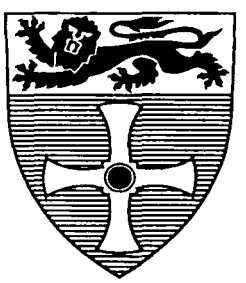
Location and Routing Optimization Protocols Supporting Internet Host Mobility

Gi Hwan Cho



Ph.D. Thesis

The University of Newcastle upon Tyne
Computing Science Department

December 1995

To my lovely children YooRi and BaDa,

and my wife MoonJa

Abstract

With the popularity of portable computers and the proliferation of wireless networking interfaces, there is currently a great deal of interest in providing IP networking support for host mobility using the Internet as a foundation for wireless networking. Most proposed solutions depend on a default route through the mobile host's home address, which makes for unnecessarily long routes. The major problem that this gives rise to is that of finding an efficient way of locating and routing that allows datagrams to be delivered efficiently to moving destinations whilst limiting costly Internet-wide location updates as much as possible.

Two concepts – "local region" and "patron service" – are introduced based on the locality features of the host movement and packet traffic patterns. For each mobile host, the local region is a set of designated subnetworks within which a mobile host often moves, and the patrons are the hosts from which the majority of traffic for the mobile host originated. By making use of the hierarchical addressing and routing structure of Internet, the two concepts are used to confine the effects of a host moving, so location updates are sent only to a designated host moving area and to those hosts which are most likely to call again, thus providing nearly optimal routing for most communication.

The proposed scheme was implemented as an IP extension using a network simulator and evaluated from a system performance point of view. The results show a significant reduction in the accumulated communication time along with improved datagram tunneling, as compared with its extra location overhead. In addition, a comparison with another scheme shows that our functionality is more effective both for location update and routing efficiency. The scheme offers improved network and host scalability by isolating local movement from the rest of the world, and provides a convenient point at which to perform administration functions.

Acknowledgements

First of all, I owe a debt of gratitude to my supervisor, Dr. Lindsay Marshall, for his help and encouragement throughout this research. His reading and commenting upon the numerous drafts have been invaluable in completing this thesis. His efforts are greatly appreciated and cannot be forgotten.

I would like to thank several members of the Computing Science Department, particularly to Professor Santosh Shrivastava and Dr. Graham Parrington, for their many helpful comments and encouragement. Further thanks go to Trevor Kirby, for his many help on use of computing facilities, and to Shirley Craig, for the kind support she does as the Department's librarian.

I am also sincerely grateful to the Electronics and Telecommunications Research Institute (ETRI) in Korea, which has provided me the scholarship and financial support during my studies. Further, I would like to express my gratitude to Dr. GilRok Oh in ETRI for his continuous stimulus to my family.

Last, but not least, I should not forget to thank my lovely children, SooBin and KyuJin, and my wife MoonJa in Korea. They have to live apart from my care for two years. My parents, brothers and sisters have supported and encouraged me in many ways during the time I have spent working on this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the early 1990s, we have seen two great revolutions in computing technology. First, portable computers which are as powerful as some desktop workstations in terms of both features and computational power began to appear, and these are now widely available and affordable. Second, there has been intense interest in wireless communication such as cellular communications, wireless LAN, wireless data networks, and satellite services. Given the likely conjunction of these two exploding trends, users of portable computers are now no longer required to remain confined within wired network premises to get network access. Users would like to carry their computers with them wherever they go and yet maintain network connections despite migration from one network to another. This trend has been appearing in a new computing paradigm – *mobile computing* or *mobile data networking.*

Portable computers are a natural development, given the tremendous success of personal computers and the strong trend in the computer industry to produce devices with decreasing size and increasing power. The generic term for this type of portable, personal computer (or device) is a *mobile host.* "Personal Digital Assistants (PDAs)" are specialized mobile hosts designed to support a limited set of tasks. There are also increasing indications that modern computing environments cannot be thought of without some form of data networking. With the availability of wireless network interface, a mobile host may be carried from one wireless network to another even while retaining its network connections; or it

may simply be disconnected from the network at its current location, temporarily moved to a new location, and reconnected to the network through either a wireless or conventional wired network interface. Mobile hosts may exchange data among themselves, or with peers beyond their immediate locale through the existing network infrastructure. As a result, mobile users are provided with the capability of accessing information anywhere and anytime.

Wireless communication systems, mostly for voice applications, have progressed enormously in the last decade. Together with the presence of mobile hosts, this has necessarily led to a new breed of data networks, hence the *wireless local network*. With the arrival of wireless networking, the field of modern telecommunications has introduced a new concept: Personal Communication Services (PCSs). PCSs are based upon the notion of tetherless access and the networks that support connections between people or between people and places, rather than merely supporting connections between places. To support this, there are two requirements; the ability of the network infrastructures (hardware and software) to locate and communicate with a called person wherever that called person may be, and the ability of the network to hand-off connections among network ports in response to user mobility. Thus, central to the notion of PCSs are specific network services customized to the unique needs of a given user, e.g. filtering and forwarding of electronic mail. Needless to say, the concepts of PCSs and mobile computing share most of the same ideas, but these are usually described in different terms from the communication and computing point of view respectively.

In mobile computing, it is unreasonable to assume that all, or even most, of the communications structure will be wireless. With the increased availability of low-cost wired networking options, the fixed and wired networks are still likely to exist and even be expanded as a basis for information repositories and processors. Actually, wireless networks should be considered as members of the ever-growing number of networks, reinforcing the need to incorporate the wireless network within the larger internetwork. A common networking protocol which supports host mobility is desirable in order to augment these networks smoothly and on a large scale. Because of its own world-wide success, the Internet will be the most likely internetwork to be used as a basis for wireless networking.

1.1 Internet Host Mobility

From the system design point of view, mobile computing can be regarded as a host mobility extension of distributed computing, simply by adding the mobility entities, such as mobile hosts and wireless networks. In technical terms, that requires the resolution of the problem of providing network accessibility to hosts which change their location relative to the rest of the network with time. Thus, locating moving hosts is the most important feature. Location inherently includes addressing the issue of moving hosts and a quite closely related issue, effective packet routing for moving destinations. These three issues, addressing, location and routing, bring several different design choices which are very closely interrelated to each other.

With its addressing and routing capability, it is widely agreed that host mobility support in the Internet should occur in the internet (IP) layer. However, the IP protocol deals badly with a dynamic network topology such as that provided by a wireless network. IP was designed with a static view in mind. On the one hand, given the existing (large) installed base of IP systems, it makes it difficult to contemplate major changes to the protocol; it is necessary to accommodate the newly required functionalities by appending or expanding. Much previous work has tried to tackle the host mobility problem within the Internet environment [7, 33, 39, 53, 55, 70, 73].

As a result of IP's reliance on the entire Internet address for mobile host identification, the most common solution adopted is that a mobile host maintains the same address as it relocates. This can be accomplished by making use of two different IP addresses – a logical identifier and a physical locator. The next consideration is the location strategy for obtaining the current location of moving hosts. One possible way is explicitly to query the current location from a location server prior to sending a packet. This brings availability and performance problems. Another method is to use the revised protocols that integrate mobile hosts into the traditional networking infrastructure, where a reference to the new location is deposited in mobility support network entities, in well known places, such as routers in the home area, or hidden places, such as in-between routers or mobile hosts. When an entity receives packets and has a new location for the packet destination, a forwarding protocol will send them to the new location.

1.2 Problem Definition

Packet routing is generally characterized as transporting information from a source to one or more destinations, so as to meet the service requirements for that information. Maintaining uninterrupted high-quality service for distributed applications in the presence of highly mobile end hosts requires the provision of a set of routing-related solutions. Most of all, a routing decision must be made based on the location information that is available. Packet routing paths depend critically on where and/or how much location information is preserved on the network as a whole. In addition, the location scheme that goes with host mobility is usually prone to scalability problems. The problems could be much more serious in a large internetwork, such as the Internet, when host mobility is spread out Internet-wide because location information has to be changed through the whole system.

With the Internet host mobility solutions, the packet routing paths that go with host mobility depend decisively on the "somewhere" which holds the information for a mobile host's physical locator. Location information therefore has to be well placed, so that it can be effectively utilized for packet routing. With insufficient location information, packets may be forwarded with a default route, such as via the home area, which makes for unnecessarily long routes. The return path from the mobile host follows a direct route, bypassing the host's home network, hence the so-called "triangle routing". As a matter of course, the cost of maintaining the location information should not outweigh its routing benefit. Moreover, if a system has too many location caches or updates, it may flood with location updates. Both of these cases eventually bring severe problems in terms of location and/or performance transparency to the mobile computing environment.

Some previous work [1, 2] has shown a theoretical trade-off between location and routing in the host mobility environment in terms of their efficiency. However, in practice, it is very important to try to optimize this situation in order to reduce the total network cost; that is, achieving better routing by sacrificing some overhead from an efficient location framework, and then providing higher performance to the system as a whole. This inspired the work in this thesis. The work is concentrated on realizing two seemingly conflicting aims – achieving optimal routing for most communication traffic whilst limiting location propagation as far as possible. The most prominent concern for achieving better routing and in the same time saving

costly location update is to find just the places where it is highly likely to be effectively utilized for packet routing. Thus, analysis of the host movement pattern and packet traffic pattern will play a decisive role. The approach used is based on our belief that routing is effectively controlled by the network infrastructure, whilst locating can effectively be handled by the mobile host itself.

1.3 Design Characteristics

To begin with, two important design choices are required in the mobile computing context. One is the time when the new location of a moving host may be propagated, which can be either *need-initiated* (whenever the location is expected to be needed) or *move-initiated* (whenever a host moves). We adopt a move-initiated system. The other is how out-of-date location caches may be reset. In the host moving procedure, it is desirable to preserve only the most recent cache entry for the initiator in order to prevent old cache entries from being used for a wrong routing. One possible way is to use a time-out to reset possibly out-of-date caches. We present a novel reclamation method, called *back firing*, where, whenever a mobility binding is updated on the previous agent, the agent clears the mobility binding on its previous agent for the host, if it had one.

To develop an effective location approach, we start with the features of the fixed infrastructure. In order to provide scalable routing support, Internet protocols make use of *hierarchical* addressing and routing schemes. The Internet itself has been growing in an hierarchical form, in order to accommodate its graceful augmentation and to provide administrative autonomy. Thus, our approach is to move some of the location and routing roles for mobile hosts to some part of the fixed network, mostly in a lower level internetwork (a regional administration domain), and then use this to localize the effect of host mobility into a designated area.

The next point to notice is that exploiting locality in a mobile computing paradigm would play a decisive role for providing an efficient location and routing. This is based on the most obvious assumption which is that mobile hosts are most likely to move around a designated region which usually contains its home subnet and the current subnet, and communicate with a limited number of source hosts (in the region) which have an interest in contacting it. These introduce two concepts

– "local region" and "patron" – for each mobile host. The local region is a set of designated subnetworks within which a host often moves, and the patrons are the hosts from which the majority of traffic for the mobile host originates. Thus, what we are trying to do is to limit the location propagation within the host's actual moving area and, if necessary, to the actual source hosts.

The two ideas above – the lower level administrative domain and local region – are joined to provide our basic schema. The top level router on each local region now acts as a redirection agent, by maintaining mobility bindings for hosts within its service boundary and providing forwarding for packets passing through it. To support this, a mobile host notifies its new location to the redirection agent whenever it moves. With one extra registration to the redirection agent, a mobile host now does not need to declare its movements outside of its local region, whilst source hosts residing outside the local region are permitted to use inaccurate location information for a mobile host. As a result, the local region provides a natural framework for localizing the effect of host mobility into a designated area, whilst most packets are still routed close to their optimal routing paths.

The patron concept is used to confine the effect of host mobility to those source hosts which are most likely to call again. Whenever a mobile host leaves or comes back to its local regions, it sends its new location to the patron hosts (i.e those that are the source hosts where the majority of traffic for the host originated). In this point, the patron service is partially a need-based location propagation. Source hosts that access a host frequently, even if it is located far from them, will keep up-to-date location information about it, and can use the location information for their next call. Then, the traffic from patron hosts, which covers most communication from outside of local region, can always achieve optimal routing.

1.4 The Simulation

Even though several other proposals for providing location and routing optimization have been made [1, 4, 8, 32, 53], a systematic study of such schemes had not been conducted prior to this thesis; most of the previous work provided only the theoretical or conceptual framework, and any results were only concerned with

the execution times at various individual internetwork components, rather than the behavior of the overall system. This is due to the fact that system behavior is extremely hard to capture without the use of a simulation environment, and that it is still difficult to model a large wireless internetworking environment.

The protocol designed in this thesis has been implemented as an IP extension and tested in a simulation environment constructed using an event-driven network simulator which is known as *netsim* [29]. The simulation was invaluable for assessing protocol correctness, as we were able to iterate and refine the design as problems were discovered. In addition, the simulation environment was utilized to perform an evaluation study of the location and routing effectiveness for our proposed scheme, and a comparison between our approach and a major contender. To do this, five different platforms were implemented according to the concepts involved. Using various simulation parameters and with various moving and calling scenarios, the location overheads were investigated, and then the encapsulation details provided as a direct outcome of the location efforts. Finally, the network occupation times were measured in order to determine the routing effectiveness of the location scheme on system performance.

1.5 Thesis Structure

The remainder of the thesis is organized as follows. Chapter two describes the general background of mobile computing, describing a model, its built-in characteristics and applications. The chapter then summarizes previous and existing host mobility related works on order to show the extent of the problem and some solutions for different constraints and design criteria.

Chapter three begins by describing the issues involved in the design of a host mobility extension for the Internet environment. It then examines the details of the Internet characteristics from the location and routing optimization perspectives and defines the mobile internetwork routing structure. The chapter also describes how to exploit the locality property of host moving and calling pattern in the mobile computing paradigm, then presents the two concepts, local region and patron.

Chapter four deals with greater detail about the control structure of the local region and patron service. Firstly, a system model for internetwork host mobility is defined along with its required functionalities. Then a home-based forwarding strategy is adopted for providing a basic host mobility solution. Each concept is now added to the basic scheme in turn. The following section describes details of the registration procedure, as a location means, and the packet routing paths for each possible moving situation, to show how the location endeavor is incorporated for routing effectiveness. This chapter also describes some system related issues.

Chapter five outlines the simulation created to conduct some evaluation of the scheme presented in terms of location overhead and routing efficiency. It then outlines some important parameters and a network model for the simulation. The implementation details of the scheme proposed are provided with following some implementation related issues.

Chapter six shows performance evaluation and comparison results of the simulation study of the effectiveness of the local region and patron concept. The first section of the chapter describes details of the simulation runs that were performed. An important parameter, symmetric rate, is then defined to formalize the moving and calling discipline of the mobile computing environment simulated. The following two sections provides the numerical results and a comparison with a major contender, in terms of the registration overhead, the encapsulation details as the direct effect of the location, and the data communication time (including direct routing details) as the eventual result of our location strategy. Finally, the rationale for certain features of our design is discussed.

The final chapter considers the conclusions of this thesis and suggests some directions for future research. Appendix A provides sample input parameters for the network configuration and system definition for the simulation carried out for this thesis.

Chapter 2

Previous and Related Work

This chapter reviews some related areas, namely, from the system model for mobile computing to its dedicated communication protocols. Past and present host mobility solutions are investigated in several dimensions. This chapter aims to give the reader a feeling of where the state of the art was when this work was started, and why and how existing proposals fall short of providing host mobility for large internetworks. Related work is cited at appropriate points later in the thesis, to compare and contrast it with our ideas.

2.1 Mobile Computing

In general, people believe that mobility will have a similar impact on the research community as distributed systems have done. The fundamental question "centralized or distributed" will now be extended to "static or mobile". In practice, mobile computing is merely a special extension of distributed computing, with host mobility. It is a consensus that many of the problems of mobile computing are indeed subsumed by distributed computing; but there are some differences. For example, location transparency is often a goal in distributed computing, whereas location awareness is a requirement in many mobile applications. In fact, mobility of users and services will be one of the main technical issues facing distributed systems of the present and even the future.

Mobile computing is a new emerging computing paradigm posing many challenging issues. A mobile host may cross the border between two different cells while being active. Moreover, the host will frequently be disconnected due to battery power restrictions. Most likely, reading and sending e-mail or querying a database will be separated by substantial periods of disconnection. Also, the host will wake up in a totally new environment in some new location far from home. These inherent features become noticeable when the host's rate of movement is high, and the network size is large. As far as possible, host mobility should appear seamless to the user. So, how do we find the current location of mobile hosts? Which features could (or have to) be absorbed into the mobile computing infrastructure? To answer this, it is useful to identify the intrinsic characteristics of mobile computing, and examine the major impacts brought by host mobility and potential mobile applications.

2.1.1 The System Model

Even though mobile computing is one of the next logical step of distributed system, this model cannot be directly used for mobile computing; a different system model is required. This is essentially caused by the fundamental features of mobile computing (see details in the next subsection). A mobile host can connect to the network from different locations at different times. Despite the fact that a physical (and logical) link between a mobile host and its access point with the fixed network varies as hosts move, there are few changes in systems with fixed hosts (and links). The communication between a mobile host and its access point has an asymmetric nature so as to reduce power consumption at the mobile host. This is also affected by a disparity of the bandwidth between wireless links and fixed links. Moreover, mobile hosts frequently operate disconnected from the rest of the network. Clearly then, mobile hosts and fixed hosts should be modeled as two distinct computing entities. The fixed network, which was designed assuming a static view of network connectivity, is augmented with mobility agents that act as access points for the mobile hosts.

Figure 2.1 shows a system model for mobile computing. This is based on the architecture developed in previous work [4, 33, 56]. The system model consists of a set of mobility entities: *Mobile Hosts* (MHs), *Mobility Agents* (MAs) and *wireless*

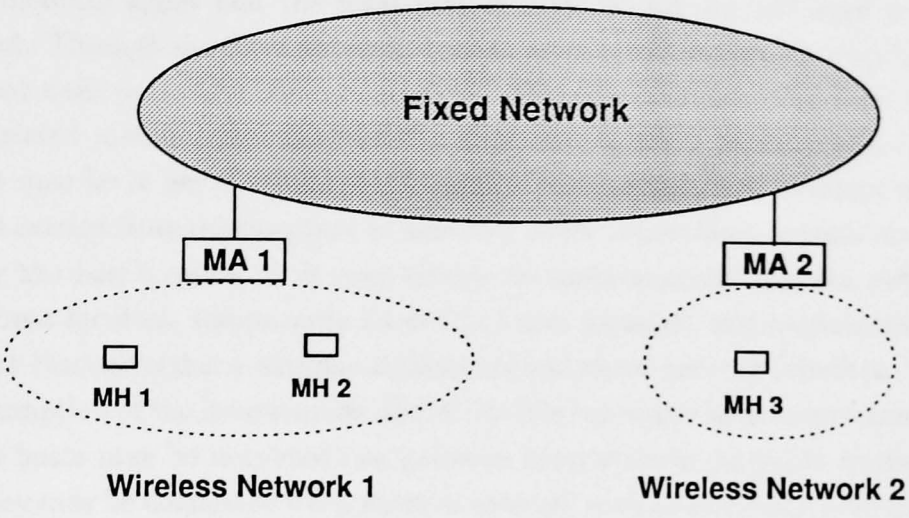


Figure 2.1: System Model for Mobile Computing

networks, in addition to the existing network entities. A mobile host is a host that can change its point of attachment from one subnetwork to another, even while retaining its network connections. The infrastructure machines that communicate directly with the mobile hosts are called mobility agents ("base stations" in a cellular network). They provide a wireless communication link between mobile hosts and the rest of the network. A *cell* is a logical or geographical coverage area serviced by a mobility agent. A mobile host can directly communicate with a mobility agent (and vice versa) only if the mobile host is physically located within the cell serviced by the mobility agent. At any time instant, a mobile host logically belongs to only one cell.

The mobile hosts rely on the mobility agents to maintain their addressability. All mobile hosts are identified with a particular mobility agent as belonging to its cell, and are considered to be local to that mobility agent. Additionally, a mobility agent has responsibility for keeping track of the addresses of hosts which are currently residing in the area it co-ordinates. These addresses may be stored as exact locations (the identifier of a cell the host is currently in) or as approximate locations. Each mobile host will be permanently registered with a mobility agent on its home subnet. A host may also register as a visitor with some other mobility agent.

Each mobility agent and the local mobile hosts within its cell form a *wireless network*. Through the fixed network, mobile hosts can establish a connection from different data ports at different locations. Wireless connection enables virtually unrestricted mobility and connectivity from any location within radio coverage. A host may be in use continuously through a wireless network interface when the host is carried from one location to another, so its connections remain unchanged during the host's move; or it may simply be disconnected from the network at its current location, temporarily moved to a new location, and reconnected to the network through either a wireless or conventional wired network interface. There is no assumption of the hardware details for mobile hosts and their connection media; mobile hosts may be notebooks or palmtop computers or portable workstations, and they may be connected via a wireless network such as infrared, radio frequency or a wired-network such as leased line and Ethernet.

