, but issues affecting the performance of Mobile IP invariably affect the data streams being transferred through higher-layer protocols as well. Two important aspects of application usability are responsiveness and raw data throughput. The responsiveness of applications receiving data over the network is affected greatly by any delays experienced by the individual packets being transferred. Raw data throughput is important for bulk data transfer of information.

In this thesis, bulk data transfer is explored using two protocols: FTP (file transfer protocol) and NFS (network file system). FTP transfer is the most common approach for bulk transfers in the Internet. Therefore, FTP has been designed to account for congestion and other problems that may be encountered and achieve fast data transfer, yet not use up all of the bandwidth. Much of this functionality comes from TCP which is used by FTP for transferring data. The goal is to determine how the performance of FTP (and TCP) is affected by the Mobile IP link.

NFS is commonly used for mounting filesystems on local area networks, and therefore was not designed with wide area networks in mind. NFS commonly uses UDP (user datagram protocol) and therefore is connectionless, although it can be configured to use TCP. Since it is used for mounting filesystems, there are many operations performed by NFS, but only bulk data transfers (initiated by a copy command from a mounted filesystem to a local filesystem) will be studied in this thesis. The goal is to determine how effective NFS is over Mobile IP, and to compare the NFS performance to the FTP performance.

1.2.5 Packet Delays at Mobility Support Agents

In certain applications, throughput may not be as important as the delays that are experienced by packets being routed through the internetwork. In particular, real-time applications need timely delivery of packets in order to hide the distance that the information is travelling from the end user and make the response seem as instantaneous as possible. This timely delivery of the packets is affected by packet loss (and subsequent re-transmission) and delays. Since there is minimal packet loss in the test-bed, this research focuses on delay.

The delays at the mobility support agents are of most interest since these are the only new routing entities in the system. Any delay characteristics shown for the mobility agents should exist regardless of the physical distance between the corresponding nodes. These delays occur at the home agent and at the foreign agent. The goal of this research is to discover the magnitude of any delay experienced, and the types of traffic flow that most encourage delay.

1.2.6 Round Trip Time

Responsiveness is studied by measuring delays and round trip times. Round trip time is determined as the time between a response request being sent out by the mobile node, and the arrival of the response at the mobile node. The time it takes for the response to arrive back at the mobile node is a good indication of the response time users are likely to experience with interactive applications. Ping traffic is ideal for measuring this type of performance and is used here. The goal is to use the ping data as a model for more complex data and estimate expected round trip times.

1.2.7 Overhead of Mobile IP

One area of interest with a new protocol is the overhead that the protocol places on the entities involved with its implementation. In particular, the bandwidth used on both the home and foreign networks is examined. These totals are compared to similar traffic being transported without Mobile IP. The goal of this research is to show the overhead in terms of extra traffic required to deliver packets to mobile nodes. The network overhead of Mobile IP is studied with both FTP and NFS bulk data transfers.

Other aspects of overhead involve the load on the agents and on the mobile node itself for protocol processing. These overheads are not studied in this thesis, but issues regarding the internals of the protocol have been investigated in [4].

1.2.8 Scalability with Increasing Mobile Nodes

Increasing the number of mobile nodes is bound to affect the performance of the mechanisms designed to support mobility. This research examines how Mobile IP performance scales with the number of mobile nodes that are in the system. All of the mobile nodes examined belong to a single home agent and migrate to the same foreign network. Throughput and packet delays are both studied as the number of mobile nodes increases from 1 to 20. Differences between forwarding traffic to many mobile nodes and a single mobile node are also studied. The goal of this research is to determine how Mobile IP performance scales with the number of mobile nodes being serviced.

1.3 Outline of Thesis

Chapter 2 provides the background information that is needed to proceed with the thesis discussion. The network model is described and the issues differentiating mobile internetworks from static internetworks are presented. Some of the solutions being developed for mobility support are also introduced, including IPv6 which has new mechanisms designed specifically for mobile nodes.

Mobile IP itself is also introduced in Chapter 2. A brief introduction to the protocol and its terminology is given. The mechanisms of the protocol are also described to show how the entities involved interact to track and provide connectivity for the mobile node.

Mobile IP is only one area of research within the larger field of mobility support. Mobility research groups are very active, and a lot of work is being done in this area. Chapter 2 also introduces some of this research, and some of the interesting topics being studied with respect to Mobile IP and mobility support in general.

Chapter 3 gives the details of the Mobile IP test-bed used in this work. The test-bed is described in terms of the hardware, software, and network configuration. This setup enables the execution of many experiments, but also restricts the types of issues that can be investigated. The design choices and goals of the test-bed are also discussed in this chapter.

Chapter 4 outlines the experiments performed. These experiments are presented by examining the metrics, experimental factors, and traffic loads. The purpose behind each experiment and a more detailed presentation of what the experiment is meant to achieve are also provided in this discussion.

Chapter 5 presents the results obtained from the experiments described in Chapter 4. Both qualitative and quantitative results are presented in both tabular form and a graphical form.

A summary of the thesis and its results is presented in Chapter 6. This chapter also presents the conclusions that can be drawn from some of the observed results. Finally, suggestions for future work in this area are presented.

The thesis also contains a glossary of terms used in the thesis. Many of the terms are familiar to anyone working with networks, but some are specific to Mobile IP and the implementation used here. The glossary is included as appendix A.

Chapter 2

Background

2.1 Introduction

Mobile IP is an extension to the popular IP networking protocol designed for routing in an internetwork. Mobile IP has been designed to provide network support for mobile nodes that travel across network boundaries, a type of movement not supported by IP. This chapter gives a brief introduction to Mobile IP and internetworking in general, along with issues presented by mobility and solved by Mobile IP.

A model for internetworking is presented which describes how the entities being studied fit together. This presentation includes the routing layers that are used to describe packet delivery at different levels of abstraction from the hardware. Both OSI and TCP/IP internetworking models are presented, but the TCP/IP protocol suite is the focus of this work.

Mobile IP is introduced by discussing the problems that mobility presents for internetwork connectivity. The discussion continues with how Mobile IP attempts to solve these problems. The details of the Mobile IP protocol are presented next. These details include descriptions of the support agents that are used, the method of packet forwarding used, and how the entities work together to provide the networking support. This discussion includes descriptions of the specific support services provided by Mobile IP.

Related research is also presented in this chapter. Related research includes research into the Mobile IP protocol, and research into mobile computing in general.

Research into multicast support with Mobile IP, enhancements to the basic protocol, and performance analysis of the protocol are some of the topics discussed. This chapter also introduces some of the terminology used for describing Mobile IP and internetworking in general.

2.2 Internetwork Model

At the heart of the Internet are the computers and networks of which it is made. In this thesis, the computers in an internetwork are referred to as *nodes*. Computers which change location (from their *home* network to some other network) are referred to as *mobile nodes*.

Mobile nodes present different problems depending on the type of media they are using for connection, and the distances that they travel. A mobile node moving a small distance on an Ethernet, or other broadcast network, does not pose any significant problems, unless it is disconnected for a long period of time. When network boundaries are crossed, however, support agents must track the mobile node and communicate with each other in order to maintain the node's network connectivity.

If a mobile node is on a wireless network, it can move over small distances without disconnecting from the network itself. This requires support agents with fixed network connections that can communicate using the wireless interface to the mobile node. Although areas of control are crossed, and some support mechanisms are required, this type of movement is handled by the broadcast nature of many local area networks.

Some of the issues with local movement, however, also apply to mobile nodes travelling over large distances. For example, internetwork movement also requires support agents for tracking the mobile node. The support agents for internetwork mobility must be more complex because the internetwork routing does not use the broadcast approach, but instead uses the address of the node itself for determining the delivery location. Therefore, when the mobile node changes location, information

Table 2.1: The OSI and TCP/IP Protocol Layers

OSI Layer Name	**TCP/IP Layers**
Application	
Presentation	
Session	Application Layer (FTP, NFS, etc.)
Transport	Transport Layer (TCP, UDP)
Network	Network Layer (IP, ICMP)
Data Link	
Physical	Link Layer (Device driver, and interface card)

can no longer reach it using its fixed location address, and a system must be set up for gathering information at the mobile node's home location and forwarding it to the new location. These issues are explored further in Section 2.4.

Mobility issues relate to the protocol layers which are used to support network connectivity at different distances and for different purposes. Commonly used network layers are shown in Table 2.1. This table shows both the OSI [28] and the TCP/IP protocol layers [7].

The main areas of interest are the lower layers which are closest to the physical medium being used. The physical layer is used to describe the physical devices and equipment used to connect the computers on the network. This includes any network cards and cable, or wireless interfaces used to transmit signals between computers.

The data link layer contains the protocols that are used to deliver information between nodes within a network. Protocols at this layer handle media access, collisions, and corrupted data, and other concerns that are directly related to the physical network itself. The networks considered in this research are shared medium networks in which any node can directly send information to any other node on the network. In other networks, nodes can also directly send information to other nodes on the network, but the medium may not be shared directly. Mobility within such a network therefore does not cause connectivity problems since as long as the node remains within the single network, it is effectively connected directly to every other node in the network and can communicate using its same network address at the different location.

The network layer contains the protocols which handle connection between nodes both within a network and across internetworks. The addresses used to identify nodes at this level commonly encode both the location of the network to which the node belongs and a number identifying the node on that network. If the node wishes to communicate with another node on its *home* network, then the packets it transmits stay within the network and are routed by the data link protocols. Packets of information destined for nodes outside the local network require extra routing and the location information encoded within the network layer address is used to deliver the packet to the destination.

The network layer in the OSI model corresponds to the IP layer in the TCP/IP protocol suite. The IP protocol was first introduced in 1974 by Vint Cerf [1], and has become the most widely used network protocol for delivering packets across internetworks. IP provides a connectionless packet delivery system independent of the underlying network hardware. Mobile IP is an extension to IP which provides additional mechanisms that remove the location restrictions on the IP addresses that are used for packet delivery. Mobile IP is designed to provide transparent mobility support across network boundaries for the protocol layers above IP.

The transport layer contains protocols which use the network layer protocols for packet delivery, but add additional features such as congestion control, connections, and the ability to distinguish between individual processes running on another node. The TCP and UDP protocols are the transport layer protocols in the TCP/IP protocol suite, and are the ones considered in this thesis.

The application layer and higher layers contain protocols for handling issues increasingly further removed from the direct delivery of packets, such as interfacing with the user, translating information which exists in different forms on different nodes, and setting up and managing connections between processes on different machines.

2.3 Internetwork Routing

The differences between network and internetwork routing are shown in Figure 2.1. This figure shows the requirements for delivering a packet to another node where the node is on the same network (Figure 2.1(a)) and a different network (Figure 2.1(b)). For simplicity, it is assumed that the networks being shown are shared medium networks.

In the case where a packet is delivered within a single network, the network layer protocol at the source node (A) recognizes that the packet is not destined for another network, and passes the packet directly to the data link layer which maps the network layer address to a data link address. When the mapping has been made, the packet is transmitted onto the shared medium. Every host on the network examines the packet and ignores it if the address is not its own. The destination node (B) will recognize its own address and pass the packet up to the higher protocol layers for processing. Location changes do not affect this delivery because regardless of where the destination node is located, it will always receive the packet and recognize its own address.

When a packet is sent from the source node (A) to another network, it may be accepted by and forwarded through several nodes on several different networks before it reaches the destination node (B). These nodes which forward the packets are known as *routers*. Routers are special nodes which have interfaces on more than one network and can forward packets from one network to another if required.

In Figure 2.1(b), a packet is sent from network A to network D. In order to get to network D, the packet must travel through networks B and C. This is done through local routing between routers within the different networks. The exact path and forwarding operations are traced in the following steps:

1. The source node realizes that the packet is destined for a node outside the local network and finds the appropriate router to which the packet should be forwarded.

2. The packet is forwarded to the router using data link protocols.

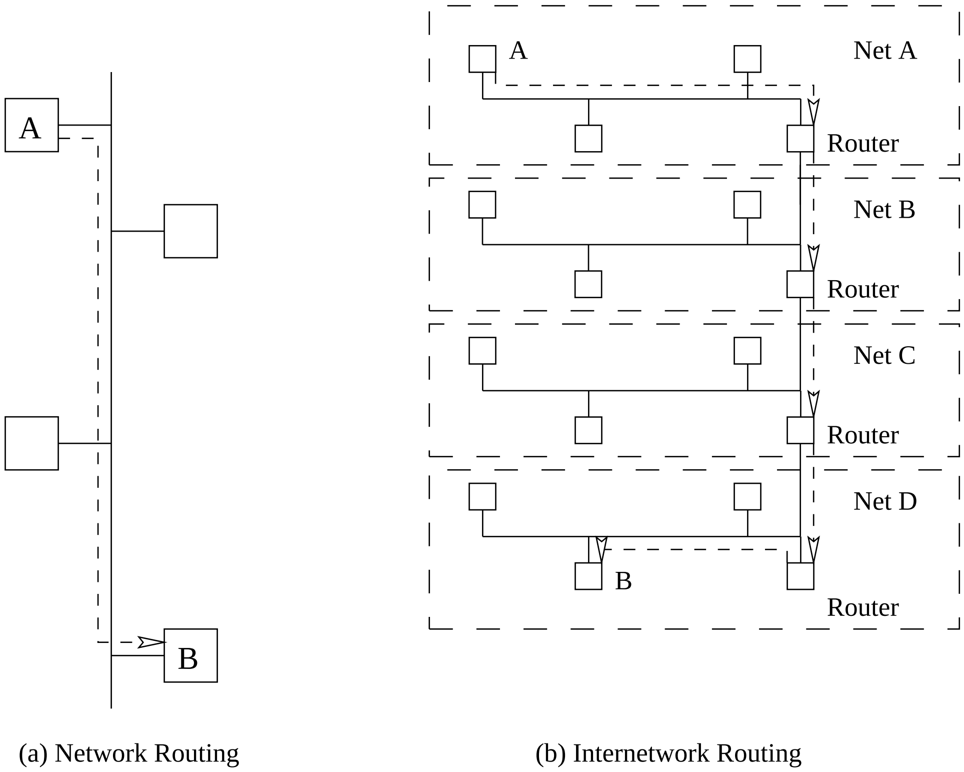


Figure 2.1: Network Routing vs. Internetwork Routing

3. The router accepts the packet and recognizes that the packet must be forwarded on another network.

4. The router determines which network to forward the packet onto, and which router on that network should receive the packet.

5. The source router forwards to the destination router which repeats the process above.

6. When a router receives the packet and notices that the destination address maps to its own network, then a lookup is done for the local address of the destination node.

7. The packet is delivered to the destination node (B) on the destination network from the router using data link layer protocols.

Table 2.2: IPv4 Address Classes

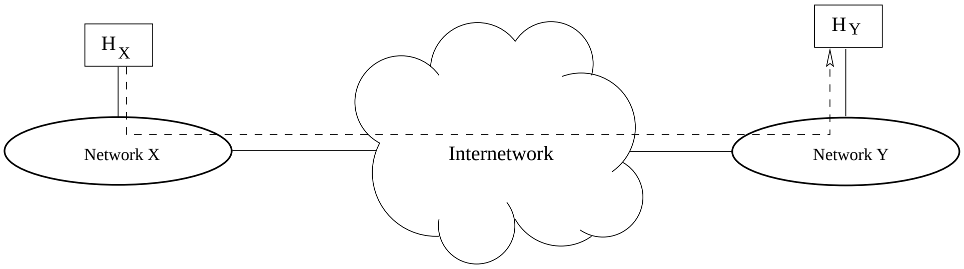
Class	Addresses
A	1.h.h.h - 126.h.h.h
B	128.0.h.h - 191.255.h.h
C	192.0.0.h - 223.255.255.h
D	224.h.h.h - 239.h.h.h

The route across the networks and through the routers is indicated by the small-dashed line in Figure 2.1(b). Network boundaries are indicated by the large-dashed lines. Note that the routers are the only nodes that have direct connections to computers on more than one network.

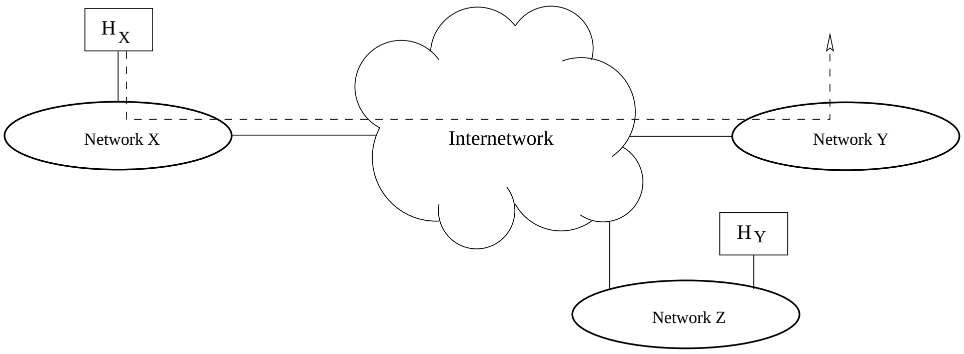
A key to the internetwork routing lies in encoding location information in the address of the node itself. In the case of IP routing, the address is a 32-bit number which is divided into both a network address and a node address. The different types of IP addresses are shown in Table 2.2. Class A addresses have 8 bits for the network address and 24 bits for the host address, class B addresses have a 16 bit to 16 bit split, and class C addresses have a 8 bit to 24 bit split. The fourth type of address, class D, is reserved for multicasting, where many nodes are identified by a single address, and, as a result, special routers are required.

2.4 Mobility Problems

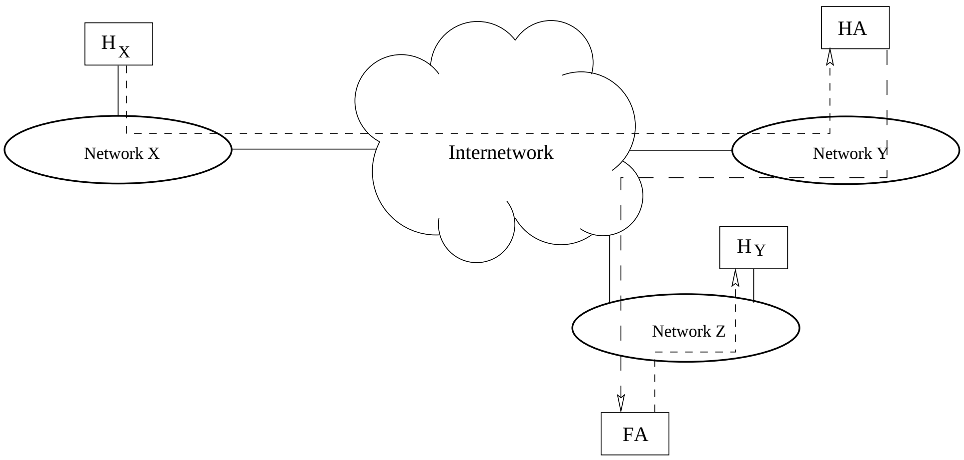
Mobility affects internetwork routing because the routing depends on the location being specified in the destination node's address. When a node moves across network boundaries (connecting to some *foreign* network), the network number in its address no longer corresponds to its location, and therefore communication to this node is impossible using its home address. To make the situation even worse, most applications rely on internetwork addresses for identifying nodes even for local traffic. Therefore the mobile node has difficulty even communicating with nodes local to the foreign network because all IP traffic still gets routed to the home network.



(a) Mobile Node at Home (on Network Y)



(b) Mobile Node Away from Home (on Network Z)



(c) Mobile IP Forwarding

Figure 2.2: Delivering Packets to a Mobile Node

The mobility problem is illustrated in Figure 2.2. Figure 2.2(a) illustrates a packet being delivered to a mobile node (H_Y) from a correspondent host (H_X). The router details have been left out of this figure, but what is shown is that the packet leaves H_X, traverses network X, and then traverses a collection of networks en route to network Y where the packet is delivered to the mobile node H_Y.

In Figure 2.2(b), the mobile node has moved to network Z. The dashed line shows the path taken by any packet destined for H_Y. Since the network encoded in the address is network Y, all of the packets are delivered to network Y even though the mobile node is no longer at that network. As a result, it is impossible for the mobile node (now resident on network Z) to receive any packets.

Mobile IP adds two more nodes located on networks Y and Z. These nodes are responsible for intercepting packets on network Y and forwarding them to network Z. This is shown in Figure 2.2(c). The Mobile IP nodes are labelled HA and FA, abbreviations for *home agent* and *foreign agent*, respectively.

When a packet arrives on network Y, the HA impersonates the mobile node and intercepts the packet. The HA then forwards this packet to the FA on network Z using a technique called *encapsulation*. When the FA receives the packet it is decapsulated and delivered to the mobile node. Details on this mechanism and other aspects of Mobile IP are given in the next section.

2.5 Mobile IP

In the 1990s, much research effort has been applied to the specification of extensions to the TCP/IP protocol suite to support mobility across networks. Some of these approaches have advocated changes to the routers which exist in the Internet today, and rely on these changes for the full functionality. Other approaches, such as IETF Mobile IP [21], are based on having a stable home base or network which can forward information to a recognized foreign network where the host is residing. These last approaches can exist in an Internet where routers know nothing about mobility.

IETF Mobile IP is the product of much discussion by the IETF Mobile IP working group. The Internet Engineering Task Force (IETF) is a collection of researchers organized into working groups for the purpose of exploring issues related to the Internet. The purpose of the IETF is to study problems in the Internet, propose solutions to those problems, and evaluate the proposed solutions in a public forum. Because of the public nature of the IETF, many of the developers of earlier mobility support protocols were involved in the development of Mobile IP, and the protocol incorporates ideas from many sources.

The IETF Mobile IP proposal is based on three main entities: a mobile node (MN), a home agent (HA) and a foreign agent (FA). These entities operate on different networks called the home network and foreign network. Nodes exchanging packets with MNs are called correspondent nodes (CN).

When mobile nodes roam the internetwork, they connect temporarily to foreign networks. When a MN connects to a foreign network, it is required to register with a foreign agent. The FA is responsible for forwarding this registration to the home agent. The FA is also responsible for accepting redirected packets from the HA and delivering them to the MN. The FA usually acts as the default router for the MN as well, which means it is involved with the traffic flowing both to and from the mobile node.

When a MN is away from its home network it informs its HA of its location and an address on the foreign network (care-of address) to which the HA can forward packets. Once a MN has registered, the HA is responsible for telling all other hosts on the home network that it will handle the MN's packets. This is done on Ethernets using a technique called *proxy ARP* [21].

Once the registration has taken place, and the home network routing is set up, the HA can begin intercepting packets bound for the MN on the home network. These packets are encapsulated and forwarded to the care-of address. In addition to forwarding packets, the HA is responsible for maintaining security information about the MN and forcing location updates to be authenticated using this security information.

There are two modes in which the IETF Mobile IP can operate: FA mode and pop-up mode. FA mode requires that there be a special node on the foreign network that is willing to accept packets on the MN's behalf using its own IP address and deliver any forwarded messages to the MN on the foreign network. Pop-up mode works without foreign agents, but requires mechanisms to be present on the foreign network to automatically assign addresses to new nodes. In this mode, the mobile node itself is responsible for receiving and decapsulating the forwarded packets. This research focuses on the FA mode of operation.

2.5.1 Support Services

In order to handle mobility, an internetwork needs to support a node when it is connected, but must also track the node when it changes location. While connected, the IP routing mechanisms enhanced with the Mobile IP protocol handle communication effectively. However, location changes require additional support outside of the routing domain. This support involves tracking new locations, keeping track of agent states, and updating agent states when mobile node connectivity changes occur. Location change support is handled in Mobile IP by *agent discovery* mechanisms and *registration services*. This section will discuss both of these services.

Agent discovery supports communication between the MN and agents on a network to which it is connected. Mobility supporting agents (HAs and FAs) periodically broadcast *ICMP (Internet Control Message Protocol) router advertisement messages* which contain mobility information. When a MN receives an advertisement message, it either updates the lifetime of its foreign agent entry, confirming its location, or recognizes that the router message has arrived from a network different from the one to which it is registered, and initiates a new registration.

Since wireless connections are possible, a MN can receive advertisements from two different networks at the same time (if the MN is within range of two different wireless access points). It is the MN's decision as to which network it registers on in this situation, and different selection strategies have been debated in the IETF forum. But, if the MN can no longer contact its old FA and if it is receiving

Table 2.3: Agent Advertisement Fields

Field	Description
Type	16 (the number for router advertisements)
Length	The length is (6+4*N) where N is the number of care-of addresses advertised.
Sequence number	Count of advertisements sent since agent started.
Registration lifetime	Maximum length registration acceptable
R	Registration required, no co-located addresses offered.
B	Busy, will not accept registrations.
H	Home agent services offered.
F	Foreign agent services offered.
M	Minimal encapsulation supported.
G	GRE method encapsulation supported.
V	Van Jacobson header compression supported.
Reserved	Reserved for future use.
Care-of addresses	A list of addresses MNs may use on this network.

advertisements from another network, then a network movement is indicated and registration procedures must proceed on this new network.

Agent advertisement messages contain many fields, which are shown in Table 2.3. Two important fields are registration lifetime and care-of addresses. The registration lifetime sent in the message is the maximum registration time that the agent is willing to support. These are the lifetimes that are used to time-out entries in the FA's visitor list and invalidate FA information at the MN.

Care-of addresses are identifiers on the foreign agent's subnet that can be used by the MN for connectivity to the internetwork. There are two types of care-of addresses: co-located, and agent-located. Co-located addresses are acquired in Pop-up mode, and are temporary identifiers for the MN itself on the foreign network. These identifiers would be taken from a sequence of IP addresses reserved for temporary use. Using these addresses allows a mobile node to connect without FA support, but requires the mobile node to decapsulate forwarded messages itself. An agent-located address is simply the IP address of the foreign agent itself. In this case, the foreign agent will receive packets destined for the MN, decapsulate them, and forward them on the foreign network. Each foreign agent must advertise at least one care-of address, but may advertise more.

If a MN is not receiving agent advertisements, or if its registration has timed-out, it can begin *agent solicitation*. Agent solicitation messages are special ICMP router solicitation messages and follow the same rules and procedures. All agents must respond to agent solicitations on their network or sub-net. An agent is required to respond with an *agent advertisement message*.

A MN needs to detect whether it is on a foreign network or on its home network. This can also be done through agent advertisements. One of the fields in the advertisement messages is used to specify whether the agent is a home agent or a foreign agent. If the agent is the home agent for the MN, then the MN must treat the first HA message as a special case and reconfigure itself for operation on the home network. While the MN is on its home network, normal IP routing is sufficient for routing its packets, and certain measures have to be taken to ensure that Mobile IP does not interfere.

When a MN detects its home network, it needs to reconfigure its routing tables so that hosts on the local network can be reached directly. Once this is done, the MN must deregister with its home agent. Deregistration causes the HA to update the MN's location information in its MN support list, and adjust its routing tables to reflect the MN's new location. When these two steps have been taken, the MN will function as a normal fixed host, and the HA will no longer intercept packets on behalf of the MN.

When a MN detects that it is on a foreign network, the MN needs to inform a foreign agent of its presence and register with a HA. A FA can refuse to service the MN if it is too busy. In this case, the MN must wait until it receives an agent advertisement message stating that the FA is willing to host the MN, or until it can acquire a co-located address. The HA must always be willing to register a MN that it services.

Registration provides the MN with the mechanisms for communicating its reachability information to the important routing entities in the protocol. The MN must be able to perform the following functions to operate properly:

- Request service for packet forwarding when connecting to a foreign network.

- Keep the HA informed of current care-of address.

- Renew registrations to prevent binding expiration.

- Deregister with the HA upon connection to the home network.

In addition to these basic capabilities, the following optional support capabilities are provided for tracking MNs:

- Multiple registrations can be maintained simultaneously by a MN. This results in duplicate packets being tunnelled and delivered to the MN via different foreign agents. When using wireless networks, this reduces packet loss during hand-offs.

- Subsets of the mobility bindings held by the MN can be deregistered. When multiple bindings are held, this is useful for de-selecting bindings over wireless interfaces for which the signal strength has become too low.

- A HA's address can be discovered dynamically if the MN is not configured with one. This capability can be implemented by broadcasting registration requests onto the home network rather than to a specific node. All HAs on the network will respond with packets denying the registration. The MN can use the source addresses in these packets to re-register with one of the responding HAs.

2.5.2 Forwarding of IP Packets

The mechanism which makes Mobile IP possible is *packet forwarding*. Packet forwarding involves intercepting a packet addressed to a host's permanent location and forwarding it to a different location. In this process the original packet must be preserved because higher-level protocols, and IP itself, depend on information in that packet for correct processing at the destination.

MN IP Header	IP Data	−− >	Link Header	MN IP Header	IP Data

Figure 2.3: IP Encapsulation within Link Layer Packets.

MN IP Header	IP Data	−− >	FA IP Header	MN IP Header	IP Data

Figure 2.4: IP Encapsulation within IP Packets.

Encapsulation is the forwarding mechanism commonly used on local area networks. Encapsulation involves taking the entire data packet and placing it inside a new data packet by adding a header onto the front of the original packet. This is precisely the mechanism by which IP traverses individual networks. The IP packets are encapsulated in network frames (see Figure 2.3) using the link layer protocol and are forwarded over the local network.

For implementing Mobile IP, a similar method of encapsulation is needed. The difference with internetwork mobility is the need to encapsulate IP packets within IP packets (IP-in-IP encapsulation). The IETF has already produced Internet standards for IP-in-IP encapsulation ([20] and [19]). This type of encapsulation has the advantage that it is easy for the packet to be redirected fully intact to the mobile node. Disadvantages include the overhead that is involved, and the inability to deliver ICMP error messages back to the original source of the packet [4]. IP-in-IP encapsulation is illustrated in Figure 2.4.

The overhead with encapsulation takes two forms. First, there is the processing time needed for encapsulating and decapsulating. Second, there is the space requirement needed for the encapsulating header. The addition of an extra IP header into the packet increases the size of the packet itself. Because of hardware and other concerns, individual networks define a value called the *maximum transmission unit* (MTU) which is the largest packet size that the network will accept. The increased size of the encapsulated packet can cause the MTU of the transmission medium to be exceeded. This causes *fragmentation* of the packet, and the overhead associated with fragmenting and de-fragmenting.

The transmission of data using IP-in-IP encapsulation is commonly referred to as *tunnelling*. A tunnel is a delivery path between two forwarding agents. Packets being forwarded using the tunnel have their addresses hidden from the network routers and they are tunnelled to their destination using the addresses of the forwarding agents. The forwarding path between the home agent and the foreign agent is called the *Mobile IP Tunnel*.

2.5.3 Encapsulation of Broadcast and Multicast Packets

The encapsulation of normal IP traffic can be handled quite easily using the IP encapsulation method outlined in the previous section. Broadcast traffic and multicast traffic, however, each present a unique set of challenges for the Mobile IP protocols. This section describes proposed methods for handling such packets. Since broadcast and multicast traffic are quite similar in the problems they pose for Mobile IP, the packets are handled the same way in the basic IETF Mobile IP protocol, and so most of the details below about broadcast also apply to multicast.

By definition, broadcast packets are sent to every host with an address on the receiving network. This includes mobile nodes. The problem is that broadcast packets are not sent specifically to the MN, so mobile routing agents may not forward the packets. It is simple enough for an agent on the home network to encapsulate the packet and forward it to a care-of address. But when the packet is decapsulated at the other end, the broadcast address does not correspond to any MN and the packet is discarded. In fact, unless the decapsulating agent is provided with the network address portion of each MN it supports, there is no way to determine whether the packet even is a broadcast packet, and therefore it must be discarded.

The solution for broadcast packets is to encapsulate them twice (as is shown in Figure 2.5). First the packet is encapsulated with a header in which the destination address is the MN and the source address is the original source. The remainder of this header is unimportant except that it must be marked to let the MN know it is

IP Broadcast Hdr	IP Data	−− >	FA IP Hdr	MN IP Hdr	IP Broadcast Hdr	IP Data

Figure 2.5: IP Encapsulation of Broadcast Packets.

encapsulated. The outer header has a source address corresponding to the mobility agent on the home network, and a destination address corresponding to the mobility agent on the foreign network. Multicast delivery is discussed further on page 30.

2.5.4 Basic Routing and Routing Enhancements

The basic routing in the IETF proposal is the most conservative in terms of expectations that are placed on the nodes involved. Basic routing involves at least four entities (MN, CN, HA, FA) and is called *triangle* routing.

In a Mobile IP scenario using triangle routing, a packet is sent from a correspondent node to the mobile node's home IP address. The mobile node is known only by its home address, so the packet is delivered to the home network. If the mobile node happens to be there, the packet is delivered normally. If the mobile node is not present on the home network, the home agent intercepts the packet for the mobile node, encapsulates it, and forwards it to the MN's foreign location. The destination address in the encapsulating header will be the care-of address for the mobile host, and the source address is usually the home agent. Triangle routing is illustrated in Figure 2.6.

Encapsulated packets are routed correctly using normal IP routing. The encapsulated packet is data like any other data. The only differences are at the encapsulation source and the decapsulation destination. The source for an encapsulated packet must produce a new header and a new packet as discussed earlier. At the destination, the host must be able to detect that this packet contains encapsulated information, and should remove the outer packet header and forward the interior packet to the destination host.

An example of routing to a mobile node is illustrated in Figure 2.7. In this figure, four types of routing are demonstrated: IP, IP-in-IP, and LAN forwarding

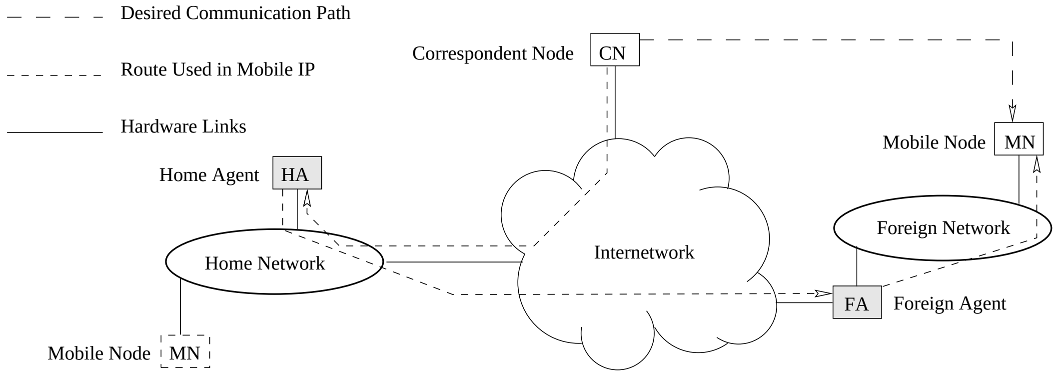


Figure 2.6: Triangle Routing in Mobile IP

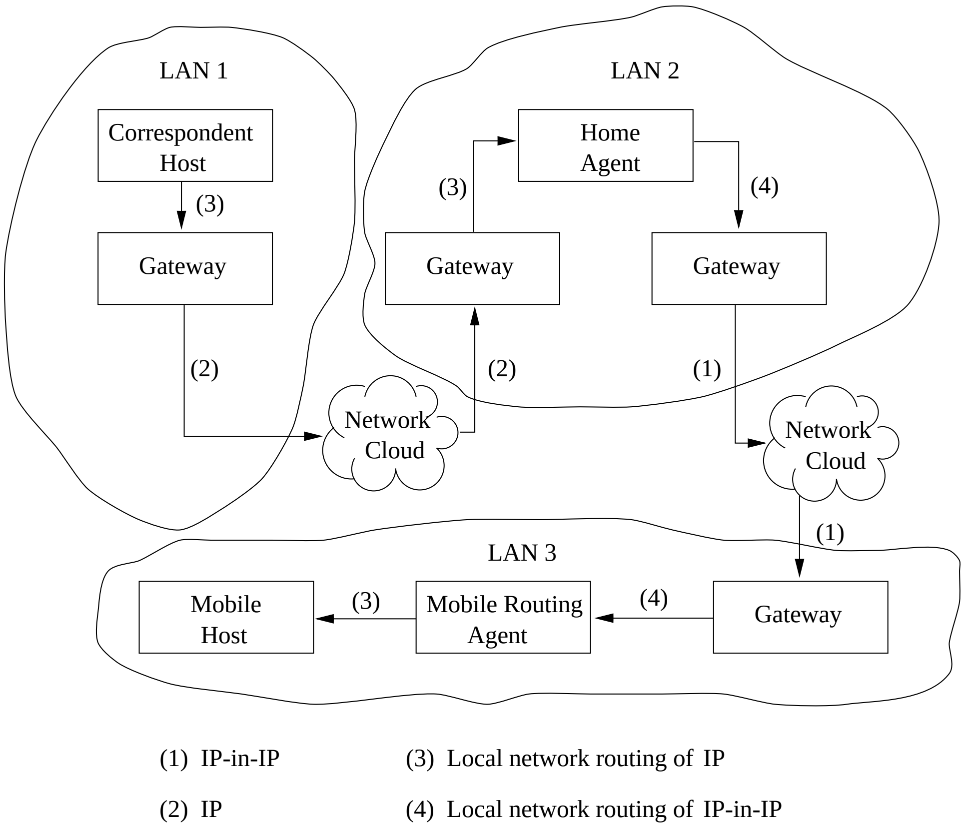


Figure 2.7: Mobile Host Routing Overview

of both IP and IP-in-IP packets. With un-enhanced mobile IP routing, a packet originating at a CN is given an IP header with its destination as the MN's home address. This packet is encapsulated in a link layer frame and forwarded to a local router (3). This router and others forward the packet using IP routing to the home network of the MN (2). The IP packet is forwarded by link layer protocols on the home network to the home agent (3). The home agent now encapsulates the IP packet using another IP header and produces an IP-in-IP encapsulated packet. This is forwarded to an appropriate local router (4) that forwards the packet on the internetwork using an IP-in-IP (or other appropriate) tunnel (1). The router on the foreign network receives the packet and forwards it to the foreign agent (4). This agent decapsulates the IP-in-IP packet and forwards the bare IP packet to the MN using local link layer protocols (3).

2.6 Related Research

The area of mobile computing is very active, and much research is being conducted into issues concerning support for mobile computers. This includes investigating wireless issues, satellite transmission issues, user tracking, and protocol support such as that provided by Mobile IP. Since the research area is so active, this section will limit the discussion to research related to the development, refinement, and measurement of Mobile IP.

2.6.1 The Early Development of Mobile IP

The development of Mobile IP protocols started in the early 1990's, with three major proposals being published in and around 1991. These proposals included Columbia University's Mobile*IP, Sony Labs Virtual IP (VIP), and IBM's protocol using *loose source routing* (also known as LSSR). Each of these approaches had its own strengths and weaknesses. The early versions of the first three are contrasted and compared in a paper by Andrew Myles and David Skellern [18].

*Mobile*IP* was developed at Columbia University by John Ioannidis [11], and is based on virtual mobile sub-nets with a network or internetwork setting. Packets are forwarded to the mobile hosts through the use of IP-in-IP (see Section 2.5.2).

Virtual IP (*VIP*) was developed at the Sony Computing Lab by Teraoka *et al.* [30]. VIP is based on propagating special routers throughout the internetwork that can cache a mobile host's location and tunnel packets using a new IP option.

IBM's proposal was designed by Perkins and Rekhter [24]. This protocol was based on loose source routing, which is implemented as an option in IPv4. The advantage of this proposal is that it uses existing standards for IPv4. Loose source routing allows the sender to specify the route that is taken to the receiving host. Therefore, a route to a network different than the mobile node's home network can be specified. Unfortunately, loose source routing is not implemented on all platforms, and is often implemented incorrectly. For these reasons, this solution to the problem was considered to be unreliable. A major contribution of [24], however, was a generic architecture that was compatible with the three major proposals for mobility support.

These three proposals and their general architecture were examined in [17] and [18]. After these proposals were investigated, they were found to have various deficiencies. This led to the development by Myles and Perkins of a new mobile protocol called MIP [16]. MIP solved many of the problems with the previous protocols but still had some problems. While MIP was being developed, David Johnson at Carnegie Mellon University independently developed the Mobile Host Routing Protocol (MHRP) [12]. MHRP used ideas which were similar to loose source routing, but it did not use IP options. MIP and MHRP were used as a basis for the development of the Internet Mobile Host Protocol (IMHP) [12] by Myles, Perkins and Johnson. This protocol has become the basis for the current IETF Mobile IP.

2.6.2 Refinements to Mobile IP

A number of refinements to Mobile IP are being investigated, which either allow it to work in a wider variety of situations, or allow it to work more efficiently. Research of this type includes Mobile IPv6, Mobile Multicast Support, Route Optimization, Foreign Agent hierarchies, and Foreign Agent load balancing.

Mobile IPv6

IPv6 is the next generation of IP protocols that is being developed to include better support for mobility and quality of service, as well as defining a much larger address space. Mobile IPv6 is a refinement of Mobile IP which is being designed for IPv6. The definitions for IETF Mobile IPv4 have been extended for IPv6 by Perkins and Johnson [23].

An additional proposal for mobility support in IPv6 was also presented by Sony [29], but it is not compatible with IETF Mobile IP. This proposal was based on the VIP proposal by Teraoka *et al.*

Route Optimization

Route optimization was first proposed by Perkins and Rekhter in 1992 (see [24]). The latest proposal for route optimization, this time for IETF Mobile IP, was recently developed by Johnson and Perkins [13]. The Mobile IP proposal is the one focused on here.

Route optimization is an enhancement to Mobile IP to reduce the overhead suffered when packets destined for the mobile node are forwarded through the home agent. Route optimization proposes changes to the protocol which allow hosts other than the home agent to cache the location of the mobile node and forward packets to it. This results in correspondent nodes being able to communicate directly with the mobile node rather than having to send the packets through the home agent (triangle or "dog-leg" routing).

Optimized routing is designed to provide multiple benefits to the MN with regards to packet routing. These benefits include reducing the number of cases where the HA is used for routing, as well as methods for ensuring packet delivery during movement on the net. These benefits arise not only from allowing the HA to cache the location of the MN, but also from allowing CNs to cache location information in a *binding cache*. The argument in favour of optimized routing is that it helps to avoid situations where the CN and MN are very close in terms of physical networks, and the HA is not close to either. In this situation, the distance travelled can be increased many times over by routing through the HA.

With short-cut routing, the HA is not required for most routing tasks, but it is still needed for registration and location tracking. The HA is always involved when the first packet between two hosts is sent and when incorrect binding cache entries are detected at CNs. There are two types of messages defined for handling these cases: binding updates, and binding warnings. When forwarding a packet the HA is required to send a binding cache update message to the sender of the packet to inform it that the MN is away from its home network. If the sender can support a binding cache, it should create an entry for the MN upon receiving an update. The HA continues to send binding cache updates as packets arrive, but will quickly back off and stop sending after the first few packets. Additional updates are sent in case the first few are lost, but updates are no longer sent once it is obvious that the CN does not support route optimization mechanisms. Binding warning messages are produced when a forwarding agent receives an encapsulated packet for a mobile node that is not registrered with that agent. In this situation, the forwarding agent sends a binding warning to the home agent If the HA receives a binding warning message, it is required to send a binding update message to the host indicated in the warning. A drawback of the short-cut routing is the requirement of HAs to deal with extra traffic in the form of these messages.

Correspondent nodes can use optimized routing to improve the efficiency of their transmissions to the MN. The binding cache entries at the CN also allow it to act as a router for the MN when it encounters packets destined for the MN. A CN is informed of an MN's location via binding cache update messages from the HA. If the CN supports a binding cache, then it will create an entry for the MN and cache its location. The binding cache entry is assigned a lifetime equal to the lifetime remaining in the MN's registration with its current FA. Once a binding cache entry times out, the CN must remove it from the cache. CNs desiring to maintain the cache entry can request a binding update from the HA, which will send the current location information for the MN if it is available.

A CN with a binding cache entry is allowed to intercept packets destined for the

MN via its home address. The CN should then encapsulate the packet and forward it as the HA would. If the CN forwards the packet to the wrong location, then it should eventually be informed of the error via binding cache update messages, and can make any corrections necessary. This facility can reduce the path length that the packet has to take if it is intercepted before the home network.

Route optimization does greatly reduce the potential overhead associated with using the home agent for forwarding, but it also introduces other requirements, such as tracking all the nodes with the location information, keeping these nodes up to date of any location changes, and handling messages that are mistakenly forwarded by nodes with faulty information. Rajagopalan and Badrinath [27] examined the costs of route optimization in terms of update overhead and other factors. Their research indicates that the update overhead is prohibitive if the ratio of location changes to connections is high, and basic routing is more desirable in this case. If the ratio of location changes to connections is low, then updates are more efficient than the basic protocol. An adaptive location management scheme was developed that adjusts the degree of route optimization according to the characteristics of the traffic and the host movement.

Multicast with Mobile IP

One of the problem areas for Mobile IP is the forwarding of multicast datagrams to roaming mobile nodes. Problems arise with multicast because the traffic is directed at many nodes, not just one. This property can cause duplication problems at the foreign network when many mobile nodes are resident at the same time. In addition, other problems can be caused in the delivery of multicast messages to scoped addresses. Scoped addresses are only used for delivery to nodes within a limited scope, or domain. Because of this, these addresses are commonly used within many domains for internal communication. Since the roaming mobile node may wish to receive these messages from its home domain, there is a problem in delivering these messages to the roaming mobile node without confusing other mobile nodes which may interpret these messages as arriving from their own home domains.

Research has been conducted in this Department into addressing some of the problems with multicast delivery with Mobile IP [5] [2]. This research has resulted in the development of a new protocol called MoM, for Mobile Multicast, that seeks to minimize the problems associated with duplicate delivery, and also deals with the scoping problem [10]. A detailed analysis of the protocol based on simulations can be found in [31].

Reverse Tunnelling

Reverse tunnelling is a recent proposal to help address routing issues in which packets from the mobile node are dropped because of topologically incorrect source addresses. As an added security measure, many gateways now check the source address of any outgoing packet to see if the source address matches the network from which it originates. If it does not, these gateways will drop the packet. This creates a problem with mobile hosts sending packets from the foreign network.

The reverse tunnelling draft [15] proposes a method by which packets to be sent onto the Internet are reverse tunnelled to the HA which then sends them using normal routing. This draft is fully compatible with RFC 2002, but raises some issues about performance. In particular, the reverse tunnelling negates advantages achieved through route optimization. Also, reverse tunnelling can effectively double the duplex traffic on the HA-FA link.

Foreign Agent Hierarchies

In RFC 2002, a single foreign agent is described that will handshake with the MN, communicate with a HA, and perform various routing tasks. Recently, the use of a single FA has been questioned. Perkins has proposed a hierarchical FA system [22] in which many FAs work together to manage MNs. This system increases the locality of state changes in the Mobile IP system as a mobile node moves between related networks in a single FA cluster. This reduces the traffic exchanged between FAs and HAs for a frequently moving mobile node.

Load Balancing

Another topic of debate is the issue of load balancing on a foreign network. This issue arises when a foreign network has many foreign agents. How is an MN to decide which one to pick? One suggestion is that the FAs deliver load information in their agent advertisements to the MNs. One way of accomplishing this is to use a priority option allowed for agent advertisements, and merely decrease the FA's priority as its visitor list grows.

2.6.3 The Mobility Research Area

A good overview of current research activities in the area of mobility support is given by Perkins in [22]. This article categorizes the research by the network layer within which the research is being done, and illustrates the wide diversity of work being done in this field.

At the physical layer, much research is being done towards expanding the bandwidth and improving the quality of wireless networks. At the link layer, many low-level issues such as compression, neighbour discovery and security are being explored. At the network layer, much research is being done on Mobile IP. This research includes route optimization, optimized registrations, source routing, IPv6 protocol integration, tunnelling inefficiencies, and mobile multicast delivery. At the transport layer, TCP and UDP performance are being studied, and in particular, problems with congestion detection, asymmetric connections, and resource reservation are hot topics for study.

Many other research interests also exist at the middleware layer and higher, such as integrating mobility with support protocols like DHCP, PPP, and others. In summary, the mobile research community is very active, and much research is being done in many areas.

2.6.4 Performance Analysis of Mobile IP

The performance analysis of Mobile IP has not been extensive to date, since the primary focus has been on getting the protocol implemented and running. Several groups are involved in such efforts, however, and performance results are starting to appear.

A research group at the University of Singapore has implemented Mobile IP on the Linux platform, and has conducted a performance evaluation of TCP over Mobile IP [6]. The performance analysis of TCP looks primarily at the workability of TCP over Mobile IP in various situations, and examines the end-to-end throughput of TCP over Mobile IP. FTP bulk data transfers are used as a traffic source. The results of these experiments indicate that fragmentation and tunnelling effects do not significantly affect the performance of TCP over Mobile IP. Significant effects happen when disconnections occur, and in addition, certain problems within the Linux kernel itself affect the performance results.

Research into the low-level implementation of Mobile IP was done by Vineet Chikarmane in this Department [3, 4]. This research demonstrated that significant queueing delays can be encountered within a mobility agent. It also showed that fragmentation can have a serious effect on the internal processing time for Mobile IP packets within the agents.

Chapter 3

The Experimental Test-bed

3.1 Introduction

The goal of this thesis is a study of the performance of Mobile IP as it operates with actual traffic on a physical network. In order to accomplish this goal, an environment that combines a working implementation of Mobile IP (in this case, the implementation of Mobile IP produced at CMU [8]) with the appropriate physical entities is required. This environment forms the experimental test-bed.

The test-bed has several important features in addition to the Mobile IP implementation itself. The most important components are the home agent, foreign agent, and mobile node. An important feature of the test-bed is the presence of multiple networks, allowing the mobile node to roam away from the home network. Applications on the correspondent nodes are needed for generating traffic, measurement software is needed to collect the data, and task-specific software was needed for processing the raw data and extracting useful measurements. All of these aspects of the test-bed are discussed in this chapter.

The test-bed is set up in the Distributed Systems Research Laboratory (DISCUS) at the University of Saskatchewan. One of the networks involved is the standard campus network and the other is a small network created specifically for the test-bed. The correspondent nodes used in the experiments are standard workstations available in the laboratory. The physical characteristics of the test-bed placed certain constraints on the experimentation work, but the existence of the test-bed with a

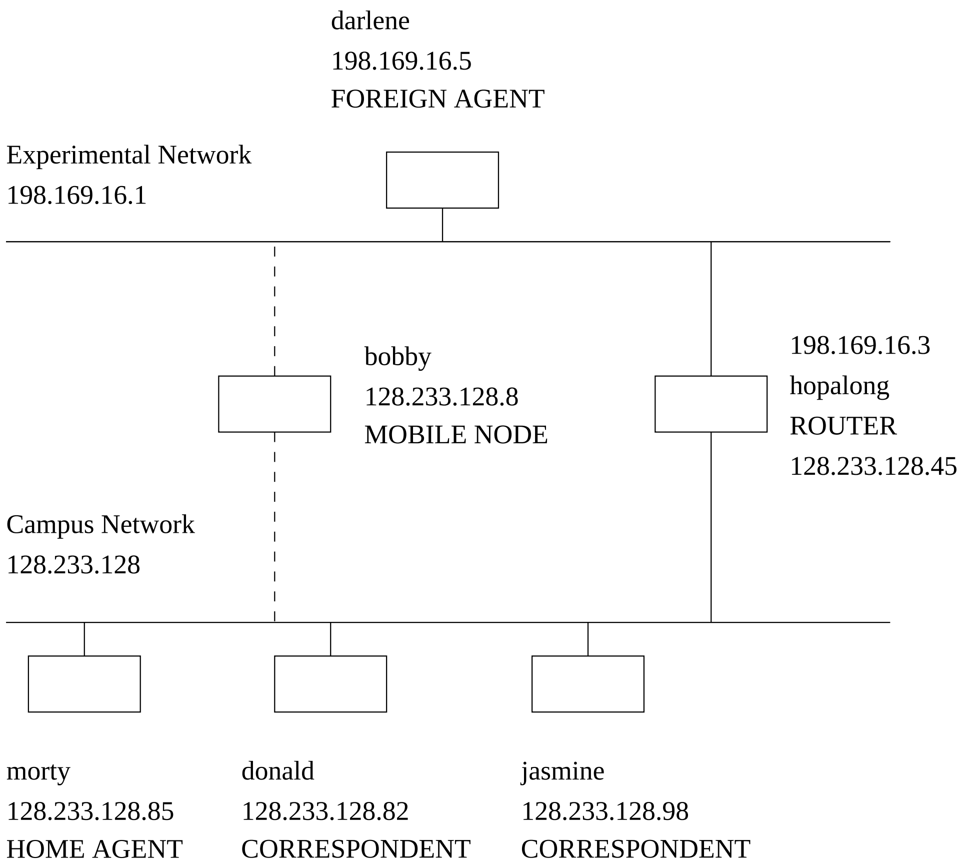


Figure 3.1: The Experimental Test-bed

working version of Mobile IP presented many opportunities for doing a complete evaluation of the protocol that did not exist in the laboratory previously.

3.2 Hardware

At a high level, the test-bed includes one foreign agent, one home agent, one mobile node, a router, and two correspondent nodes. In addition to the computers involved, two separate networks are set up to allow testing across network boundaries. This section describes the hardware in greater detail.

3.2.1 General

A topological overview of the test-bed is given in Figure 3.1, showing the layout of the test-bed, and the roles that the various pieces of hardware played in the experiments. The diagram can be roughly divided into two sections horizontally. The top section represents the foreign network, and the bottom section represents the home network, with the router, hopalong, on the dividing line. The router is a CISCO IGS-L model.

The foreign network has an address of 198.169.16.1 and a network mask of 255.255.255.249. This restricts the usable addresses on the foreign network to 198.169.16.1, 198.169.16.3, 198.169.16.5, and 198.169.16.7. Since the highest and lowest addresses have to be reserved for broadcast, this leaves only two addresses for use by specific machines. The foreign agent, darlene, is given the address 198.169.16.5, and the router, hopalong, is given the address 198.169.16.3.

The home network address is 128.233. All of the home network addresses used in the experiment lie within a sub-net of this space, 128.233.128. The computers shown on the home network in Figure 3.1 include the home agent and two correspondent nodes. The router connecting the two networks (hopalong) is given the address 128.233.128.45 on the home network.

Both networks in the test-bed are 10 Mbps thin wire Ethernets. The mobile node is moved from one network to another simply by unplugging its interface from one network and plugging it into the other. No wireless or other interfaces are in the test-bed.

3.2.2 Home Agent

The home agent, morty, is a fixed host residing on the home network. The address of the home agent is 128.233.128.85, and this address is reachable by all computers on the Internet. The home agent is an IBM P75 with an Intel Pentium processor and 16 MB ram. Software on the home agent includes *netBSD 1.2E*, *mobiled*, and *tcpdump* (see Section 3.3).

3.2.3 Foreign Agent

The foreign agent, darlene, is a fixed host residing on the foreign network. The address of the foreign agent is 198.169.16.5. The foreign agent is a Hewlet Packard 9000/433 with a 68040 33 MHz processor, 4 KB cache, and 16 MB RAM. The operating system is *netBSD 1.2E*, and the programs *foreignd*, and *tcpdump* are on the foreign agent.

3.2.4 Mobile Node

The mobile node, bobby, has the home network as its permanent home, but is physically switched from home network to foreign network and back. The mobile node's address is 128.233.128.8. The mobile node is a Hewlett Packard HP 9000/400 with a 68030 50 MHz processor, 32 KB cache, and 8 MB ram. The operating system is *netBSD 1.2E*, and the program *mobiled* is used for Mobile IP support. The application programs x-windows, ping, and FTP are on the mobile node.

3.2.5 Correspondent Nodes

There are two correspondent nodes used in these experiments. The primary correspondent node is named donald, and its IP address is 128.233.128.82. Donald is a Hewlett Packard 9000/715 running HP-UX 9.0. Donald supports ping requests up to 32000 bytes, and exports a large drive via NFS which is used for many of the experiments. Donald resides on the home network, and can communicate directly with the home agent.

The secondary correspondent node, jasmine, is a Pentium class machine with a 100 MHz Pentium processor and 32 MB RAM. The operating system on jasmine is Windows NT 4.0. Jasmine responds to ping requests of up to 64000 bytes, but does not accept FTP requests because it lacks an FTP server. Jasmine's IP address is 128.233.128.98, and it is a fixed node residing on the home network.

3.2.6 Other Nodes

Other nodes which are used, but do not appear in Figure 3.1, include skippy and dewey. Both dewey and skippy reside on the home network. Neither node had a direct effect on the experiments, but they were used to demonstrate the validity of the experimental techniques, and they are used as a starting point for launching the experiments.

Skippy was used to verify that the program tcpdump (used for collecting data) has minimal effect when used on the home and foreign agents. A test was performed in which tcpdump was run on the agents, and then on skippy. Similar behaviour was noted in all cases, and the delay was not significantly increased when the agents ran tcpdump. From this test, it can be concluded that tcpdump running on the agents does not interfere significantly with the observed results.

The other node, dewey, is used for launching various scripts. When an experiment is to be performed, telnet sessions from dewey to the various nodes are used to start the needed scripts and to do any necessary configurations. Dewey also runs a program called fakeReg that fakes registrations and makes it appear to the agents as if there are many mobile nodes when in fact there is only the one.

3.3 Software

The software used in the test-bed includes the operating systems, the measurement programs, the Mobile IP implementation, and various support programs. Specialized software is essential for providing an appropriate test-bed for these experiments. This section describes the software programs, what their purpose is, and why they are chosen.

3.3.1 Mobile IP

The most significant software in the test-bed is the Mobile IP implementation. The implementation used was developed at Carnegie Mellon University by the Monarch

Table 3.1: Significant Functionality Not Included in Monarch Mobile IP

Functionality	Description
Route optimisation	Reduces the amount of traffic having to flow through the home agent.
Reverse Tunnelling	Sends packets from the mobile host back through a tunnel from the foreign agent to the home agent.
Multiple Home Addresses	A mobile host with multiple interfaces may also have multiple address, but only one is recognized by the mobile IP software.
Multiple Registrations	A mobile host can register with only one foreign agent at a time, thus reducing the benefit gained by maintaining two connections when in overlapping network coverage.
Multiple Security Keys	Only one security key for authentication can be defined for a mobile host and home agent pairing.

research group [8]. This implementation was chosen because it is fully compatible with RFC 2002, and runs on a freely available operating system for which source code is available. The implementation used was developed for netBSD 1.1 and was ported to netBSD 1.2E for these experiments.

Table 3.1 describes Mobile IP functionality that is not included in the implementation. The table shows that the implementation is a basic one, with all the functionality needed by RFC 2002, but few of the proposed optional extensions. Despite this, the implementation is a good choice for testing the performance of the basic protocol.

The Monarch Mobile IP implementation is based on a mixed approach in which part of the functionality is implemented in the kernel, and part of the functionality is implemented in user programs. This approach was recommended by Vineet Chikarmane in his analysis of Mobile IP implementation [3, 4]. In-kernel implementation provides speed where required, while user-level implementation provides flexibility and debugging capability. In the Monarch implementation, there are three user programs: *mobiled*, *foreignd*, and *homed*, which cooperate to manage the protocol. These programs are summarized in Table 3.2.