In summary, the model for mobile computing thus consists of a *static/fixed* network comprising of the fixed hosts (including mobility agents) and the communication paths between them, and a *wireless* network associated with each fixed host (mobility agent) for communicating with the mobile hosts located within its cell. Host mobility is represented in this model as migration of mobile hosts between cells.

2.1.2 Built-in Characteristics

Mobile computing introduces a new set of issues that were not present in distributed systems with static hosts. These are mostly due to the host mobility, and to the hardware characteristics of mobile hosts and wireless networks. Thus, host mobility feature should be actively absorbed into the mobility support system, whilst hardware features may be selectively considered (resolved) by the other dimensions, such as an application. From the system design point of view, building a mobile computing system is much like building a computing infrastructure supporting host mobility. Host mobility is really the most outstanding feature. As a result, the mobile computing environment has several intrinsic characteristics as follow.

Location

In order to communicate with any particular host, it is first necessary to *locate*

the host in the network. This is due to the fact that the hosts are mobile and could be anywhere. Location is the most prominent feature attendant upon host mobility. Returning to Figure 2.1, to send a message from a mobile host MH 1 to another mobile host MH 3, MH 1 first transmits the message to its local mobility agent, MA 1, over the wireless network. MA 1 then forwards it to the local mobility agent, MA 3, of MH 3, via the fixed network. Finally, MA 3 locally forwards it to MH 3. Because a mobile host's addressability relies on its local mobility agents, in order to deliver a message, the source host, MH 1, needs first to locate the mobility agent that currently serves the moving destination, MH 3 (which could possibly move to other mobility agent's service area). Therefore, the location problem encompasses two supplementary (but closely interrelated) issues, *addressing* and *routing.*

Additionally, if the call is in progress, as the user moves from one mobility agent to another, a new frequency is assigned at the new mobility agent. The call continues to proceed using this new frequency. This process of transition between two frequencies is called the *handoff.* Basically, the handoff procedure relies on a protocol, when each mobility agent detects that a new mobile host has moved into a cell. One such protocol is the *beacon* protocol. Beaconing has two purposes; when a mobile host receives a new beacon, it knows 1) that it has entered a new cell and 2) which cell network number to use here.

Wireless medium

The wireless medium allows a mobility agent to communicate with all the mobile hosts located in its cell with a single message transmission, by *broadcasting.* The cost of such a message is independent of the number of recipients within a cell. However, because the composition of the wireless network changes dynamically as mobile hosts enter and leave the cell, it must pay special attention to utilizing the broadcasting feature. Further, host mobility is a behavior which has effects both within the fixed network as well as the wireless network; these two networks are quite different in terms of bandwidth (including error rate) and mechanism. The peer-to-peer paradigm is the basis for communication in the wired network, whilst it is broadcasting in the wireless network. At present, it is known that the wired backbone networks are faster than the wireless links by hundreds to thousands of times [41]. The cost of sending a message on the wireless and wired

portion of the network should be considered differently. Moreover, transmission of a message from a mobile host consumes more power than reception. As a result, communication within a cell needs to be *asymmetric* to reduce power consumption at the mobile hosts and better exploit the broadcast capability of the medium.

Disconnection

The ability of mobile hosts to operate while on the move requires a stand-alone source of power such as batteries. Given the limited lifetime of batteries, power consumption is a serious practical consideration at a mobile host, unlike a fixed host. Because of the geographical structure of wireless networks, mobile hosts may often disconnect from the rest of the network during moving procedure. A mobile host that disconnects in the midst of an algorithm execution may cause the execution to suspend till its reconnection. Also, the host may reconnect with a network which could be different to the one it disconnected from.

Disconnection in a mobile environment is distinct from failure. Because disconnection is voluntary, a mobile host can inform the system of an impending disconnection prior to its occurrence and execute a disconnection protocol. So, disconnections can be expected to be a regular feature of the mobile environment. Another novel operating mode of mobile hosts with the aim of reducing power consumptions is the *doze mode*. In this mode, the clock speed is reduced and no user computations are performed; instead, a mobile host simply waits passively to receive any message. If such a message is received, it resumes its regular mode of operation.

Logical structure

Many distributed algorithms depend on an underlying logical structure amongst the participants to carry out the needed communication. The main purpose of such a structure is to provide a certain degree of order and predictability. Messages exchanged within such structures follow only selected logical paths. Mobility implies that a host's location relative to the rest of the network changes with time; the connectivity of the entire network is thus modified as hosts move. So, a logical link between two mobile hosts can no longer be mapped to a fixed sequence of physical links in the underlying network. The implication of a logical link needs to be reassessed for mobile hosts; the physical connections comprising a logical struc-

ture amongst mobile hosts need to be reconfigured whenever such a host changes its location.

2.1.3 Mobile Applications

Many people believe that host mobility will not be the exception in the future; rather, it will be the norm in real life. Exactly when this will happen depends on when the infrastructure for host mobility becomes widespread. Then, what applications put pressure this trend? A workshop on mobile computing systems and applications, hold in Santa Cruz, December 1994, dealt with some of these questions [64]. As many people have done, Bob O'Hara from Microsoft observed that vertically integrated applications, such as appointment books, mail (or news) services, tended to be the most successful. Murray Mazer from OSF Research Institute suggested that another class of applications – remote information services – was going to be the fastest growing and stimulate more collaborative forms of computing. One example is a local yellow pages possibly extended with online information, such as movies currently playing at local theaters. Here, location dependent data will play a significant role in selecting relevant information (closest hospital etc). On the other hand, Marvin Theimer from Xerox PARC offered the opinion that entertainment (including games such as multi-user Doom) would be the driving force of mobile computing.

From the mobile users' point of view, two factors – interface and transparency – are very important for their perspective of mobile computing. Host mobility sometimes results in a very poor environment for applications. Users may not be prepared to tolerate bad interfaces. They will also not accept the poor performance and unannounced missing functionality, which comes with host mobility; that is, performance transparency and location transparency. Some people suppose that total transparency is never going to be possible, and that users are not expecting it anyway. Nevertheless, mobile users have to be allowed to use the stable functionality as far as possible, without any annoyance due to its movement. As an idea of the current status of mobile application, the following shows some of the products presented in [64].

IBM Mobile FileSync

This new IBM product had been inspired by the Coda file system, but differs considerably in its detailed design. It supports disconnected file access in OS/2. The support is entirely at the client end, with no changes required to existing servers. The current version of *Mobile FileSync* provides support for hoarding, as well as for step-by-step reintegration via an interactive process. These functionalities are layered entirely above the file system switch; therefore, the support for disconnected operation works with any file system below the switch.

Lotus Notes

Notes is mainly concentrated on databases for mobile computing, so as to cope with disconnection with the replication model. A client can connect to the network and obtain a replica from a server. Once a replica is downloaded, it can be disconnected from its server, so goes off-line. Considerable effort is made to hide whether you are on-line or off-line, but user control is possible via a sequence of menus. There is a full scripting language for creating filters, so that only desired information is collected from the server in any given connection.

PARC Tab

PARC Tab has a server process running on its behalf on a workstation on the wired network. Applications on a Tab can be implemented as Tcl scripts that are executed on the server. Current applications are for that of "proximate selection". One example consists of a user walking into a cell, and selecting "forward call" on his Tab: his phone calls are automatically forwarded to the room he is in. Another example consists of an application to list available printers, with nearest first: when the user walks to a different room, the display automatically changes.

Teleporting

This system was developed at the Olivetti Research Laboratory to gain experience of mobile applications. It enables the display of an application to follow a user around as he moves, leaving program execution at the original site. This ability is especially convenient when combined with an active badge system that tracks user location. If a user visits some place and presses their active badge buttons, it allows the user to interact with her existing X applications, by having X-displays migrated to the current location.

2.2 Protocols for Internetwork Host Mobility

The previous section shows that mobile computing is based on the premise of host mobility, and that the location problem for identifying moving hosts is a major part of the problem of providing seamless connectivity to the hosts. Also, the location problem includes the addressing issue for moving hosts and a related issue – effective packet routing. During the past few years, various proposals have been made for supporting host mobility on datagram-based internetworks [7, 8, 13, 17, 33, 37, 39, 43, 53, 55, 56, 61, 70, 73]. Most of these proposals have been designed to be compatible with the TCP/IP-based Internet due to its popularity; with a revised IP protocol, which is based on the fact that it is up to the network layer to be aware of host location in the interconnected networks.

In an IP-based mobile computing system, mobile hosts cannot interoperate easily because of IP's addresses and routing algorithms. An IP address consists of two parts: a *network number* that identifies the network to which the host is attached, and a *host number* that identifies the given host within that network. IP datagrams are routed to the destination based on the network number. Thus, if the destination moves in a way that necessitates changing the network part of its address, current IP routing mechanisms have no facilities for tracking the move and having the packets follow the host. In addition, changing the IP address of the host whenever it moves is difficult (or impossible while keeping existing transport-level connections open). Therefore, a solution is required for correctly routing datagrams to the host in its current location given the host's home (constant) IP address. This problem is in general also not unique to IP, since any packet routing protocol which usually uses a hierarchical addressing scheme based on network topology (or geography) faces similar problems in trying to integrate mobile hosts into the network.

A number of proposals have been made for resolving the IP addressing problem: a separation of the dual nature of an IP address into a logical identifier which is the permanent (home) IP address of the host, and a physical locator which is a forwarding (current) IP address, and a mechanism to forward packets to the mobile host's current location. In an IP-based network, two ways of forwarding packets to a moving host are known: source routing using the IP option (loose source routing), and encapsulation using a protocol packet inside each IP packet. It is

generally agreed in the research community that packet forwarding is best managed by some form of *tunneling*[1] between the source and the destination, based on encapsulation. The major differences between these proposals are the way location information is propagated and the place location information is maintained in order to trace a moving host. The below describes some previous work, with emphasis on the treatment of addressing, location and routing issues.

Mobile*IP

As an initial inspiration and experimentation which gave an impetus to begin the most recent round of IP host mobility development, Ioannidis *et al.* of Columbia University proposed a scheme for mobile internetworking, named Mobile*IP, in 1991 [33, 34]. In addition to mobile hosts, a MSR (Mobile Support Router), which acts as a gateway between a wired network and a radio cell, is added to conventional IP networks. A mobile host moves amongst such wireless cells served by MSRs. A mobile host retains the same Internet address even if the host moves to another subnetwork. All mobile hosts have IP addresses with the same network and subnetwork numbers; so these logically form a single subnetwork, called a *mobile subnetwork* although they are physically discrete. MSRs advertise route information to the mobile subnetwork on the wired network. This is based on the so-called "Embedded network" approach [20].

MSRs are responsible for forwarding traffic to and from the mobile host; there is one MSR per wireless cell, and at least one MSR per mobile subnetwork. When sending a datagram to a mobile host, a mobile host routes the datagram to the MSR that serves the wireless cell. That MSR directly delivers the datagram if the destination host is within its service boundary. If an MSR does not know which MSR is currently responsible for a destination, it sends (broadcasts) location search queries to all other MSRs of the mobile subnetwork. After getting a response from the one that is actually serving the host, datagrams are then tunneled, using an encapsulation protocol, from the MSR which received them to the MSR handling the host, decapsulated at the remote end of the tunnel, and eventually delivered to the destination. The area to which an MSR transmits

[1]Tunneling is a technique for passing packets from one part of a network to another, when the in-between routers do not know how to route the packet. This is usually accomplished by adding information to the packets so that they can traverse the part of the network that cannot properly route them.

location search queries is called a *campus*. Broadcasting, the location strategy for intra-campus mobility, is too expensive for frequent use in a large network; the campus must be small enough so that the location cost is not excessive.

If a mobile host roams away from current campus internetwork and appears in the foreign campus, it is assigned a temporary IP address, its nonce address, in addition to its original IP address. The host then notifies the nonce address to an MSR in its campus. Such an MSR is called a *designated* MSR. Because the host is now in a different campus domain, the designated MSR need not communicate with the MSRs in the host's home campus but acts as a member of the current campus only. All traffic destined for this mobile host will naturally be routed to its home network, and tunneled from there to its designated MSR using the nonce address of the host as the remote endpoint. The designated MSR delivers the datagrams using the same procedure as the intra-campus one. The work of Ioannidis *et al.* is designed primarily to support mobility within one campus; inter-campus mobility is treated as a special case.

VIP

At the same time as Mobile*IP, Teraoka *et al.* of Sony Research Labs proposed a mobility scheme known as Virtual Internet Protocol (VIP) [69, 70]. In contrast to the actual (or physical) network, a concept of *virtual network* is introduced to enable host mobility in the Internet. Each host is connected to a virtual network just as it is connected to a physical network; it never migrates in the virtual network even if it migrates in the physical network. Virtual networks are logically constructed above the physical network by assigning two different IP address to each host; a physical IP address and a virtual IP address. The IP layer then is split into two sublayers; physical IP sublayer and virtual IP sublayer. Only the virtual IP addresses are visible from the transport layer (and higher layer level protocols as the endpoint identifier), whilst the physical IP addresses are always local to the subnet the host is connected to, and are used for packet routing. A packet sent by a mobile host that is away from its home subnetwork carries both addresses; the physical IP source and destination addresses are conveyed in the conventional IP header, whilst the virtual ones are carried either as an encapsulated format or as an IP option.

Location in this scheme can be done with an address conversion from physical to

virtual or *vice versa.* To do this, all routers have a location cache, called an *Address Mapping Table* (AMT). Whenever a mobile host connects to a new subnetwork, it acquires a new physical address on that subnetwork and sends a connect control packet to its home subnetwork. The in-between routers which the control packets pass through peek into the packet header and create or update the AMT entry for the host. Likewise, a new address for a migrated host is propagated into AMTs when data packets built with the VIP option to/from the host pass through. A transit router that processes a packet holds a cache entry for the target host, and if necessary, the packet is reformatted to contain a new destination host's physical address. In the worst case, the packet sent to the migrated host is forwarded by its home subnetwork.

When a mobile host is about to disconnect, it sends a disconnect control packet to the home subnetwork, which will also broadcast a disconnect control packet on all connected subnetworks. Any VIP-capable router that receives such a packet and has a cache entry for the referenced mobile host deletes the cache entry and propagates the broadcast. Some non-VIP hosts or routers stop control packets reaching the corresponding AMT entries; some of those therefore may have an out-of-date cache entry. Later work [72] tried to improve this problem, but results in a flood of control packets, and has not solved all of the cases to do with faulty hosts or routers. In addition, the size of location caches (AMTs) is in proportion to the number of host movements, so there is scalability problem. This scheme also has a fatal deficiency in terms of compatibility with existing IP networks.

Multiple Address Approach

Wada *et al.* of Matsushita Electronics proposed a scheme based on multiple addresses [73]. As in the Sony work, each mobile host is assigned a temporary IP address whenever it moves to a new cell or subnetwork that is distinguished as its home network. The mobile host retains its home address regardless of migration. Packets are sent to a mobile host specifying its home address. Each home cell (or subnetwork) has at least one special router, called the *Packet Forwarding Server* (PFS). The PFS is responsible for tracking the temporary IP addresses between ones at the time visiting this subnetwork (or its home address) and new ones at the current subnetwork of mobile hosts. The new temporary address for a mobile host is transmitted from the host itself to its home PFS. The home PFS is then

responsible for propagating the address to all hosts or routers concerned, such as the previous PFSs which have been left by the host. It is not necessary for new temporary addresses a PFS maintains to be the latest ones.

A packet bound for a mobile host is routed to the host's home PFS. The PFS is promiscuously listening on the subnetwork, it intercepts any packets for that host, encapsulates them, and forwards them using the host's current temporary address that it maintains. Clearly, this forwarding scheme is very inefficient in a large network like the Internet due to long chains of forwarding routes. To avoid this problem, an autonomous mode is introduced to allow two hosts to communicate in a normal Internet way, by caching the location information (mobile host's current temporary address) on the sending host. Upon forwarding a packet to the other subnetwork, a PFS returns a location notification packet to the source host. Packet encapsulation then is done by the sender itself.

In this scheme, the notification packets might flood the network in proportion to the number of host moves. The size of the location cache of a host can grow without bounds. In addition, cache control for a PFS or a host is difficult because they have an out-of-date cache entry. That is, a PFS cannot delete a cache entry for a mobile host until it learns that there are no hosts or PFSs which hold the old temporary addresses of the host. If it has discarded the cache entry for the host, some packets bounds for the host may lose their route.

IP Option Approach

Perkins [7, 55, 61] of IBM developed a mobility support scheme using IP's Loose Source and Record Route (LSRR) option. At the same time, Johnson [37] of Carnegie Mellon University independently proposed a similar idea. In addition to mobile hosts, two mobility entities, Mobile Access Stations (MASs) and Mobile Routers (MRs), are added to the conventional IP network. Mobile hosts belong to only one mobile subnetwork (so MAS) at given time, and one or more mobile subnetworks go with one MR. A mobile host is assigned an IP address, and the address remains the same regardless of the host's current location. MR is responsible for keeping track of the current location of each mobile host that has been assigned an address on that subnetwork, and for advertising reachability for those mobile hosts (so MR is much like the PFS in the Matsushita work). When a mobile host moves into new cell, it informs its MR of the Internet address of the

current MAS.

When a host is away from its home subnetwork, a datagram sent to a mobile host will initially end up at its MR. The MR will try to forward them to the host's current location, it then adds an LSRR option to the datagram. When the mobile host replies to a correspondent, it also inserts a LSRR option in the outgoing datagram that specifies the address of its current MAS as transit router. When the corresponding host receives the datagram, it will reverse the recorded route on the datagram, and insert it as a LSRR option in future datagrams sent to the mobile host; such datagrams will be routed via an optimal path, without visiting the target host's MR again. If a mobile host switchs cells, the new source route is supposed to replace the old one. Despite its cleverness in support of location and routing for host mobility in the large internetwork, this scheme requires more complicated protocols than existing implementations of the LSRR IP option. In addition, the basic LSRR specification is not properly implemented in most current hosts to handle the above, also it is implemented by only a few hosts today. Routing would be always sub-optimal for UDP traffic [34, 52].

IP Mobility Support

The *Mobile-IP* (IP Routing for Wireless/Mobile Hosts) working group of the Internet Engineering Task Force (IETF) has been developing (nearly completed as of August 1995) a protocol recommendation that allows transparent routing of IP datagrams to mobile nodes in the Internet [56]. Each mobile host is identified by its home address, which is immutable regardless of its current location. When a host has moved away from its home, it is associated with a *care-of* address, which provides location information about its current point of attachment. The care-of address is either assigned to the mobile host or associated with a foreign agent, which is responsible for providing access to visiting hosts. When away from its home, the mobile host registers its care-of address with a home agent; the home agent is responsible for intercepting datagrams addressed to the host's home address and tunneling them to the associated care-of address.

When a mobile host arrives at a new location, it can listen for (or solicit) agent advertisements to determine whether a foreign agent is available. If so, the registration request to the home agent is sent via the foreign agent; otherwise, the mobile host must somehow acquire a care-of address, then directly register with

the home agent. All datagrams addressed to a mobile host are routed via the home agent; this makes the system suffer a fatal problem, that is, performance transparency, with inefficient routing[2]. The recommendation also describes various security considerations for the registration protocols, and a minimal encapsulation method for un-fragmented datagrams.

In summary, Table 2.1 shows a comparison of the previous work for supporting host mobility, in terms of addressing, locating and routing. Each of these proposals has a different characteristics in the system point of view, some of which are discussed in [52].

Criteria	Mobile*IP	VIP	Multiple Address	IP Option	IP Mobility Support
Addressing	Embedded	Temporary	Temporary	Permanent	Permanent
- Logical ID.	Home Add.	Home Add.	Home Add.	Home Add.	Home Add.
- Physical Locator	Embedded Add.	Temporary Add.	Temporary Add.	Current MA Add.	Current MA Add.
Locating	Broadcast	Location Cache (Propagating)	Forwarding Pointer (Notification)	IP Protocol	Forwarding Pointer
- Caches	None	Intermediate Routers	A Server in Home Area	Mobility Router	Home MA
Routing (Tunneling)	Mobile Host	Intermediate Routers	Forwarding Server (Mobile Host)	Mobility Router (Mobile Host)	Home MA

Table 2.1: A Comparison of Mobility Support Systems

[2]Apart from the IP mobility support, Mobile-IP group has been working for a routing optimization extension on the basic mobile IP protocol [40].