Table 3.2: Mobility Support Programs

Program	Location	Function
homed	Home Agent	Sends router advertisements on home network, maintains location information for mobile nodes, handles registration request from support mobile nodes, and performs proxy ARP for supported nodes.
foreignd	Foreign Agent	Send router advertisements on foreign network, maintains forwarding information for visiting nodes, and handles registration requests from visiting nodes
mobiled	Mobile Node	Monitors mobile node's network connectivity, solicits mobile agents if necessary, registers at home and foreign networks, turns off ARP when at a foreign network, and modifies the routing table to fit current connectivity situation.

Homed runs on the home agent and is responsible for sending advertisements on the home network, dealing with registration requests, and maintaining state information about the home agent's supported mobile nodes. *Homed* must also issue proxy ARPs on behalf of the mobile host so that other computers on the network know whether to direct their traffic to the home agent's interface or to the mobile node's interface. The homed program is executed from the user level, and is usually run in the background as a daemon. Upon starting up, it reads a configuration file containing the addresses and registration codes for all of its mobile nodes, as well as the minimum time between unsolicited advertisements and the lifetime placed on all outgoing router advertisements.

Foreignd runs on the foreign network and basically has the same responsibility as *homed*, except that proxy ARPs are not issued on the foreign network. The foreign agent also has a configuration file which states the minimum time between advertisements, the router advertisement lifetimes, and the maximum supportable lifetime. When a mobile host arrives at a foreign network, it requests a registration, and it is the *foreignd* program's task to intercept these requests, forward them to the home agent, and then act upon any reply that is sent by the home agent. If the registration succeeds, the foreign agent records state information for the mobile host and forwards the registration reply. The *foreignd* program is also usually run in the background.

Mobiled runs on the mobile node. This program is responsible for detecting router advertisements on the mobile node's network interfaces and determines the state of connection. If router advertisements are detected from the home agent, then the daemon assumes it is at home; otherwise it will actively solicit foreign agents. When movement to a new network is detected, the *mobiled* program is responsible for getting the required routing information and manipulating the routing tables so that the mobile node can resume or maintain connectivity to the internetwork. If the mobile host returns home, any routing information gained from other networks must be removed, and the routing tables must be returned to their normal fixed-host state used on the home network.

Part of the *mobiled* program's duties is to turn off ARP when the mobile node visits a foreign network. This is done in order to prevent the mobile node from replying to ARP requests for its address. If ARP was not turned off, nodes on the foreign network could assume that the mobile node is always local to their network, and packets would be lost when the mobile node moved to a different network. By turning ARP off at the mobile node, all requests for communication are directed back to the home agent, which is always up-to-date on the location of the mobile node. When the mobile node returns home, *mobiled* turns ARP back on.

Mobiled is configured on startup using a file called config-mobiled. This file contains the registration control keys for the node, as well as the mobile node's home agent's address, its requested lifetime, and the minimum rate for sending out agent solicitation messages. This file is read when *mobiled* starts up. *Mobiled* is usually run as a background daemon and therefore remains operational throughout the time period when the mobile node is activated.

The user programs have to be run with root permission. This is because they use raw sockets to deliver router advertisements and agent solicitations. All messages

Table 3.3: Configurations Used for Test-bed Daemons

Daemon	Setting	Value
foreignd	hold time	10.0 Secs.
	advertisement interval	1.0 Secs.
	FA lifetime	1000 Secs.
homed	replay protection	nonce
	min. advertisement interval	1.0 Secs.
	router lifetime	600 Secs.
mobiled	starting delay	2 Secs.
	min. solicitation interval	1.0 Secs.
	min. home advertisement rate	2.0 Secs.

sent to the networks are sent to the general broadcast address 255.255.255.255.

Communications between the home agent and foreign agent are protected by either time stamps or nonces. Time stamps work by having the home agent compare its clock with the mobile node's clock value on the timestamp. If their difference is too large, the registration is rejected; otherwise it is accepted, and the home agent thereafter synchronizes its mobile nodes with time updates. Nonces provide protection by exchanging random codes between the home agent and its mobile nodes. Each time one sends a message to the other, the sender's message must contain the last code-word it received from the receiver. In this way, communications are protected from replay attacks by malicious hosts that may duplicate packets from either the home agent or mobile node in an attempt to introduce faulty state information.

Each of the application layer programs is configured for use in the test-bed, and these configurations are shown in Table 3.3. The foreign agent daemon is configured to hold onto unratified registration requests for 10 seconds, advertise on the foreign network once per second, and grant a registration lifetime of 1000 seconds to any visiting mobile node. The home agent daemon is configured to use nonce for replay protection, advertise once per second, and issue router advertisements with a lifetime of 600 seconds. The mobile node daemon is configured to listen for 2 seconds before deciding whether it is home or not, solicit once per second if it detects a loss in connection, and assume it is disconnected from its home network if it does not receive advertisements from the home agent within 2 seconds.

The in-kernel modifications themselves are restricted mostly to files within the net and netinet directories (a file in the conf directory is also modified slightly so that the newly created files are included in any kernels). Two new files are added to the kernel: mobileip.c and mobileip.h. These files contain all the code for tunnelling packets, sending out ARP messages, and manipulating kernel state information regarding mobility. There are also other files changed in these directories in order to integrate the mobile IP functionality into the existing networking code. The exact files changed vary with the operating system.

All communication between the daemons and the kernel code is carried out through the use of ioctl commands, some of which are newly defined for the Mobile IP kernels. These new commands are described below:

RTM_MIPADD Adds a mobile IP record to the kernel tables. This either records a visiting mobile node in a foreign agent kernel, or an away mobile node in a home agent kernel.

RTM_ARPON Turns on the ARP facilities. Allows the sending of ARP messages.

RTM_ARPOFF Turns off ARP by not allowing any ARP messages to be sent onto the network.

RTM_MIPCLEAR Removes all state information about mobile nodes supported.

RTM_MIPHOMECOUNT Returns the count of mobile nodes at the home agent.

RTM_MIPDELETE Deletes a specific mobile node entry from the kernel tables.

RTM_MIPCHANGE Changes information in a mobile node entry.

Other commands that already exist are used by *mobiled* to manipulate the routing tables at the mobile node itself.

More information about the Monarch Mobile IP implementation can be found in [8], where more features of the implementation are discussed, along with configuration and installation instructions.

3.3.2 NetBSD 1.2E, 1.3

NetBSD is a UNIX operating system that has been developed by many developers and is available on the Internet free of charge. Because of its Internet development, netBSD changes frequently, and consequently incorporates the newest design decisions and refinements to the UNIX standard. An example of this is the networking code which incorporates the latest standards for IP, TCP, UDP, and other protocols.

The biggest advantages of netBSD are that it is free and the source code is freely available. This enables the establishment of a test-bed environment without a large expenditure for customized operating systems. The freely available source code facilitates adding in functionality such as Mobile IP and helps with refining the implementation during the lifetime of the test-bed.

3.3.3 Tcpdump

The main tool for measuring the traffic through the mobility agents is *tcpdump*. *Tcpdump* allows a computer user to monitor traffic on the network by listening to every packet on the network and extracting bits of information from the packets for real-time analysis or storage on local media. *Tcpdump* works by taking advantage of a packet filter that is built into the operating system as a pseudo-device. The *tcpdump* program reads from this pseudo-device and formats the data.

The *tcpdump* program is run on each of the agents. The program is run with the –w option which causes the packets to be written to disk in their raw form, thereby cutting down on immediate processing with filters. Filter processing to get the needed information is done after the traces are taken by reloading the traces into *tcpdump* using the -r option.

3.3.4 netUtilization

Tcpdump is also used as the basis for creating a system that monitors the network utilization. This system consists of two programs. The first program, *tcpdump*, reads in frames from the network and passes them onto the second program, which then extracts the timestamps and the sizes from the frame headers. With these timestamps and frame sizes, the second program, *netUtilization*, is able to count the number of bytes transferred in each second, and then perform a quick calculation to get the utilization over that time period.

This utilization program is used to evaluate the network overhead of Mobile IP when various traffic is sent between the correspondent node and the mobile node. This program is also run on both mobility agents, but is not used when other performance measurements are being conducted at the agents.

3.3.5 fakeReg

In order to simulate the interaction of the mobile node's traffic with the traffic of other mobile nodes a method was devised to create "fake" mobile nodes. A program was created to fake registrations to the foreign agent. This program, called *fakeReg*, is run on a home network node. Registration requests are generated with the required information for a fictitious node, and sent through the router to the foreign agent. When the foreign agent receives these requests, it processes the information and forwards the request to the home agent, which sends back a reply indicating whether or not it has approved the request. A successful reply results in the mobile node being registered at the foreign agent, and a successful reply is sent back to the mobile node. Since the mobile node is fictitious, this reply is lost, but the node remains registered until the registration times out. In this way, the registration of up to 20 extra mobile nodes can be faked. Once the nodes are registered, traffic can be sent to them, and this traffic is forwarded harmlessly onto the foreign network where it gets discarded.

3.4 Limitations

Although the test-bed allows Mobile IP to be tested in a real-life environment, it has its limitations. These limitations include the types of experiments that can be performed, the factors that can be varied, and the range of variation. Some of these limitations are discussed here.

3.4.1 Proximity

The test-bed is a short-distance environment located in a single laboratory. The networks used for testing purposes are only one router apart topologically, and less than a metre apart physically. This provides negligible internetwork routing delay, which is not a realistic scenario. In reality, a mobile node will most likely operate on a foreign network which is at least two routers away from the home network, often more. This will require routing across at least one other network, and therefore the traffic will be affected by the characteristics at that network (such as MTU), possible congestion, and possibly low bandwidth. When a mobile node is much farther away, these problems will be magnified due to the number of networks that need to be crossed, and additional delay will be caused by the physical distance itself.

It is assumed that most of the delay resulting from the distance that the mobile node is from home will come as a result of crossing routers, not as a result of propagation delays over physical distances. From the data examined in this thesis, the delay at a router will be shown, and this delay can be extrapolated across many routers, if so desired, to get an approximation of delay over a larger distance. It is speculated that unless slower network links are encountered, the delays from the routers will be more significan than the delays from physical distance.

Since the test-bed does not allow for the exploration of delays caused by long-distance connections, the effect of distance has been removed from consideration in these experiments, and the delays that are focused on are those occurring directly at the agents and on the home and foreign networks themselves. In a real situation, these delays may be overshadowed in some cases by delays due to distance, but the same characteristics being studied at the agents should be present given the same traffic, regardless of distance.

In addition to the extra delay, there is an increased chance of packets being lost, corrupted, or discarded when they are routed through many networks. Because the home network and foreign network are directly connected in the test-bed, this possibility is unlikely. This lack of packet loss simplifies many of the experiments and removes this factor from consideration. However, it also removes a source of potential problems for the Mobile IP protocol, and the higher-layer protocols which operate using Mobile IP. An example of how this could affect results is with NFS traffic.

The version of NFS used in the test-bed transports data using UDP. New versions of NFS exist that allow the use of TCP, but this capability was not exploited in the test-bed. UDP maintains little state information, and if packets are lost or re-ordered, UDP has no method for detecting this. Within the home network, this is of no concern, since there should be no packet loss. Once the mobile node moves to a foreign network, however, these datagrams must now be forwarded across networks where packet loss may occur. In this way, the transparency of the network connectivity on the foreign network is compromised because a new factor (packet loss) is introduced for UDP traffic which did not exist while the mobile node remained on its home network.

3.4.2 Number of Mobile Nodes

Because of limitations in the number of computers available, and in the number of addresses available on the foreign network, the number of real mobile nodes which would be used in the experiments was limited. This reduces the type of experiments that can be performed because real interaction cannot happen with more than one mobile node at a time. Also, situations involving mobile nodes sending data to each other cannot be explored.

3.4.3 Applications on the Mobile Node

The types of applications available on the mobile node used for testing was also limited. In particular, graphical programs such as web browsers and video-conferencing were not installed on bobby. The installation of such programs natively on the mobile node would allow for more detailed analysis of multimedia and real-time traffic over the Mobile IP link. In addition, the web allows an easy method for testing connectivity to nodes far beyond the networks in the test-bed.

3.5 Advantages

Even with its limitations, the test-bed provides a realistic environment for running Mobile IP. In fact, the protocol must be used as it would be if deployed throughout the internetwork. No hard-coding of network routes, faked traffic, or faked nodes are needed for the basic experiments performed. The key advantage of the test-bed is that it allows a functioning Mobile IP system to be studied as it behaves in real life, not just on paper, in simulations, or in theory.

Chapter 4

Experimental Design

4.1 Introduction

This chapter describes the experiments that were carried out to assess Mobile IP performance and usability. All experiments were performed in the test-bed environment described in Chapter 3. The experiments are introduced here by discussing the purpose behind each experiment, the metrics examined, the factors that were varied, and the workloads that were used.

4.2 Purpose

Much effort has gone into designing and implementing Mobile IP, but performance results for Mobile IP are just starting to appear. The purpose of these experiments is to examine a working implementation of Mobile IP and determine how well it works in a real environment rather than a theoretical one. To this end, the test-bed described in Chapter 3 was used to perform the experiments presented here.

The first task when evaluating Mobile IP is to determine what works with it, and what tests are possible. This motivates a set of experiments designed to test the operability of various applications and protocols, and the operability of Mobile IP itself. This evaluation is quite high level, and amounts to the following questions. Does the application work correctly when the node is mobile? When doesn't it work? When doesn't Mobile IP work? Why doesn't Mobile IP work in certain situations?

How are higher level protocols affected?

Most of these questions can be answered by attempting to run various applications in various network configurations and observing what happens. These experiments are described as operability experiments. The purpose of these operability experiments is to show what can and can't run with Mobile IP, and also in what situations Mobile IP itself will not run.

The issue of how higher level protocols are affected leads to a second category of experiments, namely the performance experiments. Mobile IP operates on a packet-by-packet basis. Many higher-level protocols operate at a higher level of functionality, which can cause the traffic at the packet level to be bursty, intense, or fairly uniform in nature. The higher layer protocols can also affect the size of datagrams being transferred, and therefore the size of the packets being transferred. These variations in packet traffic can affect how Mobile IP performs in the delivery of the traffic to the mobile node. In turn, the performance of Mobile IP in delivering these packets will affect the performance of the higher-level protocol. The purpose of examining the performance of Mobile IP with difference traffic loads is to study the interaction between the traffic load and Mobile IP, and show the effects of such interaction on the higher layer protocols.

4.3 Metrics

The experiments were designed to measure Mobile IP performance using certain metrics. These metrics are presented in this section. A summary of the metrics used is presented in Table 4.1. The table shows the name of the metric, any units of measurement associated with the metric (if applicable), and a brief description of the metric. Some of the metrics presented, like system operability, have no units associated with them, but rather are qualitative examinations of Mobile IP.

Table 4.1: Summary of Metrics Used for Evaluating Mobile IP

Metric Name	Units	Description
Successful Execution	Boolean	Whether or not an application was able to be started and used when Mobile IP was providing the network connectivity for the mobile node.
System Failure	N/A	Severity and frequency of system failures.
Round Trip Time	milliseconds	The time elapsed between sending a query to a correspondent node and receiving the reply.
Delay	milliseconds	The elapsed time between an agent receiving a packet and the packet being forwarded by the agent.
Network Load	Ratio	The ratio of the average Mb/s of traffic sent on the network to the total bandwidth of the network.
Throughput	KB/s	The rate of information transfer from the correspondent node to the mobile node.

4.3.1 Successful Execution

Mobile IP was tested using various applications. The successful execution metric is simply a boolean (true or false) indicating whether or not the application is able to run successfully using Mobile IP for packet delivery. Successful execution involves having the application run for an extended period of time and exchange packets with correspondent nodes. Success is indicated by several things, including whether or not the application starts up, how long it executes, how well it executes, and how well it handles movement across networks.

Execution success was evaluated by attempting to start an application on the mobile node. If it starts, the application is run over an extended period of time and general usability is evaluated from an end user's perspective. During this test, the mobile node was moved from one network to another to see if Mobile IP mechanisms for tracking across networks are transparent to the application.

4.3.2 System Failure

System failure is the failure of Mobile IP to provide the internetwork connectivity for the mobile node. Such failures occured in the test-bed environment because of equipment failure and bad configuration of the Mobile IP daemons. Examining the system failures involved noting what conditions caused the system to fail and what needed to be done to get the system working again. The frequency of the system failures and the ability of the system to recover from failure were also important for measuring the impact of a system failure.

4.3.3 Round Trip Time

Round trip time is an indicator of how responsive a real-time application is over Mobile IP. Real-time applications which respond to input from the user will suffer if the delay between the user's actions and the response is too large. A major contributor to this delay can be the round trip time to another node required to produce the response. Therefore, at least for user interfaces, the round trip time is an important metric for system usability.

Round trip time is measured as the number of milliseconds it takes a sequence of bytes to reach a correspondent node and then return to the mobile node. The correspondent node upon receiving this request responds with the traffic that is measured. The time between the sending of the request to the correspondent node and the arrival of the response is the round trip time.

The round trip time is affected by the location of the mobile node with respect to the correspondent node and its home agent. It is also affected by the characteristics of the packet stream, the speed of the nodes involved, characteristics of the networks involved, congestion at routers, and congestion on networks. By varying the location of the mobile node, the effect of using Mobile IP can be examined. By varying datagram sizes, and therefore the characteristics of the packet stream, scalability with traffic density can be examined.

The most significant factors in the round trip time are the throughputs on the networks between the nodes, and the distance between those nodes. These are physical factors which cannot be overcome. In the test-bed environment, the physical distance has effectively been removed from the equation, and the maximum throughputs of the test-bed networks remain constant through all experiments, so this factor is also removed. Therefore, measurements of the round trip time should indicate how the use of Mobile IP will affect real-time interaction with correspondent nodes.

4.3.4 Delay at the Home Agent

Delay at the home agent is meant to show how much time packets spend at the home agent while being forwarded to the mobile node. Delay is measured as the time between the packet's arrival at the IP incoming queue for the home agent, and its removal from the IP outgoing queue at the agent. Queueing delays at the hardware interface are therefore not recorded in this delay. What is recorded is the time to copy the packet information within kernel memory, the time for fragmenting and de-fragmenting the packets, the time to perform the IP-in-IP encapsulation, and the queueing delays at the IP layer. Since re-fragmentation at the home agent may occur, the delay is defined as the time until the last fragment of a packet has left the outgoing queue of the home agent. Delay is measured in milliseconds.

The delays on individual packets forwarded through the home agent may be different because of the amount and type of processing that must be done to forward the packet, and any queueing effects that may be present at an agent. Delay frequency plots show the packet delay distribution. Spikes in such a plot clearly show delays that are common to many packets, and therefore indicate a similar characteristic causing the delay for these packets. The frequency plot is used to show the delay metric as it is observed for the home agent in the test-bed.

4.3.5 Delay at the Foreign Agent

Delay at the foreign agent is the amount of time that a packet spends in the foreign agent being processed at protocol layers higher than the physical layer. The delay is measured as the time between the last fragment's arrival on the IP incoming queue, and the removal of the packet from the IP outgoing queue. Things affecting the delay include the de-fragmenting of the packet, queueing delays at the foreign agent's interface, and Mobile IP processing of the packet. Queueing delays at the foreign agent can also be complicated by traffic from the mobile node since the mobile node uses the foreign agent as its default gateway.

Delay at the foreign agent is measured in milliseconds. Frequency plots are used to show the distribution of the delays at various amounts of time, and effectively show amounts of time that are common delays experienced at the agent.

4.3.6 Network Load

Network load is used to show the impact of Mobile IP traffic on the networks that the mobile node uses. Network load is based on the ratio of the measured amount of mobile IP traffic in Mb/s to the total capacity of the network in Mb/s. Therefore, a ratio of 0.7 would indicate that 70% (7.0 Mb/s) of the network's 10.0 Mb/s capacity is being used by Mobile IP traffic when this measure is taken.

The network load is affected by the number of bytes used for encapsulating the packets, any re-forwarding that needs to be done on the networks involved, and the aggressiveness of the higher-layer protocols in securing bandwidth for transmitting the data. Encapsulating overhead increases if the packets need re-fragmenting at the mobility agents and therefore the number of encapsulation headers per incoming packet rises from one to two. Re-forwarding of packets on the home and foreign network effectively doubles the amount of bytes that must be transferred on the affected network to send the packet onto its destination. Finally, the aggressiveness of the higher-layer protocol affects the network load because a less aggressive protocol will maintain a more constant network load but will take longer to transfer the bytes. A more aggressive protocol will try to deliver the bytes quickly, but use up extra bandwidth to accomplish this.

Another aspect of network overhead is the number of bytes that are needed for the transfer of state information for the correct operation of Mobile IP. In the test-bed, this state information is small since there is only one mobile node, and the home agent is required to send out advertisements only once every second. If the number of mobile nodes increases, or the frequency of advertisements and registrations increases, the impact of this type of traffic will also increase. In the test-bed, the network load produced by this traffic is negligible.

Some of the network load can be examined analytically using the amount of the traffic that gets duplicated, and the number of bytes added to existing traffic. The extra network load actually produced by applications being executed on the mobile node require observations to be made on the network itself. These observations are made by examining the packets being transferred on the network and totalling the number of bytes being transferred by Mobile IP. These totals are updated once per second, and then an average over many seconds is taken to determine the average load for that time period.

4.3.7 Throughput

Throughput is measured as the amount of data (in bytes) transferred per second between two nodes in one direction. Only data bytes are counted for throughput results, and not total packet size. Throughput is a good metric for illustrating how well the underlying protocol layers deliver the information being transferred by applications. In particular, bulk data transfers are greatly affected by the attainable throughput because this determines the length of time that a transfer will take. Throughput over the Mobile IP link is measured for traffic travelling from a correspondent node to the mobile node. Bulk data transfers are used for generating the traffic being measured. The throughput is measured both internally by the application, and externally by timing the execution time of the application performing the bulk data transfers. The second method is less accurate than the first, but large data transfers are used to try and minimize the relative impact of the time to start and terminate the application itself.

4.4 Experimental Factors

The factors in these experiments include the traffic types, the applications, the number of mobile nodes, and the location of the mobile node.

4.4.1 Traffic Types

Three types of traffic are studied in this thesis: ping traffic, FTP, and NFS. Ping traffic is used because of the ability to easily control the datagram size being transferred. In addition, the traffic is not affected by measurement activity at the transport layer or higher. NFS traffic has less control, and brings transport layer protocols into the picture. FTP introduces congestion control mechanisms, windowing, and other more sophisticated transport layer functionality to the traffic stream.

Ping traffic was varied in size from 64 to 11748 bytes. Varying the size of the datagrams allows for the effects of fragmentation and queueing to be studied independently and incrementally, using previous experiments as base cases. Ping datagrams of 64 bytes do not suffer fragmentation. Pings of 2938 bytes suffer fragmentation at the correspondent node and re-fragmentation at the home agent for one of the packets in the datagram. Pings of 2958 byte suffer re-fragmentation for both packets at the home agent. Pings of 2968 bytes show re-fragmentation for two packets and none for the third, but queueing effects start to emerge. With datagram sizes of 5828 bytes and 11748 bytes, most of the packets are re-fragmented, and queueing becomes significant.

4.4.2 Applications

Various applications were used for generating traffic over Mobile IP. Primarily four types of traffic were used for operability and performance testing. X-windows traffic is generated using graphical-oriented programs remotely from the mobile node. X-windows applications are a desirable addition to a UNIX environment, and Mobile IP would not be adequate if graphical interfaces such as X-windows could not run effectively over the link. The graphical, real-time nature of the X-windows traffic is sensitive to delays in the traffic stream.

Telnet is a method commonly used for connecting to and using remote computers. The ability of telnet to work over Mobile IP constitutes a basic functionality issue that is essential for demonstrating adequate network connectivity.

FTP is used for bulk data transfer. The majority of the traffic from FTP is transferred in large datagrams from the correspondent node to the mobile node. Some small amount of traffic flows in the reverse direction for regulating the flow, and issuing requests to the FTP server.

NFS traffic is used on local area networks for maintaining distributed files systems. As a result, many UNIX-based nodes rely heavily on NFS to provide remote file system support for the end user's environment. Usually NFS-mounted filesystems are not exported beyond the confines of a local area network. Mobile IP provides a method for crossing the network barrier and delivering the environment to nodes operating on foreign networks. Effective NFS delivery means location-independent usability of all NFS-mounted filesystems, and therefore facilities and information local to the home network.

4.4.3 Number of Mobile Nodes

The number of mobile nodes is an important factor for scalability measurements. Because of the previously mentioned limitations in the test-bed, only one real mobile node can be provided for testing. However, simulated mobile nodes can be used to provide extra traffic over the Mobile IP link approximating the effect of increasing mobile node populations. The experiments reported here used 20 simulated mobile nodes to provide interference traffic for the real mobile node. Results with 1 and 21 mobile nodes can be compared to show any difference that may have occurred when more nodes were added to the networks.

4.4.4 Network Connection of Mobile Node

Perhaps the most important factor of all is where and how the mobile node connects to the internetwork. Four different connection states were identified for this work: home network, foreign network with no Mobile IP, foreign network with Mobile IP, and disconnected or in transit. Each connection state places its own restrictions on the traffic routed to and from the mobile node.

When the mobile node is on the home network, the Mobile IP mechanisms do not interfere with traffic flow unless the mobile node fails to receive router advertisements from its home agent. While on the home network, the mobile node behaves as any other fixed node, and can communicate directly with correspondent nodes on the home network.

The move to a foreign network entails a decrease in performance due to packets being forwarded through a router. To test these effects the mobile node was placed on the foreign network as an actual node on that network. The permanent IP address for the mobile node was set to an address on the foreign network. In this way, the exact same node is used for performance measurement at each location.

Mobile IP adds another layer of protocol interaction, and this is investigated by having the mobile node move to the foreign network and register with the foreign agent. When the mobile node is connected in this manner all traffic to and from the mobile node passes through the foreign agent, and all traffic to the mobile node passes through the home agent. This is the third connection state examined.

The simplest type of network connection is none at all. When the mobile node is in transit between networks, no traffic can be routed to or from it. This situation obviously does not allow for sophisticated performance analysis, but disconnection can cause application failure if it lasts long enough.

4.5 Instrumentation

Most of the data that is used in the experiments is gathered using *tcpdump*. This program gathers its data using a packet filter that is built into the operating system kernel. The packet filter monitors all data travelling in and out of the network interface of the node on which the filter is installed. Using such a filter, all traffic on a network can be monitored.

The probes for the packet filter are located between the IP queues and the interface queues. Because of this location, measurements taken on a host will record the timestamp of a packet leaving the host after it has left the IP queue, but before it has left the interface queue. Similarly, an incoming packet will be examined after the interface queue, but before the IP queue. Therefore network queuing can be examined using such a filter, but not interface queuing. In the test-bed, the packet filters are used on the home and foreign agents to monitor traffic going into and out of these nodes.

The timestamp on the filter has microsecond resolution. This amount of resolution is needed for examining delays at the agents since many of the delays experienced by packets at the agents are below 1 millisecond. Delay at an agent is computed by subtracting the timestamp when the packet entered the agent (enters the probe at the incoming queue) from the timestamp when the packet exits the node (enters the probe at the outgoing queue).

Much of the statistical information about traffic packet is extrapolated from packet traces taken with the tcpdump filter. The traces taken by tcpdump are post-processed using special scripts to extract the needed data. The necessary data can include the timestamp, the source, the recipient, the type of protocol, or the size of the packets in the trace.

4.6 Workloads

4.6.1 FTP Traffic

FTP traffic was generated using a script file to perform consecutive FTP transfers. Each FTP transfer obtained a 10 Mbyte file from the correspondent node and placed it on a local filesystem at the mobile node. A 10 Mbyte file was chosen because it is large enough to reduce the significance of startup effects, it gives TCP enough time to effectively start managing the bandwidth, and it reduces the significance of errors in the measurement techniques for both FTP and NFS. All of these benefits occur because the transfer time for a 10 Mbyte file is much larger than any of the other effects noted. 100 consecutive FTP transfers were done for each connection state.

4.6.2 NFS Traffic

NFS traffic was generated using a copy command to transfer a file from an NFS-mounted filesystem on the correspondent node to a local filesystem at the mobile node. As with FTP, the file transferred was 10 Mbytes in size, and 100 transfers were made with this file.

4.6.3 Ping Traffic (Single Mobile Node)

The ping application was used to generate traffic for two purposes. First, the datagram size can be varied for the ping traffic, and therefore the effects of increasing datagrams size can be examined. Second, ping traffic can easily be sent to real and fake mobile nodes so that the scalability of Mobile IP with the number of receiving nodes can be studied.

Ping traffic to a single mobile node was generated by sending a ping request from the mobile node to a correspondent node. These requests were sent out once per second, except for 64 byte pings which were sent out twice per second. 4000 pings were sent out for each datagram size. The resulting responses generated traffic through the home and foreign agents with datagrams the same size as the request datagrams.

As described earlier, the ping sizes used for generating traffic to the mobile node were 64, 2938, 2958, 2968, 5828, and 11748 bytes. These sizes were chosen to demonstrate different amounts of re-fragmentation at the home agent, and to show how much re-fragmentation can affect forwarding. The close size of the 2938-byte, 2958-byte, and the 2968-byte packets show how differing fragmentation effects can cause large changes in forwarding characteristics even with small changes in packet size.

The fragmentation of packets at the home agent is a source for the delay experienced there and should be correlated with increases in delay experienced as datagram size increases. 64-byte datagrams provide a stream of packets which do not require re-fragmentation at the home agent, and which do not get fragmented at the correspondent host itself. 2938-byte datagrams get fragmented at the correspondent node because the MTU (Maximum Transmission Unit) on the home network is only 1500 bytes. The two packets are of size 1480 and 1458 bytes. When the packets are forwarded through the home agent, the Mobile IP encapsulation headers increase the size of the first packet to 1500 bytes, causing it to be re-fragmented. After encapsulation, the size of the second packet is only 1478 bytes, and it is not fragmented by the home agent. At 2958 bytes, the datagram gets fragmented at the correspondent node into packets of size 1480 and 1478 bytes, both of which get re-fragmented at the home agent. With 2968 bytes, three packets are sent from the correspondent node, with the first two requiring re-fragmentation but not the last one, because it is only 8 bytes long.

With the larger datagram sizes, the correspondent host produces many packets, most of which get re-fragmented at the home agent. A 5828 byte datagram results in four packets from the correspondent node, three of which get re-fragmented. A 11748-byte datagram yields eight datagrams, seven of which get re-fragmented at the home agent. Thus it is apparent that delays caused by re-fragmentation, and problems resulting from re-fragmentation should be most apparent with the larger datagram traffic.

Although the ping traffic being studied in detail is the traffic going from the correspondent node to the mobile node, there is also traffic travelling in the opposite direction. For every response datagram from the correspondent node, there is first a request datagram from the mobile node of equal size. Although this traffic does not suffer from the re-fragmentation at the home agent, and there is no forwarding of this traffic between the home and foreign agents, this traffic affects the performance because it is routed through the mobile node's default gateway, the foreign agent. The traffic travelling from mobile node to correspondent node has the same properties as that coming the other way except for re-fragmentation.

4.6.4 Ping Traffic (Multiple Mobile Nodes)

When multiple mobile nodes are simulated, the traffic level is much higher, but generated in the same manner. With multiple mobile nodes, the ping datagram size was 64 bytes (512 bits). Therefore, no fragmentation delay was experienced. The ping requests were sent two times per second, therefore 42 separate requests were sent out per second since each simulated mobile node gets a ping request sent to it from the correspondent node. There are also two replies (from the correspondent node to the mobile node) sent out per second. Therefore the total traffic in bits per second including headers was 29568. This is a very small percentage of the capacity of the networks involved (10 Mb/s), and therefore poor performance due to network bandwidth limitations should be minimal.

Ping requests sent from the original mobile node generated responses from the correspondent host at a rate of two per second. Pings to the fake mobile nodes were sent from the correspondent node to the fake addresses, and this traffic was routed using Mobile IP if the nodes have been registered. No return traffic is generated.

Chapter 5

Experimental Results

5.1 Operability

5.1.1 Successful Execution

The results of application testing are presented in Table 5.1. This table shows the applications tested, and the results that were observed for those applications. The results include whether or not the application would start, the duration of time for which the application remains operational, how the application reacts to host relocation, and the quality of service (QOS) observable for this application while using Mobile IP.

X-windows was studied here as an application. This is because X-windows is a very popular application and it operates using protocol layers above IP. X-windows communicates with other nodes on the internetwork to display the remote applications on the mobile node's visual display.

Programs executed remotely include *Netscape Navigator*, *MPEG play*, *xterm*, and *xedit*. All applications were started from the command line of an *xterm* session running on a remote computer. Once the programs were started, normal user interaction such as web browsing and file editing was performed.

X-windows ran successfully on the mobile node regardless of its connection to the internetwork. Throughout the experiments X-windows remained operational, although individual programs died, or suffered delays when the mobile node moved across networks because of their disconnection from the X-windows server on the

Table 5.1: Success Results for the Applications Tested

Application	**Execution**	**Duration**	**Relocation**	**QOS**
X-windows	Successful	Indefinite	No Effect	Slight Delay
Ping	Successful	No Limit	Packet Loss	Substantial Delay
telnet	Successful	Indefinite	Potential Connection Loss	Slight Delay
FTP	Successful	Indefinite	Slow Recovery From Movement	Apparent Reduction in Throughput
NFS	Successful	Indefinite	Service Disruption, Quick Recovery	Significant Delay

mobile node. X-windows remained running even through periods of disconnection from the internetwork and quickly recovered when full network connectivity was resumed.

The quality of service demonstrated by X-windows over Mobile IP is comparable to that of the mobile node on the home network. Screen updates suffer slight delays over Mobile IP, delays that can be noticed when moving windows on the screen. Real-time echoes for user actions on application displays are instantaneous in most cases, and only rarely are delays noticeable.

Delays are more noticeable when the *ping* application is used for testing. The *ping* application is essential to many of the experiments performed here. Fortunately, *ping* is operational with Mobile IP, and may run indefinitely on the mobile node. The *ping* application itself is unaffected by network conditions provided it can resolve the correspondent node's address and start sending the pings. If network problems do occur, *ping* merely reports them as lost packets or increased delay on the packets. Although the networks in the test-bed are adjacent to each other, and packet loss, if any, is expected to be minimal, *ping* actually reported loss sometimes exceeding 1% for its traffic loads. Some of this packet loss can be directly attributed to large delays exceeding the time-out for the datagram.

Relocation across network boundaries affects *ping* by causing packet loss while the mobile node is disconnected. Every packet sent out while the mobile node is disconnected gets lost and is recorded as such by the *ping* application. Once the mobile node reconnects at its destination network, Mobile IP and *ping* recover quickly.

Telnet was tested both to and from the mobile node. Connection was possible in both cases and connections were maintained for an extended time. Setup times for the *telnet* connections seem to be longer over Mobile IP than they are when the mobile node is on the home network, but *telnet* sessions are usually set up within five seconds in any event.

Once set up, the sessions were fully usable for a long time, but could be interrupted by movement between networks. The movement between the networks resulted in disconnection of the mobile node, and if the mobile node is disconnected long enough (more than five minutes), the *telnet* session can die.

The quality of service for *telnet* over Mobile IP is good. Some character echo delay was noticeable when the mobile node experienced a high traffic load, but the delays were non-disruptive for the adequate use of the *telnet* application.

NFS was also tested for operability with Mobile IP. NFS was included in this study despite it being more an operating system enhancement than an application. The workability of NFS is important for presenting a consistent user environment both on and off the home network. The work in this thesis is based upon UNIX systems, and since NFS is used for distributed filesystems for UNIX systems, the operability of NFS is crucial for maintaining the home network environment if filesystems are distributed and used by the mobile node. Therefore, NFS is a crucial tool for UNIX that needs to be compatible with Mobile IP.

NFS was tested by having the mobile node mount a filesystem from a correspondent node on the home network. The mounting of the filesystem was successful both on the home network and the foreign network. On both networks, NFS filesystems were correctly mounted upon reboot of the system and were unmounted or mounted successfully while the system was running.

NFS seems to be stable on the mobile node both on the home and foreign network. Problems arise, however, when the node is disconnected from the source of the NFS filesystem. In this situation, the application accessing the mounted filesystem becomes blocked and unusable until connection to the NFS source is restored. On the test-bed, it was observed that NFS recovered very quickly (within seconds) upon re-connection to the network.

The quality of service for NFS was good. Basic operations like changing directories, moving files, renaming files, and editing files proceeded normally. Significant delays were noticed, however, with file system copies from a NFS-mounted filesystem to a local filesystem. These delays are discussed further in Section 5.4.

In general, Mobile IP is able to provide an environment consistent with that of a fixed node on the home network. All the applications tested work with Mobile IP and the quality of service is acceptable. The end-user at the mobile node suffers only small inconveniences except for when the mobile node is actually disconnected from the network.

5.1.2 System Failures

Table 5.2 shows the system failures that were encountered in the test-bed and how these failures affected the Mobile IP environment. Many of the failures were caused by events external to the Mobile IP system itself, but it is interesting to study how these problems affected Mobile IP. The table presents the failures in three columns. First a descriptive name is given to the problem. Second, a brief description of the problem is given. Third, a description of how the problem affects Mobile IP is given. The data shown in Table 5.2 is also discussed further in this section.

The first system failure discussed is router failure. The router being used in the test-bed had hardware problems which caused it to become unusable and require a reboot before routing successfully again. This caused many problems but also allowed for a study of Mobile IP's behaviour under such adverse conditions. A router failure occurs when the router no longer responds to network traffic directed

Table 5.2: System Failures in the Test-bed

Type of Failure	Description	Effect on Mobile IP
Router Failure	Router fails to respond to network traffic	Mobile node becomes disconnected because the foreign agent can no longer contact the home agent. If left long enough, this can cause the ignoring advertisement failure.
Mobile Node Ignores Advertisement	After a series of unsuccessful attempts, the mobile node will ignore further advertisements from a foreign agent but does not solicit other advertisements either.	Mobile node remains disconnected from internetwork until it leaves the foreign network. Upon re-connection to any network, registration proceeds normally.
Solicitations not Sendable	Daemon resets system in such a manner that agent solicitation messages result in generation of a "Network Unreachable" error.	Mobile node can no longer solicit an agent. In fact, the mobile node cannot even reply to advertisements that it receives. Reboot of the mobile node is required.
False Loss of Home Agent	Mobile node does not hear advertisements from its home agent before the last one times out.	Mobile node assumes it has left home network, clears routing table, and begins solicitation.

at it. Essentially, this cuts off all forwarding over the router, and therefore all communication between the foreign and home networks.

When Mobile IP is faced with a router failure, it cannot tell what has happened, but just tries to deal with it. Naturally with the router not operating, traffic can no longer be forwarded from the home agent to the foreign agent, and the mobile node cannot send traffic beyond the foreign network. Eventually, the registration times out, and the mobile node tries to re-register through its foreign agent. The foreign agent cannot forward the registration to the home agent, and it fails. The mobile node now waits an ever-increasing amount of time until it tries to register again. If routing is restored, the mobile node registers with the foreign agent and packet forwarding is restarted.

If the above problem persists for a long time, the mobile node starts ignoring advertisements from the foreign agent. This presents a problem, because even when routing is restored and the foreign agent can communicate with the home agent, the mobile node refuses to register with the foreign agent sending out advertisements; therefore re-connection is impossible. The only way to fix this problem is to remove the agent advertisements through disconnection of the mobile node. The mobile node now thinks it has left the foreign network, clears its information, and begins solicitation. Upon re-connection to the foreign network, the mobile node successfully registers through the foreign agent. This problem is shown as the second failure in Table 5.2.

The third type of system failure encountered on the test-bed involved the internal network setup on the mobile node. Things such as routing tables are modified when the mobile node changes its connection state to the internetwork. For instance, when the mobile node leaves a network its routing table is cleared, and when it arrives at a new network, default gateways and routes to other important nodes are set up. In most cases this re-configuration is exactly what is needed.

Occasionally on the test-bed the mobile node configured itself in such a way that it could no longer send or receive any network traffic whatsoever. This situation was indicated by a "Network Unreachable" message appearing whenever the mobile node daemon tried to send out agent solicitations. It is unclear what caused this state to occur, but reboots were necessary to restore operability of the mobile node.

The fourth type of system failure encountered on the test-bed was the false loss of the home agent when the mobile node was resident on the home network. This type of failure is actually not critical in that network connectivity is not lost for an extended time, but it does cause extra processing for the mobile node and home agent, and increases administrative traffic for Mobile IP. The problem arises when the mobile node has a home agent time-out configured to be less than the time between home agent advertisements.

When the time-out at the mobile node is less than the time between advertisements from the home agent, the mobile node considers itself to be disconnected from the home network every time the time-out period elapses. This starts the mechanisms in the mobile node for leaving a network, which include clearing out the routing table and soliciting for an agent. Upon receiving the solicitation, the home agent quickly advertises its presence, and the mobile node resets itself to its home network configuration. Internetwork communication for the mobile node can be adversely affected while it considers itself disconnected, because it no longer has gateways and name servers configured. Because of this problem, packets may be lost, and a large amount of extra processing is generated at the mobile node. This problem is easily fixed by changing the settings in the configuration file of either the mobile node daemon or the home agent daemon so that the home agent advertises multiple times before the mobile node times out.

The problems mentioned in this section represent possible problems that can be encountered with Mobile IP. Although the problems seem to be caused by events external to Mobile IP, the connectivity of the mobile node can be greatly affected. Further investigation into some of the failures may be needed to prevent future problems with the protocol (at least in the test-bed). In addition, the way Mobile IP handles some of the problems may need to be addressed.

5.2 Throughput

Throughput over Mobile IP was tested by transferring a 10 Megabyte file from the correspondent node to the mobile node. Throughput was measured for NFS by dividing the size of the file by the time it took to deliver the file to the mobile node. Throughput for FTP is calculated by the FTP application itself. For each traffic type 100 transfers of the file were examined.

Each traffic type was examined with the mobile node in three configurations. The first configuration had the mobile node on its home network. This is a normal packet transfer within a network, and therefore was the least complex of the three configurations. The second configuration had the mobile node on the foreign network, but with a permanent IP address on the foreign network. This configuration showed the performance across the network boundary, without Mobile IP overhead. The third configuration involved having Mobile IP deliver packets to the mobile node on the foreign network.

5.2.1 NFS

NFS throughput results are shown in Table 5.3. The results are based upon averaging the values obtained from 100 experimental runs. The average value for the throughput (in KB/s) is given in the middle column of the table. The first column shows the lower limit of the 95% confidence interval for the average. The third column shows the upper limit of the 95% confidence interval.

These results show that the throughput with the mobile node on the foreign network is not much less than with the mobile node on the home network. The throughput decreases substantially, however, when Mobile IP is used for packet forwarding. The average throughput on the home network is 411.085 KB/s, while the average throughput on the foreign network is 335.414 KB/s (a reduction of 18.4%), and the average throughput over Mobile IP is 156.773 KB/s (a reduction of 61.9%). This decrease is shown graphically in Figure 5.1. The performance on the home network, foreign network, and over Mobile IP are clearly shown by the spikes at 400 KB/s, 335 KB/s, and 155KB/s, respectively.

The confidence interval with the mobile node on the home network overlaps with the confidence interval where the mobile node is on the foreign network. Therefore it cannot be said with 95% confidence that there is a difference between the average throughput to the home network and the average throughput to the foreign network.

Table 5.3: 95% Confidence Intervals for NFS Throughput (KB/s)

Node Location	Lower	Mean	Upper
Home Network	331.458	411.085	490.712
Foreign Network (Mobile IP)	130.124	156.773	183.422
Foreign Network (No Mobile IP)	270.129	335.414	400.699

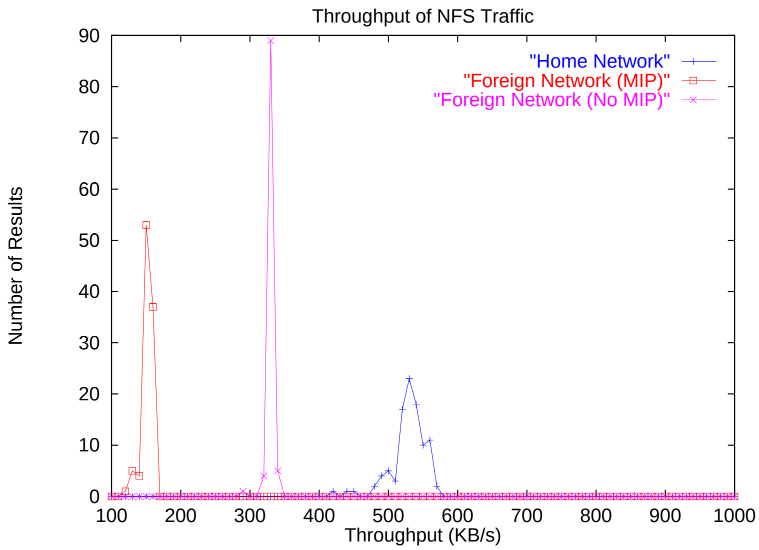


Figure 5.1: NFS Performance

The confidence interval for the Mobile IP traffic does not overlap with either of the other confidence intervals, and therefore, it can be said with 95% confidence that the throughput of NFS over Mobile IP is less than the throughput of NFS without Mobile IP.

Figure 5.1 shows the frequency curves for throughput observations between 0 and 1000 KB/s. The vertical axis represents the number of observations (out of 100) at the throughput value, and the horizontal axis measures the throughput itself. This figure shows the full distribution of throughputs observed at each location. The observed values for each mobile node location are tightly focused on a narrow range of throughput values, thereby generating the tall and thin spikes seen in the performance plot. The distinct spikes in the figure show little variance for throughput at the same location as compared to the variation across locations.

Table 5.4: 95 % Confidence Intervals for FTP Traffic (KB/s)

Location	Lower	Mean	Upper
Home Network	429.522	534.370	639.218
Foreign Network (Mobile IP)	274.661	341.421	408.182
Foreign Network (No Mobile IP)	339.485	406.030	472.574

5.2.2 FTP

The throughput results for FTP traffic are summarized in Table 5.4. This table shows the lower and upper values for the confidence intervals, and well as the average throughput obtained. Values are given for each of the three test cases examined. The results reveal a large amount of variability with FTP performance, and the confidence intervals show large overlaps with each other.

On the home network, an average transfer had a throughput of 534.370 KB/s with the confidence interval extending from 429.522 to 639.218 KB/s. On the foreign network, the average throughput dropped to 406.030 KB/s, with the confidence interval ranging from 339.485 to 472.574 KB/s. With Mobile IP added, the average dropped to 341.421 KB/s, with the confidence interval ranging from 274.661 to 408.182 KB/s. The results are shown graphically in Figure 5.2.

The confidence intervals indicate that it cannot be said with 95% confidence that the FTP performance on the home network is different from the FTP performance across the gateway. It also cannot be said that the FTP performance across the gateway is different form the FTP performance over Mobile IP. But, it can be said with 95% confidence from our results that the average throughput for FTP is lower when using Mobile IP than when the information is delivered directly to a mobile node on its home network.

Figure 5.2 shows the frequency curves for throughput observations between 0 and 1000 KB/s. The vertical axis represents the number of observations at the throughput value, and the horizontal axis measures the throughput itself. In contrast to the NFS results, this data shows that the observations are not tightly coupled

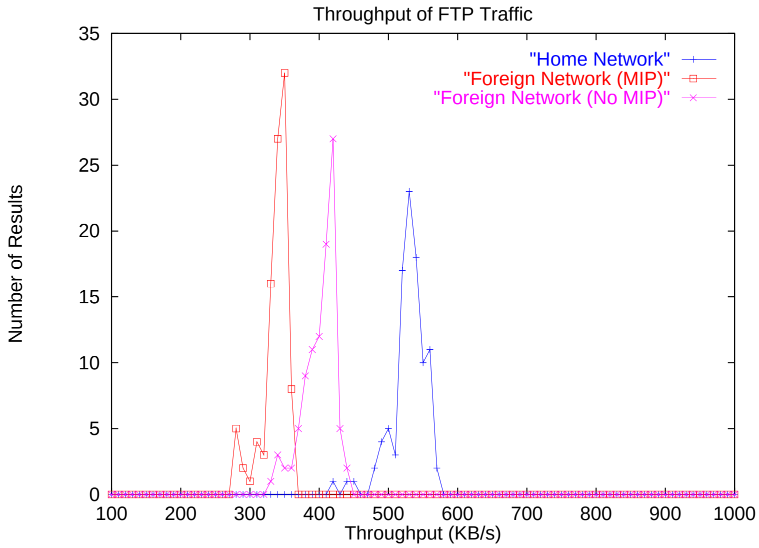


Figure 5.2: FTP Throughput Performance

around a mean (the spikes are short and wide). This indicates that the throughput of FTP traffic has more variation than that of NFS traffic on the test-bed. Significant overlap is shown between the observations at the different locations, indicating other factors are affecting the throughput as much as the location changes.

The difference in variation between NFS and FTP is substantial given that both types of traffic were used on the same network, with the same file. The cause for the variation is not known for sure, but part of the variation may be a result of the aggressiveness of these protocols. The experimental results have shown that FTP consumes the bandwidth more aggressively than NFS. A more aggressive protocol, consuming more bandwidth if it can, is more likely to be susceptible to ambient traffic on the network. Since the home network in the testbed was not isolated, ambient traffic did exist. It is likely that the ambient traffic would have affected the FTP traffic more than the NFS traffic, and this shows up in the plots as a greater variability.

These results indicate that FTP traffic is less susceptible to the problems introduced in our test environment. The higher performance in all situations tested also indicates that the mechanisms in FTP effectively maximize the bandwidth that is available in the Mobile IP tunnel.

5.3 Round Trip Time

The round trip time for packets travelling between two hosts can have serious effect on real-time interaction but has limited effect on non real-time applications unless the round trip time exceeds time-out periods. In this section, the round trip time for ping packets is studied. Because of the delay effects at the mobility agents, the round trip times are likely to vary as the ping datagram size increases. This relationship is the key interaction studied.

The round trip time results are presented in three ways. First, the percentiles for the round trip times are presented for all datagram sizes. The percentile graphs are produced by ordering the round trip times from smallest to largest, and then plotting the times against the percentage of packets with round trip times less than that. The 50th percentile shows the median round trip time, and the 100th percentile shows the maximum round trip time. In the results presented here, times up to the 95th percentile are shown. Second, the mean round trip times are examined for the different traffic types. Finally, the median values are examined.

5.3.1 Distribution of Round Trip Times by Packet Size

The cumulative distribution for ping round trip times is given in Figure 5.3. The types of traffic shown are ping datagrams of size 64, 2938, 2958, 2968, 5828, and 11748 bytes. The data in Figure 5.3 shows that the distributions are quite similar for the first 50% of the packets. For all datagram sizes, the curves rise steeply, and 50% of the values are below 100 milliseconds. The function for 11748 bytes levels out sharply at 60%, and the distribution climbs slowly until 1000 milliseconds, below which are approximately 75% of the values. The next 20% for 11748-byte datagrams have round trip times between 1000 and 1500 milliseconds. The function for 5828 bytes begins to level out at 80%, so 80% of the round trip times for 5828-byte datagrams had round trip times less 50 milliseconds. From the graph it is apparent that approximately another 10% of the datagrams have round trip times between 50 and 1000 milliseconds, and most of the remaining 10% have round trip times

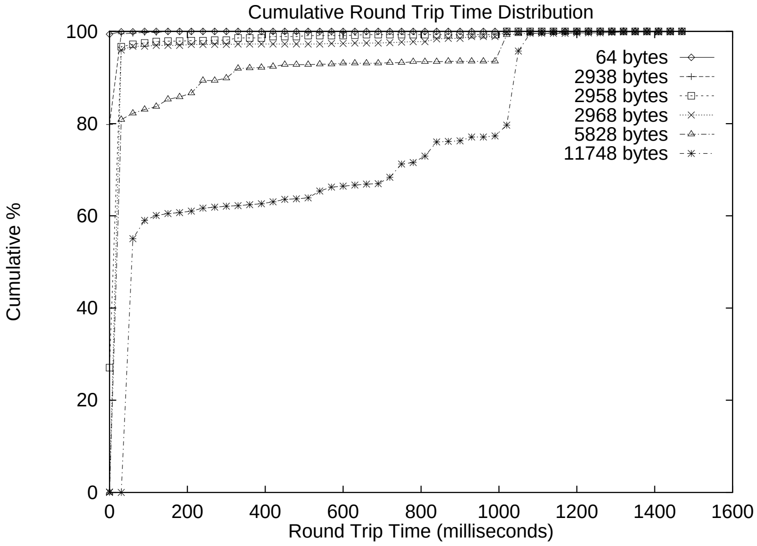


Figure 5.3: Cumulative Distribution for Ping Round Trip Time

less than 1100 milliseconds. All of the other datagram sizes have distributions that rise sharply to 95% within 50 milliseconds, and then the remaining are spread out between 50 and 1500 milliseconds.

Figure 5.4 shows the cumulative round trip time distributions for the smaller datagram sizes in an expanded view. In this diagram, a clear distinction can be seen between the datagrams that experience re-fragmentation (those above 1480 bytes) and the datagrams that do not (64 bytes). The plot for the 64 byte datagrams climbs sharply up to approximately 98% of the values having round trip times less than 10 milliseconds. Smaller datagram sizes yield plots with large verticle steps indicating that the variance for the round trip times is very small, and every round trip time is extremely close to the average. The round trip times for the datagrams of size 2938 bytes through 2968 bytes are concentrated around 20 milliseconds for approximately 95% of the values in the distribution. This plot shows that the introduction of fragmentation has a significant effect on the round trip time (moving the distribution out from less than 10 milliseconds to approximately 20 milliseconds).

These plots show that for small datagram sizes, the major determining factor for round trip time was the fragmentation effects at the home agent. Datagrams of 64 bytes had no fragmentation at the home agent and completed the round trip

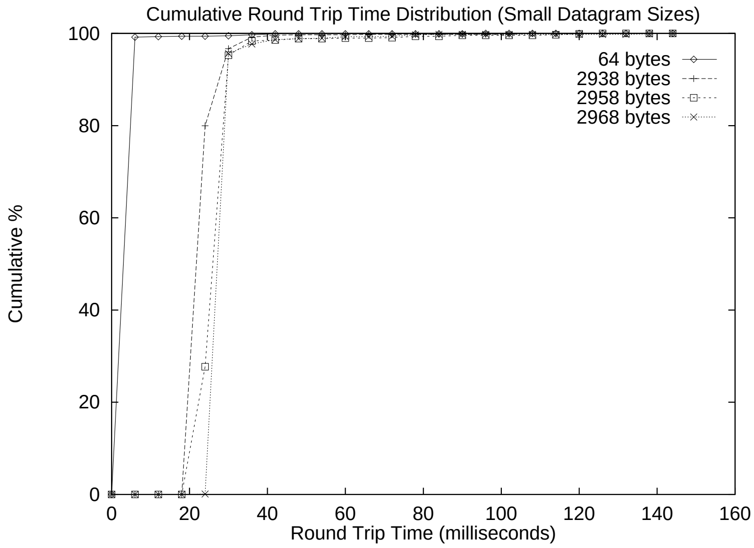


Figure 5.4: Cumulative Distribution for Ping Round Trip Time (Size <= 2968 Bytes)

within 7 milliseconds. Datagrams between 2938 bytes and 2968 bytes experienced fragmentation and showed delays much higher, around 30 milliseconds. The much larger datagrams, however, seemed to suffer from delays that cannot be explained by re-fragmentation alone. These are most likely due to the queueing effects noted in Section 5.4. These other effects greatly influenced the round trip times and increased them to beyond 1000 milliseconds for many of the 11748 byte datagrams.

The maximum round trip time for the various datagram sizes are shown in Table 5.5. The largest values are anomalously high and represent a very small fraction of the traffic. The maximum round trip times, like all other percentiles, increase as the datagram size increases. The maximum value for 64 bytes is the lowest at 101.188 milliseconds, although this is much larger than the 95% percentile of 7.672 milliseconds. The maximum round trip time for 2938 byte datagrams was only 179 milliseconds, while the maximum for 2958 byte and 2968 byte datagrams were 1031 milliseconds and 2027 milliseconds respectively. It is unclear why these maximum values change so drastically between 2938 and 2968 bytes. The times for 5828 bytes and 11748 bytes are the largest at 2247 milliseconds and 4390 milliseconds respectively.

The drastic change in the maximum values from 2938 to 2968 bytes is most likely

Table 5.5: Maximum Round Trip Times Measured Over Mobile IP According to Packet Size

Packet Size	Maximum RTT (milliseconds)
64	101.188
2938	178.988
2958	1031.26
2968	2027.136
5828	2247.004
11748	4390.124

caused by changes in the re-fragmentation that is required at these packet sizes. With 2938 bytes, only one packet per datagram needs re-fragmenting, and so there is little chance that the home agent will become overwhelmed with packets going into and out of its interface. With 2958-byte datagrams, two packets need re-fragmenting at the home agent, and therefore the chance of these packets interfering with each other increases substantially. With 2968-byte datagrams, the numbers of packets arriving at the home agent is increased to 3, and therefore there is more chance of interference between incoming and outgoing packets at the home agent. Beyond 2968 bytes, the increase in maximum delay tapers off indicating that a critical shift in the way that the incoming packets interact with the outgoing packets may lie within the 2938-byte to 2968-byte region.

5.3.2 Average Round Trip Times

The average round trip times show the delays that can be expected for any particular datagram of a certain size. Table 5.6 compares the average round trip times for traffic using Mobile IP, traffic sent to the foreign network but not using Mobile IP, and traffic restricted to the home network. The home network traffic is the base case and shows what can be expected with no forwarding of any kind. Traffic sent to the foreign network shows the effect of forwarding through a router. Mobile IP traffic shows the worst-case scenario, where Mobile IP forwarding is including in the round trip delivery path.

Table 5.6: Average End-to-end Performance Results

	Round Trip Time (milliseconds)		
Packet Size	On FN (with Mobile IP)	ON FN (without Mobile IP)	On HN
64	7.763	8.271	6.559
2938	30.397	20.467	17.166
2958	44.108	20.593	17.154
2968	59.079	20.523	17.806
5828	140.429	28.660	24.223
11748	472.407	45.753	38.912

The data shows that when packet size is small, little delay is added by either the router or the Mobile IP tunnel. The smallest packet size actually experienced worse end-to-end performance on the foreign network without Mobile IP than on the foreign network with Mobile IP. This is likely an affect of ambient traffic. Since the packet size is small, the Mobile IP effects are minimal due to lack of re-fragmentation, and ambient traffic effects are more noticeable. As the packet size increases, and the fragmentation effects become larger, the delay effects of Mobile IP and network traversal become more apparent. The packets delivered through Mobile IP exhibit the worst performance while the packets delivered directly without tunnelling show only slightly reduced performance.

In the average case, the effects of extreme performance values can be shown. In the performance tests, round trip times as high as 4390.124 milliseconds were recorded for traffic with packet size 11748 bytes. Both the value of the extreme results, and the frequency with which extreme values occur, increased as the packet size increased.