2.3 Location and Routing Optimization

The key service for providing seamless connectivity to mobile hosts is the creation and maintenance of a packet forwarding tunnel between a known location (possibly through the home agent, or by querying a location directory) and the host's current agent. Clearly, packet routing paths going with host mobility depend critically on where and/or how much location information is preserved on the network as a whole. For example, let us suppose a mobile host is visiting some subnet. Even packets from a source host on this same subnet must be routed through the Internet to the mobile host's home agent on its home subnet, only then to be tunneled back for delivery to the mobile host's current foreign agent. This causes significant delay in delivering the packets, and an unnecessary burden on the networks and routers along this path. We will describe this in more detail in subsection 3.2.2. Even though the location and routing method is of such significance for handling host mobility in the internetworking environment, little work has been devoted to improving their effectiveness.

Two proposals address the location problem but do not consider implementation details. Awerbuch *et al.* of MIT University proposes a formal model for the tracking of mobile users, named *regional directories*, where each directory is based on a hierarchical decomposition of the network into regions [1]. This approach reflects spatial locality in that frequently-accessed directories are cheaper to access than far away ones. Two operations, *find* and *move*, are defined for host location. If a mobile host processes a move operation from a new location at distance d away, only the $log\ (d)$ lowest levels of the hierarchy of regional directories are updated to point directly to the new location. Directories at the higher levels continue to point to the old location. In order to provide access for some hosts that use those remote directories, a forwarding pointer is left at the old location. This directs the find operation. Nearby hosts from the host must locate it by inspecting their local directories whilst hosts under remote directories have to use higher level directories which may have the old location and hence have to use the forwarding pointers on the host's previous location. This work shows a theoretical examination of the two operations' cost. One valuable result of this work is a formal proof that caching of localized mobility is very helpful for tracking mobile destinations. This is one reason supporting the present work. Another result is

that there is always a tradeoff between optimizing ease of tracking versus ease of searching. It should be applicable to most location strategies. This theoretical work is most of interest to research and commercial efforts directed at tracking mobile entities.

Imielinski *et al.*'s proposal of Rutgers University relies on the ***mobility profile*** which reflects the host's mobility pattern [31, 32]. Partitions are defined depending on the user profile for each host by grouping the cells (or location servers) amongst which the host moves frequently and by separating the cells (or location servers) between which the host relocates infrequently. Whenever a mobile host moves, it tells its new location to base stations within its partition. Three strategies can be used to locate a host within a partition: broadcast, lists and forwarding pointers. If a host crosses partitions, it tells its current partition to the outside world. Sources within the destination's partition can directly locate its current position whilst sources outside the destination's partition must utilize a certain degree of knowledge about its whereabouts (so the destination would be somewhere in a given partition). The partition concept increases the move cost, but on the other hand, it decreases the location search cost; clearly this is a tradeoff, which depends on the host's mobility pattern. This scheme is conceptually quite similar to Awerbuch's work, in terms of utilizing the locality property of host mobility and of taking advantage of *incomplete information*, even if this is user-oriented for location and the former is system-oriented. However, these works do not consider the impact of temporal locality on the calling pattern. Neither of these schemes above deals with routing efficiency in conjunction with location scheme.

As an orthogonal approach to the above, two systems have been proposed that stress routing efficiency. Dupont of the Motorola Wireless Data Group presents a *local mobility management* concept that limits the scope and frequency of location updates for a moving host within local area boundaries [24]. A designated router, called a local mobility router, acts as a local directory to isolate local mobility events from the rest of the world; a host informs only the router about its local moves. The home location directory also does not always need to know the precise location of a host (that is, it may know the address of the local mobility router in which the host is currently located). The router then acts as a bridge to tunnel any incoming datagrams which have incorrect addresses for the destination. These local mobility routers are extended to a hierarchical mobility management model,

which is analogous to Awerbuch's hierarchical model. This concept provides only partial solutions; any local moves are hidden from others, but datagrams are firstly forwarded by a router on the home area.

Yuan of NEC proposes a concept, called a *friend network* to accommodate the host calling pattern for effective routing information propagation [75]. A friend network is a special set of networks which generates the majority of traffic for a mobile host, so may need to communicate with the mobile host in the future. For any movement of a host, new location information is propagated to those hosts located in the home network and friend networks; so datagrams generated from those hosts are routed directly. This system also provides partial and conceptual solutions: it does not consider the host moving pattern, which is a very important parameter in the host mobility environment.

Stressing the practical aspect, some work [8, 17, 40, 53] has concentrated on location and routing optimization. These schemes all have similar basic mobility support schemes, mainly borrowed from the IETF work [56]. The work from Blaskwell *et al.* of Harvard University [8] and that from Johnson *et al.* [40, 53] share the same idea. These schemes are based on the location notification message whenever one entity determines that another entity might have incorrect (or empty) location information for a mobile host (the location notification idea is originated from Wada *et al.*'s work [73], as described in the previous section). Their notification procedure details differ in which entity takes charge of propagating the new location, and which entities will cache the location information.

For both schemes, a mobile host is assigned a home IP address, and can be reached via the home address (agent) regardless of its current location. In Blaskwell's approach, when the correspondent host sends its first packet to a mobile host, which is away from its home area, the packet should be forwarded via its home agent. The agent then informs the sender the fact that the host is mobile. The correspondent host asks the home agent to keep it informed of the mobile host's location. The home agent remembers all correspondents that have subscribed to mobile host location updates. So long as this subscription is maintained, the home agent informs the correspondent hosts of the mobile host's current agent address each time the mobile host registers a new location. The correspondent caches the location updates from the mobile host's home agent, installs the appropriate

routes in its IP routing table, and thereafter encapsulates packets bound for the mobile host directly to its current foreign agent.

Johnson's system uses lazy notifications to inform other nodes that a mobile host's binding has changed. If a network entity receives a packet that it must tunnel to some mobile host, it is likely that the source node of the packet has an incorrect binding (so the packet has been tunneled to this node) or no binding for the destination host (in the case of a normal packet). In either case, if this entity determines that a new binding might improve packet routing, that is, the tunneling on this entity makes for an unnecessarily long route for the packet, it then may send a binding notification to the source node of the packet. Here, the location notification is issued not only by the home agent but also by the previous agent. As in Blaskwell's work, a correspondent host acts as a cache agent. Recent work [53] also permits any intermediate agents to function as a cache agent. When an intermediate cache agent snoops on the notification, this cache agent can use the notification to acquire a binding for the mobile host. If a packet passes through the intermediate cache agent which has a location cache entry for this packet's destination, then the cache agent should tunnel the packet to the mobile host's current location.

Because a mobile host continues to move around, cache agents end up with out-of-date cache entries for the host. If a cache agent or local agent receives a packet that was tunneled directly to this node but the agent is unable to forward the packet (no location cache entry or visitor list entry for that packet) or it finds a routing loop, it should then tunnel the packet to the host's home agent. It is regarded as a *special tunnel*; the packet must be routed using only normal IP routing. This notification concept could improve routing efficiency, but there are several problems with this approach:

1. The network could be flooded with location notification messages. Even if later work introduces a back-off mechanism, it is in proportion to the product of the number of host movements and the number of calls and even the number of cache agents, for each mobile host.

2. The location cache on each correspondent host can grow too big. It is somewhat analogous to the number of destination hosts to which the correspon-

dent has made calls. This situation is the same for the cache agent; where the size is proportional to the number of mobile hosts it has served.

3. The first packet, which issues the location notification, is always tunneled through an sub-optimal route. Depending on the calling and moving discipline, many notifications are not actually (or practically) utilized for the packet routing.

4. As moves and calls progress, cache agents are apt to be out-of-date; all previous agents for a mobile host have to be notified of its new location. If not, the forwarding path would bring long chains - these make rather long routes[3]

Nonetheless, the work is an interesting practical attempt at solving the location problem in conjunction with routing efficiency. They were developed at roughly the same time as the LROP protocol, which is presented in this dissertation (see early work in [17]). Their main contribution is the concept of need-based location notification to the correspondent; also important is their emphasis on routing efficiency being a crucial factor in designing the location scheme in terms of the system's effectiveness. In chapter 6, we will compare Johnson's approach with our scheme, using some simulation results.

2.4 Summary

This chapter presented details of mobile computing and related previous work. In fact, mobile computing can be regarded as a host mobility extension of distributed computing, with new mobility entities, such as mobile hosts, mobility agents and wireless networks. A mobile host may change its location relative to the rest of the network with time; it brings several intrinsic characteristics which affect the system design. Amongst them, location of moving hosts is the most prominent feature. In practice, the host mobility problem can be formalized with the problem of maintaining location information for identifying moving hosts. From a technical

[3]In the Johnson *et al.* work, a timeout scheme plays a part in deleting these out-of-date cache entries, but the problem still remains.

point of view, the location problem includes the addressing issue for moving hosts and a closely related issue, the effective packet routing for moving destinations.

It is generally known that addressing problems can be resolved by making use of two different addresses: a logical identifier and a physical locator. A mobile host then maintains the identifier as it moves, but changes its physical locator, because the IP protocol relies on the entire IP address. The location issue should be resolved with the revised protocols that integrate mobile hosts into the traditional networking infrastructure, as most previous work has done. To do this, most common approaches cache the host's new location, i.e., a reference to the new location is deposited somewhere, in the known places [33, 56, 73], mostly the home agent, or the unknown places as well [38, 53, 70], such as in-between routers or mobile hosts. When a cache agent receives packets and has a location cache entry for the packet destination, a tunneling protocol will forward them to the packet's current location.

However, the location cache tends to have a tradeoff between location efficiency and routing efficiency. If the location propagation is too limited, most packets may be tunneled with a default reliance such as destination's home agent, which make for an unnecessarily long route. On the other hand, if a location scheme has excessive location caches or updates, it may flood the system with location updates. This situation eventually results in location and/or performance transparency problems to the mobile computing framework. This thesis will deal with how this issue can be effectively managed from the system performance point of view. At the same time as our work was undertaken, some efforts [8, 40, 53] was directed at location and routing optimization. Their works share the same idea; the location notification scheme whenever one entity determines that another entity might have an incorrect (or empty) location information for a mobile host, or that a new binding might improve packet routing. This approach is based only on the expectation that a host which has sent a packet will send to the same destination again, however, it ignores the fact that the destination keeps moving.

Chapter 3

Approaches to Location and Routing Optimization

In the previous chapter, we examined a number of mobile computing issues, which varied in terms of features, applications and technical approaches. This chapter presents our approach to location and routing optimization for providing internetwork host mobility. The first section shows the Internet issues involved in embodying host mobility. In the next section, we shall be looking again at what has to be considered in a location and routing optimization scheme from the networking perspective. The third section of this chapter exploits the locality property of the host movement and calling pattern as a basis for our elaboration, and then presents two concepts, *local region* and *patron host*. In each description, some design issues are considered from the protocol point of view.

3.1 Host Mobility in the Internet

The Internet is the collection of networks and gateways that use the TCP/IP protocol suite and function as a single, cooperative virtual network. It has grown to become a major component of the network infrastructure. As of the middle of 1994, the Internet consists of over 31,000 networks, with one new network being added to the system approximately every 10 minutes. The number of computers connected through the Internet exceeds two million, but by how much is unknown

due to the incredible rate of growth. Over 20 million people can be reached by electronic mail and have access to resources via the Internet. Monthly traffic on the U.S. NSF backbone alone is about 10 terabytes [44]. Moreover, the increasing commercial uses of the Internet on a profit-making basis is likely to create even faster growth in the future. The Internet protocol suite is now referred to as the standard for computer communications. As the Internet continues to grow, so does the larger global Internet: the set of networks using multiple network technologies that can intercommunicate. Increasingly, the TCP/IP Internet has provided the glue for this larger infrastructure.

Wireless communication systems, mostly for voice applications, have progressed tremendously in the last decade. However, it is unreasonable to assume that in the future all, or even most, communications will be wireless. The wired mesh of existing networks is likely to continue to exist and even expand. Portable hosts exchange data among themselves and with existing non-portable services such as databases, file servers, and printers. As in the case of their wired counterparts, users of the wireless network need to communicate with peers beyond their immediate locale, which generates the need to incorporate the wireless network within the larger Internet. Hence, the wireless networks should be considered as members of the ever-growing number of networks. This in turn means that wired internetworking should be the basis of the design of the wireless networks; naturally, it calls for the use of the Internet protocol suite as the foundation of the wireless environment.

3.1.1 Internetworking

An *internetwork* is a collection of networks interconnected by routers along with protocols that allow them to function logically as a single, virtual network. The term "the Internet" refers specifically to the connected internetwork which uses the TCP/IP protocols. Like the ISO (International Organization for Standardization) OSI (Open Systems Interconnection) reference model [36], the TCP/IP protocol suite uses the layering principle. The protocol is organized into four conceptual layers that were built on a fifth layer of hardware [21]. It is a central concept of layered protocols that layer n at the destination receives exactly the same object sent by layer n at the source. Machines are only physically connected

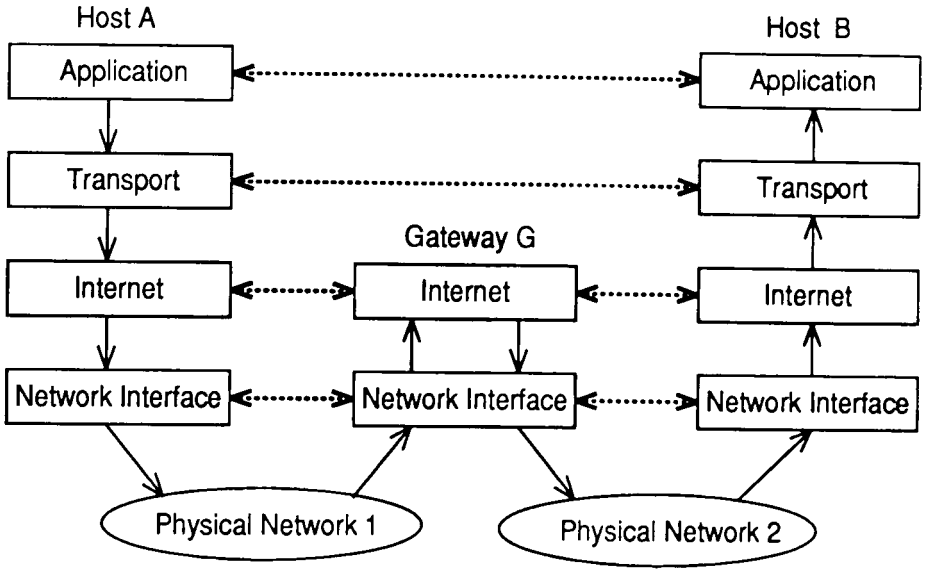


Figure 3.1: An IP Gateway Internetworking

at the physical layer, but each layer behaves as if it were directly connected to the corresponding layer of its peer machine. Figure 3.1 depicts an internetworking with the 4 conceptual layers. The following briefly describes the four layers.

At the highest layer, users invoke application software that accesses services which are available Internet wide. An application interacts with the transport layer protocol(s) to send or receive data; it first has to choose the style of transport needed, which can be either a sequence of individual messages or a continuous stream of bytes. The application layer includes a couple of typical computer communication programs, such as *telnet* and *ftp* on top of TCP, and *RPC (Remote Procedure Call)* protocol on UDP.

The primary duty of the transport layer is to provide end-to-end operation from one application program to another. This layer may regulate the flow of data. It may also provide reliable transport, ensuring that data arrives without error and in sequence. Multiplexing independent virtual connections between multiple application programs accessing the Internet at the same time over the same logical link is also a function of this layer. The transport layer consists of two different protocols, TCP (Transmission Control Protocol) and UDP (User Data-

gram Protocol). TCP provides the reliable bidirectional data stream service. It is connection-oriented in the sense that before transmitting data, participants must establish a connection. UDP provides unreliable datagram delivery. It essentially serves connectionless-mode communication services.

The internet layer handles communication from one machine to another across the underlying Internet. To do this, the internet layer provides naming, addressing, and routing functions. It accepts a request to send a packet from the transport layer along with an identification of the machine to which the packet should be sent. It is also responsible for providing the path that packets should follow when going from one machine to another across the underlying Internet by maintaining routing information and using the addresses to decide how to forward packets. Internet Protocol (IP) is a typical internet layer software. It defines the IP datagram as the unit of information passed and provides the basis for a connectionless, best-effort packet delivery service which means that it is allowed to drop packets with impunity, whilst making an earnest attempt to deliver packets.

As the lowest layer, the network interface layer is responsible for accepting IP datagrams and transmitting then over a specific network link, either a local area network to which the machine attaches directly or a network consisting of packet switches that communicate with hosts using HDLC (High level Data Link Control). It may use any of several subnetwork dependent protocols, such as Ethernet or FDDI (Fiber Distribution Data Interface).

Our interest is thus in how individual hosts are interconnected to form the Internet. Machines are connected through an Ethernet or point-to-point links to form a connected network (subnetwork or subnet); they share the same physical hardware. The simplest way to bring together the connected networks is through the use of a *repeater* or *bridge*. A repeater simply duplicates the electrical signals in order to increase the physical range of a network. On the other hand, a bridge can store and forward complete packets; it is a kind of router because it chooses whether to pass packets from one physical subnetwork to another. Even though these two permit the connected networks to be geographically large, they do not allow connection of networks with different physical hardware. They also use physical addresses, so propagating information about the location of each machine does not scale well.

An advanced way to interconnect any subnetworks is through a *gateway*. A router

is a general term which is applied to any special purpose machine responsible for making decisions about which of several paths network traffic will follow. It chooses a route for individual packets. Similarly, a gateway stands for any machine or device that connects two or more machines, especially if the machines use different protocols. It sits on the boundary between networks that are aware of the addressing and routing conventions in each of the networks they border.

When used with TCP/IP, the router refers specifically to an IP gateway (or simply gateway) that routes datagrams using IP destination addresses, and that allows machines with different network interface protocols, such as Ethernet, point-to-point and ATM, to communicate with a common IP protocol. Figure 3.1 shows a gateway's function in the 4 conceptual layers of TCP/IP protocol stack. In the layering view, a gateway provides the network layer connectivity. Datagrams pass from gateway to gateway until they reach one that can deliver the datagram to the destination host directly. In this thesis, the term gateway refers to an IP gateway, and it is used interchangeably with the term router.

3.1.2 Why the Internet Layer should take (and does) charge of Host Mobility

A mobile host is generally supposed to move around amongst the subnetworks, while retaining its network connections. In practice, host mobility is much like support for fault tolerance; it has to be built-in functionality in the network protocol. As shown in the previous chapter, the primary aim for mobile internetworking is to hide host mobility from applications, which never need to know about it and want nothing above the transport layer to be changed. Moving up the TCP/IP protocol stack, host mobility can be supported by one (or some combination) of the four conceptual layers. Therefore, it is very useful to examine the most appropriate layer to provide transparent host mobility.

If applications have to take care of host location, each time an application wants to communicate with another, it must obtain the current address of its peer. One possible way is to query the current location for the peer application (which may have moved) through a centralized name server or broadcasting mechanism. This is clearly impractical in large internetworks because the network will flood with

location queries. Moreover, it is impossible to provide on-line moving, which is a desirable feature in mobile computing, without modifying the application program itself. The peer application program would be required to know the moving host's current address for every transmission, in order to preserve existing communication channels.

The transport layer provides end-to-end operation from one application program to another. As the application program does, it also relies on the IP address for peer host identification; hence, the addresses of a pair of communicating hosts must remain constant for the lifetime of a transport connection. Therefore, the transport layer approach for host mobility suffers the same problem as the application layer solution. However, some special areas, for example, multimedia applications typically having strict constraints on delay, delay jitter and throughput (which rely on connection-oriented protocols and resource reservation), may be supported by a transport layer approach, in order to utilize the connection-oriented characteristics in support of host mobility [11, 12, 41, 74].

The network interface layer can also to handle host mobility. Here, any two subnets are connected with a bridge. [66] described a bridge-based scheme in which each bridge maintains a location cache. If a bridge finds a cache entry for the destination host of an ongoing packet, it forwards the packet to the destination. Otherwise, the logical structure of bridged networks (assumed to be a spanning tree) is utilized for broadcasting the packet. This scheme has several deficiencies, such as the dependency of routing on network structure, and immediate location notification. A bridge also inherently restricts the interconnected networks to having the same type of network interface protocol. Nonetheless, the network interface layer provides very useful features for managing host mobility: it can monitor the quality of the network interface service, including connection or disconnection to (from) the communication medium, and report the fact to higher level protocols. It would be desirable to use these features in other layers' approaches.

According to the TCP/IP conceptual layering architecture, host-to-host communication is an internet layer service, while end-to-end communication is the responsibility of transport layer. The transport layer and the higher layers do not see the notion of host and it is up to the internet layer to maintain host location in the network. The internet layer hides the different hardware addresses at the

network interface layer, and the exact location of a host on an internetwork. It can be thought of as two sublayers: the subnet layer, representing the internet-layer issues that occur in a connected network, such as physical (data-link) to logical (network) address mapping, and the internetwork layer, which is responsible for handling routing and internetworking.