The average round trip times are shown in Figure 5.5. This figure shows that traffic produced on the home network and forwarded to the foreign network without Mobile IP is not significantly affected by increases in datagram size. The plots of these two traffic types seem quite flat compared to the round trip time plot of the Mobile IP traffic, which rises steeply as the datagram size increases. The plot seems

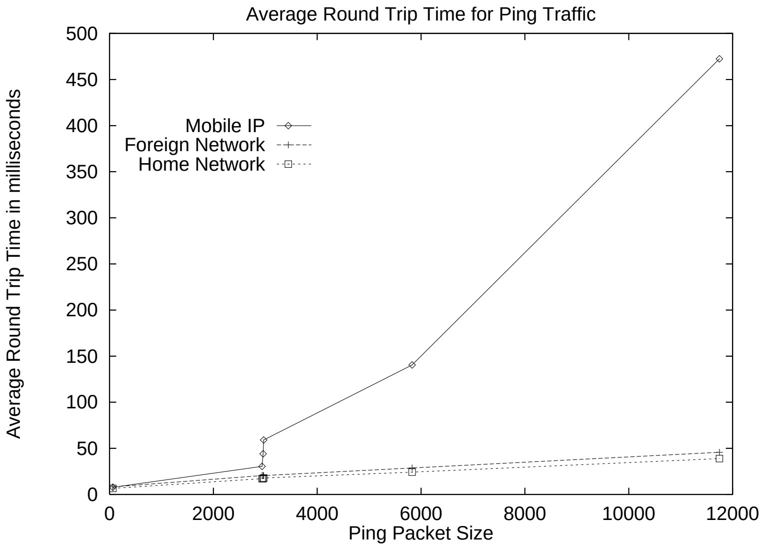


Figure 5.5: Average Round Trip Time vs. Ping Size

to suggest that beyond 2968 bytes the increase in round trip time is linear; however a step-like function is more likely, as the most significant jumps in delay are likely to occur at datagrams sizes where more fragmentation starts to happen.

5.3.3 Median Round Trip Times

The median round trip times for the different mobile node locations are shown in Table 5.7. The median results show the 50th percentile round trip times that were suffered by the datagrams. These results show that much of the average round trip time is produced by a smaller percentage of extreme values for the larger datagrams.

The median values are shown for the various packet sizes on the home network, foreign network, and over Mobile IP. The median values for Mobile IP traffic are always the largest, for each datagram size, while the other two values are practically indistinguishable for all datagrams less than 11748 bytes. The difference in round trip time of 7 milliseconds for the 11748 byte traffic is the most significant difference between performance on the home network and across the router without Mobile IP. The median values for all connection types rise as the datagram size increases. This is similar to what happens with the average values, although the rise is less steep in the case of the median values.

Table 5.7: Median End-to-end Performance Results

	Median Round Trip Time (milliseconds)		
Packet Size	On FN (with Mobile IP)	ON FN (without Mobile IP)	On HN
64	7.428	6.676	6.508
2938	29.252	19.200	16.600
2958	30.116	19.360	16.648
2968	31.272	19.908	17.256
5828	43.172	26.528	24.012
11748	83.996	45.412	38.536

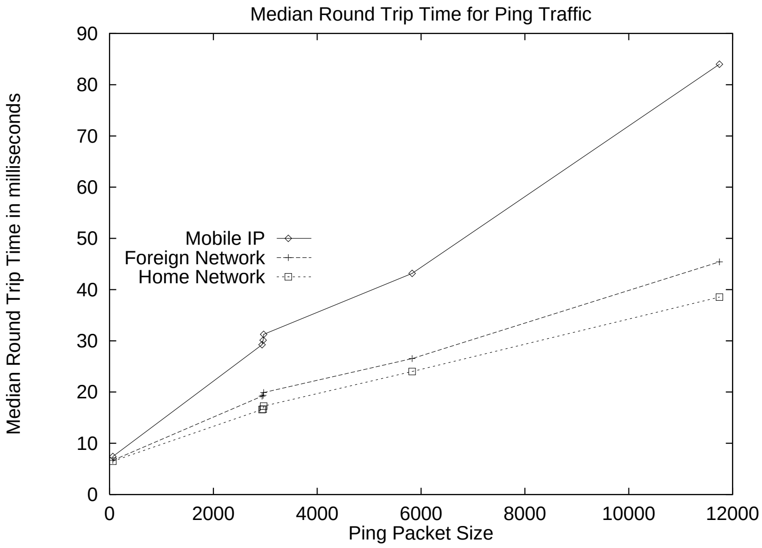


Figure 5.6: Median Round Trip Time vs. Ping Size

The medians are presented graphically in Figure 5.6. This figure shows that there is an increase in round trip time for all the traffic types, but the increase for the Mobile IP traffic is the most pronounced. The rise in round trip time is roughly constant for both traffic within the home network and traffic to the foreign network as datagram size is increased from 64 bytes (median round trip times of 6.508 and 6.676 milliseconds) to 11748 bytes (median round trip times of 38.536 and 45.412 milliseconds respectively). The rise in the Mobile IP delay from 7.428 milliseconds at 64 bytes to 83.996 milliseconds at 11748 bytes is also roughly constant.

5.4 Delays

5.4.1 Home Agent Delay

As the packets are forwarded through the home agent, they are delayed by processing and queueing. The types of processing include normal IP layer processing, encapsulation within another IP packet, possible fragmentation, and any table lookups that need to be performed to get the routing information for the mobile node. The packet is also delayed by queueing at the input queue before processing and at the output queue after processing. Because of the mechanism used to measure the delays in these experiments, the packet is considered to be sent once it is placed in the output queue.

The data presented here shows the delay on individual packets as they are forwarded through the home agent. Each packet coming into the home agent will be at maximum 1480 bytes because of fragmentation on the home network. Each packet seen by the home agent, however, is actually only a fragment of a larger packet (in most cases) sent from the correspondent node. This larger packet will be termed the datagram, and the results presented here are in terms of the datagram size. It is useful to examine the behaviour as datagram size increases, because the size of the datagram affects traffic burstiness, and the amount of fragmentation that must be done at the home agent. The datagram sizes examined here are 64, 2938, 2958, 2968, 5828, and 11748 bytes.

The delay at the home agent is a measure of the time between a packet fully arriving at the Ethernet interface and being transferred into the input queue, until it has been processed and forwarded onto the output queue. This is measured by tcpdump running on the home agent. Placing tcpdump on the home agent does introduce some additional load on the home agent, but it is assumed that this does not significantly affect the results. This assumption is based upon separate trial runs with tcpdump running on a machine other than the home agent. The results with this configuration are not significantly different from those with tcpdump on the home agent.

Table 5.8: Delays at Home Agent as Datagram Size Increases

Size (Bytes)	Delay (milliseconds)		
	Average	Minimum	Maximum
64	0.131	0.107	4.774
2938	1.084	0.118	59.899
2958	2.114	0.868	498.286
2968	3.745	0.100	508.296
5828	23.509	0.112	563.913
11748	85.834	0.117	2037.15

Delay as Packet Size Increases

A reasonable hypothesis for the effect of datagram size would be that it is minimal up to the size at which the packets begin to fragment. Since all of the fragments forwarded should be less than or equal to 1480 bytes, the forwarding times should be similar regardless of datagram size. This hypothesis was tested in the results presented in the following sections.

Table 5.8 shows a summary of the delays experienced as the datagram size increases. The columns in this table show the size of the datagram (in bytes), and the average, minimum, and maximum delays (in milliseconds) experienced by an individual packet. The times are presented in milliseconds because delays in this range are common with network traffic. A ping within a single LAN will usually be performed within 1 millisecond.

Datagrams of size 64 bytes (the standard ping datagram) experienced an average delay of 0.131 milliseconds, with a minimum of 0.107 milliseconds. This delay is a base case for the forwarding of packets through the home agent, since no fragmenting is done at the home agent, and little queueing should be experienced since only two packets of size 64 bytes are arriving per second. The results show a best-case time of 0.1 milliseconds should be expected for forwarding a packet through a home agent like the one in the test-bed.

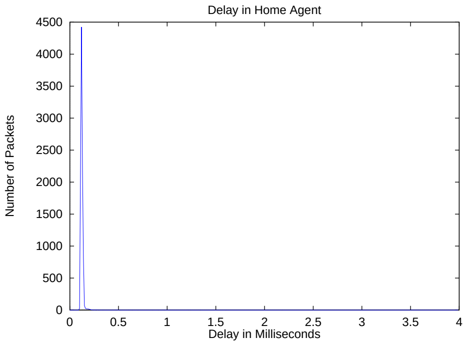
The average delays increased substantially as the datagram size increased, but the minimum delay was constant around 0.1 milliseconds in every case except when the packet size was 2958 bytes. This behaviour is examined more closely later in this section, when actual packet traces are examined. The large minimum delay is a result of every packet in the datagram requiring re-fragmentation at the home agent. The maximum delays increased sharply as the datagram size increases, and substantial delays were recorded for datagrams as small as 2938 bytes.

Datagrams of size 2938 bytes are fragmented once at the correspondent node and arrive at the home agent as two packets. The average delay of 1.084 milliseconds shows that the effect of fragmentation is significant and it increases the delay by an order of magnitude over the best case scenario. Extra delay is encountered with datagrams of size 2958 bytes. Despite the small increase in size (20 bytes), datagrams of size 2958 bytes produce packet delays double that produced for datagrams of size 2938 bytes. This rapid growth occurs again at 2968 bytes, when the delay again roughly doubles. These increases in delay are most likely a result of the different fragmentation characteristics of datagrams with sizes 2938, 2958, and 2968 bytes on a LAN with maximum frame size of 1514 bytes (which corresponds to 1480 bytes of IP data).

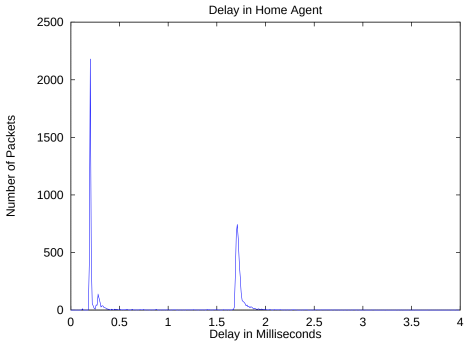
The Distribution of Packet Delays

The distribution plots in Figure 5.7 show the packet delays for each datagram size. The plots show the number of packets as a function of the delay suffered. Spikes in the curves indicate amounts of delay that are common to many packets in the trace. The variability of the delays can be inferred from the prominence of the spikes in the plot. Less prominent and wider spikes mean more variability.

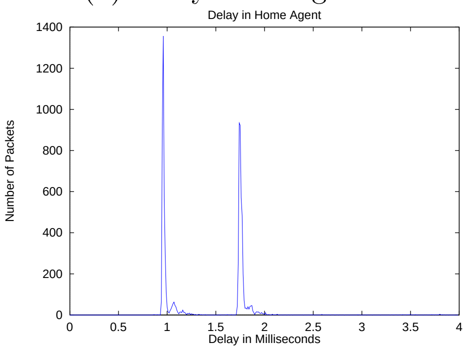
The distribution plots in Figure 5.7(a) through (f) are ordered by increasing datagram size and show a trend towards more variability and higher delays as the datagram size increases. With smaller datagrams, the spikes in the distribution are well defined and there are not many of them. This indicates that certain delays are very common for the packets that compose the datagrams. As the datagram size increases, the spikes become less defined which indicates that more variability is being introduced in the delays. The number of spikes also increases, and this shows that the number of different packet categories sharing a similar trait (which is leading to similar delay values in the plots) increases. The increase in the number of spikes as the datagram size increases is logical given the increase in the number of packets composing the datagram.



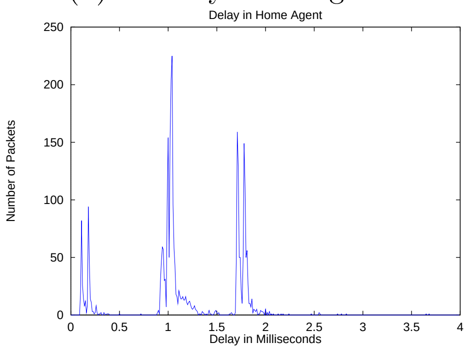
(a) 64-byte Datagrams



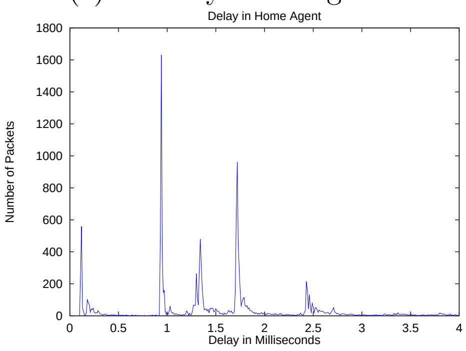
(b) 2938-byte Datagrams



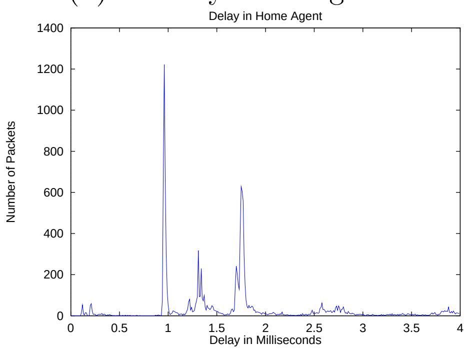
(c) 2958-byte Datagrams



(d) 2968-byte Datagrams



(e) 5828-byte Datagrams



(f) 11748-byte Datagrams

Figure 5.7: Packet Delays at the Home Agent as Datagram Size Increases

It is hypothesized that much of the delay in forwarding packets is due to re-fragmentation at the home agent and Mobile IP encapsulation. The amount of re-fragmentation that needs to be done is totally dependent on the size of the packets that arrive at the home agent. Any packet larger than 1480 bytes will need to be re-fragmented at the home agent because when the extra 20 bytes is added for the encapsulation header, there are no more bytes left for the Ethernet frame header that must be added.

The encapsulation time is more or less constant for all packets that are forwarded, except for hash table lookups for finding mobile node information. The difference in the encapsulation overhead happens when there are two packets produced for one incoming packet. In this case, the encapsulation overhead is doubled.

The frequency plot in Figure 5.7(a) illustrates the delay at the home agent with a ping size of 64 bytes. A ping packet of 64 bytes does not suffer fragmentation at the correspondent node, or at the home agent. Therefore, these results are the best estimate for the home agent's processing time for a packet. Note that the spike at 0.12 milliseconds is very defined in the plot, and there is only one such spike in the range shown. This indicates that all packets being forwarded for this traffic are experiencing similar delays at the home agent, and there is not much variance. This plot shows that 0.12 milliseconds is a good estimate for the delay that a simple packet will experience at the home agent.

With a datagram of size 2938 bytes, two packets are forwarded for each datagram. The first fragment is maximally sized, and must be re-fragmented at the home agent. The second fragment is small enough that it just gets forwarded normally, like the 64 byte packets. The distribution plot in Figure 5.7(b) shows two prominent spikes in the distribution. The larger spike at 0.2 milliseconds represents a delay twice that suffered by the 64 byte packets. The smaller spike at 1.7 milliseconds represents a significant increase in delay from the 0.1 milliseconds established for 64 bytes, and the 0.2 established for 1480 bytes. This increase is due to the time required to re-fragment the packet and release two fragments onto the network. Since the delay is calculated based upon the last fragment leaving the home agent, this delay also includes the time it takes to queue the second fragment behind the first.

With a size of 2958 bytes, both packets need to be re-fragmented at the home agent. The distribution plot in Figure 5.7(c) shows two prominent spikes at 1.7 milliseconds and 1.0 milliseconds. Although the spikes are still well defined, there is some variability apparent in the figure at the base of the spikes.

Figure 5.7(d) shows the performance when datagrams of size 2968 bytes are sent to the mobile node. Datagrams of this size are split into three packets at the correspondent node. These packets are reflected in three spikes in the plot. Two of the packets require re-fragmenting. These packets would correspond to the spikes at 1.0 and 1.7 milliseconds. The delays on these packets are the same as those shown in Figure 5.7(c). The third spike shows a concentration of packet delay around 0.2 milliseconds, and indicates a group of packets with minimal processing as shown in Figures 5.7(a) and (c). It is worth noting that the spikes in this plot are not as well defined as in previous plots, and therefore the variability in the delays is higher.

The final two plots show the behaviour when the datagram size increases drastically to 5828 bytes and 11748 bytes. Datagrams of these sizes correspond to four packets and eight packets respectively. The corresponding plots in Figure 5.7 have five distinct spikes and three distinct spikes respectively. In both of the plots, the variability in the delays has increased substantially.

Datagrams of size 5828 bytes exhibit concentrated delays as high as 2.5 milliseconds. Delays are also concentrated around 0.2, 1.0, 1.5, and 1.7 milliseconds. Four of the five packets require re-fragmentation at the home agent and would correspond to the four higher-delay spikes. The packets not requiring fragmentation would correspond to the delay spike around 0.2 milliseconds.

Datagrams of size 11748 bytes result in much variability in the delays, as illustrated in Figure 5.7(f). The three main spikes shown in this plot are concentrated around 1.0, 1.5, and 1.7 milliseconds. A small number of observations are also concentrated around 0.2 milliseconds. Much smaller and more spread out spikes occur at intervals above 1.7 milliseconds, and the average delay experienced with datagrams of this size is much larger than that with the smaller sizes.

Packet Delay Traces at the Home Agent

The datagrams sent to the mobile node are split into multiple packets at the correspondent node for all datagrams larger than 64 bytes. The position of a packet in a

Table 5.9: Delays at the Home Agent (64 Byte Datagrams)

Timestamps		
Input Queue	Output Queue	Delay
879113700.083815	879113700.083936	0.000121
879113700.182027	879113700.182160	0.000133
879113700.283625	879113700.283744	0.000119
879113700.381982	879113700.382183	0.000201
879113700.483861	879113700.483980	0.000119
879113700.582108	879113700.582258	0.000150
879113700.683652	879113700.683769	0.000117

datagram seems to affect the delay that packet experiences at the home agent. In this section, traces of packets being forwarded through the home agent are presented, and the delays on individual packets are explored. Evidence for the correlation between packet position and delay is inferred from these traces.

Typical delays for 64 bytes packets are shown in Table 5.9. These delays correspond to the large spike in Figure 5.7(a). The table shows the timestamp when the packet arrives at the home agent, the timestamp when the packet is forwarded by the home agent, and the difference between these (the processing time). The results show that the delays are very similar in magnitude for the packets examined.

A corresponding tcpdump trace for some of these packets is shown in Figure 5.8. The packet trace shows the mobile node (bobby) sending an echo request to the correspondent node (donald). The resulting response is sent to bobby's IP address, but routed through the home agent (morty). The Mobile IP packet destined for the foreign agent shows up as a packet from morty to the foreign agent (darlene). Because of the limited amount of traffic being forwarded through the home agent, this trace shows a best-case scenario for forwarding packets through the home agent. The packets being forwarded here do not need re-fragmenting, and they also do not have other forwarded packets to contend with for processing.

The delay measurements shown in Table 5.9 are for the packets flowing into and out of the home agent with 2938 bytes datagram traffic. This trace shows that the

```
879113700.083173 ip 98: bobby > donald: icmp: echo request
879113700.083815 ip 98: donald > bobby: icmp: echo reply
879113700.083936 ip 118: morty > darlene: ip-proto-4 84
879113700.181499 ip 98: bobby > donald: icmp: echo request
879113700.182027 ip 98: donald > bobby: icmp: echo reply
879113700.182160 ip 118: morty > darlene: ip-proto-4 84
```

Figure 5.8: Trace of Ping Traffic on the Home Network (64 Byte Datagrams)

Table 5.10: Delays at the Home Agent (2938 Byte Datagrams)

Timestamps		
Input Queue	Output Queue	Delay
879117634.000773	879117634.002512	0.001739
879117634.002446	879117634.002649	0.000203
879117634.500356	879117634.502071	0.001715
879117634.502007	879117634.502213	0.000206

packet delays alternate from large to small, and it turns out that the first packet of the datagram is delayed the most. The larger delay is relatively constant around 1.7 milliseconds, and the smaller around 0.2 milliseconds.

This large discrepancy in the delays indicates that the forwarding of one packet is greatly affected by the processing that needs to be performed on it. From the trace in Figure 5.9 it is clear that the first packet of the datagram gets re-fragmented at the home agent. It is reasonable to assume that the re-fragmentation process introduces the delay. However, the actual packet trace shows that the processing time at the home agent to fragment and forward the first piece of the fragmented packet is small, and that queueing and media contention delay the second half of the packet and produce the extraordinary delay.

The Mobile IP forwarding begins at time 34.500356 with the first reply packet arriving at morty from donald (the correspondent node). This packet is large enough to be fragmented into two separate fragments by the home agent. The first fragment is forwarded at time 34.500522 and the second at time 34.502071. There are two things of significance to note in this trace. First, the forwarding of the first

```
34.493646 ip 1514: bobby > donald: icmp: echo request (frag 30068:1480@0+)
34.495495 ip 1492: bobby > donald: (frag 30068:1458@1480)
34.500356 ip 1514: donald > bobby: icmp: echo reply (frag 24133:1480@0+)
34.500522 ip 1514: morty > darlene: ip-proto-4 1480 (frag 47503:1480@0+)
34.502007 ip 1492: donald > bobby: (frag 24133:1458@1480)
34.502071 ip 54: morty > darlene: (frag 47503:20@1480)
34.502213 ip 1512: morty > darlene: ip-proto-4 1478
```

Figure 5.9: Trace of Ping Traffic on the Home Network (2938 Byte Datagrams)

fragment of the first packet is very fast, less than 0.2 milliseconds. Second, the forwarding of the second fragment occurs after the arrival of the second packet from the correspondent host at time 34.502007. Since the home agent is required to process the packet from the correspondent host, and there is contention for the shared media, this packet arrival may contribute largely to the delay suffered by the second fragment of the first packet. This type of interaction is common in these traces.

The fragment information has been left in Figure 5.9 so that the size of the packets being forwarded can be shown. Notice that the first packet of size 1480 bytes is maximally sized for transmission on the media and gets re-fragmented into a packet of size 1480 bytes and another of size 20 bytes by the home agent. This re-fragmentation is evidence of the extra 20 bytes in header information required by the encapsulation method used by Mobile IP in the test-bed.

A packet size of 2958 bytes has an increased average delay per packet due to the re-fragmentation of both packets forwarded through the home agent. Because of the re-fragmentation, the minimum delay experienced in this case is over 0.8 milliseconds.

Even though both packets suffer the same fragmentation effects, the delays are quite different. As before, the packet suffering the largest delay is the first one. In this case, though, the difference in delay is caused by the arrival of another packet at the home agent. The trace in Figure 5.10 shows this happening. The first ping datagram shown here is composed of two packets, one of 1480 bytes and another of 1478 bytes. The first fragment from the home agent is sent out quickly (within 0.2 milliseconds of the packet's arrival). The second fragment, however, is not forwarded as fast, and during its processing, another packet arrives from the correspondent host, thereby further delaying the fragment's transmission. Typical delays for the first packet are 1.7 milliseconds.

Typical delays experienced by packets of 2958 byte datagrams are shown in Table 5.11. All packets forwarded for datagrams of this size experience significant but not equal delay at the home agent. A datagram of size 2958 bytes gets split into two fragments. These delays shown in Table 5.11 show that the first packet gets delayed the most although both packets get delayed significantly. In this table, it can be seen that the first packet is usually delayed 1.7 milliseconds and the second packet is delayed approximately 0.9 milliseconds.

Both packets require re-fragmenting at the home agent as shown in Figure 5.10. The first packet is sent from donald at time 10.089474 and the first fragment from this packet appears at time 10.089646. This represents a quick processing time, as was shown with 2938 bytes. The second fragment, however, is delayed until 10.091247. This delay coincides with the arrival of the second packet at time 10.091170, and the delay effects for the first packet of the datagram seem consistent with those of the previous datagram size. The second packet is delayed by the completion of the first fragment's forwarding, re-fragmentation, and queueing at the home agent's output queue. The second fragment does not experience interference with more incoming packets, and the resulting delay is consistently lower.

From the results shown with these datagram sizes, two effects seem to be happening. First, the re-fragmentation of a datagram does affect its delay, and re-fragmented datagrams will consistently have a delay of at least 0.9 milliseconds on the second fragment. Secondly, the arrival of other packets at the home agent affects the delay, and packets suffering interference of this sort experience delays of at least 1.7 milliseconds on their second fragment.

Table 5.12 shows common delays with 2968 bytes of information in the datagram. With this much data, three packets are produced at the correspondent node. The first two packets are maximally-sized Ethernet frames which must be re-fragmented

Table 5.11: Delays at the Home Agent (2958 Byte Datagrams)

Timestamps		
Input Queue	Output Queue	Delay
881146210.089474	881146210.091247	0.001773
881146210.091170	881146210.092134	0.000964
881146210.589463	881146210.591239	0.001776
881146210.591161	881146210.592261	0.001100
881146211.089531	881146211.091291	0.001760
881146211.091215	881146211.092191	0.000976
881146211.589615	881146211.591376	0.001761
881146211.591300	881146211.592261	0.000961

```
10.089474 ip 1514: donald > bobby: icmp: echo reply (frag 947:1480@0+)
10.089646 ip 1514: morty > darlene: ip-proto-4 1480 (frag 28072:1480@0+)
10.091170 ip 1512: donald > bobby: (frag 947:1478@1480)
10.091247 ip 54: morty > darlene: (frag 28072:20@1480)
10.091415 ip 1514: morty > darlene: ip-proto-4 1480 (frag 28073:1480@0+)
10.092134 ip 52: morty > darlene: (frag 28073:18@1480)
```

Figure 5.10: Trace of Ping Traffic on the Home Network (2958 Byte Datagram)

at the home agent. The third packet contains only 8 bytes of data. The delays shown in Table 5.12 show that the behaviour is similar to that of 2958 bytes for the first few packets and then the third packet is forwarded with minimal delay. A slightly less common case is also shown here where the delay on the third packet is about 1.0 milliseconds. The types of delay experienced are heavily determined by whether or not the outgoing packets from the home agent interfere with incoming packets from the correspondent node.

The two common cases experienced with these datagrams are shown in Figure 5.11. In the first half of the trace, the first packet from donald is re-fragmented, but the second packet arrives before the second fragment can leave the home agent. After this delay, the rest of the fragments are forwarded as was common with 2958 bytes. The final packet from donald is received and forwarded after the other fragments are gone, and little delay is experienced. In the second case, both the second

Table 5.12: Delays at the Home Agent (2968 Byte Datagrams)

Timestamps		
Input Queue	Output Queue	Delay
880717407.212375	880717407.214096	0.001721
880717407.214034	880717407.214964	0.000930
880717407.216484	880717407.216594	0.000110
880717407.712471	880717407.714263	0.001792
880717407.714134	880717407.715129	0.000995
880717407.714210	880717407.715246	0.001036

```
07.212375 ip 1514: donald > bobby: icmp: echo reply (frag 27998:1480@0+)
07.212542 ip 1514: morty > darlene: ip-proto-4 1480 (frag 28175:1480@0+)
07.214034 ip 1514: donald > bobby: (frag 27998:1480@1480+)
07.214096 ip 54: morty > darlene: (frag 28175:20@1480)
07.214254 ip 1514: morty > darlene: ip-proto-4 1480 (frag 28176:1480@0+)
07.214964 ip 54: morty > darlene: (frag 28176:20@1480)
07.216484 ip 60: donald > bobby: (frag 27998:8@2960)
07.216594 ip 62: morty > darlene: ip-proto-4 28
07.712471 ip 1514: donald > bobby: icmp: echo reply (frag 28000:1480@0+)
07.712642 ip 1514: morty > darlene: ip-proto-4 1480 (frag 28179:1480@0+)
07.714134 ip 1514: donald > bobby: (frag 28000:1480@1480+)
07.714210 ip 60: donald > bobby: (frag 28000:8@2960)
07.714263 ip 54: morty > darlene: (frag 28179:20@1480)
07.714418 ip 1514: morty > darlene: ip-proto-4 1480 (frag 28180:1480@0+)
07.715129 ip 54: morty > darlene: (frag 28180:20@1480)
07.715246 ip 62: morty > darlene: ip-proto-4 28
```

Figure 5.11: Trace of Ping Traffic on the Home Network (2968 Byte Datagram)

and third packets arrive from the correspondent node before the second fragment is forwarded by the home agent. In this case, a considerable delay of 1.0 milliseconds is experienced by the final packet, since it arrives before most of the other data is forwarded, and it must queue up behind the other data packets. The delays experienced here appear to be mostly queueing delays caused by contention for the media bandwidth.

With larger packet sizes, the same effects are noticeable, but there is more traffic at the home agent, and therefore the potential for even longer delays per packet.

Table 5.13: Delays at the Home Agent (5828 Byte Datagrams)

Timestamps		
Input Queue	Output Queue	Delay
879119982.673755	879119982.675471	0.001716
879119982.675405	879119982.676357	0.000952
879119982.677478	879119982.679951	0.002473
879119982.678730	879119982.680034	0.001304

Table 5.13 shows a typical trace of delays for datagrams of size 5828 bytes, which are broken up into 4 packets to be forwarded through the home agent. Because of home agent re-fragmenting, these 4 packets translate into 7 fragments from the home agent to the foreign agent. The resulting interference and forwarding characteristics are shown in Figure 5.12.

Figure 5.12 shows a trace of 5828-byte datagrams. The packets being shown here are in groups of four, three of which need re-fragmentation. In what is a reasonably typical case, the first fragment is forwarded quickly, and then the next fragment arrives from the correspondent host, after which the next three fragments are forwarded by the home agent. Next, the final two fragments arrive, and then the final three fragments are forwarded from the home agent. Most of the packet delays experienced in this scenario are above 1.0 milliseconds, and some exceed 2.0 milliseconds.