Based on its addressing and routing capability, the internet layer could hide from higher protocols the fact that hosts move, *i.e.*, give to higher level protocols the abstraction that the network address remains unchanged. In conformance with that architecture, host mobility functionality should belong to the internet layer. Then, the problem that has to be resolved in this layer now consists of addressing the mobile hosts, and locating and routing to them effectively while maintaining their addressability as they move.

3.1.3 IP Address Structure

The internetworking environment should provide a universal communication service. To do this, it needs to establish a globally accepted method of identifying the hosts attached. In general, each host has a unique identifier by which it is reachable from any other host. Host identifiers are often classified as *names, addresses,* and *routes.* The terms, names, addresses, and routes, are three interrelated issues in every discussion of the internetworking. Perlman [54] suggests that a name is a location-independent characteristic of a network entity, an address is a function of the location of the destination station, and a route is something that depends on both the source and destination. From the layering perspective, names, addresses, and routes really refer to successively lower level representations of host identifiers. People usually prefer names, whilst protocol software works better with addresses. Either could have been chosen as the host identifiers.

In the Internet, a name is represented as a string which reflects organizational hierarchical conventions, such as peepy.newcastle.ac.uk, or kiet.etri.re.kr. Those names are called Fully-Qualified Domain Names (FQDNs). The Domain Name Service (DNS) [50] is a hierarchical, distributed method of organizing the name space of the Internet. It translates names to IP addresses (for example, kiet.etri.re.kr to 129.254.33.9). Addresses in the Internet are 32 bits long, and have a two-level

hierarchy: the network number, which may be 1, 2 or 3 bytes long, identifying a subnetwork, and the host number, which is the remainder of the address, identifying a host on that particular subnetwork. An address is normally written as four decimal number (each representing a single octet) separated by dots, as in 128.240.150.136. Routing is based on some information, mainly addresses, carried in the packet header. Each router examines part of the packet header (e.g., the destination address, source-route fields, sometimes source address and various quality or type of service (QOS or TOS) fields if policy-based routing is in effect, etc.), determines what the next-hop router is going to be, and delivers the packet to it.

IP datagrams clearly are routed based on the network number in the IP address. So, if a host moves to a new network, its network number would be changed; as a result, packets bound for the host could not be delivered without extra redirection support. Noteworthy from the above description is the early binding of an address to a route. The originating host specifies the destination host's address and thereafter no re-evaluation of the binding takes place. This is essentially the problem faced by a designer of IP-based wireless networks. In the [21], this address *versus* routing problem is summarized as:

If a host moves from one network to another, its IP address must change.

Here it is worth pointing out that an IP address has a underlying assumption; all hosts having addresses with the same network number are connected to the same physical network (e.g., Ethernet). Network numbers are assigned to organizational units, such as universities, companies, government agencies etc. They are unique in the internetwork as a whole. The network number thus allows for routers to keep track of where to send a packet by inspecting only the network number of the destination address in the packet. For some organizations, it is possible to waste a lot of addresses for those owning network numbers. Thus the idea of subdividing the address space at a lower level emerged, and this is done via so-called *subnetting* [49]. Subnetting is done by expanding the network number to cover part of the host portion of the address, so parts of the host number are used to indicate sub-networks within the organization. The subnet number is unique in the network identified by the network number. The host number is unique in the subnetwork identified by the network number and the subnet number. Thus, a

host is uniquely identified by a network address. As we see below, the multi-level hierarchical structure of IP addresses makes the task of route distribution and maintenance scalable (and autonomous) because each hierarchy level only needs to know about itself, and the levels directly above and below it.

3.2 Location and Routing Considerations

In the previous section, we described some advantages of the hierarchical structure of the IP address, from the point of an addressing convention by encoding network information in an Internet address. As everything has a good and bad side, there are also some disadvantages. The most obvious weakness is that addresses primarily refer to networks, not to hosts, as a routing end point. Clearly, with regard to movement, it is hosts that move not networks (in the mobile network, network itself would move around). This seems to be even more critical when applied to mobile computing environments, in which a host frequently moves from one network to another. This situation leads to several issues in handling host mobility. From the routing viewpoint, an IP address is no longer an identifier for a mobile host. That is, an IP address no longer implies the location information for moving hosts, so it now cannot function as a routing basis for those hosts. In this section, we discuss these issues from the routing point of view.

3.2.1 Routing in the Internet

The hierarchical structure of the IP address is a very important feature for understanding the Internet routing structure, and, as we shall see later in this thesis, it also affects the way we handle host mobility. Initially it aims to scale internetworking well by providing the abstraction of address clustering. In practice, this abstraction allows routers to keep minimal routing information, and make their routing decisions efficiently. Each level in the address hierarchy need only concern itself with the portion of the address that is relevant at that layer. This allows organizations to do their own internal routing (with subnet or host number), whilst routing between organizational networks is done based only on destination network, not on destination host. A gateway only needs to maintain network num-

bers, not full IP addresses. In addition, a gateway only specifies one step along the path to a destination network; it only points to gateways that can be reached directly across a single network. That is, it does not maintain the complete path to the destination. In this way, routing information about specific hosts is confined to the local environment in which they exist – machines, which are far away from them, route packets without knowing such details.

The routing path may be determined at the initiation of a communication, to give connection-oriented routing, or by continuous hop-by-hop decisions, that is, connectionless routing. As described in the previous subsection, the internet layer takes charge of this routing duty, and uses hop-by-hop routing. Choosing the next-hop router is accomplished by consulting routing tables in each intermediate router, using the destination network address of the packet as the search key. Once the packet arrives at a router in the final destination subnetwork, the router must resolve the network address of the destination host into the hardware address for that host because the network interface layer internally uses the hardware address. There is no relationship between the network address and the hardware address (48 bit flat format in Ethernet). In the case of Ethernet, the Address Resolution Protocol (ARP) [57] was designed to achieve this address mapping.

The IP routing algorithm is table-driven, and a datagram is delivered from gateway to gateway based on the gateway's routing table, until it reaches one that can deliver it directly to its final destination. The IP routing table initially preserves minimal (local) information, such as directly connected gateways or hosts, and only maintains network (and subnet) number of the IP address. It contains only partial routing information; many routings depend on *default* routes to possible distant destinations. Really, having information about all possible destinations in all gateways is impractical – however, partial information introduces a problem which may make some destinations unreachable from some sources (refer chapter 13 in [21]). Moreover, the Internet topology is rapidly growing or changing with failure or movement. In this situation, in order to provide an Internet-wide routing service, the Internet uses an architectural approach that allows groups to manage local gateway autonomously, adding new network interconnections and routes without changing distant gateways. Figure 3.2 shows the Internet routing architecture, which consists of two layers, a core system and a set of autonomous systems.

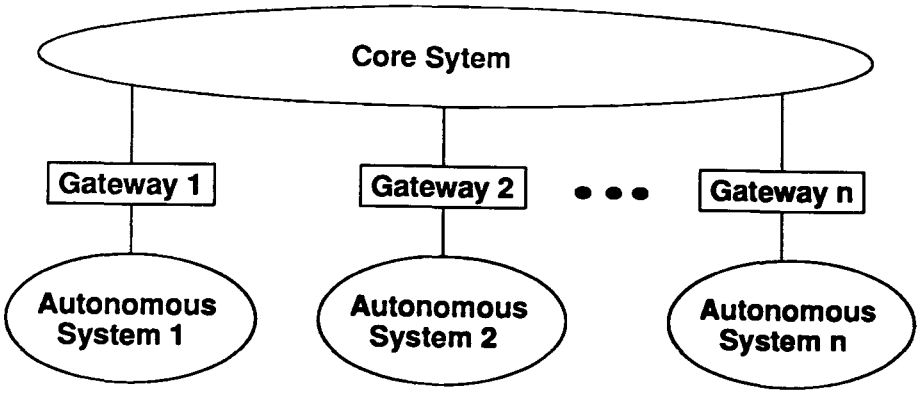


Figure 3.2: The Internet Routing Architecture

Both systems include multiple networks and gateways. The core system, as its name shows, is the glue that holds the Internet together and makes universal interconnection possible. It is controlled by the Internet Network Operations Center (INOC), and provides reliable, consistent, authoritative routes for all possible destinations. It does not use the default route. All core gateways exchange routing information periodically amongst themselves so that each has complete information about optimal routes to all possible destinations, and keep their route table up-to-date. [30] describes the core system concepts and the formal GGP (Gateway-to-Gateway Protocol) specification.

However, the core system itself cannot grow to accommodate an arbitrary number of groups, because a group will have an arbitrary complex structure, so some networks will not directly attach to the core system. In practice, groups, mostly organizations, have multiple networks and gateways. So, it is desirable to provide a way for each group independently to manage them, in order to accommodate graceful augmentation of the Internet; each collection of networks and gateways managed by one administrative authority is considered to be a single autonomous system.

An autonomous system is free to choose its internal routing architecture, but must collect information about all its networks and designate one or more gateways that will pass the reachability information to other autonomous systems. The protocol is described in [48], the EGP (Exterior Gateway Protocol) specification. EGP

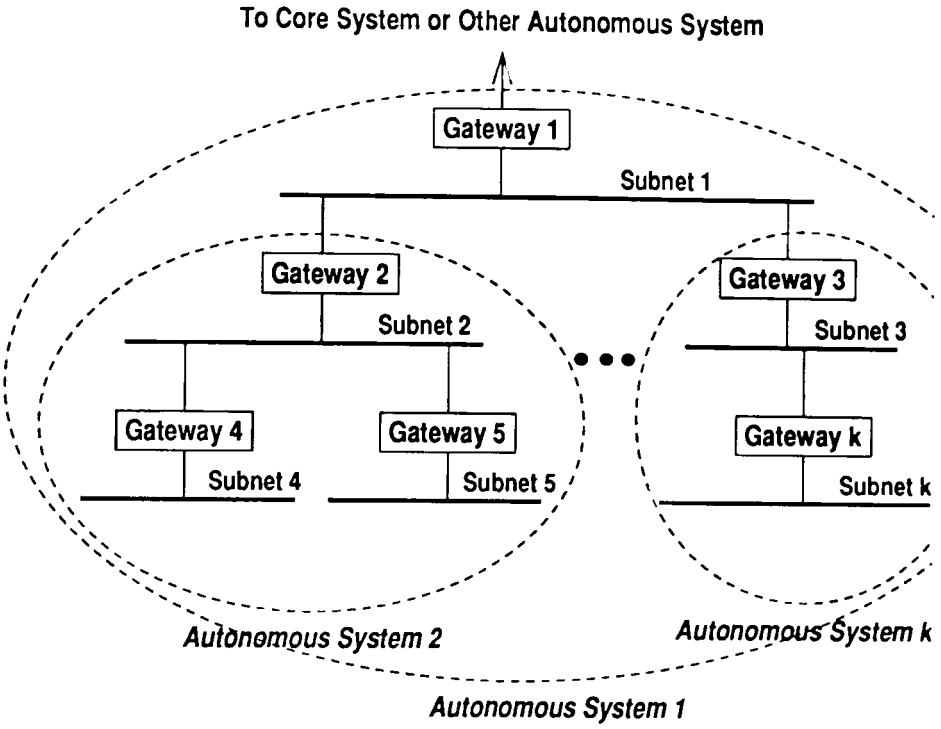


Figure 3.3: Autonomous Systems

allows gateways to advertise only the reachability of those destination networks within the gateway's autonomous system; it does not include any information about the distance metrics between them. This restricts the topology of any internet using EGP to a tree structure in which a core system forms the root. There is only one path from the core system to any network. The tree-shaped topology goes with the historical evolution of the Internet [19], and still remains as its base structure, even though it has some shortcomings, such as single points of failure and load sharing problems.

Again, autonomous systems are hierarchically grouped into an autonomous system, depending on their administrative ties. Figure 3.3 depicts an example of a hierarchical structure of autonomous systems. Gateways in an autonomous system should learn about network statuses and their connection changes within the system quickly and reliably. This is done by exchanging network reachability and routing information. As a logical counterpart of EGP, this protocol is called IGP

(Interior Gateway Protocol). The most widely used IGP is the RIP (Routing Information Protocol), which is described in [28]. Finally, the host relies on gateways to update its routing table. If a gateway cannot route or deliver a datagram, such as in the cases of failure (or disconnection) of the destination and congestion of an intermediate gateway, or if the gateway detects a host using a nonoptimal route, it needs to instruct the original source host to take action to avoid or correct the problem. To do this, IP includes the ICMP (Internet Control Message Protocol) control and error message protocol as an integral part [59].

3.2.2 Location of Moving Hosts

The IP protocol's main role is for internetworking along with an addressing convention and routing functions that span more than a single network. When the originating host specifies the destination host's address, a binding between the address and its route is established, and thereafter no re-evaluation of the binding takes place. In a static network, there is little re-evaluation of the binding; the network topology is rarely changed, probably only when network component, link or router, has a failure (or malfunction). Therefore, the routing protocols[1] in current IP implementations, such as ICMP and RIP, convey the changes into the related routers before the next possible change takes place.

In a mobile computing environment, mobile hosts are expected to move from time to time, and in a way that necessitates changing their address – moves to other locations in different parts of the hierarchy. Unfortunately, current IP routing mechanisms cannot decouple the host tracing function from an IP address – it needs another facility to trace a moving host, and then to deliver the packets following the host based on the location information. This is the most important issue faced in the IP-based approach for supporting host mobility. In practice, this can be formalized as a location problem, which includes an addressing convention for identifying mobile hosts, and acquisition and/or preservation of location information. In order to support location, it is inevitable that we must re-structure the routing scheme as well, according to the address convention adopted. The loca-

[1]The IP specification [28, 30, 48, 59] only conceptually specifies routing functions and gateway protocols. In this dissertation, routing protocol is described broadly taking into consideration its specifications and the current IP implementation as well.

tion issue seems similar with the mapping elaboration between name and address, which is already complicated in the traditional static network. That is, the mapping "what" to "where" is changed to the one "known-where" to "current-where".

An important result by Cohen *et al.* [20] identifies three possible address resolution schemes in conjunction with supporting host mobility in the Internet. In the Permanent IP-Address Scheme (PAS), each mobile host has a permanent IP-address from the initial (home) administration address space. Whenever a host moves, some hosts or routers (at least the old mobility agents) are informed of the new mobility agent's address as a current location; they then forward packets to the current location using the location information. The Temporary IP-Address Scheme (TAS) is that a temporary address is assigned dynamically every time a host connects with a mobility agent. The location information is managed by supporting a directory system or the source mobility agent broadcasts a query to find the current location. In the Embedded Network Scheme (ENS), each mobile host has a permanent IP-address and an embedded network address which consists of the current mobility agent address and a temporary address. The gateways maintain a mapping between IP-addresses and the embedded network address, and use this to forward any on-going packets, if needed.

Much previous work [33, 39, 55, 56, 70, 73] made use of one of the schemes above; ENS for [33], TAS for [73] and PAS for the others. Because the IP address implicitly contains its route, the location acquisition strategy is mostly decided by the addressing scheme adopted; broadcasting for ENS [33], an address consultant for TAS [73] and a forwarding pointer (sometimes a location cache) for PAS [39, 55, 56, 70]. It is clear that broadcasting or the address consultant are inappropriate schemes for Internet-wide host mobility because of high cost. In the research community, it is widely agreed that the forwarding pointer is among the fastest, and most useful in the IP environment[2].

The remainder of the location problem is where and/or how much location information will be preserved on the system as a whole. Obviously, a routing decision

[2]Therefore, we will hereafter limit our discussion to the forwarding pointer scheme. We merely mention the others for completeness. Also note that a location cache is usually used for the cases of maintaining the location information in many and unspecified routers or hosts, whilst a forwarding pointer is for preserving the information in the fixed known routers, such as the home agent.

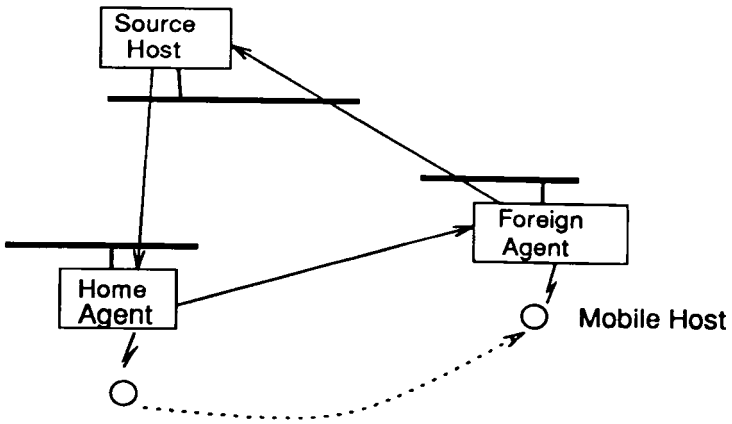


Figure 3.4: The Triangle Routing

must be made based on the location information that is available; packet routing efficiency depends critically on how effectively a packet comes across its current address. Ideally, the changing (current) address of the mobile hosts must be propagated to just the routers handling that host's traffic as possible as it can be done. If a source host knows the whereabouts of the destination mobile host, tunneling can occur directly, bypassing the mobile host's home agent, thus giving *direct routing.*

In considering the current IP address's role, the most common (simple) placing method is to hold the current location for moving hosts on the host's home agent. In this case, an IP packet must be sent to the mobile host's home agent where it is tunneled to the mobile host's current location, resulting in *triangle routing.* Triangle routing was named for the situation where the return path from the mobile host follows a direct route, bypassing the host's home network, hence creating a triangular round-trip route. If the source host is another mobile host and it moved to other place after it sent a request packet but before getting a reply, then this results in tetragonal routing. However, it is a symmetrical problem, and it is sufficient to deal with only half the problem. Figure 3.4 shows this situation.

Triangle routing is undesirable – the increase in the network utilization and high sensitivity to network partition – because of the unnecessarily long route. It also makes a bottleneck on the home agent. Triangle routing is mainly caused by

poor investigation of the relation between the location strategy and its routing effectiveness. The routing results in a mobile support system may usually appear in the form of performance transparency to mobile applications. For example, if an application has real-time constraints on the amount of allowable delay, it may be necessary for the routers to forward the packets ahead of the deadline. If this does not happen, the application cannot be applied to a mobile host environment, even if it was working well in the fixed network. Therefore, routing issues should be investigated from the system performance point of view, in order finally to provide a stable computing environment.

As a result, in devising a location scheme, it is important to keep in mind that excessive location preservation can be wasteful of network resources, whilst on the other hand insufficient location propagation leads to inefficient routing. This tradeoff is especially important in the Internet environment, as host mobility demands frequent (and widespread) updates. The arguments that mobile computing now faces are how to distribute location information, and then how to utilize the information effectively, in order efficiently to deliver packets to moving destinations whilst still limiting costly location updates as much as possible. These are the main motivations for our work, and our steps towards a solution start from the next section.

3.2.3 Mobile Internetwork Routing Structure

As presented above, the addressing hierarchy reflects some of the topology of the network; also, the format of an address is usually selected to facilitate this routing process. The information present in the routers is essentially a description of the precise topology of each hierarchical level. Also, the address of a host is administratively determined by its location. The larger the routing domain (the set of addresses over which routes are computed), the more computational (and network) resources are needed in the routers themselves, and to exchange routing information. The routing protocols must converge a lot faster than the time between changes. If the links go up and down faster than the protocols can converge, routing may not be possible even though physical paths exists. As shown in the previous subsection, the Internet routing architecture makes use of two level hierarchies: the core systems and autonomous systems. Each of these system has

protocols responsible for determining and distributing routing information among routers within the system and between the systems.

The mobile computing architecture, shown in the Figure 2.1, is that of a mobile network overlaid on top of a static network, using the latter as its basic communication infrastructure. Even if an IP address takes advantage of a separated form of its logical and physical nature in order to trace the physical location of moving host, each of these is still an IP address in order to keep its reachability via a static network. A mobile host usually keeps up its physical addressability through the mobility agent currently serving it. A packet routing path bound for a mobile host is mainly decided on the fixed network. This provides certain key features that affect the design of a mobile internetwork structure.

A host may move around within an administration authority area, or, sometimes, may regularly cross between administration areas. Each part of this area is maintained in conjunction with the IP address's formation. The hierarchical structure of the IP address, which is designed to facilitate routing, creates a locale in terms of sharing the knowledge of the whereabouts of the individual hosts of that subnet. As the router usually makes use of it for the routing decision, hosts not having an address on the given subnet(s) are considered external to that subnet(s). Therefore, it would be useful to utilize the hierarchical nature of IP addresses in order to exploit an effective routing for host mobility. This is based on the fact that most of the routing attribute come from the IP address's convention.