When the packet size increases to 11748 bytes, the behaviour becomes very complex. The highest delay experienced with 11740 bytes is 2037.715 milliseconds, with the lowest being 0.117 milliseconds. Average delay per packet fragment is 85.834 milliseconds. These delays are quite large considering delays higher than 30 milliseconds are significant enough to observe with our test-bed. Delays experienced with 11748 bytes are very large and would be noticeable in a real-time application.

```
82.673755 ip 1514: donald > bobby: icmp: echo reply (frag 29208:1480@0+)
82.673907 ip 1514: morty > darlene: ip-proto-4 1480 (frag 60773:1480@0+)
82.675405 ip 1514: donald > bobby: (frag 29208:1480@1480+)
82.675471 ip 54: morty > darlene: (frag 60773:20@1480)
82.675635 ip 1514: morty > darlene: ip-proto-4 1480 (frag 60774:1480@0+)
82.676357 ip 54: morty > darlene: (frag 60774:20@1480)
82.677478 ip 1514: donald > bobby: (frag 29208:1480@2960+)
82.678730 ip 1422: donald > bobby: (frag 29208:1388@4440)
82.679255 ip 1514: morty > darlene: ip-proto-4 1480 (frag 60775:1480@0+)
82.679951 ip 54: morty > darlene: (frag 60775:20@1480)
82.680034 ip 1442: morty > darlene: ip-proto-4 1408
```

Figure 5.12: Trace of Ping Traffic on the Home Network (5828 Byte Datagrams)

Delay Characteristics for NFS

As with the ping traffic, the delays at the home agent depend on the traffic coming from the correspondent node. In particular, the size of the packets, and the number of packets in a burst have a large effect on the delays. Standard NFS traffic is sent in datagrams of 8192 bytes. These are the datagrams observed in the experiments. As shown in the previous section, the delays increase substantially as the datagram size increases. Therefore, a datagram size of 8192 bytes should produce significant delays.

Figure 5.13 shows the distribution of packet delays at the home agent for the NFS traffic. This graph looks almost identical to the graphs for 11748 bytes and 5828 bytes below the 2.0 millisecond delay mark. Above that mark, the behaviour is slightly different, with the delays concentrating around 2.5, 4.5, and 8.0 milliseconds for 5828-byte pings, NFS traffic, and 11748-byte pings respectively. This data shows a gradual progression to higher delays as the datagram size increases. The NFS data also shows that this delay affects NFS traffic the same as it affects the ping traffic.

A tcpdump trace of NFS traffic on the home network is shown in Figure 5.14. This trace shows that the traffic from the correspondent host tends to be bursty (a result of the 8192 byte datagrams). The bursts happen in response to requests from the mobile host for more data. This is similar to the case with the ping traffic in

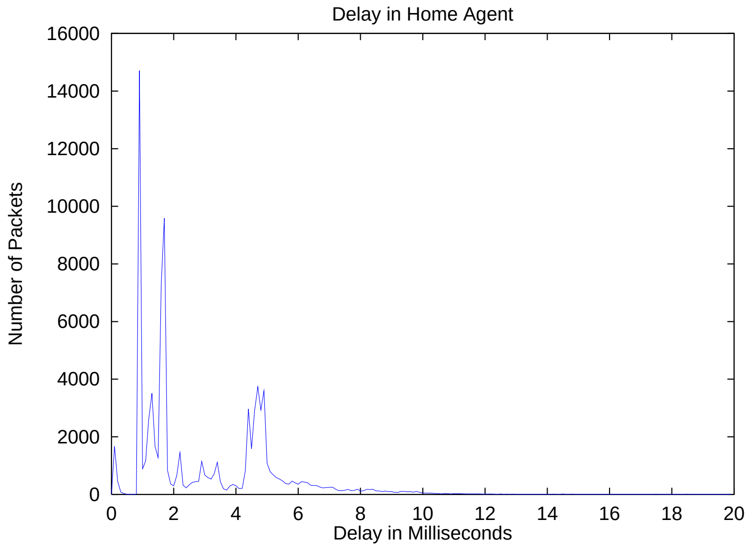


Figure 5.13: NFS Traffic Delays up to 20 Milliseconds

which an echo request results in a reply from the correspondent node.

The most concentrated bursts are in the third through the sixth packets. The inability of the home agent to process the packets fast enough to forward them before the next one arrives from the correspondent node results in larger delays for the last four packets than there is for the first two packets. It is noteworthy though that when the home agent resumes forwarding of the packets, its queue empties quickly.

When the home agent is keeping up with the traffic, all queued data is forwarded before the next request arrives from the mobile host. If the home agent is unable to complete forwarding of an NFS datagram before the next request arrives from the mobile host, then significant interference is caused by excess traffic from the correspondent node, and the queueing delays climb significantly.

The largest delay experienced at the home agent was 645.385 milliseconds. This value is quite large and indicates abnormal activity. During the time in which this value was obtained, the traffic flow from the correspondent node was especially heavy. The packets were queued up long enough to result in the mobile node issuing another request to the correspondent node before the first one was serviced. Eventually in the trace, the home agent was able to catch up, but large delays were recorded for the packets being forwarded. The most common delays were focused

```
17.181241 ip 154: bobby > donald: 112 null
17.185474 ip 1514: donald > bobby: reply ok 1472 null (frag 39747:1480@0+)
17.185624 ip 1514: morty > darlene: ip-proto-4 1480 (frag 24527:1480@0+)
17.187112 ip 1514: donald > bobby: (frag 39747:1480@1480+)
17.187175 ip 54: morty > darlene: (frag 24527:20@1480)
17.187335 ip 1514: morty > darlene: ip-proto-4 1480 (frag 24528:1480@0+)
17.188055 ip 54: morty > darlene: (frag 24528:20@1480)
17.189210 ip 1514: donald > bobby: (frag 39747:1480@2960+)
17.190629 ip 1514: donald > bobby: (frag 39747:1480@4440+)
17.191913 ip 1514: donald > bobby: (frag 39747:1480@5920+)
17.192436 ip 934: donald > bobby: (frag 39747:900@7400)
17.195281 ip 1514: morty > darlene: ip-proto-4 1480 (frag 24529:1480@0+)
17.195985 ip 54: morty > darlene: (frag 24529:20@1480)
17.196724 ip 1514: morty > darlene: ip-proto-4 1480 (frag 24530:1480@0+)
17.197424 ip 54: morty > darlene: (frag 24530:20@1480)
17.198162 ip 1514: morty > darlene: ip-proto-4 1480 (frag 24531:1480@0+)
17.198860 ip 54: morty > darlene: (frag 24531:20@1480)
17.198921 ip 954: morty > darlene: ip-proto-4 920
17.217661 ip 154: bobby > donald: 112 null
```

Figure 5.14: Trace of NFS Packets Through the Home Agent

around 0.9, 1.6, 4.0, and 6.0 milliseconds. Because of the large delays suffered by a few packets, the average delay for NFS traffic is actually 8.79 milliseconds.

Delay Characteristics for FTP

The delay characteristics for FTP packets over the Mobile IP link are shown in Figure 5.15. The data shows the familiar pattern of small delays concentrated around 0.1 and 1.7 milliseconds, but the distribution has a heavy tail with many delays much larger than 2 milliseconds. There are several areas of the curve in which delays are concentrated more than others, but these areas are not as pronounced as in other traffic. The delays are distributed in an almost uniform fashion from 6.0 through 16.0 milliseconds and taper off after that. The largest delay experienced at the home agent is 1433.346 milliseconds, while the minimum delay is 0.106 milliseconds. The average delay experienced at the home agent is 11.444 milliseconds. This is much smaller than the average delay for 11748 byte datagrams, and even smaller than the average delay for 5828 byte packets, but exceeds the average delay for NFS packets.

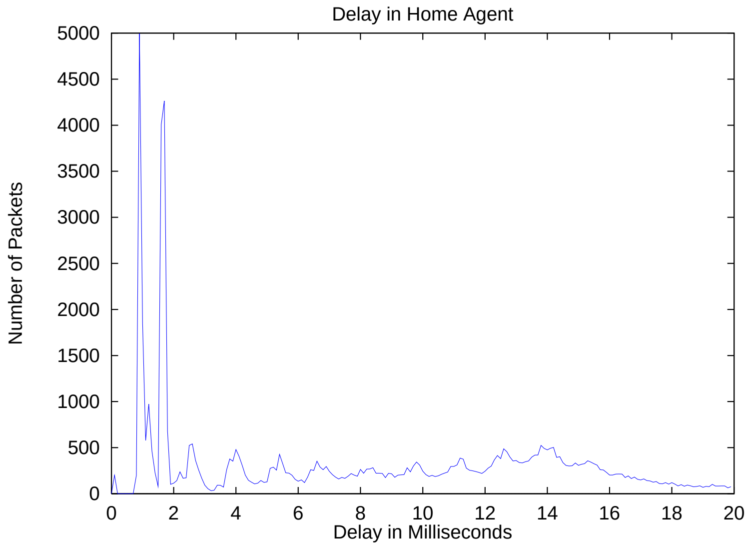


Figure 5.15: FTP Traffic Delays up to 20 Milliseconds

Similar to the NFS traffic, the FTP traffic suffers from congestion at the home agent. The largest delays occur when there are large traffic bursts from the correspondent node. Due to the congestion control mechanisms of FTP, congestion at the home agent is detected by the nodes at either end of the connection, and this affects traffic flow. When congestion is detected, the advertised window size from the mobile node decreases to 0 bytes and there is a pause in packet sending. The window size is quickly updated to 8192 bytes, and then rises to 17520 bytes again within 200 milliseconds, and the traffic continues to flow normally as before the break in packet forwarding. In the cases of extreme congestion observed, some retransmission from the correspondent node is observable.

Most of the traffic flows steadily through the home agent, with packets consistently being delayed no more than 16 milliseconds. The higher throughput with respect to the NFS traffic is most likely attributable to the windowing ability of TCP to deliver the packets to the mobile host, combined with the requirement of NFS for the mobile host to send specific requests for each piece of the NFS file being transferred. Through the use of the mechanisms provided in the TCP protocol, the FTP transfer is able to exploit the forwarding capabilities of the home agent fully and recognize cases of extreme congestion, but although the throughput is higher (as will be shown in later results), greater delays are suffered because of frequent congestion at the home agent.

5.4.2 Foreign Agent Delay

When the packets are forwarded through the Mobile IP agents they are delayed both at the home agent and at the foreign agent. The delay at the foreign agent is caused by the decapsulation process, the recombining of fragmented packets from the HA, and queueing delays. The delay at the foreign agent is computed as the difference between the time that the last fragment of a packet is delivered to the foreign agent and time that the restructured packet is put into the foreign agent's output queue.

The delays at the foreign agent are also affected by the behaviour at the home agent. If the home agent is forwarding packets very quickly, congestion can happen at the foreign agent. Conversely, if the home agent is experiencing congestion, the packets forwarded through the home agent arrive more slowly at the foreign agent, and congestion is reduced.

This section shows how packets are delayed at the foreign agent. Delays will be shown by presenting average delays as datagram size increases and the frequency distribution curves as datagram size increases. Actual traces of the packets being forwarded through the foreign agent are also given to illustrate how the packets are delayed.

Delay as Packet Size Increases

This section shows how the delay at the foreign agent was affected by changes in the packet size. Several traces were taken with packets sizes 64, 2938, 2958, 2968, 5828, and 11748 bytes. The results of these experiments are presented in Figure 5.16. These graphs show the distribution of delays for each packet size that was tested.

The distributions are shown by plotting the number of observations that occurred against the delays that were observed. The x-axis shows the delays in milliseconds.

Each graph consists of 100 intervals, and the graph shows the number of observations that were observed in each interval.

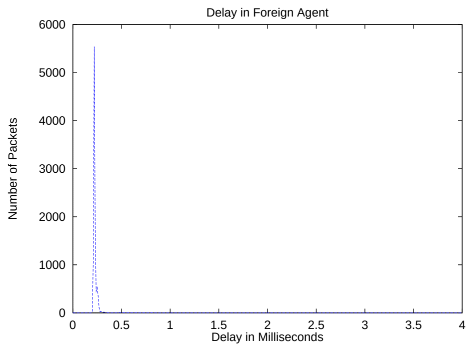
Figure 5.16 is composed of six graphs, (a) through (f). Each graph shows the delay experienced by a set of pings of a given size. The graphs are ordered from smallest datagram size to largest. This ordering shows how the complexity of the traffic and its delay both increase as the datagram size increases.

In Figure 5.16(a), the delays experienced with packets of size 64 bytes are shown. This graph shows the basic behaviour of Mobile IP. In this experiment one 64 byte fragment is sent out twice per second, and the packet is forwarded through Mobile IP. It is evident that the vast majority of the delays are between 0.2 and 0.4 milliseconds. About 12,000 observations are observable within 0.04 milliseconds of each other. This graph shows that the delays are reasonably low (around 0.2 milliseconds) and have little variability (most of the observations are within a 0.1 millisecond area). It is noteworthy that the delays experienced with these datagrams are at least twice the amount of delay experienced by the same packets at the home agent. Since the processing with packets this small does not include the processing of fragmented packets, much of the extra delay experienced may be attributed to the difference in speed between the home agent and foreign agent (the home agent is faster).

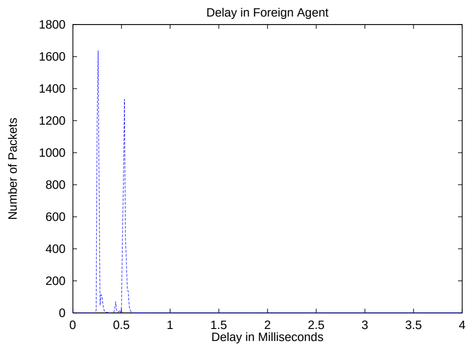
One other factor involved with the foreign agent but not the home agent is the traffic travelling from the mobile node. Since the foreign agent is the gateway for the mobile node, it must forward all traffic from the mobile node as well as to the mobile node. This places an additional burden on the foreign agent that is not present at the home agent, and congestion can happen at the foreign agent from traffic travelling in both directions simultaneously.

With larger packets, fragmentation starts to have an effect, as it did with the home agent. The graph in Figure 5.16(b) shows the delays at the foreign agent for datagrams of size 2938 bytes. In this graph two concentrations of packet delay show up prominently. These concentrated delays correspond to the processing times of fragments that are created at the home agent.

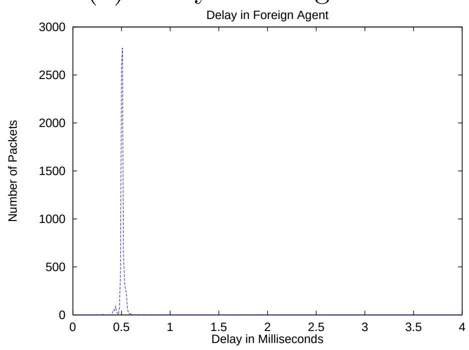
The first packet of the original datagram is large enough to get re-fragmented at



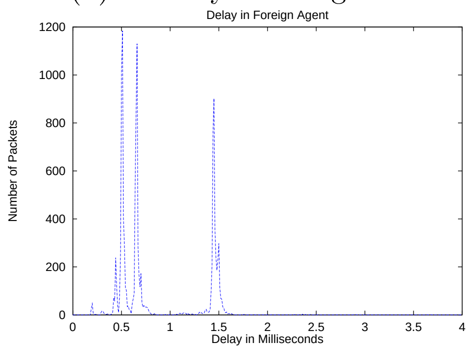
(a) 64-byte Datagrams



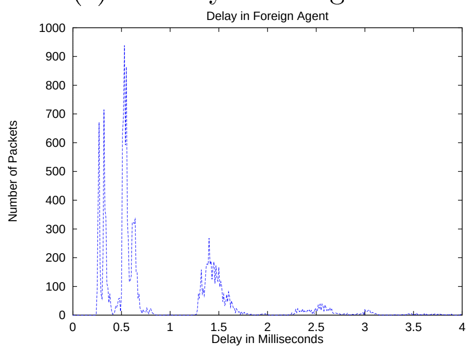
(b) 2938-byte Datagrams



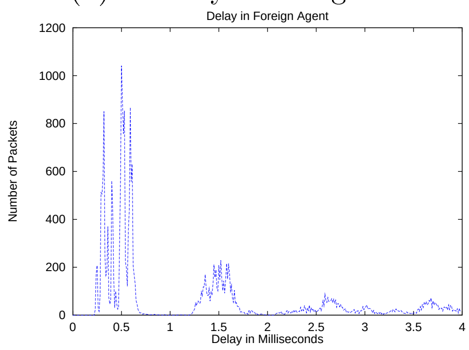
(c) 2958-byte Datagrams



(d) 2968-byte Datagrams



(e) 5828-byte Datagrams



(f) 11748-byte Datagrams

Figure 5.16: Packet Delays at the Foreign Agent as Datagram Size Increases

the home agent into fragments of sizes 1480 and 20 bytes. When this fragmented packet arrives at the foreign agent it is re-assembled and then forwarded to the mobile node. This takes time, and thus it shows up in the graph as a greater delay, one of approximately 0.5 milliseconds. The smaller fragment does not suffer the re-fragmentation, and therefore it is forwarded in the normal time demonstrated for the smaller packets above. This data seems to suggest that the additional processing required to re-combine fragments adds roughly 0.25 milliseconds of delay, effectively doubling the delay at the foreign agent.

Figure 5.16(b) also shows little variation in the delay experienced at the foreign agent. The frequency spikes in the figure are very pronounced, indicating that delays do not differ much between packets that are experiencing the same type of processing at the foreign agent. A slightly larger variation is shown for the processing of the fragmented packet as compared to the processing of the non-fragmented packet. As with the smaller packet sizes, there is some variation, and there are some larger delays, the largest being 2.056 milliseconds. These delays are few though, considering that only 0.6 % of the packets take longer than 1.0 milliseconds to process at the foreign agent.

With a ping size of 2958 bytes, both packets from the correspondent host are re-fragmented at the home agent. This behaviour is shown in Figure 5.16(c) by a single spike centred around 0.525 milliseconds. This shows that each packet fragment is suffering the same delay since it is undergoing the same processing. In addition, there is little variance in the delays experienced by the packets, as again, the spike in the distribution curve is very pronounced.

The results for 2958 bytes also confirm that approximately 0.5 milliseconds is required to forward fragmented packets through the foreign agent. Since both packets are fragmented, and there is only one delay spike, this indicates that the re-combining of the fragments produces the delay characteristic for these packets. The delays shown here also correspond to the delays shown for the fragmented packet with 2938 byte datagrams.

Figure 5.16(d) shows the delay at the foreign agent with a datagram size of 2968 bytes. This graph shows that the packet delays are concentrated around the three values 0.500, 0.700, and 1.450 milliseconds. With a ping size of 2968 bytes three packets are produced at the correspondent host, two of which are re-fragmented at the home agent. The performance graph clearly shows two spikes around the 0.525 millisecond range experienced by the fragments of the 2958 bytes packets. This delay is the same as that experienced by all of the re-fragmented packets for the smaller datagram sizes.

From this, one would expect that these values reflect the processing time for the fragments that were re-fragmented at the home agent. At least the lower delay seems to suggest that a similar delay is being experienced as was experienced with smaller datagram sizes for one of the packets making up the 2968 datagram. The slightly larger spike at 0.7 milliseconds may correspond to the other packet being re-fragmented, or may correspond to the smaller packet experiencing slight queueing delays. Traces of the packets show that this delay is for the second fragment packet, and the slightly larger delay is caused by the untimely arrival of the third smaller packet before the second packet can be forwarded. This causes the extra 0.2 milliseconds of delay for the second packet, and causes the delay on the third packet to increase to 1.5 milliseconds because of queueing behind the second packet. It is assumed that the delay for the third packet is mostly queueing delay, since it does not require fragment processing, and other non-fragmented packets for smaller datagrams are forwarding in less than 0.3 milliseconds.

With larger datagram sizes from the correspondent host, the delays experienced are larger and are more influenced by the arrival of other packets at the foreign agent. With a datagram of size 5828 bytes, three packets of 1480 bytes and one of size 1388 bytes are produced at the correspondent node. In this scenario, the first three packets get re-fragmented, but the last packet does not. The graph in Figure 5.16(e) shows delays concentrated around 0.275 and 0.550 milliseconds with concentration also apparent around 1.5, 2.5 and 3.0 milliseconds.

The packet delay again greatly depends on the arrival times of the packets at the foreign agent. The most common case has the third and fourth packets each taking around 1.5 milliseconds to forward, with the first and second packets taking much less. In this case, the fourth packet arrives during the processing of the third packet and is therefore delayed until the third packet is processed.

If fragments do not arrive at the foreign agent during the processing of the previous fragment, the delay is much less. A typical case has delays of 0.536, 0.696, 0.764, and 0.276 milliseconds for the four packets respectively. The larger delays around 3.0 milliseconds are produced when the arrival of the second, third, and fourth packets all interfere with the processing of the previous packet.

It is noteworthy that the delay spikes in Figure 5.16(e) are not as pronounced as they are in plots for smaller datagrams. This indicates a larger variation experienced by the packets for 5828 byte datagrams. The delay spikes below 1.0 milliseconds still demonstrate little variability when compared to the spikes beyond 1.0 milliseconds, but there is a lot more variability than that experienced around similar delays with smaller datagrams. The delays above 1.0 milliseconds are much less concentrated around certain values than the distributions for smaller datagrams. The distribution also shows that there are many more higher delays present than with smaller datagrams.

Figure 5.16(f) shows the delays for packets in a datagram containing 11748 bytes. The figure shows the typically large spikes around 0.500 milliseconds. These represent the best-case forwarding of fragmented packets that have to be re-combined. Although packets experiencing the best-case delays still represent the majority of the packets, the number of packets experiencing delays beyond 1.0 milliseconds is very large.

The spikes around 0.5 milliseconds correspond to the forwarding times of the first few re-fragmented packets that experience minimal interaction with other packets arriving from the home agent. Later packets interfere with each other, and eventually the delay climbs to values commonly reaching 5.0 milliseconds or more.

Smaller, less defined spikes are also present at 1.5, 2.6, and 2.7 milliseconds. These spikes are very poorly defined, and illustrate a large amount of variability with the higher delays being experienced. Since datagrams of size 11748 are made up of 8 packets, there are large packet bursts whenever one of these datagrams is forwarded through Mobile IP. These 8 packets translate into 15 fragments arriving at the foreign agent for processing. With this large number of fragments arriving in a burst, the chance of fragment arrivals interrupting previous fragment processing is much greater than with smaller datagrams. Also, the combinations of packet arrivals are very diverse, and can account for the variability in the delays shown in the distribution plot.

As with datagrams of size 5828 bytes, the delay distribution has a tail. The tail for 11748 bytes is much more pronounced than that for 5828 bytes, though. In fact, large delays exceeding 10 milliseconds are commonly experienced by packets for the 11748-byte datagrams, but are not common for the 5828-byte datagram packets.

Packet Delay Traces at the Foreign Agent

The delays at the foreign agent are shown in this section through traces of Ethernet frames, and the delays that are experienced by individual packets being traced. These traces demonstrate the queueing delays that are affecting packets at the foreign agent.

Table 5.14 shows some delays experienced by packets at the foreign agent. Most of the packets experience a delay of 0.2 milliseconds. However, as shown in the table, sometimes much more significant delay (e.g., 7.1 milliseconds) is experienced. The distribution in Figure 5.16(a) demonstrates that the occurrences of such large delays are rare with the small datagram traffic.

Figure 5.17 shows a tcpdump trace of some of these packets. There are two sets of packets being shown in this trace. First, the packets from the mobile node (bobby) to the correspondent node (donald) are shown. This traffic demonstrates the foreign agent acting as the default router for the mobile node. The packet leaving the mobile node and the one leaving the foreign agent are identical in this trace (see the first two packets). This is because the names shown in the trace are determined by the IP address in the packet, which does not change; only the Ethernet address changes with the forwarding.

The trace shows that packets being forwarded without Mobile IP from the mobile node to the correspondent node are actually delayed longer in the foreign agent than the packets going in the other direction. The reason for this is not clear, and it is counterintuitive because there is less processing to be done for packets travelling from the mobile node to the correspondent node. It is possible that the table look-ups and processing for normal forwarding of IP packets takes longer than the internal processing for Mobile IP forwarding. This may be the case since a foreign agent merely has to look up the mobile node in its visitor list and dump the packet onto

Table 5.14: Delays at the Foreign Agent (64 Byte Datagrams)

Timestamps		
Input Queue	Output Queue	Delay (secs)
879091899.038600	879091899.038828	0.000228
879091899.139424	879091899.146512	0.007088
879091899.238624	879091899.238852	0.000228
879091899.340336	879091899.340560	0.000224
879091899.438788	879091899.439036	0.000248
879091899.541004	879091899.541248	0.000244
879091899.638680	879091899.638912	0.000232
879091899.740300	879091899.740536	0.000236

```
99.036332 ip 98: bobby > donald: icmp: echo request
99.036728 ip 98: bobby > donald: icmp: echo request
99.038600 ip 118: morty > darlene: ip-proto-4 84
99.038828 ip 98: donald > bobby: icmp: echo reply
99.135636 ip 74: morty > darlene: ip-proto-4 40
99.135864 ip 54: skorpio3 > bobby: . ack 468045 win 33580
99.137148 ip 98: bobby > donald: icmp: echo request
99.137484 ip 98: bobby > donald: icmp: echo request
99.138468 ip 189: bobby > skorpio3: ack 1 win 17520
99.138856 ip 189: bobby > skorpio3: ack 1 win 17520
99.139424 ip 118: morty > darlene: ip-proto-4 84
99.146512 ip 98: donald > bobby: icmp: echo reply
99.236328 ip 98: bobby > donald: icmp: echo request
99.236720 ip 98: bobby > donald: icmp: echo request
99.238624 ip 118: morty > darlene: ip-proto-4 84
99.238852 ip 98: donald > bobby: icmp: echo reply
```

Figure 5.17: Trace of Ping Traffic at the Foreign Agent (64 Byte Datagrams)

the network, whereas when the packet is going the other way, the next hop router has to found as well. Regardless of the reason, a delay of 0.4 millisecond seems to be common for the traffic from the mobile node to be forwarded through the foreign agent.

The traffic from donald to bobby that is forwarded through Mobile IP is also shown in this trace. The packets labelled donald→darlene are ones forwarded

Table 5.15: Delays at the Foreign Agent (2938 Byte Datagrams)

Timestamps		
Input Queue	Output Queue	Delay (secs)
879095738.156568	879095738.157100	0.000532
879095738.158708	879095738.158980	0.000272
879095738.663404	879095738.663940	0.000536
879095738.665552	879095738.665816	0.000264
879095739.165436	879095739.165996	0.000560
879095739.167612	879095739.167880	0.000268
879095739.665820	879095739.666376	0.000556
879095739.668016	879095739.668276	0.000260

through Mobile IP from the home agent to the foreign agent. The time to forward the packet through the foreign agent is calculated as the difference between the time that the HA→FA packet is detected and the FA→MN packet is detected. In the trace given, this time is around 0.2 milliseconds for most packets. The trace also shows an example of a packet that requires more than 0.2 milliseconds being forwarded. This packet is forwarded while there is other traffic being sent to and from the mobile node. Although it is the only apparent difference in the traffic trace between when the packets are forwarded in 0.2 milliseconds and when they are forwarded in 7.0 milliseconds, it is unclear how this extra traffic would affect delay, since there are no direct queueing effects noticeable in the trace (i.e., the extra traffic does not appear between the packet arriving at the foreign agent and it being forwarded to the mobile node).

A trace of delays for datagrams of size 2938 bytes is given in Table 5.15. This table shows that the delays on the first and second packet of each datagram are common across all the datagrams. In almost every case, the first packet is delayed approximately 0.55 milliseconds, and the second packet is delayed around 0.25 milliseconds. This occurs since the first packet experiences re-fragmentation because of its size (1480 bytes), but the second packet does not, because its size is only 1458 bytes.

The forwarding of 2958 byte datagrams through the foreign agent is characterized

Table 5.16: Delays at the Foreign Agent (2958 Byte Datagrams)

Timestamps		
Input Queue	Output Queue	Delay (secs)
881121961.408072	881121961.408576	0.000504
881121961.410332	881121961.410836	0.000504
881121961.908204	881121961.908652	0.000448
881121961.910420	881121961.910976	0.000556
881121962.407892	881121962.408312	0.000420
881121962.410108	881121962.410628	0.000520
881121962.908216	881121962.908648	0.000432
881121962.910416	881121962.910940	0.000524

```
61.908024 ip 1514: morty > darlene: ip-proto-4 1480 (frag 16859:1480@0+)
61.908204 ip 60: morty > darlene: (frag 16859:20@1480)
61.908652 ip 1514: donald > bobby: icmp: echo reply (frag 62018:1480@0+)
61.910268 ip 1514: morty > darlene: ip-proto-4 1480 (frag 16860:1480@0+)
61.910420 ip 60: morty > darlene: (frag 16860:18@1480)
61.910976 ip 1512: donald > bobby: (frag 62018:1478@1480)
```

Figure 5.18: Trace of Ping Traffic at the Foreign Agent (2958 Byte Datagrams)

by a common delay for almost all of the packets. All of the packets forwarded for 2858 byte datagrams require the recombination of fragments at the foreign agent, and they are of a similar size. Because of this, the delays shown in Table 5.16 are consistently between 0.4 milliseconds and 0.55 for most packets. This delay is similar to the delay of 0.5 milliseconds that was experienced at the home agent for this same traffic.

A typical trace of 2958 byte packets is given in Figure 5.18. This trace shows two packets arriving from the home agent and being forwarded through the foreign agent. For each packet, the two fragments arrive at the foreign agent, and then a single packet is forwarded approximately 0.4 milliseconds later. The two fragments arriving at the foreign agent are combined to form the packets leaving the foreign agent. This data shows that this extra processing effectively doubles the delay for the Mobile IP packets.