Now, let us combine the Internet routing architecture and the mobile computing architecture. Figure 3.5 depicts the mobile internetwork routing structure. It consists of three hierarchical levels – internetwork, local internetworks (dashed circle) and wireless networks (dotted circle). The internetwork includes the core system and autonomous confederations (a group of autonomous systems), whilst a local internetwork may be an individual autonomous system in the Internet routing architecture. Mobility agents play an intermediary role between the local internetwork and the wireless network. Thus what we are trying to do is to move up some routing duties for mobile hosts to some part of the fixed network (in the mobile computing architecture), usually a local internetwork. As a result, many of the routing features of the IP address can be utilized for routing facilities of mobile hosts. This approach makes possible our intention to distribute the mobile

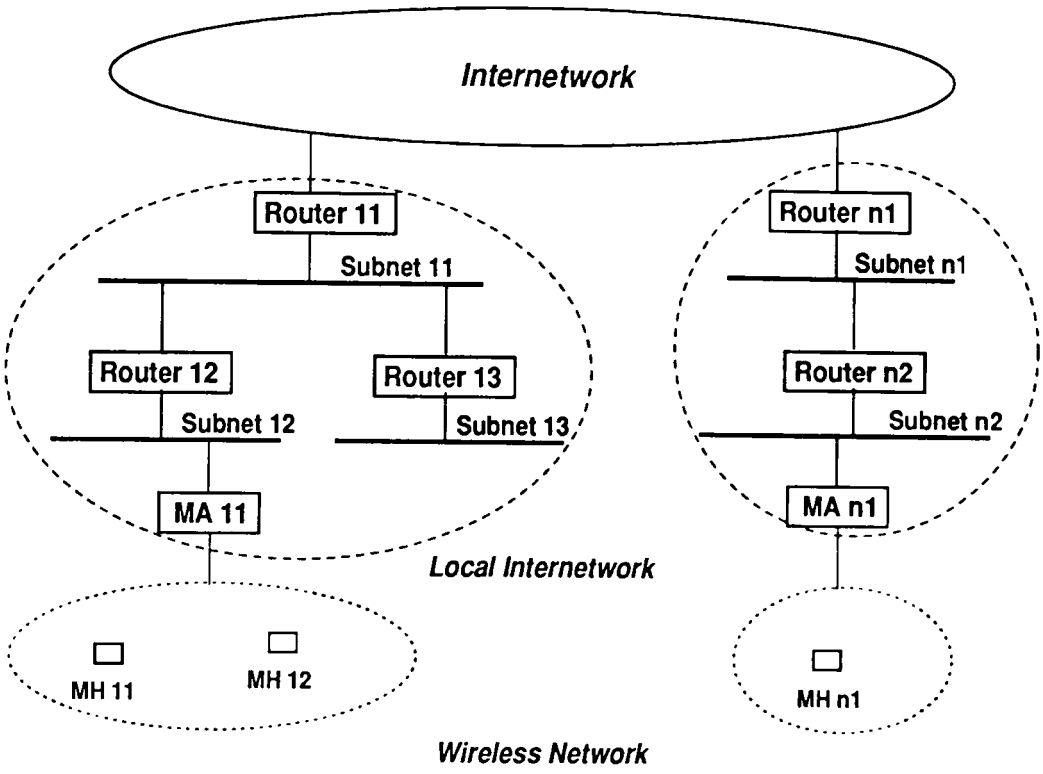


Figure 3.5: Mobile Internetwork Routing Structure

host's current address just to the places where it is highly likely to be utilized. In the next section, this idea is elaborated further to show how to use host mobility and packet calling patterns to improve routing efficiency.

3.3 Locality in mobile computing

The route for reaching a particular mobile host may be changed not only every time the host initiates a new connection, but also while communication with the host is taking place. Whenever a host moves, it has to generate a significant amount of location update data in order to keep providing a location transparent service to others. The Internet may require more frequent location updates than today's cellular telephone systems since hosts will be operating not only in wide area (macro-cellular) environments, but also in urban (micro-cellular) and in-building

(pico-cellular) environments. Mobile hosts in pico-cellular environments will often move across cell boundaries. In addition to its frequency, another problem occurs in the fact that host mobility is spread out Internet-wide. Changing all the location information throughout the Internet is obviously unreasonably expensive. On the other hand, maintaining the current location simply on the home agent only will bring unreasonably lengthy routes – triangle routing.

To achieve better routing and at the same time save valuable network bandwidth (especially on wireless links), and therefore to provide a flexible and scalable scheme, we feel that analysis of the host movement pattern and the packet traffic pattern will play a decisive role. Each host may move from one place to another, bringing with it route changes on the fixed network. Each traffic pattern may have a corresponding optimal routing. Thus, the most obvious assumption is that mobile hosts are most likely to move around between the mobility agents within a region, which usually contains its home subnet and the current subnet. Likewise, a host is most likely to communicate with a limited number of source hosts which have an interest in contacting it; a considerable number of these would be in its home region or the current region, and few others elsewhere.

Locality in a mobile computing environment, can be looked at both from spatial and temporal points of view, and with respect to the movements of hosts and the calls that are made to them. Even if it is difficult to predict how often a mobile host will move and how often a source host will access the host, in real life, moving and calling usually have some kind of pattern. In some work in wide-area internetwork applications, it has been observed that there exists a strong locality trend in the traffic pattern for way hosts and networks are accessed over the Internet [10].

Intuitively, a *local region* can be defined for each mobile host by including those subnetworks between which the mobile host moves often and omitting subnetworks into which the mobile host rarely ventures. A local region may reflect administrative domains within the same geographic area because the Internet protocol makes use of the hierarchical addressing and routing scheme based on geography or network topology, and the Internet has been augmented in conjunction with the administrative autonomy. Practically, it would be an autonomous system or a group of autonomous systems, where its home agent or current agent is overlaid underneath the system(s). For the same reason, a local region may also include

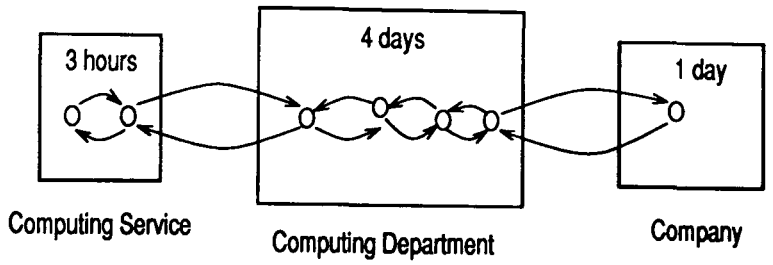


Figure 3.6: A Host Moving Pattern

a considerable number of source hosts, which frequently interact with the mobile host; though it is not a necessary condition for a local region. Here, a local region naturally reflects spatial locality by taking into account the temporal locality of host movement pattern as well as packet calling.

For example, Figure 3.6 shows the weekly routine of a professor who works mainly at the computing department, goes to the computing service for about three hours, and visits a company to co-operate on a project for one day. She makes a number of local moves within each of the areas. Many of the calls to her machine are from the colleagues in the department, the computing service or even the company. Obviously, it is reasonable to define the computing department as a local region for the professor. If necessary, the professor may also assign the company as a local region. Thus, each mobile host has a different local region depending on its interests. A local region for a mobile host can be determined either by monitoring a host's movements and accesses to it or by getting information directly from the host's user in advance. As a result, a local region would be used to provide a means of maintaining a degree of knowledge about the mobile host's whereabouts, so as to cut down the cost of location and routing.

In the Internet, because a hierarchical address space is chosen and where mobile hosts change address during migration, the routing is suboptimal unless location information can be propagated to the router closest to the source host, or to the actual source host. If location updates are sent only to those hosts that actually need to communicate with the mobile host, the location overhead can then be gracefully limited, with most source hosts achieving optimal routing. Here, an optimal route means the route a packet would normally take between two

stationary hosts using conventional IP routing protocols.

A host tends to communicate with others that are interested in it. In the internetwork environment, even if the potential set of sources that could communicate with a host may be large, only a small set of sources would actually communicate on a regular basis. [10] shows that this reference locality is relatively stationary; any change to this set will come very slowly and only after a period of time. However, the interested group should be somewhat influenced by the host mobility pattern, that is, depending on where the host is currently staying. For each mobile host, we establish *patron* hosts which are the source hosts where the majority of traffic for the mobile host originated[3], and which are therefore highly likely to call again. For example, let us assume a calling scenario from a sequence of source hosts to a mobile host: **B, A, B, F, T, A, F, F, B, A, K, B, A, A, B, F, B, A, F, ...** In this example, the host would certainly select A, B and F as its patron hosts. A host would determine its patron hosts by monitoring the calls it receives.

It should be advantageous to let the local region mutually co-operate with the patron concept for the purpose of improving their effectiveness. Because the local region provides a certain degree of host moving abstraction to the outside world, it is desirable to consider only the hosts outside the local region; potential patrons are recorded only when the host stays within the local region. When a mobile host leaves (or crosses) its local region, it notifies its new location to the patron hosts. We will refer to this as the *patron service.* Then, the patron hosts can use the location information for their next call to the host. This condition takes advantage of the spatial locality of the destination host's movement pattern and the temporal locality of the calling pattern itself, in order to pass the new location to the source hosts most likely visit again.

The next chapter specifies the detailed control structure for the local region and patron, with emphasis on how these concepts work together to achieve a unified scheme which provides nearly optimal routing (bypassing the default reliance on routes passing mostly through the home agent), with effective location (hiding host mobility locally as far as possible, and, if need be, informing only those source hosts that, otherwise, may need a lengthy route to get to a mobile host).

[3]If the source host is another mobile host, the patron will be symmetrically formalized.

It is important to point out that location updates that sometimes result in sub-optimal packet delivery for infrequently visited source hosts may be acceptable. In addition, if a scheme has source hosts that tolerate the use of stale location information, the number of location updates could be significantly decreased. These features should be properly incorporated into the control frame of the local region and patron concepts. Additionally, as described above, a mobile host will notify its new location to some of the location caches each time it moves, and to the patron hosts whenever it crosses a local region, so providing move-initiated location updates. Another possible way is to be informed of the new location whenever a source needs it; that is, a need-initiated update. The tradeoff between these two strategies is a very important choice for the location effectiveness; its details are discussed in chapter 6.

3.4 Summary

This chapter presented some Internet related issues in support of host mobility and our basic approaches for location and routing optimization. Wireless networks should be considered as members of the ever-growing number of existing fixed networks. This naturally calls for the use of the Internet TCP/IP protocol suite as the foundation of wireless networking, as due to its own world-wide success; it is the most prevailing standard internetwork communications. With its addressing and routing capability, it is widely agreed that host mobility support in the Internet should occur in the internet (IP) layer. Unfortunately, the IP protocol deals badly with a dynamic network topology such as that provided by a wireless network. The problems that have to be resolved for host mobility in the Internet consists of addressing the mobile hosts, and locating and routing to them effectively while maintaining their addressability as they move.

A common approach is to use two separated IP addresses (a logical identifier and a physical locator) with a forwarding mechanism in association with location caches held elsewhere. Clearly, the packet routing path depends on the "elsewhere", which holds the mobile host's current location. Thus, our concern for effective location is to find just the places which it is predicted, can be effectively utilized for packet routing. One piece of evidence comes from the features of fixed infras-

tructure. A packet routing path bound for a mobile host is mainly decided on the fixed network part because the physical locator is still an IP address. The Internet protocol makes use of hierarchical addressing and the routing structure in order to reduce the size of the routing tables that must be maintained at each router and exchanged between routers, and simplify the routing decisions at each router. This hierarchical nature is very useful in the mobility support system, in order to exploit some knowledge of the whereabouts for the individual moving hosts. Also, the Internet has been growing in a hierarchical form, in order to accommodate its graceful augmentation, and to provide administrative autonomy. Our approach now is to move the locating and forwarding roles for mobile host to some part of the fixed network, and then to localize the effect of host mobility into a designated area.

In addition, exploiting the locality in a mobile computing paradigm seems to be vital for realizing our intention. The starting point of our approach comes from the assumption that mobile hosts are most likely to move around between the mobility agents within a designated region, and communicate with a limited number of source hosts which have an interest in contacting it. This tendency is very general and covers most communication scenarios in the real world. With the locality property, two concepts, *local region* and *patron*, are defined for each mobile host; the local region is a set of designated subnetworks within which a host often moves, and the patrons are the hosts from which the majority of traffic for the mobile host originated. Thus, our scheme will try to propagate the location information within the local region and, if necessary, to the patron hosts. In next chapter, we shall show how these two concepts are controlled to achieve better routing and in the same time saving costly location distribution in the Internet host mobility environment.

Chapter 4

A Location and Routing Optimization Protocol

In the previous chapter, we examined the issues involved in location and routing optimization for internetwork host mobility, and introduced two concepts, "local region" and "patron service", based on the locality features of host moving and calling patterns. This approach naturally goes with the Internet's hierarchical structure with regard to addressing, routing and configuring. In this chapter, we shall show how the local region and patron concepts are incorporated to realize a unified scheme for efficient host location as well as optimal routing. We first define a system model for internetwork host mobility and its required functionalities, then describe a basic scheme. Building on the basic scheme, the two concepts are added in turn, and the the registration and routing details are described. Finally, additional considerations for the scheme are discussed.

4.1 The Setup

In order to complete our description, the system model for mobile computing, detailed in subsection 2.1.1, is now slightly changed to incorporate the mobile internetwork routing structure, described in subsection 3.2.3. The fixed network is replaced with the Internet, and some subnetworks are arranged between the Internet and the wireless network. Figure 4.1 depicts a sample configuration for

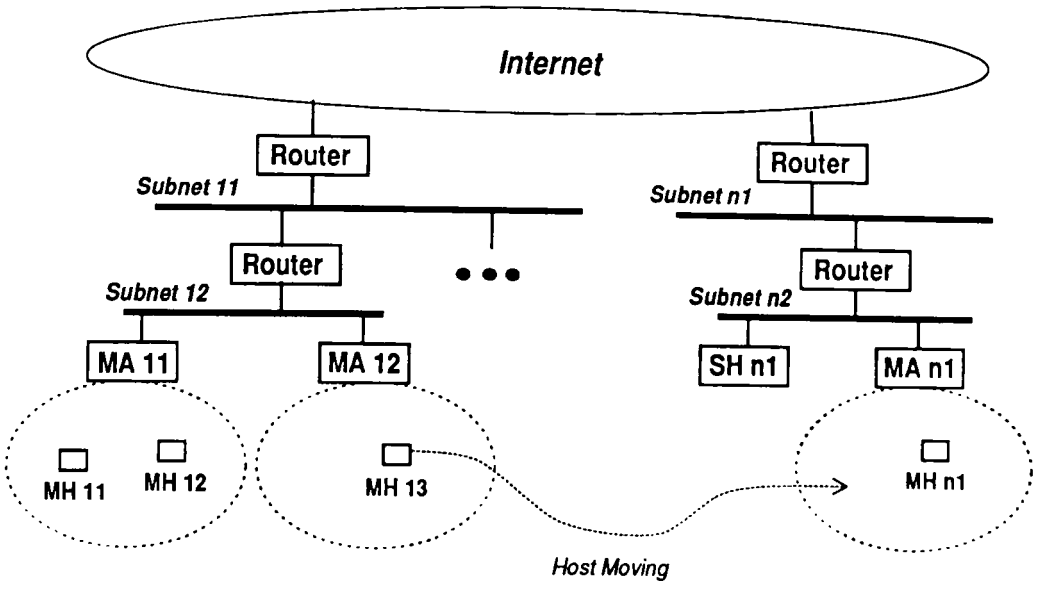


Figure 4.1: A Sample Configuration for Internetwork Host Mobility

internetwork host mobility. This configuration allows arbitrary mixing between mobile hosts and Static Hosts (SH). The existing static hosts and normal network routers should be able to communicate with the mobility entities. Therefore, the burden of supporting host mobility should be borne entirely by the mobility entities, principally the mobility agent.

In the figure, mobile hosts MH 11 and MH 12 are served by mobility agent MA 11 and MH 13 is served by MA 12 on the same subnet, subnet 12. This subnet connects with another subnet, subnet n2, via two groups of intermediate routers through the Internet where mobile host MH n1 is served by MA n1 which coexists with a static host SH n1. Each group of intermediate routers is connected by some subnet – subnet n1 and subnet n2 for the right side – and maintained by an administrative authority. Many of our examples concentrate on using this local internetwork. The figure shows that MH 13 leaves from a cell served MA 12, and joins a cell under MA n1.

A mobile host is assigned a permanent (home) IP address in the same way as any other host and this remains fixed regardless of where the host is attached, thus acting as its *logical identifier*. When a host connects to a subnetwork, it

finds and then registers with a mobility agent, which becomes its *current agent*. If the mobile host is initially (therefore permanently) registered with this agent we shall refer to it as its *home agent*, otherwise we shall call it a *foreign agent*. The *previous agent* is what we call a home (or foreign) agent which a host has just left. A mobility agent is a router providing normal routing functions, and connected to intermediate routers and wireless networks. It has the ability to tunnel a packet to a foreign agent, which currently serves the packet's destination host, for eventual delivery to other hosts. Therefore, a mobility agent's address may be used to represent a mobile host's current location, that is to act as a *physical locator*.

A mobility agent also maintains a set of mobility bindings for mobile hosts, which have a direct or indirect relation with itself. A mobility binding is the association of a mobile host's home location with its current location. This is made by recording their IP addresses: the mobile host's own address and the current agent's address. The mobility bindings are mainly designed to help with forwarding (tunneling) any ongoing packets to moving hosts, so these are sometimes called *forwarding pointers*. The home agent maintains a *home list* identifying all mobile hosts it is configured to serve. The current agent, but not the home agent, records the mobility bindings in force for each mobile host that it is currently serving as a *visitor list*. In addition, the home agent and some foreign agents maintain a *forwarding list* that records the mobility bindings in force; that is, at the home agent for each of its hosts that is away from home, and at the previous agent for each host that is not currently registered in its visitor list but that visited this agent before. Actually, the home list (or visitor list) and the forwarding list represent a mobility binding at the home (or previous) agent. But, for simplicity of presentation, these are separately named according to their purpose; so the home list or the visitor list is assumed to be null for the current agent's address field. In addition, these lists are sometimes referred as a location cache. The contents of the finite cache space may be maintained by any local cache replacement policy. One possible way to implement the lists is to use the same table that is already used to handle the existing host-specific ICMP redirect message type [59], but with the different flag.

The following list identifies the basic functions[1] of the mobility entities, without

[1]Most of these features are drawn from previous work, especially the recommendation which has been discussed in the IETF Mobile-IP group [56].

giving details of how to implement them. Based on these functions, we build a basic scheme in the next section. This will be extended for a location and routing optimization protocol for an internetwork host mobility environment. However, our scheme will be described as independently of these functions as possible.

- A beaconing and/or solicitation protocol enabling the mobile host to identify itself to a mobility agent, and then obtain network connectivity. Beaconing is where mobility agents periodically advertise their identities, and solicitation where mobile hosts multicast to find prospective agents.
- A registration protocol to create a set of mobility bindings between a mobile host and its current mobility agent. Depending on its point of attachment, the mobile host registers its location information with its home agent, current agent, and previous agent (if any).
- An encapsulation protocol that tunnels the packet to the mobility agent currently serving the destination mobile host. The mobility agent decapsulates the packet and eventually delivers it to the desired host.

4.2 Basic Scheme

The basic scheme is defined as a home-based forwarding strategy. All packets destined for a mobile host are routed through that host's home agent. If the host is served by the home agent, it directly delivers the packets through its wireless interface. If the host is currently served by a foreign agent, the home agent then encapsulates each packet to the host's current agent. The current agent decapsulates the packet and sends it to the target host. To allow this, each time a mobile host moves to another location, the mobile host invokes a set of registrations to create the mobility bindings on the related agents: the home agent, the foreign agent it has just left, if any, and the current agent.

Firstly, the corresponding host's entry in the home list is reset, so as to indicate that the host has moved away from the home agent. A new entry on the forwarding list is created with the host address and the current agent's address, to denote that the host now stays with the current agent. If the previous agent is different

from its home agent, the visitor list's entry of the previous agent is deleted, so the host is no longer served by this agent. An entry is created in the forwarding list on the agent. The previous agent makes use of the entry in order to intercept any packets delivered to a host which had been a visitor but has moved on, and then forward them to its new location. Finally, an entry is added to the visitor list of the current agent. These agents – home, previous and current – are then all aware of the same mobility binding for the mobile host.

When a foreign agent receives a packet addressed to itself, the agent decapsulates it if the packet has been tunneled, and consults its visitor list (current agent) or forwarding list (previous agent). The agent will deliver the packet directly to a mobile host named in its visitor list, or again tunnel the packet to the foreign agent named in its forwarding list. In the case of a home agent, it acts as a normal router delivering the packet to the host listed in its home list. It also looks up a forwarding list for mobile hosts that are away from home, and encapsulates the packet to the foreign agent currently serving the mobile host, which finally delivers it locally to the mobile host.

Here, as a preserver of the physical locale, the forwarding pointer brings about another problem; forwarding pointers generally require the reclamation of old pointers which have been superseded. If not, a sequence of forwarding pointers is left along the path of a moving host; the pointers are successively linked from the oldest previous agent for which the host first left the home agent to the current agent. Packets arriving at these agents would be forwarded with out-of-date forwarding pointers, and would result in unnecessarily long forwarding paths. This situation is very serious when there are frequent host movements. Considering the reasons above, it is desirable to preserve only the most recent forwarding lists from the initiator. To reset the out-of-date forwarding pointers, one possible method is to use a timeout. In this work, we introduce the concept of *back firing*, where whenever the current agent updates an element of the forwarding list of the host's previous agent, the latter agent clears the forwarding list entry for the host on the previous agent, if it had one. This prevents needlessly long route paths, and goes with the spirit of move-initiated registration; its details are discussed in chapter 6.

In addition, mobile hosts have tight constraints on power consumption, and are

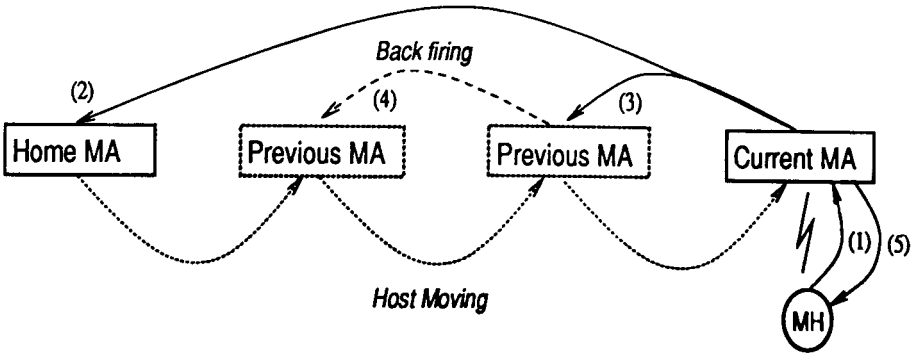


Figure 4.2: A Registration Example with the Basic Scheme

connected to the rest of the network via a wireless link. The wireless medium has relatively low-bandwidth compared with a wired one, and has the property that transmission of a message from a mobile host consumes more power than reception. To adapt to this asymmetric communication, we propose an *indirect registration model* where much of the responsibility for location propagation is shifted from the mobile host to the supporting network infrastructure, that is, the mobility agent. In the case of registration, a mobile host simply tells its current mobility agent its home agent address and previous foreign agent address if it had one. The mobility agent will try to deliver the registration packet to the corresponding agents on behalf of the mobile host. It only needs to pass the acknowledgement packet, which indicates success or not, to the host.

Figure 4.2 shows an example of the registration entities, the indirect registration procedure and the back firing concept adopted for the basic scheme. A dotted rectangle means that the entity is accessed only if it is available. When a mobile host finds a new mobility agent, after connecting with the agent, it sends a registration request (1) which includes a set of registry destinations depending on its movement history. Based on the registration request from the host, the current agent firstly tries to register with the home agent[2] (2), then with the previous agent (3) if it had one. If necessary, the previous agent resets the mobility binding on its previous one, as a back firing (4). After confirming these all succeeded, the current agent records the host as a visitor, and replies with the registration result

[2]The home agent has to be firstly registered because it usually has some important responsibilities, such as the authentication and the recovery from failed registrations.

to the mobile host (5).

4.3 Local Region and Patron Control

4.3.1 Local Region Control

As the first step in building our new protocol, we introduce an additional mobility entity, called a Mobility Router (MR). It is a router (or gateway) located somewhere between the Internet and the mobility agents. A mobility router is connected to the mobility agents directly or indirectly. The mobility router and mobility agent form a local internetwork, which is usually an administration domain, such as a university, a company or even a country. In accordance with the mobile internetwork routing structure in subsection 3.2.3, a mobility router maintains a routing table which keeps the addresses for all subnetworks under the router in a topology, possibly implicitly for the benefit of the IP address's hierarchical structure. Hence, a mobility router knows all the addresses of mobility agents within its local service area (usually a local internetwork, or a local region). For simplicity of description, it is assumed that each mobility router has one connection outside its local service area[3].

A mobility router has similar functionalities to a mobility agent, except that it does not have a wireless interface. It maintains some mobility bindings and has the ability to tunnel packets to the current agent; tunneling from a mobility router is sometimes known as *redirecting*, in order to differentiate it from that done by a mobility agent.

When a mobile host first joins the network or its user's interest area changes, the mobile user will be asked to define a local region. A hierarchical relationship is then assumed between the mobility agents and mobility routers which make up the local region. This relationship is represented as a tree, called a redirection tree. The leaf nodes consist of the mobility agents, and the other nodes are the mobility routers. The root node has the special duty of redirecting packets passing through

[3]In practice, a router may have multiple network connections to the others for the purpose of fault protection, whilst a single connection is still general for simplicity of routing management.

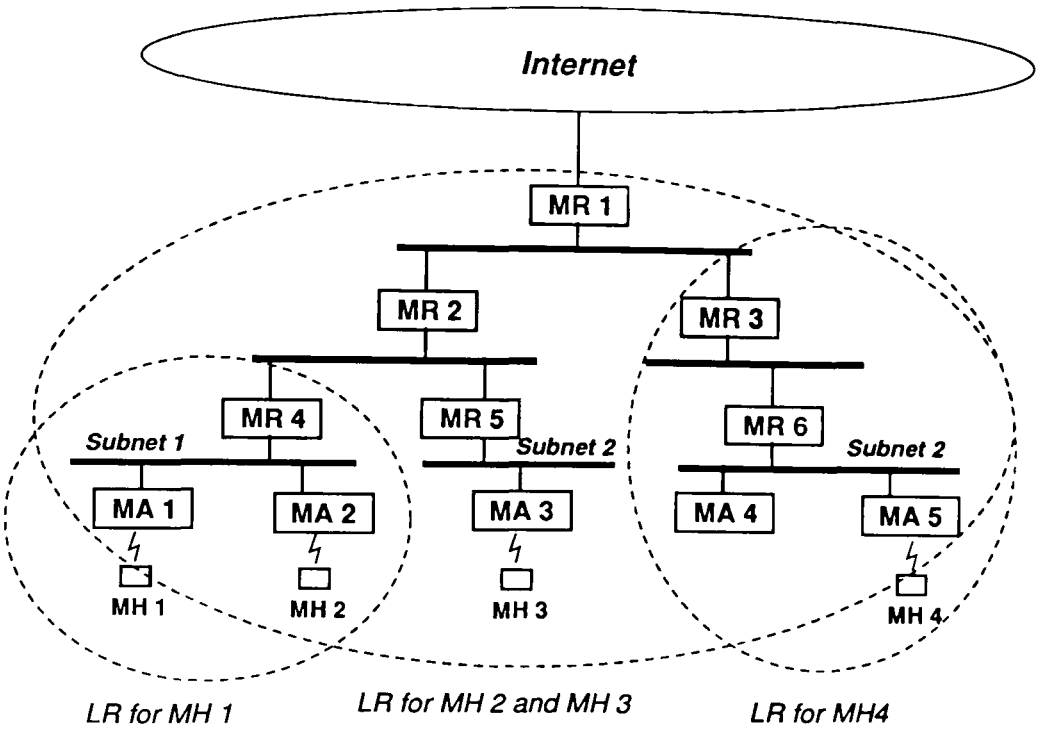


Figure 4.3: Mobility Routers and Local Regions

itself to the mobile host. We say that this root acts as a *Redirection Agent* (RA) for this local region. The intermediate nodes (mobility routers), if available, simply serve as normal routers for this local region, even if they are assigned as redirection agents for other local regions. Although a mobile host's users define its own local region, the redirection agent must be assigned by an authorized network manager who is familiar with the network topology of a given administration domain.

Figure 4.3 depicts the mobility routers and local regions for mobile hosts 1, 2, 3 and 4. Here, MH 2 and MH 3 share the same LR, and perhaps, the whole administration domain of the given organization, for example a university. MR 1 serves as their RA. MH 1 defines a sub-domain, such as a faculty or department, as its LR, and MR 4 might serve its RA. Similarly, MH 4 has MR 3 as its RA. The redirection agent maintains a list - the *redirection list* - to preserve the current location information for the hosts that have appointed this as their redirection agent. Now, each time a host moves, the mobile host must create an additional

mobility binding in its redirection agent, and its home agent, previous agent and current agent, just as in the basic scheme. The redirection agent uses the redirection list for redirecting any packets, which pass through it, to the current location of the host.

The initially designated LR for a mobile host (the region in which a mobile user mainly moves around) is called the primary (or home) LR, and its corresponding RA is called the primary (or home) RA. A mobile host may want to define only one LR, or may need multiple LRs for several areas based on the user's interests. Let us consider the case of moving a host to a foreign network outside the primary LR for a certain period, maybe a few weeks. It would be reasonable to define a new local region as well as the primary LR. Here, the local region which was assigned at the time the host has just moved out of the primary LR is called a secondary LR; likewise, its corresponding RA is a secondary RA. A current LR (or RA) stands for an additionally defined LR (or RA), which is neither the primary LR nor the secondary LR. Each of these LRs is controlled in the same way; the redirection lists on these RAs are maintained to trace the local region which currently serves a moving host. Their registration and control details for each of the cases of moving a host amongst these LRs are described in the next section.

Consider the case of moving a host somewhere outside its local regions for a short time, without assigning a new local region during that time. When host mobility is controlled by the local region framework and when the host moves around outside any existing local region, there are two possible modes to be considered: *redirection* mode if a host is moving (or working) with the benefit of a local region, and *direct* mode otherwise. For a host under direct mode, the forwarding lists of previous MAs are used to trace the host by using the foreign agent at the time the host moved out of its local region; needless to say, the previously defined local regions would help to reach the foreign agent. In direct mode, another method, such as a timeout, seems to be more reasonable than back firing, for reclaiming out-of-date forwarding pointers. In this dissertation, we only look at the case of the host moving in redirection mode.

If a host moves within its local region and a packet comes from a source host outside its local region, its redirection agent will intercept the incoming packets, and forward then to the mobility agent currently serving the host, bypassing the

possibly lengthy route via its home agent. With the benefit of the redirection facility (although it comes with one extra registration overhead to the redirection agent(s) per host move), a host does not need to declare its movements to any agent outside its local region. Source hosts residing outside the local region are permitted to use inaccurate location information for a mobile host. As a result, the local region provides a natural framework for localizing the effect of host mobility in a designated area, whilst most packets are still routed close to their optimal paths. In the next subsection, the patron concept is introduced to improve these benefits further.

4.3.2 Patron Control

If a host moves outside its primary LR, i.e. the physical locale is different to the home address, the forwarding path for the packets sent by a source host outside the LR becomes much longer because it will be via the primary LR and most of the packets would be forwarded by the primary RA. By definition, the majority of calls to a host are still from what we called the patron hosts - those that are highly likely to visit the host again, and that have their home addresses outside the host's primary LR. Each mobile host keeps track of those source hosts in its *patron list* built from the receiver's standpoint. This could be done by monitoring the source host of incoming packets, and by managing the calling information based on frequency of the source hosts' calls. Only high frequency visiting hosts are recorded as patron hosts. In practice, a mobile host may use the IP address structure for the source host to determine whether a packet has came from a source host outside its local region or not.

Whenever a mobile host leaves or comes back to its local region, a mobile host additionally registers with the redirection agents, the primary RA and/or the secondary RA, and if necessary, the current RA (see next section for details). When the host tries to register with the previous RA it has just left (in this time, the host know that it is still under the RA), the RA should determine if the host has just crossed one of its local region boundaries. One possible way to decide this is for the redirection agent to use the structure of IP addresses by checking the network part of the current MA address, to see if it is within its service boundary. If so, the (previous) RA sends a notice (cross_LR) packet to the

host. On receiving the notice packet, the mobile host sends its new location to the hosts on its patron list, so developing the *patron service*. The patrons then record the host's current location information in their *calling list*, and thus will use the new location information for subsequent calls. Thus the calling list is a mobility binding maintained by a mobile host from a sender's standpoint, for holding the new location of those hosts with which they have most frequently interacted.

The patron service takes place whenever a host movement crosses its local region boundary. For a host in redirection mode, a redirection agent is assigned for each of the defined local regions. Also a patron list is maintained for each of the defined local regions. Therefore, when a mobile host has multiple local regions, as the patron services progress, it is required a complicated method to redirect consistently packets sent from those patrons to the host. In this dissertation, we limit for simplicity the patron monitoring to only when the host is staying in its primary LR, and the patron service only for the time the host crosses between the primary LR and the secondary LR. But this can be extended to further crossing between the LRs. When a host leaves the primary LR, the patron service updates or creates the calling list entries with the host's new location. On the other hand, when a host rejoins the primary LR, the patron service would reset the corresponding entries on patron hosts. For further movements from the time a host left the primary LR, packets from the patrons are always forwarded to the current location by the redirection agents of related local regions, even if this is slightly inefficient for some extra redirections.

As a result, source hosts that access a host frequently and that are located outside the host's local region, even those that are far from the host, will keep up-to-date location information. The traffic from the patron hosts, which covers most communications from outside its local region, can always achieve optimal routing, whilst traffic from infrequently visited source hosts needs to be tunneled, possibly by the home redirection agent. Because patrons are decided based on call history, and the expectation is that they will call again soon, the patron service can be regarded as a need-based location strategy. However, location updates, to the patron host(s), as in the case of the redirection agent, are initiated based on host movement, so are move-initiated registration.

The local region takes advantage of both the calling pattern as well as the host

Mobility entities	Mobility bindings	Usage
Redirection agent	Redirection list	Local region control
Mobility agent		Basic scheme
(Home agent)	Home list, Forwarding list	
(Foreign agent)	Visitor list, Forwarding list	
Mobile host		Patron control
(Sender standpoint)	Calling list	
(Receiver standpoint)	Patron list	

Figure 4.4: Mobility Entities and Their Mobility Bindings

movement pattern to define its dedicated boundary, i.e. the area where a host's internal movement should be told to the system. The patron concept then makes use of the local region to confine the costly external location propagation to those sources which would seem most usefully to utilize the location updates. As we will see in chapter 6, it is clear that the effectiveness of location and routing with each concept depends on the dedicated host movement and calling pattern.

Figure 4.4 summarizes the relationship between mobility entities, mobility bindings and their use for the local region and the patron service. The resulting system needs three additional location caches: the redirection list, calling list and patron list, and more than one extra registration – the redirection agent(s), cross_LR notice and patron service – depending on from/to where the host has moved. The cache size of the redirection list is restricted to the total number of mobile hosts (resident or visitor) within its serving area (local region), which is small in comparison with the size of the Internet. In the case of the calling list and the patron list, the size is limited by the number of hosts that actually frequently communicate with each other.

In addition, it is important to pay attention to preserving consistency between the mobility bindings. If mobility entities have different location information for a mobile host, the location strategy must be more complicated to tolerate the inconsistency. The current MA has charge of the registration for each host move.

It confirms the registration result to the host in an all-or-nothing fashion. For the patron service, a mobile host uses the same patron list both when it leaves the home LR and when it returns. The list is refreshed only while the host is in the home LR. Here, it is sometimes unreasonable to get replies from all patrons for the patron service via the Internet; so the current MA only sends the patron packets, and does not wait for their reply. If this packet has been lost in the middle of its way, the corresponding patron would have a different mobility binding from the other patrons. As shown in the next section, the redirection facility protects the system from this problem [4].

4.4 Location and Routing Operations

In this section, we describe the registration procedure, as a means of location, and the packet routing paths to show how effectively the location can be used for routing efficiency. A host may move around within its home LR, and sometimes cross out of the LR, if necessary, it then defines a new local region – the secondary LR (actually, this should be done by an authorized network manager). The host then carries out a patron service based on the calling statistics on the home LR, in order to tell its new address to the patrons. The new location for the host should be reflected in the redirection lists on the primary RA and the secondary RA. Therefore, a host may register with different mobility entities depending on from/to where the host moves. Packets bound for a host have different routing paths depending which local region the host is currently staying in, which local region the source belongs to, and whether the source is a patron for the host or not.

[4] In practice, when a patron service resets the mobility bindings established by the previous service (that is, when a host returns to its home LR), a supplementary method is needed, such as a timeout, to make sure the reset gets carried out; in this case, the redirection scheme would not solve the problem.

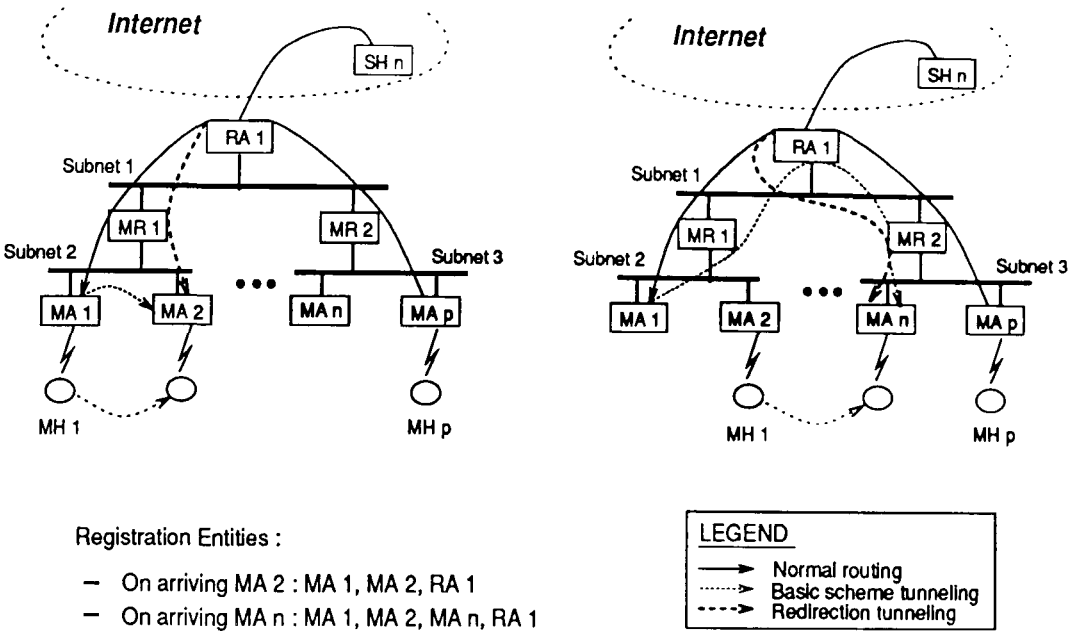


Figure 4.5: Routing Paths (Moving within the Home LR)

4.4.1 Moving within the Home LR

Let us look at the routing operation when a host moves around within its home LR. The host registers its new location with the home MA, the previous MA, if it had one, the current MA, and the primary RA, for each move. Figure 4.5 depicts the location and routing operations. Initially, a mobile host, MH 1, is registered with MA 1, as its home agent. MH 1 has its primary RA, RA 1, in accordance with the initial assignment, which consists of subnetworks 1, 2 and 3. Suppose MH 1 moves to MA 2 from MA 1, on the left side of figure. After identifying its new mobility agent, MA 2, MH 1 registers its movement with its home agent, MA 1, and with the primary RA, RA 1. MA 2 then adds MH 1 to its visitor list. For the next movement from MA 2 to MA n, on the right side of the figure, the registration process is similar, except that it includes a registration with the previous agent, MA 2.

We consider only the routing paths for incoming packets to a mobile host; those for outgoing packets can be explained in a similar fashion. When a host moves

within its local region and a packet comes from a source host inside its local region, the location and routing role of the redirection agent can be explained using the redirection tree concept. The routing path of a packet depends on where the source, destination and destination's home agent are situated with respect to the source's redirection agent. If the source or the destination and the destination's home agent are in the same subtree, packets may be tunneled by the home agent or its redirection agent (see the left side of the figure). If the source host is in the same subtree as the destination host and the destination's home agent is in the other subtree, packets always pass through the destination's home agent via the redirection agent, so the routing path is longest in the local region (see the right side of the figure).