Table 5.17: Delays at the Foreign Agent (2968 Byte Datagrams)

Timestamps		
Input Queue	Output Queue	Delay (secs)
880728584.736176	880728584.736668	0.000492
880728584.740912	880728584.741528	0.000616
880728584.741052	880728584.742464	0.001412
880728585.733280	880728585.733792	0.000512
880728586.723384	880728586.723704	0.000320
880728586.729644	880728586.729840	0.000196

With 2968 byte datagrams, the delays for a datagram consisting of two large packets and one small packet are shown. Table 5.17 shows typical delays for packets making up datagrams of this size. The first case shows considerable delay for all of the packets in the datagram. This becomes more apparent when these values are compared to the second case. The first packet in the first datagram is delayed 0.492 milliseconds, compared to 0.512 in the second datagram, and so the first packet is actually delivered faster in the first case. The delay on the second packet changes from 0.616 to 0.323, a substantial decrease in delay. The difference in the third packet is even more, with delay changing from 1.414 to 0.196 milliseconds. This data shows that in some cases the first two packets are delayed the usual amount because of fragmentation, and the third is forwarded very quickly. In other cases, though, the second and third packets are delayed substantial amounts.

With 5828 bytes, there are four packets to be forwarded, resulting in eight fragments reaching the foreign agent. Typical delays for these packets are shown in Table 5.18. In these traces, it is apparent that the first and second packets are delayed only small amounts because of the processing needed to de-fragment them. The third and fourth fragments of the datagram suffer much larger delays of around 1.5 milliseconds.

The largest ping size tested is 11748 bytes. This size is large enough to see very complex behaviour in the delays at both the home and foreign agents. With datagrams of this size, the first 7 packets are re-fragmented, and therefore suffer

Table 5.18: Delays at the Foreign Agent (5828 Byte Datagrams)

Timestamps		
Input Queue	Output Queue	Delay (secs)
879098109.446768	879098109.447096	0.000328
879098109.450008	879098109.450572	0.000564
879098109.452512	879098109.453936	0.001424
879098109.453556	879098109.455084	0.001528

Table 5.19: Delays at the Foreign Agent (11748 Byte Datagrams)

Timestamps		
Input Queue	Output Queue	Delay (secs)
880928904.862964	880928904.863488	0.000524
880928904.865636	880928904.866168	0.000532
880928904.868196	880928904.868700	0.000504
880928904.877168	880928904.877672	0.000504
880928904.879436	880928904.880932	0.001496
880928904.880308	880928904.884072	0.003764
880928904.882756	880928904.885384	0.002628
880928904.883772	880928904.886496	0.002724

from the usual delays. A typical trace of these packets is given in Table 5.19. It is interesting that the first four packets are all forwarded quickly, and it is not until the fifth packet that significant delays occur. A partial explanation for this is that the forwarding through the foreign agent depends on the forwarding through the home agent. Therefore, if there is congestion at the home agent, the packets should arrive more slowly at the foreign agent, and therefore it should be better able to cope with the traffic load. Once some queueing delays start at the foreign agent, though, they usually affect all packets until the datagram is done.

Figure 5.19 shows a trace of a 11748 datagram through the foreign agent. In this trace, the first two Ethernet frames contain the first two fragments of the forwarded packets. These fragments are re-combined and forwarded as the next packet. This pattern continues for the next three packets, and therefore the forwarding rates are all similar. In each case, there appears to be at least 2.0 milliseconds between the

```
04.862788 ip 1514: morty > darlene: ip-proto-4 1480 (frag 60127:1480@0+)
04.862964 ip 60: morty > darlene: (frag 60127:20@1480)
04.863488 ip 1514: donald > bobby: icmp: echo reply (frag 52533:1480@0+)
04.865480 ip 1514: morty > darlene: ip-proto-4 1480 (frag 60128:1480@0+)
04.865636 ip 60: morty > darlene: (frag 60128:20@1480)
04.866168 ip 1514: donald > bobby: (frag 52533:1480@1480+)
04.868032 ip 1514: morty > darlene: ip-proto-4 1480 (frag 60129:1480@0+)
04.868196 ip 60: morty > darlene: (frag 60129:20@1480)
04.868700 ip 1514: donald > bobby: (frag 52533:1480@2960+)
04.876976 ip 1514: morty > darlene: ip-proto-4 1480 (frag 60130:1480@0+)
04.877168 ip 60: morty > darlene: (frag 60130:20@1480)
04.877672 ip 1514: donald > bobby: (frag 52533:1480@4440+)
04.879280 ip 1514: morty > darlene: ip-proto-4 1480 (frag 60131:1480@0+)
04.879436 ip 60: morty > darlene: (frag 60131:20@1480)
04.880176 ip 1514: morty > darlene: ip-proto-4 1480 (frag 60132:1480@0+)
04.880308 ip 60: morty > darlene: (frag 60132:20@1480)
04.880932 ip 1514: donald > bobby: (frag 52533:1480@5920+)
04.882592 ip 1514: morty > darlene: ip-proto-4 1480 (frag 60133:1480@0+)
04.882756 ip 60: morty > darlene: (frag 60133:20@1480)
04.883772 ip 1442: morty > darlene: ip-proto-4 1408
04.884072 ip 1514: donald > bobby: (frag 52533:1480@7400+)
04.885384 ip 1514: donald > bobby: (frag 52533:1480@8880+)
04.886496 ip 1422: donald > bobby: (frag 52533:1388@10360)
```

Figure 5.19: Trace of Ping Traffic at the Foreign Agent (11748 Byte Datagrams)

arrival of the last fragment for one packet and the arrival of the first fragment of the next. This seems to be enough time for the foreign agent to forward its packets with no queueing effects.

With later packets, there is a smaller interval between packet arrivals. The trace shows that the fifth packet arrives less than 0.8 milliseconds after the last fragment of the fourth packet. This causes interference, as the foreign agent spends time processing the new arrivals and does not forward the fourth packet until the fifth's arrival has been processed. The queueing seems to affect later packets as well, since the fifth and sixth packet both get delayed until the third packet has arrived, and then all three are forwarded in a burst.

Delay Characteristics for NFS

The NFS traffic delays suffered at the foreign agent are presented graphically in Figure 5.20. It is apparent from the graph that the largest concentration of delay is around 0.5 milliseconds. There are also quite large spikes in the distribution between 1.0 and 4.0 milliseconds, with the concentrations of packets diminishing quickly after 4.0 milliseconds. This traffic demonstrates much variability, although the data suggests that many packets experience similar delay characteristics, causing the 0.5 millisecond spike.

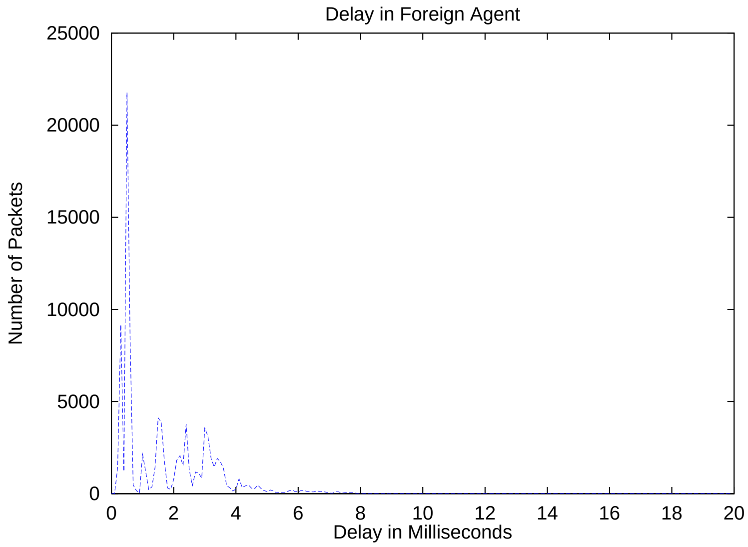


Figure 5.20: NFS Traffic Delays up to 20.0 Milliseconds

Table 5.20 shows a typical set of delays experienced at the foreign agent. In this table, the first few packets are delayed a minimal amount because the output queues are clear and this creates the large spike at 0.5 milliseconds. However, as the queue fills up, the delays increase. By the third packet the delay has reached 1.5 milliseconds, and by the fourth it is at 3.108 milliseconds. The delay does not rise much above 3.0 milliseconds in most cases.

The actual packet trace corresponding to these delays is shown in Figure 5.21. In the trace, the ordering of the packets from the home agent and foreign agent can be seen. The initial few packets in a burst are quickly processed and forwarded, but later packets get delayed, and eventually the foreign agent reaches a state where it

Table 5.20: Large Delays at the Foreign Agent (NFS Traffic)

Timestamps		
Input Queue	Output Queue	Delay (secs)
879026526.440984	879026526.441308	0.000324
879026526.444140	879026526.444760	0.000620
879026526.446684	879026526.448184	0.001500
879026526.448044	879026526.451152	0.003108
879026526.450048	879026526.452500	0.002452
879026526.450576	879026526.453684	0.003108

```
26.438012  ip 1514: morty > darlene: ip-proto-4 1480 (frag 49464:1480@0+)
26.440984  ip 60: morty > darlene: (frag 49464:20@1480)
26.441308  ip 1514: donald > bobby: reply ok 1472 null (frag 32629:1480@0+)
26.443968  ip 1514: morty > darlene: ip-proto-4 1480 (frag 49465:1480@0+)
26.444140  ip 60: morty > darlene: (frag 49465:20@1480)
26.444760  ip 1514: donald > bobby: (frag 32629:1480@1480+)
26.446512  ip 1514: morty > darlene: ip-proto-4 1480 (frag 49466:1480@0+)
26.446684  ip 60: morty > darlene: (frag 49466:20@1480)
26.447868  ip 1514: morty > darlene: ip-proto-4 1480 (frag 49467:1480@0+)
26.448044  ip 60: morty > darlene: (frag 49467:20@1480)
26.448184  ip 1514: donald > bobby: (frag 32629:1480@2960+)
26.449876  ip 1514: morty > darlene: ip-proto-4 1480 (frag 49468:1480@0+)
26.450048  ip 60: morty > darlene: (frag 49468:20@1480)
26.450576  ip 954: morty > darlene: ip-proto-4 920
26.451152  ip 1514: donald > bobby: (frag 32629:1480@4440+)
26.452500  ip 1514: donald > bobby: (frag 32629:1480@5920+)
26.453684  ip 934: donald > bobby: (frag 32629:900@7400)
```

Figure 5.21: Trace of NFS Traffic at the Foreign Agent

no longer can forward any packets until the burst from the home agent is over.

Table 5.21 shows a series of delays for a NFS packet experiencing minimal delays in forwarding. The key with this packet trace is that the packets from the home agent are delivered with large gaps between them. These gaps seem to give the foreign agent enough time to finish processing one packet before it has to work on the next one. The trace in Table 5.21 shows the timestamps on the arrival of the packets from the home agent. The important arrival in terms of the Mobile IP processing is that

Table 5.21: Small Delays at the Foreign Agent (NFS Traffic)

Timestamps		
Input Queue	Output Queue	Delay
879026526.923384	879026526.923908	0.000524
879026526.926180	879026526.926716	0.000536
879026526.928780	879026526.929452	0.000672
879026526.931520	879026526.932156	0.000636
879026526.934072	879026526.934716	0.000644
879026526.936124	879026526.936384	0.000260

of the last fragment needed to form the Mobile IP packet. In these arrival times, there are differences of 2.796, 2.600, 2.740, 2.552, and 2.052 milliseconds. When compared with differences of 2.544, 1.360, 2.004, 0.528 milliseconds for the packets in the previous trace which were experiencing queueing delays, it seems that this time lag may be a factor, but that it does not seem to be the sole factor influencing the increase in delay. The time lags in the second trace, however, seem to be enough time to keep the output queue clear, so no delay is added in queueing.

In general, the delays with the NFS packets are similar in nature to those experienced by the ping traffic with larger packet sizes. If we examine the packet delays for the packets of size 11748 bytes there is a similarity in the distribution of packet delay. The main difference is that some of the packets with 11740 bytes were delayed slightly longer than the NFS packets, leading to a spike in the curve at 6.0 milliseconds.

Delay Characteristics for FTP

The delay characteristics for FTP at the foreign agent are shown in Figure 5.22. The distribution of the delays shows the expected concentration of delays below 1.0 milliseconds, but it also shows a heavy tail to the distribution that is more pronounced that either the one shown in Figure 5.20 for NFS traffic, or the one shown in Figure 5.16(f) for ping traffic with a packet size of 11748 bytes. The two most significant concentrations of delay lie around 1.7 and 0.9 milliseconds. Beyond

2.0 milliseconds, the delays are mostly uniformly distributed from 2.0 through 11.0. After that, the distribution drops off rapidly with only the occasional delay being larger.

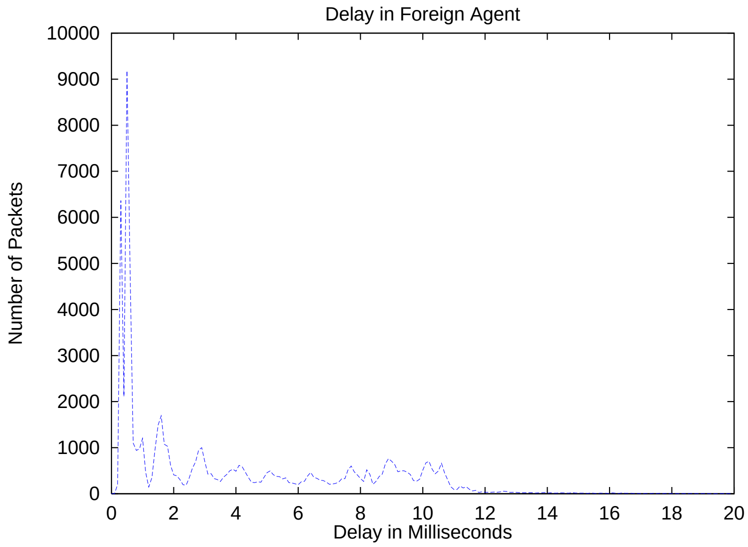


Figure 5.22: FTP Traffic Delays up to 20.0 Milliseconds

The largest delay experienced at the foreign agent is 22.784 milliseconds, and the smallest delay is 0.2 milliseconds. The average delay experienced by FTP traffic through the foreign agent is 4.031 milliseconds. These values represent relatively small delays when compared to those exhibited by the traffic being forwarded through the home agent. In general, because of the longer delays suffered at the home agent, the packet arrivals to the foreign agent are steady, yet give the foreign agent enough time for processing so that the queueing delays are not large. The delay trace in Figure 5.22 is typical of the delays experienced at the foreign agent.

In this trace, rapid processing of packets is followed by packets experiencing some delay. From the trace it appears that the output queue needs about 10 milliseconds to clear itself once congestion has occurred. In this trace, the delay jumps sharply when the packets arrive in a large burst from the home agent. During the burst from 0.189336 to 0.195868, five packets arrive in 6.55 milliseconds. The agent is kept busy handling the burst of incoming traffic, and no fragments are forwarded from the foreign agent until after the burst. Therefore, the packets that are forwarded

Table 5.22: Delays at the Foreign Agent (FTP Traffic)

Timestamps		
Input Queue	Output Queue	Delay (secs)
878857381.178540	878857381.178876	0.000336
878857381.181028	878857381.181656	0.000628
878857381.183652	878857381.184452	0.000800
878857381.186740	878857381.187392	0.000652
878857381.189336	878857381.196084	0.006748
878857381.190556	878857381.197068	0.006512
878857381.192028	878857381.199548	0.007520
878857381.193228	878857381.200432	0.007204
878857381.194556	878857381.201780	0.007224
878857381.195868	878857381.203028	0.007160
878857381.206340	878857381.207000	0.000660

after the burst are delayed about 6.55 milliseconds.

5.5 Network Load

The overhead on the home network can be measured in three ways: first, how many extra bytes need to be transmitted on the network to transmit files via Mobile IP compared to not using Mobile IP; second, how many more frames need to be transmitted on the home network to deliver the same amount of data; and third, how much does the network utilization increase when Mobile IP is used to deliver the packets instead of normal IP.

The number of bytes and packets transmitted is empirically derived from the size of the original packets involved and the MTU of the network. The utilization of the network bandwidth depends on how well the protocols react to increased traffic due to Mobile IP, and whether or not some performance is sacrificed in order to keep Mobile IP traffic from dominating the bandwidth. Profiles of NFS and FTP transfers are presented in graphical form to show how the network bandwidth is used both with and without Mobile IP. In addition, load values are shown in tabular form to show how the Mobile IP traffic affects the network it is on. Network overhead will be discussed with respect to NFS and FTP. The experiments focus on these two protocols, because they have the most complex behaviour of those tested here.

5.5.1 Mathematical Analysis of Home Network Traffic

A node in a network can be in one of three positions: in the interior, alone on the edge, and with others on the edge. An interior node has no connections to other networks and does not act as a router to other networks. A node alone on the edge of a network has an interface to at least one other network, and is the only router on the first network. A node with others on the edge of a network acts as a router to other networks, but is not the only router on the network.

If the home agent is placed on the edge of the network (as a router to another network), and if all packets destined for the foreign agent use the home agent as their router, then there is no extra traffic generated on the home network, as the Mobile IP traffic does not get forwarded onto this network. If the home agent is on the edge of the network and another router is used to send packets to the foreign agent, then all of the mobile IP traffic will be forwarded on the home network, but the original traffic from the correspondent node may or may not traverse the home network. If the home agent is an interior node, then all traffic from the correspondent node will traverse the home network, and all Mobile IP traffic to the foreign agent will traverse the home network. This last case is the worst-case scenario. This case was studied in the test-bed, and it is the basis for the analysis presented here.

The following equations present a mathematical analysis of Mobile IP impact on the home network (the analysis done here would apply equally as well to the foreign network). The impact of Mobile IP on the home network is determined to be the number of extra bytes and frames that need to be transmitted on the home network when Mobile IP is used. The calculation of the extra traffic is straightforward based on any re-fragmentation that must be done. The analysis presented here is carried out with respect to a specifc datagram size (5828 bytes), and depends on the position of the home agent in the network.

$$
\begin{aligned}
DataSize &= 5828\ bytes \\
NumWholeFrags &= floor(DataSize/1480) \\
&= 3 \\
Remainder &= DataSize - NumFrags * 1480 \\
&= 5828 - 3 * 1480 \\
&= 1388\ bytes \\
MobileIPTraffic &= NumWholeFrags * (1514 + 54) + Remainder + 34 \\
&= 3 * (1568) + 1388 + 34 \\
&= 6126\ bytes
\end{aligned}
$$

This calculation is based on an MTU of 1500 bytes (not counting frame headers). Since the IP header takes 20 bytes, this leaves 1480 bytes of data. Therefore, the number of maximally sized ethernet frames produced at the correspondent host is the datagram size divided by 1480 bytes. The remainder is placed into its own packet. Based on this, the number and size of the packets sent from the correspondent node can be calculated.

Some of the Mobile IP traffic comes from extra headers that are added to the packets at the home agent because of IP-in-IP encapsulation. Each of these headers is 20 bytes. If the original packet size plus these 20 bytes is larger than 1480 bytes, then the packet must be re-fragmented at the home agent. Re-fragmentation results in another 20 bytes of traffic because two fragments are now encapsulated instead of one. With the Ethernet header, the extra header information imposed by Mobile IP equals 54 bytes per re-fragmented packet.

The computation above shows the Mobile IP traffic generated for a 5828-byte datagram on the home network. It should be noted that this is for 5828 bytes of data, not total bytes transmitted on behalf of the correspondent node. Actual bytes transmitted over the home network to a mobile node at home (i.e., no Mobile IP) would include headers, and would total $5828 + 4 * 34 = 5964$ bytes , an increase of 136 bytes, or 2.3%.

Table 5.23: Network Load by Traffic Type and Connectivity Type

			Network Load on Home Network	
Traffic Type	Mobile IP	Completion (secs)	Average	Maximum
NFS	Yes	489	.33	.71
NFS	No	197	.34	.40
FTP	Yes	208	.57	.76
FTP	No	153	.43	.55

Because of fragmentation at the home agent, 5828 data bytes from the correspondent node results in 6126 bytes from the home agent. Therefore, to transmit 5828 bytes using Mobile IP, at least 12090 bytes of traffic traverses the home network if the home agent must re-transmit the packets to the appropriate router. If the home agent is the appropriate router, no extra traffic is generated on the home network.

With datagrams or concentrated packet bursts smaller than 1480 bytes, the overhead is slightly less because the packet does not have to be re-fragmented at the home agent. However, in terms of total bytes transferred on the home network, re-fragmentation is not as significant a factor as the position of the home agent.

5.5.2 Observed Network Load on Home Network

An overview of overhead for NFS and FTP traffic is given in Table 5.23. This table shows the type of traffic presented, whether or not Mobile IP is used, how long the transfer takes in seconds, the average level of network usage, and the maximum load achieved. The completion time reflects how long the protocol takes to complete a 10 megabytes file transfer. These values are averages over 100 transfers.

The large difference in completion time between FTP and NFS would be mostly because of differences in how these protocols decide when to send information from the correspondent node to the mobile node. With FTP, windowing is used, and with the window size usually around 14000 bytes, a large amount of information can be sent without acknowledgments. With NFS, a request must be sent out for

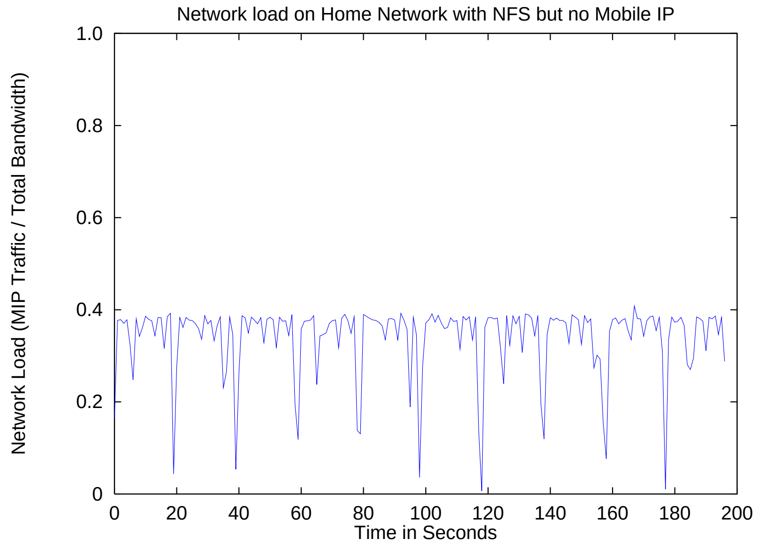


Figure 5.23: NFS Traffic on the Home Network Without Mobile IP

each piece of data transferred. The reply contains 8192 bytes of data. No further replies are sent until they are explicitly asked for by the receiving node. Therefore the NFS traffic is slower because it can only send information in discrete chunks of 8192 bytes, whereas FTP uses a sliding window that constantly updates the amount that can be sent with every acknowledgement received.

The average and maximum network usage show the ratio of Mobile IP traffic to total bandwidth. These last two values are calculated by averaging samples that are taken once per second. Each sample takes the total number of bits in Mobile IP packets during that second and divides this by the total bandwidth in terms of bits per second.

The plots in Figures 5.23 through 5.28 show the traffic levels on the home and foreign networks. These figures are measured on the network as a sequence of ten transfers are performed, with and without Mobile IP on the home network and with Mobile IP on the foreign network (because home network traffic does not reach the foreign network).

Figures 5.23 and 5.24 show network traffic levels generated by NFS over time. Figure 5.23 shows the traffic without Mobile IP. This traffic is generated by an NFS transfer on the home network from the correspondent node (donald) to the mobile

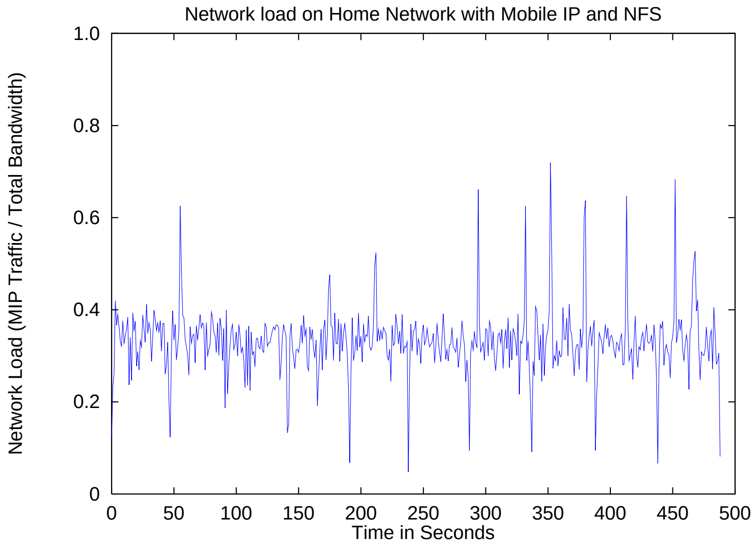


Figure 5.24: NFS Traffic on the Home Network With Mobile IP

node (bobby). The individual transfers taking place over time are evident. The time between transfers is clearly marked by large reductions in the network usage level at that time. While the NFS transfer is occurring, the traffic level shows large variation below the 0.38 level, but almost no variation above this level.

NFS traffic with Mobile IP is shown in Figure 5.24. This figure demonstrates much more variability than that shown in Figure 5.23. The boundary between one transfer and the next has become blurred, and is no longer as apparent. Although the average level of traffic is still around 0.34, the variability causes any individual second's worth of traffic to be far above or below this level. It is evident from comparing the two graphs that Mobile IP introduces a lot of variability into the traffic, and causes extreme spikes in the network usage high above values normally seen for NFS traffic of this sort. Since the total amount of NFS data transferred in Figures 5.23 and 5.24 are the same, the difference in completion time (shown at the bottom of the graphs) shows that Mobile IP greatly increases the time required to perform data transfers using NFS.

The overhead of FTP traffic is shown in Figures 5.25 and 5.26. The individual transfers are clearly distinguished from each other in the plot by deep troughs, as they were for NFS traffic. Again, the variability in network load is quite low, and

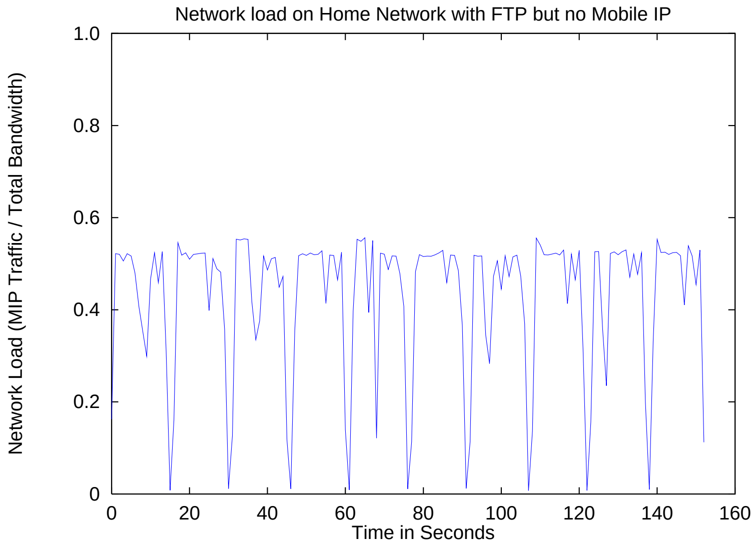


Figure 5.25: FTP Traffic on the Home Network Without Mobile IP

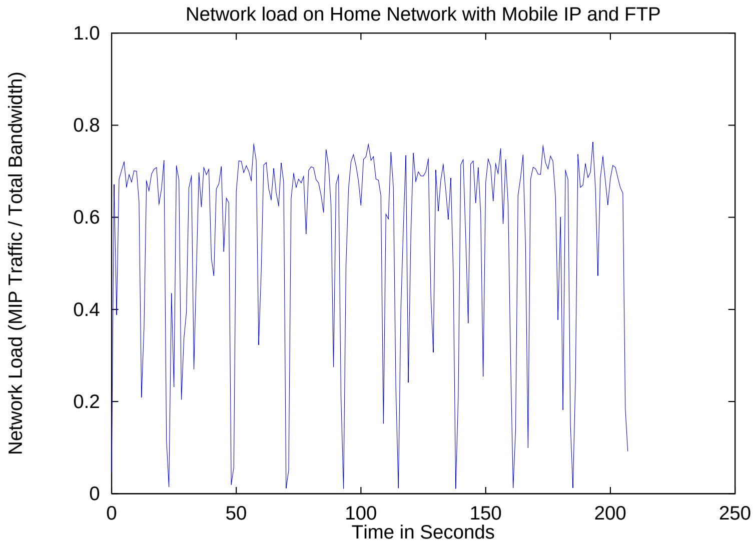


Figure 5.26: FTP Traffic on the Home Network With Mobile IP

most of the variability reduces the network load from the transfer. These figures do show that the impact of FTP traffic is much greater than that of NFS traffic. NFS transfers have a ratio of 0.38, but the FTP transfers shown here have an average ratio over 0.45. This represents a significant increase in bandwidth usage (10%) as compared to NFS, although the FTP transfers take less time to complete.