In the right side of the figure, the source host, MH p, sends packets to MH 1, which is currently served by MA n. The packet should be bound for the home agent, MA 1. When the packet arrives at RA 1, it is intercepted by RA 1, which should have a redirection list entry for MH 1. Then RA 1 tunnels the packet to MA n (see the redirection tunneling path), thus bypassing the host's home agent. Then, MA n uses its visitor list entry for MH 1 to forward the packet locally to MH 1. Without the redirection function, all packets would be forwarded by the home agent, MA 1, using the forwarding list that is maintained by MA 1 (see the basic scheme tunneling path). With the benefit of redirection, the packets passing through the redirection agent are intercepted and forwarded by the agent, without going via the home agent, so are delivered with an optimal route.

If a host moves within its local region and the packet comes from a source host outside its local region (SH n in the figure), the incoming packets would be intercepted and forwarded to the current MA by the redirection agent, bypassing the possibly lengthy route via its home agent; these packets are delivered as if by the normal routing mechanism. Therefore, a host which moves around within its local region does not need to notify its movements outside its local region. Source hosts residing outside the local region are permitted to use inaccurate location information for a mobile host, as far as the location information indicates an address within the redirection agent's service boundary. As a result, the local region provides an elegant framework so that most traffic in the system is routed as close as possible to its optimal path, whilst the location propagation of the host's movements within a local region is confined to that region.

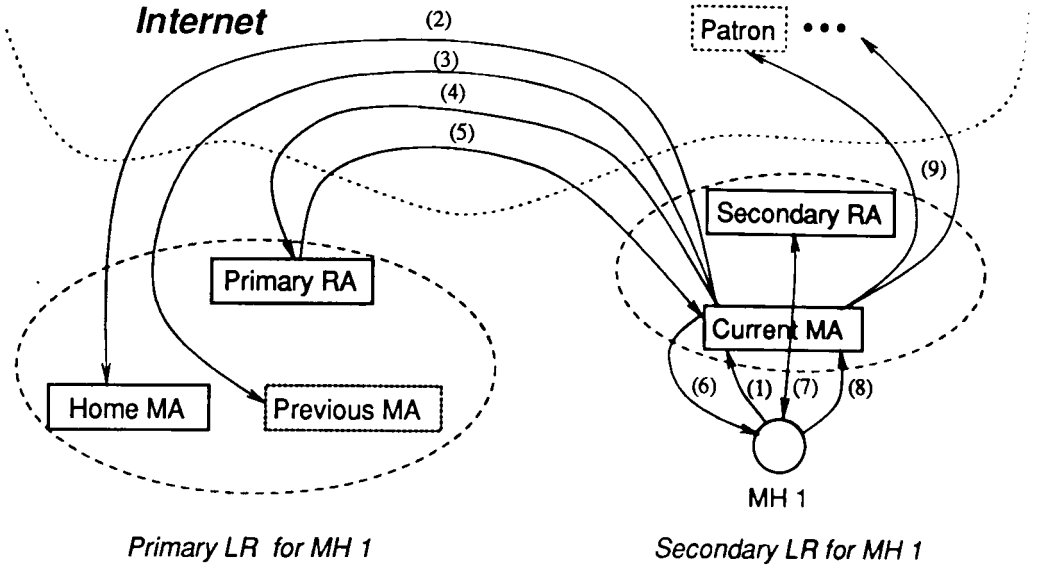


Figure 4.6: Registration Sequence (Patron Service)

4.4.2 Crossing the Home LR

Now, let us assume that a host has crossed its home LR boundary; once again, and so will start a patron service. Figure 4.6 shows the sequence of the registration procedure when a host, MH 1, crosses out of its home LR. After connecting with the current agent, the host sends a registration request to the agent (1), based on the moving history that it has preserved; the request may include some registration destination's addresses, such as the primary RA, the home MA and the previous MA. The current agent processes the registration for each destination in turn – the home agent (2) where an entry of the forwarding list is newly built up, the previous agent (3) where the visitor list entry for the host is reset and an entry in the forwarding list is newly created (thus, the back firing from the previous agent is deliberately omitted), and finally the redirection agent (4) where the corresponding entry in the redirection list is updated to point to the current agent.

Here, the primary RA, as a previous RA for the host, would find that the current mobility binding for MH 1 is out of its service boundary. Therefore, it determines that the host has just crossed its local region, and replies with a cross_LR notice packet to the host (5). On receiving the cross_LR notice, including the confir-

mation of registration success (6), the host requests the definition of a new local region, here it has a secondary RA. The RA creates a new redirection entry for the host and returns the result (7). Finally, the host executes a patron service to the current MA (8), if it has patrons. Finally, the current MA tries the service to each patron hosts in turn (9) on behalf of the host. When a host returns back to the home LR, the registration procedure is analogous to the process just described.

4.4.3 Moving within the Secondary LR

Let us examine the case of a host moves around within the secondary LR. Figure 4.7 depicts the location and routing operations after a host has connected with its secondary LR. Now, whenever a host moves around within its secondary LR, it has to register with the home MA, the previous MA if it had one, the current MA and the secondary RA. In the figure, those entities are shown as a dark rectangles (the previous MA is deliberately omitted). The number of registration entities is the same as at the time the host moves around in the home LR. This is still one more when compared with the basic scheme. The secondary RA is now an extra registration, just as if it were the primary RA in the case of the home LR. The patron hosts should maintain the MA's address sent to them at the time the host joined its secondary LR, as the host's current location, until the host returns back to the home LR.

In the basic scheme, packets are routed to the host's home agent, regardless of where they originated. With the benefit of the redirection facility, packets from a source within the host's local region, the primary or secondary LR, are routed as when the host is in its home LR. Moreover, some packets from a patron host within the secondary LR are directly delivered to the current MA, without being tunneled by the secondary RA, though only if the patron and the current MA are in the same subtree with the secondary RA as a centre (refer to the left side of Figure 4.5). Packets from a patron host outside the secondary LR are tunneled to the mobility agent – the one for which the patron service has taken place when the host first joined the secondary LR. Therefore, this agent has a responsibility for directing these packets to their tunneling end point, we will call it a *guide MA*.

When the guide MA receives (and decapsulates) packets tunneled by the patrons,

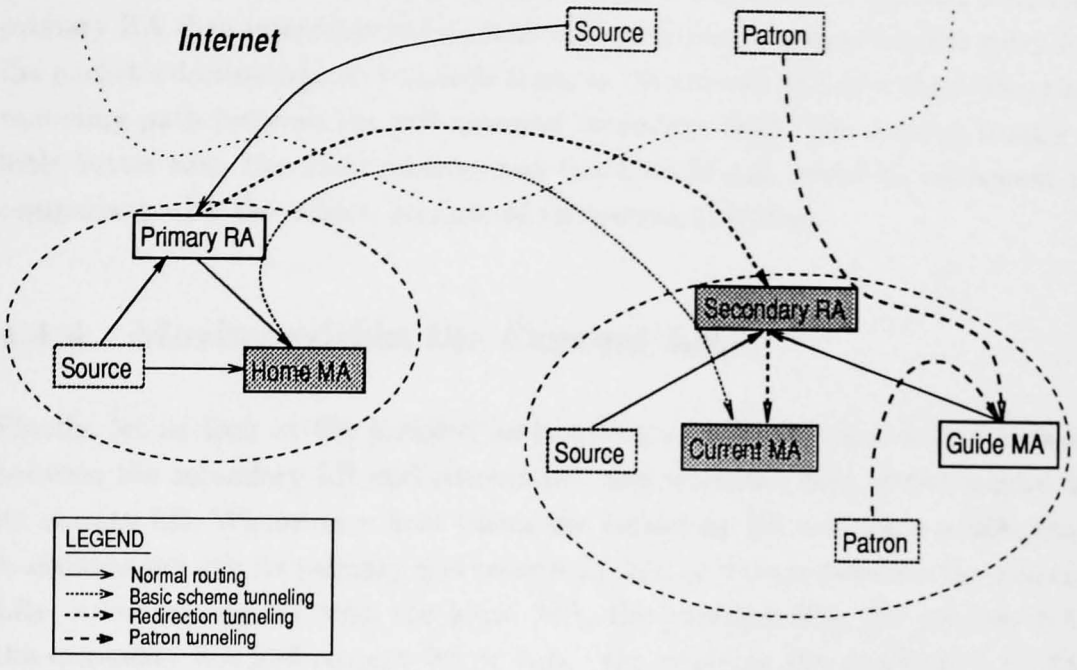


Figure 4.7: Routing Paths (Moving within the Secondary LR)

if the MA has a visitor list entry for the destination, it directly delivers them. If the MA has a forwarding list entry, it re-tunnels the packet to the current MA. If the guide MA has no mobility binding for the destination, it sends the packet using the normal routing mechanism (i.e. the packet heads for the destination's home agent). When the packet arrives at the secondary RA, the RA intercepts and redirects the packet to the current MA because it should have a redirection list entry for the packet (see the redirection tunneling paths between the secondary RA and the current MA). Therefore, the routing path for the patrons (so most traffic from outside its local region) is much close to the optimal routing even if the host has moved out from the home area.

As a result, when a host moves around within the secondary LR, most packets can still achieve optimal routing with only one extra registration on the secondary RA just as when the host moves around in the home LR. Also, it is not necessary to tell the outside world about movements within the secondary LR, except for the home MA. A source, which is situated somewhere in the Internet but is not a patron to the mobile host, would use the normal routing mechanism to send packets. The

packets are intended for the home MA, so they will arrive at the primary RA. The primary RA then intercepts the packets because it has a redirection list entry for the packet's destination, and tunnels them to the current MA (see the redirection tunneling path between the primary and secondary RA). The routing is only a little better than the basic scheme, but this kind of call would be infrequent in comparison with the others, because of the patron definition.

4.4.4 Moving within the Current LR

Finally, let us look at the location and routing operations when a host crosses between the secondary LR and current LR, and when the host moves around in its current LR. Whenever a host leaves the secondary LR and joins a MA, that is situated outside its primary and secondary LR, or crosses between the current LRs, it should register with the home MA, the previous MA, the primary RA, the secondary RA and current RA in turn. On receiving the registration by the secondary RA, the RA, as the previous RA, will find that the host has crossed its service boundary, so it sends the cross_LR notice to the host. When registered with the current MA, the host then assigns a new local region (the current LR), and registers with the current RA. In this cross_LR notification, the patron service will not be carried out; the patrons still have the old mobility binding, which points to the guide MA in the secondary LR.

When a host moves around within a current LR, the registration procedure and packet routing paths are quite analogous to those at the time the host moves around in the secondary LR. Figure 4.8 depicts the registration entities and the routing paths when a host has moved to the current LR. In comparison with the basic scheme, the extra registration overhead still requires one more place, the current RA, including the home MA, the previous MA and the current MA (these are shown as a dark rectangles, but the previous MA is deliberately omitted). Packets from a source within the host's local regions, the primary, secondary or current LR, are redirected by the corresponding RA; their routing paths would be close to optimal, whilst the basic scheme forwards the packets at the home MA only.

However, packets from patron hosts will still be sent to the guide MA in the

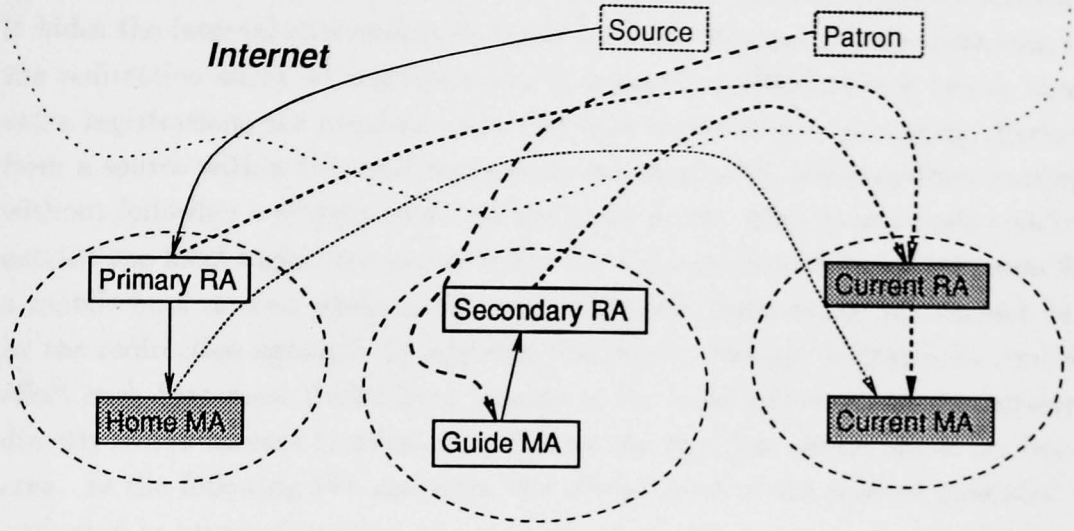


Figure 4.8: Routing Paths (Moving within the Current LR)

secondary LR, because a patron service has not taken place after then. When the packet arrives (and is decapsulated) at the guide MA, it then is sent to the destination's home agent because the guide MA has no binding for the destination (of course, if the guide MA is the previous agent the destination has just left, the packet would be directly tunneled using the forwarding list). When the packet arrives at the secondary RA, the RA intercepts and redirects the packet to the current RA as it has a redirection list entry for the destination (see the redirection tunneling paths between the secondary and current RA). When the packet arrives at the current RA, it may tunnel the packets, if necessary, to the current MA because the secondary RA could have an old binding for the host (note that the host registers with the secondary RA with the cross_LR notice, but not for internal moves within the current LR). This one-step indirect routing of the packets from patron hosts is a bid to limit the patron service to the case of crossing the home LR[5], to save costly registration on the secondary RA for frequent moves within a local region.

In summary, the local region, mostly through the redirection agent, plays a decisive

[5]Clearly, patron services each time that a cross_LR notice is generated are too expensive from the point of communication overhead. Moreover, the set of patrons for a mobile user may vary with time and its locale in which it currently stay.

role in the location and routing operations in support of internetwork host mobility. It hides the internal movements from outside with only one extra registration on the redirection agent within those area it currently resides, even if two or three extra registrations are required when the host crosses the local regions. Packets from a source within the local regions are delivered with nearly optimal routing, without following a lengthy route via the home agent. Also, source hosts residing outside the local region are permitted to use incomplete location information for a mobile host to send packets; these are correctly forwarded to the current MA by the redirection agent(s). In addition, the patron concept heightens its routing effect such that most traffic from outside of the local regions is nearly delivered directly to the current location, even though the host has moved out of the home area. In the following two chapters, the effectiveness of the scheme presented is evaluated in terms of location and routing optimization, by using a simulator.

4.5 Other Considerations

The main philosophy of our scheme is to confine the effect of host mobility to the area where the host stays most of the time, and to those source hosts which are most likely to call again. It is based on our belief that location and routing efficiency is the responsibility of the network infrastructure, and that location can be further improved by partially sharing its duty with the mobile host itself. These approaches fit naturally within the existing Internet, and are well accommodated with the locality properties of host moving and packet traffic in which we are primary interested. There are two important issues which have an influence on the scheme.

Orphan packets

In the system model, a mobile host may move around between the mobility agents even in the midst of a data transfer. Here, packets tunneled from the home agent after the host disconnected with the previous agent would be delivered to the previous agent until the home agent stops forwarding to this agent, that is until a registration takes place on the home agent. The previous agent has an entry on the visitor list for the packet's destination, but the corresponding wireless network can not deliver the packets because it has no network connectivity with the host.

These packets which lose their route are called *orphan packets*. It is an inevitable part of the host moving procedure due to overheads such as leaving a cell, joining a new cell, and processing registration, whilst incoming packets are still arriving the previous agent. Likewise, the same situation takes place at the home agent and the redirection agent.

In many cases, the systems need to forward the orphan packets to the new mobility agent, and thus eventually to the mobile host. In contrast to a host state hand-off, this can be regarded as an application hand-off, and its procedure depends on the quality of service of the application. Some applications may ignore the orphan packets, on the other hand, some applications may not permit any packets to be lost. However, an application never knows the new address of a moved host, and the orphan packets address the current mobility entity as their destination address. Therefore, each mobility entity is assumed to have an ability to buffer the orphan packets when it cannot forward them any more, and to retry sending the packets. If a new route (that is, a forwarding list entry) for the destination is created, the entity must try to tunnel them to the new location. If the wireless network resumes network connectivity with the host[6], the corresponding mobility agent would directly deliver the orphan packets to the host. This is why the previous forwarding list is meaningful within a local region even though a source host never uses the previous mobility agent address in our location strategy.

Fault Tolerance

Robustness is another great concern for the designers of mobile computing infrastructure. The two main problems faced are the loss of registration packets and crashes of entities that maintain location caches. In our approach, the current agent which executes as a proxy for a mobile host during the registration uses an all-or-nothing method to update the corresponding mobility bindings. This protects against crashes of those agents and losses of the registration packets in the middle of registration. When a mobility entity restarts from a crash, it must minimally recover the mobility bindings for the hosts under its control. To do this, it is assumed that each mobility entity has the facility to retrieve mobility bindings from another entity using techniques found in distributed systems or databases.

[6]In the real world, it is regarded as a temporary disconnection due to network faults, which is common for a wireless medium

In addition, disconnection in a mobile environment should be treated as distinct from failure as it is a voluntary act. A mobile host can inform the system of an impending disconnection prior to its occurrence. The detached participant must then execute a disconnection protocol in order to function in a stand-alone mode by downloading any data that it may need later. The detaching participant may have to offload any data or state information so that the host can smoothly integrate with the system, even via a different attachment point than it disconnected from. To do this, we assume that the host's home agent has a responsibility for responding to disconnection requests [42]. The home agent will provide buffering for incoming packets during disconnection. When the host connects again, it hands off the stored state to the new current agent.

Chapter 5

Design and Implementation of the Simulation

In the previous chapter, we described in detail the control structure of the local region and patron service, by describing the registration procedures and packet routing paths. In this chapter, we present the simulation built to evaluate the effect of the scheme proposed in terms of location overhead and routing efficiency, and then eventually system performance. The various simulation parameters are described along with a description of the simulator. An internetwork model for the simulation domain is then defined. The implementation details of the scheme are set out, and some implementation related issues are discussed.

The simulation has proved to be a valuable and effective emulation of an Internet-wide host mobility. It was invaluable for assessing protocol correctness, as we were able to iterate and refine the design as problems were discovered. In fact, it is very difficult to capture overall system behavior without the use of a simulation environment. The simulation was even more essential as the University does not have any wireless networking hardware. The simulation does not provide a fine-grained model of each network entity's internals as we are primarily interested in the behavior of the location update and packet routing of the internetwork host mobility environment.

5.1 Simulation Objective

With this simulation, we tried to provide a firm basis for the various location and routing optimizations, by investigating the related issues, such as mobility workloads, location propagation strategies and location cache's places, which affect the system performance as a whole. The first aim of the simulation is to justify that the design approaches suggested in this thesis is correct and suitable for location and routing optimization in support of Internet host mobility. The simulation is also utilized to find out how much the location overhead for LROP scheme is required and how the tunneling roles are distributed between the location cache agents. It then shows how much the location efforts is incorporated to improve routing efficiency, so the effectiveness of the host mobility support system.

To do this, the LROP scheme is compared with its basic scheme and another scheme (two concepts, local region and patron, are individually investigated to shows its effect). The simulation carry out based on two main dimensions: the number of identified event's occurrences as operating costs of the corresponding schemes and the network occupation time as accumulated networking infrastructure overheads. Three evaluation categories are then presented: the registration overhead, the encapsulation details as packet tunneling effects of the registration, and then the network occupation time for data packets eventually to show the results of the location (registration) and routing optimization. Finally, the number of direct routing is given to provide some rational of the design choices.

5.2 The Simulator

The simulator employed is based on the *netsim* (or *mitsim*) [29], release version 2.2, developed by the MIT LCS Advanced Network Architecture group in 1993. It can simulate anything that can be modeled by a network of components that send messages to one another. The simulator is written in the C programming language, and uses an event driven technique in order to manage processes and exchange packets between network components. It supports an arbitrary number of network nodes and internetworking topology for them. Datagrams are routed as in the Internet, by choosing a route based on the routing information avail-

able in the given router. An arbitrary number of mobile hosts can move among the mobility agents. The internetwork configuration, the network parameters for each network entity, and the simulation parameters for calling and moving control are configurable at run time. The simulator gathers extensive statistics during each run, and presents a summary upon completion. It also supports an X window interface to allow interactive use, and to display varying amount of status information as the simulation proceeds.

The simulator has been broken up into two layers – one for the core simulator routine and one containing the components used as basic building blocks. The core routine only provides the means to schedule events and to communicate with the user. Its most significant role is as an event manager, which triggers events on each component based on effective times by sorting them into an increasing order. A component then becomes active and, after doing something according to its event type, if necessary, it produces the next event(s) and puts them into the event manager. There are three event classes: command, normal and private. A command event does something to initialize the simulator, such as creating a component, connecting the components, and building routing information. A normal event has to do with running the simulator; such as send, receive and ready. Command and normal events are applied to all components. Also, private events can be defined by each component, in order to arrange their own operations. For example, three important private events were given to the mobile host component: move, call and patron. These events are mainly controlled by the moving and calling discipline specified details in section 6.2.