When Mobile IP is introduced (Figure 5.26), the variability and the usage levels increase substantially. Although most of the usage is still below the 70%, there is a

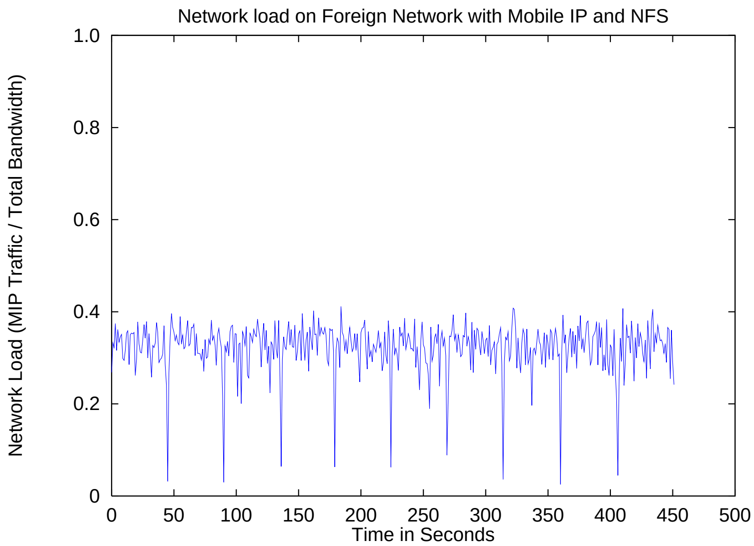


Figure 5.27: NFS Traffic on the Foreign Network With Mobile IP

lot more above this mark than with no Mobile IP present. The average bandwidth usage with Mobile IP is 0.58, which is a high level of average network usage.

5.5.3 Observed Network Load on Foreign Network

The load on the foreign network is essentially the same as that on the home network (see Figures 5.27 and 5.28). The average traffic level of about 0.37 is very comparable with the level on the home network for the same traffic. Some extra overhead is caused on the foreign network due to traffic from the mobile node to the correspondent node. This traffic in the reverse direction gets duplicated on the foreign network because the foreign agent is used as the default gateway.

In general, the network usage plot for the traffic on the foreign network has a shape similar to the same traffic on the home network except that there is a lot more variability. The traffic level is usually within 0.05 of the average while the transfer is occurring, and between transfers, the traffic level drops steeply to levels below 0.05.

The trace in Figure 5.28 shows the load on the foreign network for the FTP traffic. The performance is the same as that on the home network, except that the variance is a little larger, and the low traffic levels between NFS transfers are less pronounced. As on the home network, the average traffic level is centred around

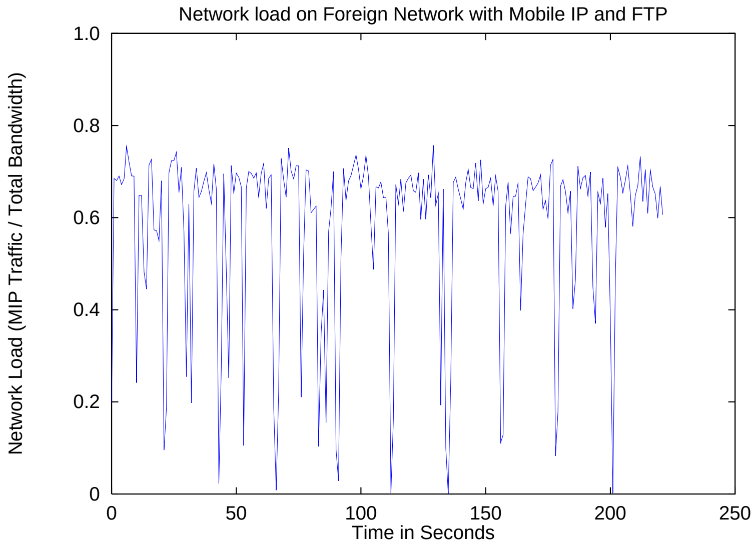


Figure 5.28: FTP Traffic on the Foreign Network With Mobile IP

0.70 during the transfers, but sporadically reduces to low levels during and between transfers. There does not seem to be any apparent effect from the packets travelling from the mobile node to the correspondent node.

5.6 The Impact of Multiple Mobile Nodes

This section will studies the impact of ambient traffic on the performance of ping traffic over the Mobile IP link. Ambient traffic is introduced in four forms: traffic on the home network, traffic between the home and foreign agents, traffic to a single mobile host, and traffic to multiple mobile hosts. The ambient traffic used for these experiments is ping traffic issued four times per second. The ping traffic is generated by twenty ping programs running on the correspondent node, each outputting one ping request every 0.25 seconds. Thus, there is on average one ping request every 1/80 th of a second. The datagram size used for these experiments is 64 bytes.

The ping traffic is generated on the home network by pinging the correspondent host (donald) from the home agent (morty). The ping traffic between the home and foreign agents is created by pinging darlene (the foreign agent) from morty. The pings to the other mobile hosts are made to fictitious addresses on the home

Table 5.24: Round Trip Times with Ambient Traffic (milliseconds)

Ambient Traffic Type	Minimum	Maximum	Average
Home Agent to Foreign Agent	5.704000	30.208000	13.066948
Traffic on Home Network	7.484000	30.308000	13.014123
Traffic to a Single Mobile Host	10.236000	117.520000	43.536879
Traffic to Multiple Mobile Hosts	7.800000	118.932000	44.259681

network. These do not actually correspond to hosts, but they have been registered as mobile nodes on the foreign network.

The results presented show the round trip times experienced by ping traffic from the mobile node to the correspondent node and back. This traffic is sent out while the ambient traffic is flowing on the networks in order to create interference for the pings.

Table 5.24 shows the results from these experiments. The average, minimum, and maximum round trip times are shown for the ping traffic with each of the four different ambient traffic types. The ambient traffic that is sent using Mobile IP seems to have had the greatest effect on the performance of traffic to the mobile node. Table 5.24 shows that the average round trip time with Mobile IP is 43.5 milliseconds with ambient traffic to a single node and 44.3 milliseconds with ambient traffic to multiple mobile nodes. In contrast, the average round trip time with ambient traffic between the home agent and the foreign agent is 13.1 milliseconds, and the average round trip time with ambient traffic on only the home network is 13.0 milliseconds.

Figure 5.29 shows the frequency distribution of the round trip times for the four test environments. The distributions show that the round trip time distribution with traffic on the home network is identical to the distributions with traffic between the home and foreign agents.

The other two traffic sets, with traffic sent to a single and many mobile nodes, also exhibit similarities with each other. The round trip times in these cases are generally much larger than in the other two cases. The highest concentration of values is at 40 milliseconds, but the variability of round trip times is high, from 20

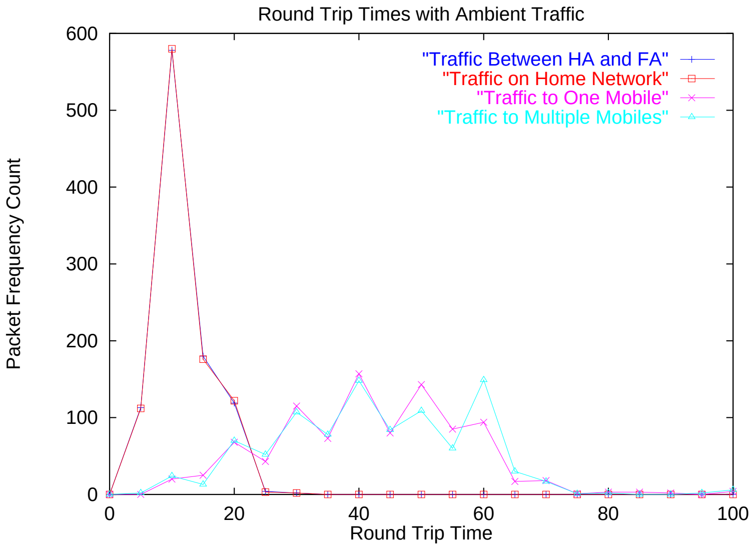


Figure 5.29: Round Trip Times With Ambient Traffic

through 60 milliseconds.

From the data presented in this section, it is apparent that ambient traffic over the Mobile IP link has a more substantial impact on the performance of Mobile IP then either ambient traffic on the home network, or ambient traffic between the foreign and home agent. It does not matter whether this traffic is destined for a single host or many hosts.

Chapter 6

Summary and Conclusions

6.1 Summary

Mobile IP is a new protocol that has been developed to support mobility in an internetwork while maintaining a single IP address for reachability. This solves a problem with the TCP/IP protocol suite because the IP addresses carry the location of the node as part of the address itself. The problem is that since location is encoded in the address, without Mobile IP, a node that changes location has to also change its IP address in order to maintain connectivity.

Mobile IP changes the environment considerably for mobile nodes. With Mobile IP, mobility support agents are set up on networks for forwarding a mobile node's traffic to its current destination. The agent located on the home network (the home agent) is responsible for intercepting traffic and forwarding it. Agents on foreign networks (foreign agents) are responsible for receiving the forwarded traffic from the home agent and delivering it to the mobile node. Through correct synchronization of the agents and the mobile node itself, Mobile IP is able to track the movements of a mobile node and continue packet forwarding over subsequent relocations.

The work presented in this thesis studied the performance implications of Mobile IP. These performance implications were measured in several ways: operability, failures, throughput, round trip time, network usage levels, and individual packet delay. Several types of experiments involved sending traffic to the mobile node and measuring the impact on the home and foreign networks. The results of the experiments were presented in graphical and tabular form.

All experiments presented here were performed on an experimental test-bed in the DISCUS laboratory at the University of Saskatchewan. This test-bed consisted of two networks joined by a single router. A mobile node was allowed to migrate between these networks with one of them being the mobile node's home network and the other being a foreign network. A home agent and foreign agent were also provided on their respective networks. Correspondent nodes, with which the mobile node exchanged traffic, were placed on the home network .

Traffic in the test-bed was supplied by FTP, NFS, various X-windows applications, and ping. The variety of traffic types provided different opportunities for testing Mobile IP's capabilities. These traffic types were combined with various locations for the mobile node to provide most of the experimental work in this thesis.

Mobile IP was examined in an attempt to determine both the cost for the end users and the networks involved. The evaluation for the end users focused upon application functionality issues and system failures, as well as performance issues such as throughput, delay, and round trip time. loss, and delays. The cost to the networks involved was examined by looking at the extra traffic imposed by Mobile IP on both the home and foreign networks.

Delays were examined by exchanging several different traffic types between the mobile node and its correspondent nodes while the mobile node was on a foreign network. The results of these experiments showed that bursty traffic (large amounts of packets arriving at the home agent in a short period of time followed by periods of little or no traffic) experienced large delays at both the home agent and foreign agent. Sparse traffic was forwarded with a minimal delay of 0.2 milliseconds.

Packet delay depended on the way the packet arrived at the home agent for forwarding. This was most evident when large datagrams were sent as ping traffic to and from the mobile node. The delays rose steadily with the rise in the datagram size, and significant delays above 100 milliseconds became common when datagram sizes larger than 5000 bytes were used. When the datagrams were small, little interference occurred between the packet arrivals, and delays were small. As the datagram size increased, and therefore, the density of the resulting packet burst at the home agent increased, the delays also increased because of congestion at the home agent.

Re-fragmentation at the home agent was found to delay packets at least 0.5 milliseconds. When contrasted with the 0.2 millisecond delay experienced by packets which were not re-fragmented, this represented a significant increase. As ping datagram size increased to 2968 bytes, delays of 1.4 milliseconds and more became common because of the queueing of packets at the home agent. With a datagram size of 5828 bytes, delays above 2.0 milliseconds were more common than with smaller datagrams, but most of the delays were still below 2.0 milliseconds. With 11740 bytes, significantly larger delays of greater than 10 and even 100 milliseconds were experienced. This progressive rise in delay was combined with a rise in the ratio of packets being re-fragmented to packets not being re-fragmented. Higher delays also corresponded to an apparent increase in interference between arriving packets and outgoing packets at the agents.

Packet delay through the foreign agent was less than through the home agent. In this test-bed, the foreign agent was actually a slower computer than the home agent, and this suggests that the delay at the agents was not caused by processing overhead, but was more likely caused by queueing delay. In addition, the delay at the foreign agent seems to have been affected by the delay at the home agent. When the packets were delayed at the home agent, they had larger inter-arrival times at the foreign agent. Therefore, the packets arrived at the foreign agent with sufficient time between the arrival of each packet to allow for processing and forwarding. As a result, queueing delays were reduced. When the packets were forwarded quickly through the home agent, delay increased at the foreign agent due to small interarrival times for the packets. Overall, however, packet delay still increased on both agents as datagram size increased.

Re-fragmentation was shown to be a significant factor in the increasing delay of larger datagrams. Significant delay was found at the home agent and the foreign agent when packets forwarded between these entities were re-fragmented at the home agent. Re-fragmenting also significantly increased the frame count on the home and foreign networks.

Other traffic present on the Mobile IP link had a large effect on end-to-end performance. Traffic over the Mobile IP link affected delay more than either traffic on the home network or independent traffic between the home agent and foreign agent. This suggests that many mobile nodes operating over a single Mobile IP link will interfere with each other in a significant way. The experiments for studying ambient traffic used three types of traffic: traffic on the home network, traffic between the home agent and foreign agent, and Mobile IP traffic to other mobile nodes. When the traffic was placed on the home network, it caused little delay to the Mobile IP traffic. When the traffic was between the home and foreign agents, it also produced little delay on the Mobile IP traffic. When the traffic was itself Mobile IP traffic over the HA-FA link, however, this had a noticeable impact on the traffic to the mobile node. In addition, it did not matter whether the ambient traffic was to a single recipient, or to many different recipients on the foreign network.

In summary, the impact of Mobile IP on the end user involves a reduction in performance, but not much of a reduction in functionality. Our results indicate that significant packet delays can be suffered with Mobile IP traffic. The round trip times and end-to-end throughput experienced by the protocols we tested were all affected by Mobile IP. In general, delays increase at least 10% in our test-bed, and end-to-end throughput decreased significantly. For NFS transfers, throughput decreased 56.26%, and for FTP transfers throughput decreased 36.06%. When using applications involving X-windows traffic, little delay if any was noticeable, however.

The impact on the support networks was studied by examining the extra traffic load that was added to the networks by Mobile IP. This effect was examined by calculating the extra bytes per packet that needed to be transmitted and the observed network load of FTP and NFS transfers. Analysis showed that the protocol overhead per packet was 20 bytes if the packet did not require re-fragmentation, and 74 bytes if the packet did require re-fragmentation. In terms of total bytes transferred, the worst-case scenario would result in an extra 6126 bytes being transferred for a 5828 byte datagram. Thereby approximately doubling the amount of traffic required on the home network to deliver the information to the mobile node.

The experimental observations regarding the performance impact of Mobile IP supported the analytical results. In the experiments presented here, the impact in terms of network bandwidth was examined. The traffic source for the experiments were FTP and NFS file transfers. In terms of average bandwidth used per unit time, the requirements for the Mobile IP traffic were comparable to those without Mobile IP, and were slightly less with NFS traffic (33% vs. 34%), and slightly more with FTP traffic (57% vs. 43%). In terms of peak impact on the network, both FTP and NFS consumed large amounts of bandwidth (76% and 71% respectively), whereas the same transfers without Mobile IP consumed much less bandwidth (55% and 40% respectively). In both the NFS and FTP cases, the transfers not only had higher peak bandwidth requirements, but they took a longer time (148% more time for NFS and 36% more time for FTP).

6.2 Conclusions

The results show that Mobile IP can effectively deliver traffic to a mobile node when it is away from its home network. Mobile IP mechanisms can track movement across networks and maintain connectivity for popular application programs and traffic types. In particular, X-windows, FTP, and NFS traffic can be transmitted over Mobile IP with little effect on the end user. The environment that a user experiences on his/her home network, complete with the user interface, and NFS-mounted filesystems, can be preserved on the foreign network through the use of Mobile IP. The results also indicate that Mobile IP is able to operate with a variety of machines, and interact correctly with different operating systems.

Mobile IP forwarding was shown to produce significant delays, however, and these significantly affect end-to-end performance of traffic for the mobile node. In addition, these delays appear to increase with the burstiness of the traffic source. The delays at the home agent support steps to minimize the home agent's role through techniques such as optimized routing. With optimized routing, delay overhead is reduced in many ways. First, the physical distance the packet has to travel can be greatly reduced if the correspondent node and mobile node are closer together than either is to the home agent. Second, since the packets are encapsulated at the correspondent node, a large packet (such as a 11748 byte ping packet) will only be encapsulated once, instead of 8 times as is the case when the home agent is used.

The reduction in encapsulation work would result in two beneficial effects: less encapsulation for the same number of bytes, and fewer packets re-fragmented at the home agent. In the test-bed, when a large packet was sent from the correspondent node, it was sent as several maximally-sized Ethernet frames. When these frames reached the home agent, each one had to be encapsulated separately. This encapsulation increased the size of these packets, and caused them to be re-fragmented at the home agent. This entire process caused extra work at the home agent and on the home network (as the number of packets coming out of the home agent was close to double that going in, when large packets were encountered).

From the network overhead results it can be concluded that both NFS and FTP traffic have large peak bandwidth usage when forwarded over Mobile IP. These results suggest that while file transfers without Mobile IP operate within a range which does not saturate the ethernet, Mobile IP forwarding adds enough extra traffic to cause network saturation. This situation can be even further exaggerated if multicast traffic and broadcast traffic are unicast to the roaming mobile nodes. If a network wishes to support many mobile nodes, it would be advisable to place home agents and foreign agents at the edge of the network, so that the forwarding of packets on the home and foreign networks would be reduced for most traffic. A further extension of this idea would be to have each router on the home network be a home agent and have a master home agent that the mobile node would communicate with. When the mobile node registers, the registration would be handed off to the router that would be used for sending packets to the foreign network. In this way Mobile IP traffic would never be seen on the home network.

In conclusion, Mobile IP has been shown to effectively provide mobility support without requiring address changes at the mobile node. The applications tested worked well with Mobile IP, and there was little effect on the end-user. In short, the goal of transparency seems to have been achieved. Specific performance results have pinpointed areas in which bottlenecks exist for Mobile IP. The agents in particular can cause considerable delay and overhead, and ideas for optimising the involvement of agents need to be explored to reduce these effects.

6.3 Future Work

Mobile IP has been examined with respect to delay characteristics in a limited environment. Future research is needed to determine the performance of Mobile IP in a larger environment. In particular, increasing the distance between the home and foreign networks used should cause substantial increases in the traffic delay. Increasing the distance travelled could also introduce traffic loss, which may be exaggerated by Mobile IP. Packet loss was not significant in the experiments performed in our test-bed due to the small distance between the home and foreign networks. The characteristics of Mobile IP in a WAN environment need to be investigated.

Future work is needed in the area of wireless networks. The performance of Mobile IP in a wired environment has been shown to be acceptable for most needs. In a wireless environment, signal quality, overlapping signals, and significant packet loss are substantial considerations. Combined with the increased bandwidth usage of Mobile IP, these could cause significant problems. The interaction between Mobile IP and wireless networks needs to be studied to determine how effectively Mobile IP operates and optimizes the delivery of traffic to a wireless mobile node.

This thesis has restricted its studies to a test-bed in which all entities use the same Mobile IP software, and both the mobile node and the mobility agents use the same operating system. In order to accurately show Mobile IP performance in the Internet, the impact of combining many operating systems, and multiple implementations of Mobile IP over a large area needs to be examined. A better appreciation of how Mobile IP supports environments based on operating systems other than UNIX, and programs other than those common on UNIX platforms is needed.

Multicast traffic has not been studied in this thesis. The combination of multicast traffic with mobility creates certain problems that need to be addressed [2]. Solutions to these problems have been proposed as part of a new protocol called MoM [10]. This protocol needs to be implemented and tested as a solution for multicast support.

Appendix A: Glossary

Care-of Address An address that a mobile node acquires from a foreign network to use as a forwarding address while visiting that network. Packets to the mobile node are encapsulated by a mobile routing agent and sent to this address. Facilities are provided at the care-of address for delivering the packets to the mobile node.

Correspondent Host (CH) A correspondent node that is not acting as an internetwork router.

Correspondent Node (CN) A host or router that wishes to communicate with a mobile node. Correspondent nodes know the permanent address for the mobile node and send information to this address.

DHCP Dynamic Host Configuration Protocol: A protocol used for automatically configuring a node with an IP address. A group of IP addresses at a network are set aside for temporary use. When a node needs an address, it communicates using DHCP to a DHCP server which issues one of these addresses to the node for temporary use.

Foreign Agent (FA) A node on the foreign network that is responsible for receiving tunnelled packets from the HA, decapsulating these packets, and forwarding them to the MN.

Foreign Network (FN) A network other than a mobile node's home network that the mobile node uses for connection to the Internet.

FTP File Transfer Protocol: An application and protocol designed for bulk data transfer. FTP uses tcp to set up a connection between to two nodes in which commands can be passed from the client node to the server node. Based on these commands, many actions such as sending and retrieving data files can be initiated. The FTP application uses TCP for transferring the data files.

Home Agent (HA) A node on the MN's home network that is responsible for intercepting packets delivered to the MN and forwarding them to the MN's care-of address through a tunnel using encapsulation.

Home Network (HN) The network whose network prefix corresponds to the mobile node's home address (i.e., the mobile node's home address has been assigned on this network).

Home Address A mobile node's permanent address on its home network. This is a unique identifier used to correspond with a mobile node.

ICMP Internet Control Message Protocol: A protocol for sending error messages to nodes in an internetwork. This protocol operates at the IP layer using IP addresses. It is used to report on errors encountered in delivering IP packets to their final destination.

IETF Internet Engineering Task Force: A group of professionals whose purpose is to research and propose stanards for use on the Internet.

IP Internet Protocol: The protocol used for transferring packets in an internetwork. IP allows networks boundaries to be crossed by encoding the destination network in the IP address. IP does not provide sophisticated traffic flow management, and is a connectionless protocol. The current version of IP is IPv4. An extended version (IPv6) is being developed with an expanded address space of 2^{128} addresses, and many other advanced features.

Mobile Host (MH) A mobile node that is not acting as an internetwork router.

Mobile Node (MN) A host or router that will vary its location by crossing network boundaries while still requiring full connectivity through its permanent Internet address. These nodes must be aware that they are mobile and must be able to interact with mobile routing agents (see below) so that these agents can set up special packet routing needed to deliver the MN's packets to its temporary locations on foreign networks (see below).

Mobile Routing Agent (MR) A node on the Internet which can intercept packets on behalf of the mobile node and ensure their delivery to the mobile node on its foreign network. These routing agents are aware of both the permanent and care-of addresses associated with a mobile node and implement the routing between these two addresses.

Mobile Security Association (MSA) A security relationship set up between two mobile entities that promotes correct entity identification. This helps to ensure that a node is the mobile entity that it says it is. *Packets* are the collections of data that are transmitted on networks according to a set of rules called *network protocols*. Packets have two parts: the *header* and the *payload*.

Mobility Binding An association of a care-of address with a home address.

MTU Maximum Transmission Unit: The maximally-sized frame allowed on a network.

NFS Network File System: A method for sharing filesystems with a network. A NFS server exports its file systems which are mounted by NFS clients. These clients may interact with the filesystem causing commands to be sent to the server using the NFS protocols. Bulk data transfers may be done using the NFS protocol. NFS usually uses UDP as its transport layer protocol.

Packet Encapsulation The inclusion of an entire packet of information as the data portion of another packet.

Packet Header A sequence of bits placed at the beginning of the packet and used to specify where the packet is to go, who sent it, and other protocol-specific information. The *payload* is the data being transferred within the packet.

PPP Point to Point Protocol: A protocol commonly used in dial-up connections to transfer data across the dial-up connection.

Router A network node that keeps traffic flowing on a network or internetwork by forwarding information to other routers. The basic task of a router is to decide where to send a piece of data it receives for another host and then send it there [7].

TCP Transmission Control Protocol: Part of the TCP/IP family of protocols. A transport layer protocol with sophisticated flow control, congestion control, and error handling built in. TCP is a connection-based protocol which can provided a stable link between two processes residing on separate nodes or the same node.

Tunnel The path followed by a packet while it is encapsulated. A mobility agent will encapsulate a packet to the mobile host by enclosing the entire packet with an outer IP header that specifies the care-of address at the other end of the tunnel. This tunnelled packet is delivered using normal IP mechanisms, and is decapsulated at the other end of the tunnel.

UDP User Datagram Protocol: Part of the TCP/IP protocol family. A connectionless transport layer protocol providing dtagram traffic flow between two processes arbitrarily located within the same internetwork.

Visitor List A list of visiting mobile nodes maintained by a routing agent on foreign networks. This list contains information such as the home agent for the mobile node, a time-out value for deleting this entry from the list, and possibly the network interface to which the node is connected, and its network address.

References

[1] V. Cerf and R. Kahn. A Protocol for Packet Network Intercommunication, *IEEE Transactions on Communications*, 22(5):637–648, May 1974.

[2] V. Chikarmane, R. Bunt, and C. Williamson. Mobile-IP based Multicast as a Service for Mobile Hosts In *Proc. SDNE '95: Second International Workshop on Services in Distributed and Networked Environments*, 11–18, Whistler, B.C., Canada, June 1995.

[3] V. Chikarmane. *Network Support for Mobile Hosts in a TCP/IP Internetwork.* MSc. Thesis, Department of Computer Science, University of Saskatchewan, 1995.

[4] V. Chikarmane, C. Williamson, and R. Bunt. Implementing Mobile IP Routing in 4.3 BSD Unix: A Case Study. *Internetworking: Research and Experience*, 6(4):209–227, 1995.

[5] V. Chikarmane, C. Williamson, R. Bunt, and W. Mackrell. Multicast Support for Mobile Hosts Using Mobile IP: Design Issues and Proposed Architecture. *Mobile Networking and Applications*, 1998 (to appear).

[6] F. Choong. TCP Performance in Mobile IP. MSc. Thesis, Department of Electrical Engineering, National University of Singapore, 1997.

[7] D. Comer. *Internetworking with TCP/IP*, Volume 1. Prentice-Hall Inc., Englewood Cliffs, NJ, second edition, 1991.

[8] D. Johnson and D. Maltz. *The CMU Monarch Project IETF Mobile IPv4 Implementation User's Guide.* http://www.monarch.cs.cmu.edu, February 1997.

[9] G. Forman and J. Zahorjan. The challenges of mobile computing. *IEEE Computer*, 27(4):38–47, April 1994.

[10] T. Harrison, C. Williamson, W. Mackrell, and R. Bunt. Mobile Multicast (MoM) Protocol: Multicast Support for Mobile Hosts. In *Proc. ACM/IEEE MOBICOM '97 Conference*, 151–160, Budapest, Hungary, September 1997.

[11] J. Ioannidis, D. Duchamp, and G. Maguire. IP-Based Protocols for Mobile Internetworking. In *Proc. ACM SIGCOMM '91 Conference: Communications Architectures and Protocols*, 235–245, Zurich, Switzerland, September 1991.

[12] D. Johnson, A. Myles, and C. Perkins. *The Internet Mobile Host Protocol (IMHP).* Internet Draft, Mobile IP Working Group, IETF, February 1994.

[13] D. Johnson and C. Perkins. *Route optimization in mobile IP: Version 07.* Internet Draft, Mobile IP Working Group, IETF, November 1997.

[14] T. Berners-Lee, R. Cailliau, J-F. Groff, and B. Pollermann. World-Wide Web: The Information Universe. Electronic Networking: Research, Applications and Policy, 2(1):52–58, Spring 1992.

[15] G. Montenegro. *Reverse Tunneling for Mobile IP: Version 2.* Internet Draft, Mobile IP Working Group, IETF, March 1997.

[16] A. Myles and C. Perkins. *Mobile IP extensions (MIP)*, October 1993. Internet Draft.

[17] A. Myles and D. Skellern. Comparing Four IP-Based Mobile Host Protocols. *Computer Networks and ISDN Systems*, 26:349–355, 1993.

[18] A. Myles and D. Skellern. Comparison of Mobile Host Protocols for IP. *Internetworking: Research and Experience*, 4(4):175–194, December 1993.

[19] C. Perkins. *Minimal Encapsulation Within IP.* RFC 2004, Mobile IP Working Group, IETF, October 1995.

[20] C. Perkins. *IP Encapsulation Within IP.* RFC 2003, Mobile IP Working Group, IETF, October 1996.

[21] C. Perkins. *IP Mobility Support.* RFC2002, Mobile IP Working Group, IETF, October 1996.

[22] C. Perkins. Mobile Networking in the Internet, *Mobile Networking and Applications*, 1998 (to appear)

[23] C. Perkins and D. Johnson. *Mobility support in IPv6: Version 03.* Internet Draft, IPv6 Working Group, IETF, August 1997.

[24] C. Perkins and Y. Rekhter. *Short-Cut Routing for Mobile Hosts.* Internet Draft, IBM T.J. Watson Research Center, July 1992.

[25] J. Postel, editor. *Internet Protocol: DARPA Internet Program Protocol Specification*, RFC791. USC/Information Sciences Institute, September 1981.

[26] J. Postel, editor. *Transmission Control Protocol: DARPA Internet Program Protocol Specification*, RFC793. USC/Information Sciences Institute, September 1981.

[27] S. Rajagopalan and B. Badrinath. An adaptive location management strategy for Mobile IP. In *Proc. MOBICOM '95*, 170–180, Berkeley, California, USA, November 1995.

[28] A. Tanenbaum. *Computer Networks*. Prentice-Hall, Inc., Englewood Cliffs, N.J., third edition, 1996.

[29] F. Teraoka and K. Uehara. Mobility Support in IPv6 Based on the VIP Mechanism. In *Network Technology: INET'95*, Honolulu Hawaii, June 1995.

[30] F. Teraoka, Y. Yokote, and M. Tokoro. A network architecture providing host migration transparency. In *Proc. ACM SIGCOMM 91*, pages 209–220, Zurich, Switzerland, September 1991.

[31] C. Williamson, T. Harrison, W. Mackrell, and R. Bunt. Performance Evaluation of the MoM Mobile Multicast Protocol. *ACM/Baltzer Journal on Mobile Networks and Applications (MONET)*, 1998 (to appear).