The core also includes the I/O routines and various tools (lists, queues, hash tables, etc.) that can be used to build components. It also provides the means of displaying the topology of the network and parameters of its operation, and to allow the user to modify the network and control parameters the simulation. Some of this interface has been modified to show the host mobility aspect of our simulation.

The model being simulated and the action of the components is entirely determined by the action routines controlling the components, not by the framework of the simulator. Each instance of a component has a set of data structures and action routines. The data structure is used to store any information needed by the

components, as well as a set of standard information needed by the simulator for every component. The components schedule events for one another to cause things to happen, and send events to one another via the event manager. When an event for a component fires, the corresponding action routine is called. Therefore the action routine defines the behavior of a component.

5.2.1 Component Description

Three different types of components have been built for this simulation: network entity – *Internet, Ethernet, Wirelessnet*, mobility supporter – *Mrouter, Magent* and mobile host – *Mhost.* All components of the same type share some common action routines and network parameters but they have different parameter values which identify their characteristics (the values used are mainly drawn from existing work [12, 29, 41]) as shown below. Each component also has some private routines to identify its own actions. An action routine is called for each event type that the event manager causes to happen to a component. When the simulator starts to run, it inputs these various configurable parameters for each component and the configuration structure between the components.

Internet

A component to simulate a conceptual Internet model and its interface to other components. In this simulation, it acts like a full duplex point-to-point link. When an Internet gets an event from a mobility supporter, it schedules the event for the next component after a delay which is calculated as the network occupation time of the event. During that delay, it considers itself to be busy. When a component sends an event to the Internet, it marks itself as busy until it gets a ready event back from the Internet, and puts the packet on a queue. The Internet never gets another packet from a component that is busy, though it may get packets from other components. The Internet component can be connected to any number of Mrouters.

Ethernet

A component to simulate a typical subnetwork, Ethernet, and its interface to other components. If this component is busy transmitting a packet when it gets another

packet from a different mobility supporter, it puts the packet on a queue and tries to send it later. This behavior mimics reality, where the various interfaces wait for the network link to be free before transmitting. Any number of Magents can connected to an Ethernet component.

Wirelessnet

A component that acts very much like an Ethernet, though it has different network parameters from the Ethernet. It also has some private routines to treat packets which have lost their route due to host moves or disconnections. The Wirelessnet component can be connected to only one Magent, but it can be connected to any number of Mhosts.

A network entity type has two configurable parameters:

	Internet	Ethernet	Wirelessnet
Link speed (Kbit/sec) :	128	100000	1000
Latency (μsec) :	1000	500	10000

Mrouter

This component routes packets between network entities. When it gets an event from a network entity, it consults the routing module to figure out where to send to next, then it schedules the event for the next step after a delay. During that delay, it considers itself to be busy, and any other incoming packets are put on a queue to be processed later. In addition, it contains a queue for each attached network entity, and if that entity is busy, it will queue packets destined for it and not send them until the entity is ready. In addition, an Mrouter maintains the redirection list which records the redirection information sent from an Mhost, and if necessary, encapsulates packets passing through itself using the list. An Mrouter component can be connected to a set of Ethernets or the Internet.

Magent

This component acts very much like an Mrouter. But an Magent has the responsibility of processing a sequence of registrations. To do this, it preserves the complicated registration routines and their related states. Three mobility bindings, the home list, the visitor list and the forwarding list, are maintained. An Magent can do encapsulation and decapsulation as well using the forwarding list.

An Magent component is connected to a Ethernet and a Wirelessnet.

The mobility supporter type has two configurable parameters:

Delay in processing a packet (μsec) : 2000
Speed of each component (μsec/Kbyte) : 1

Mhost

This component simulates a mobile end system. An Mhost can be disconnected from the Wirelessnet it is currently connected to and can then be connected into another Wirelessnet. Whenever an Mhost initiates a move, it issues a registration packet to its current Magent. The Mhost has to make the decision for the entities for which registration takes place, even if the Magent actually processes the registration sequence on behalf of the Mhost. To assist in this decision, some historical information is preserved, such as home, current and previous Magent addresses for call and move control, and home, current and previous Mrouter addresses for local region control. The Mhost generates a sequence of data packets. The time interval in between successive moves and calls is based on a Poisson input parameter. Mhost also maintains two lists for the patron service control: patron list and calling list. An Mhost component can be connected to a Wirelessnet only.

Three parameters are configurable for Mhost:

Mean packet processing time (μsec) : 500
Packet processing time variation (μsec) : 100
Poisson input for packet generation (pkts/μsec) : 0.0000001

The network parameters for each component and their topological structure, including related routing information, are specified in a configuration file (see appendix A).

5.2.2 Simulation Parameters

In mobile computing, there are two important events: "calls" which refer to the number of data packets a mobile host has sent, and "moves" which stand for the number of moves the host made, in a given period of time. The *call to mobility*

ratio[1], CM_{ratio}, the ratio of the rate (reciprocal of the mean) of calls to the rate of moves, is a critical characteristic in this simulation model. It is defined as the average number of calls made by a mobile host per move, and is used for regulating the frequency of calls and moves for each mobile host. Another important parameter is the symmetric rate, S_{rate}, which captures the moving and calling discipline. It defines the different direction rate of calls and moves' destination (see subsection 6.2 for details). In fact, CM_{ratio} and S_{rate} are experimental parameters but, in our simulation, are explicitly set (and configurable) for each simulation run.

Each mobile host generates call and move events randomly. In general, the number of events in some unit time is often modeled as a random variable which has a Poisson distribution. Given the mean number of events in a fixed time interval, which is an input parameter for the Mhost component in this simulation, the time interval between successive calls can be worked out with the Poisson value. Likewise, the time interval between successive moves can be calculated by the product of CM_{ratio} and the Poisson value. Each Mhost component repeatedly fires the call or move event with the time interval computed, and runs a corresponding routine.

There are several other aspects of the simulation which are governed by a uniform random distribution. Most prominent is the choice of the destination of call and move. In the case of host moving, an Mhost first randomly chooses a local region. Then the prospective wireless network, a Wirelessnet, is chosen randomly within the local region already decided. For a call, a source host randomly chooses a destination local region, then a destination host within the local region. Both choices are done with considering the calling symmetric rate which captures the moving and calling discipline. A random perturbation for the packet processing delay on Mhosts is computed using an input parameter – packet processing time variation. In addition, the size of the data packet is randomly chosen between 32 bytes and 8192 bytes. The simulator accepts a seed number whenever it starts to run, so that the simulation is reproducible by using the same seed number for each run, that is, with the same sequence of random numbers.

[1]This term was initially introduced in [4, 31], but they defined calls from the receiver point of view, that is, the number of calls made to a host.

5.3 Network Model

In order to simulate a scheme in the internetwork environment, the network configuration needs to be big enough in terms of the number of links and nodes. However, it may be restricted due to simulator constraints and/or computing power limitations. We here are interested in verifying clearly how efficiently the suggested scheme works and, later, for comparison with another scheme. To do this, the network model used for our simulation is shown in Figure 5.1. In addition to Internet, it consists of 23 subnetworks, mobility routers, 25 wireless networks and mobility agents, and 60 mobile hosts. In our simulation, each Wirelessnet is initially connected to a set of mobile hosts with from two to five mobile hosts.

The local region is deliberately defined by the configuration file, as shown in Appendix A.3. It is used to regulate the registration process and the host moving and packet calling discipline. In our experiment, 4 local regions were sufficient to examine the various cases for all schemes we have simulated. For simplicity, we assume that all mobile hosts within a local region have the same redirection agent, which sits on highest level of the local region; that is, MR 10, MR 20, MR 30 and MR 40. Two local regions amongst them are made with a 4 deep subnetwork hierarchy from internet to mobile host, one has 3, the rest have 2 levels. This difference in network hierarchy (depth) is important for generalizing the effect of the local region.

Even though the Internet component is modeled as an entity which stands for an internetworking abstraction, addressing and routing are analogous to the current implementation of the Internet protocol. For the sake of completeness of the internetworking convention, every component, even the Internet component, is assigned an address shaped like an IP address. Each component identifies the others using their address. As described in subsection 4.1, the Magent address is used to indicate an Mhost's current location, at least for those currently within the service boundary of the Magent.

Whilst the address is used as the identifier for network entities, it is also used to choose the next route by the mobility supporters. Each mobility supporter routes packets passing through itself as a gateway would. In addition to the normal

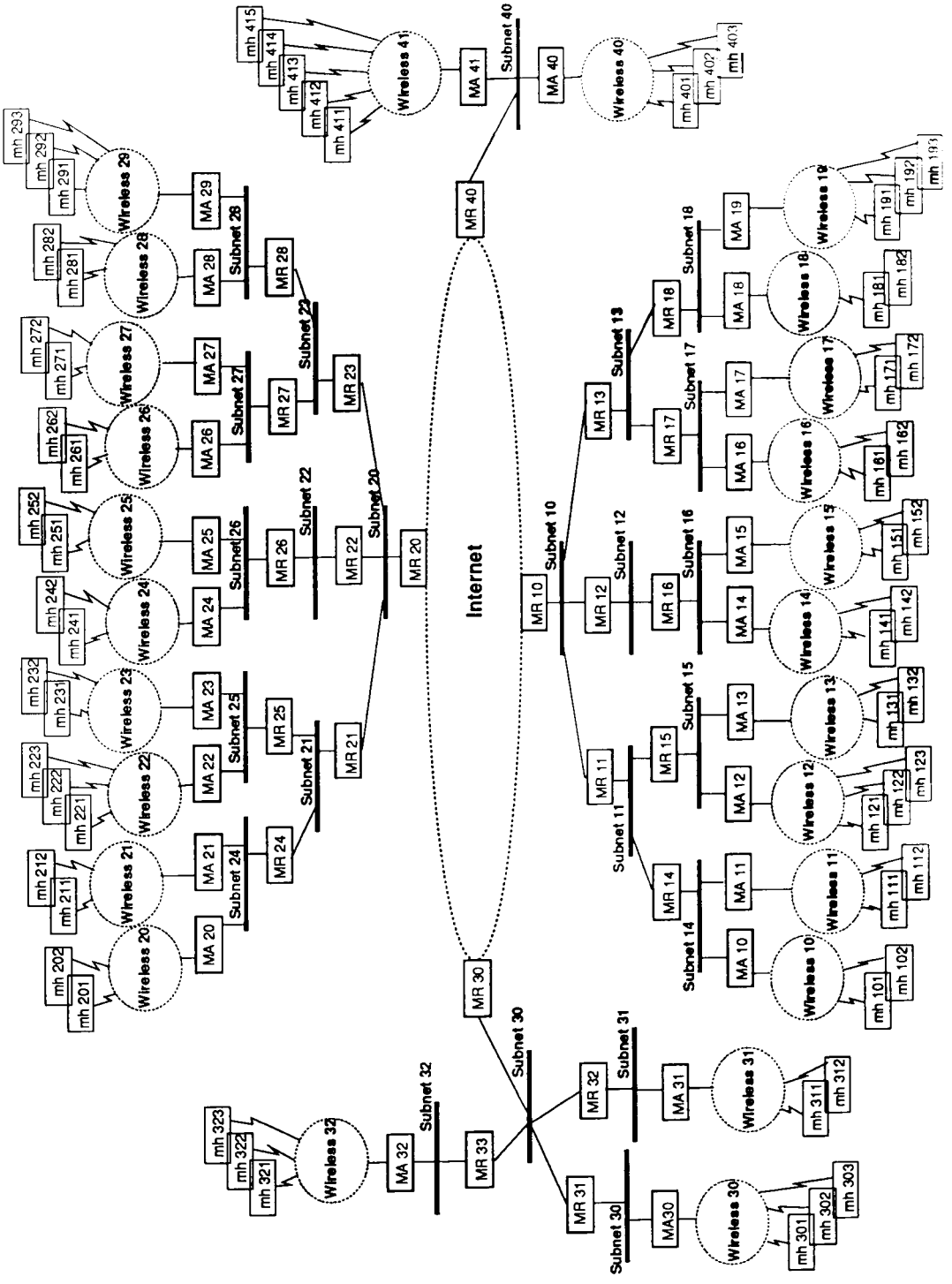


Figure 5.1: The Network Model for the Simulation

routing mechanism, the mobility supporters and the mobile hosts have the ability to do encapsulation and/or decapsulation. On receiving a packet, they first check the forwarding list, redirection list or calling list respectively. If they find an entry for the packet's destination, the packet is encapsulated to a mobility supporter which might serve the destination host. The network number of the mobility supporter's address, as a tunneling end point, is used for a routing decision to deliver the packet. After the mobility supporter decapsulates the packet, the host number of the host's address is utilized for eventual delivery of the packet. The routing information is initially set from the configuration file, as shown in Appendix A.4 and A.5.

5.4 Implementation Description

In order to carry out the simulation, we implemented two main protocols: a registration protocol to create a set of mobility bindings, including the patron service, and a tunneling protocol that forwards packets to the Magent currently serving the destination mobile host. Based on their primary roles, we shall simply refer to these two protocols as LROP (Location and Routing Optimization Protocol). The registration procedure is invoked by two different events, the move event and the cross_LR event which is issued when a host crosses its local region, so they are referred to as move registration and patron registration respectively. Another important protocol is a beaconing and/or solicitation protocol enabling the mobile host to identify itself to a mobility agent, and then obtain network connectivity. It is essentially an interior routing protocol within a wireless network. This protocol is beyond the scope of this work (and independent of LROP), so it is assumed that network connectivity is implicitly accomplished during the moving procedure in this simulation.

Two different approaches are proposed for incorporating tunneling information into the existing IP packet format: a new IP option code and the encapsulation approach. The new IP option code approach [70] utilizes the option processing facility of the IP protocol, which is primarily for network testing or debugging. On defining an additional type of IP option (the IP protocol has eight types), the source and destination addresses of the IP header are moved into the IP option

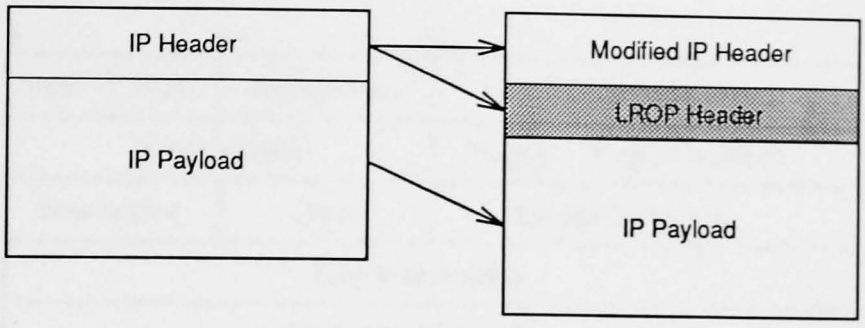


Figure 5.2: Building the LROP Packet

field, and the existing IP header is changed to the tunneling information. It needs extra processing in every router the packet goes through. Also, the total length of IP options is defined with 40 bytes as its upper limit, so little space is left for other IP options.

With the encapsulation approach, one simple way is IP within IP [8, 33, 73]. A duplicated IP header is inserted into the original packet immediately following the existing IP header, and some fields of the existing IP header are modified to reflect the tunneling. This increases the network overhead by copying unnecessary fields of the IP header. Even if it is logically possible to do nested encapsulation, the appended information in the header is limited to the size of an IP header. Johnson [39] proposed a more efficient variation of the IP within IP method, that is, a protocol header encapsulation. This does not add a complete new IP header, but rather modifies only the fields necessary in the existing IP header and builds a protocol-dependent header between the existing IP header and the payload. Figure 5.2 illustrates the encapsulation process for LROP protocol using the protocol header encapsulation.

Some fields of the existing IP header are copied into the new header, and then altered to reflect the tunneling. Once the LROP header is added, the packet uses only normal routing mechanisms for delivery to the designated destination – packet tunneling. When the destination receives the packet, the LROP header is removed from the packet, and the original IP header is reconstructed. The packet is then directly delivered to the mobile host, or re-tunneled to another destination if this is necessary. Our implementation uses this method for tunneling because it

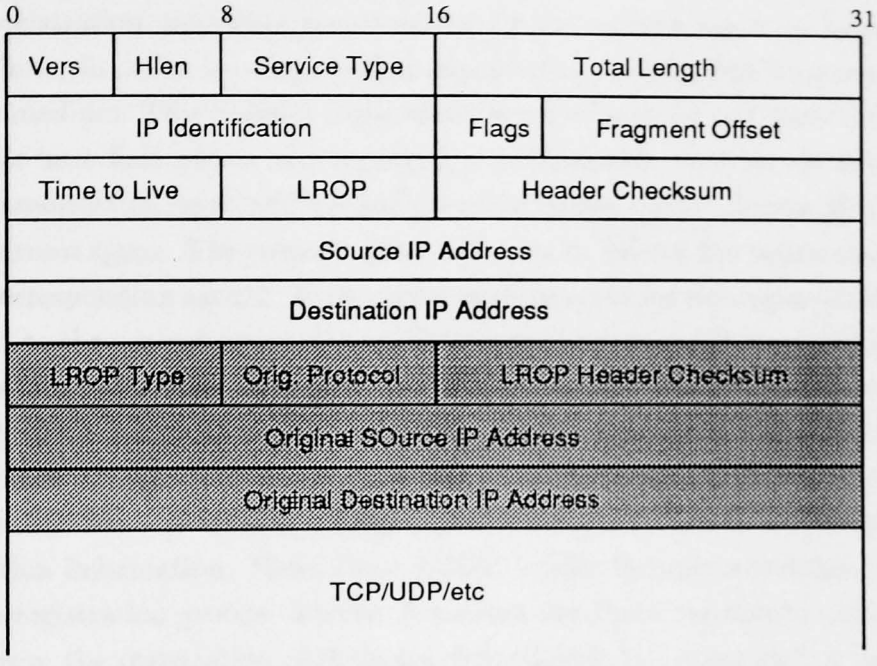


Figure 5.3: The Packet Format for Encapsulation

is among the most flexible, and has considerable savings in space overhead in the packet. Moreover, it allows a single extension format to handle the registration as well as the encapsulation, as described below,

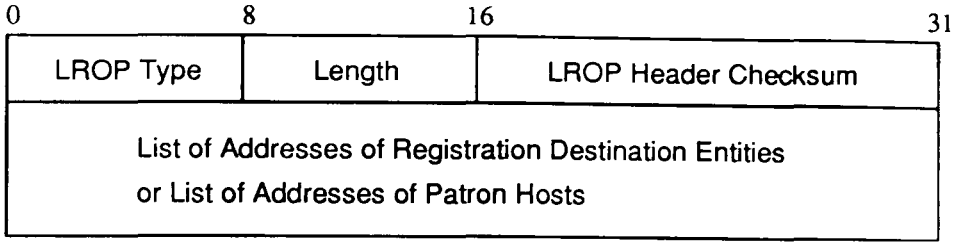
Figure 5.3 shows the packet format for encapsulation. The shaded portion in the figure shows the LROP header, and the other part is the IP packet. The mobility agent or mobility router, which preserves a mobility binding for the packet's destination, partially copies the existing IP header into the LROP header (protocol number and source and destination IP addresses). It then modifies the destination address field of the existing IP header to that of the foreign agent which is currently serving the destination host, and the source address field of the IP header to its own IP address. The protocol number in the existing IP header is replaced by the IP protocol number indicating LROP. The total length of the IP header is increased by the size of LROP header, 12 bytes, and then the header checksum of the IP header is modified to reflect the changes to the IP header. Finally, the LROP type is set to the encapsulation code, *data_encap*, 40, and the LROP header checksum is computed.

With the indirect registration model (see details in the section 4.2), the sequence of the registration procedure is carried out by the current agent on behalf of the mobile host, in order to adapt to the asymmetric communication nature of the wireless medium. This indirect registration sequence can be split into three parts. A mobile host first passes the registration information, such as its home agent address, redirection agent address and previous foreign agent address, if it had one, to the current agent. The current agent then tries to deliver the registration packet to the corresponding agents. Each corresponding agent for the registration returns its result to the current agent after carrying out the appropriate registration work. On receiving all replies from the destination agents (that is, in all-or-nothing fashion), if the registration has succeeded, the agent passes a reply packet to the mobile host. If the host does not receive the reply within the expected time, it will retry the registration. Each registration group needs a different set of registration information. Here, three LROP header formats are defined for each of these registration groups. Figure 5.4 shows the three registration headers. As it is shown, the registration packets are defined with the same format as that for encapsulation, with the same protocol number as LROP in the IP header, but with a different LROP type in the LROP header.

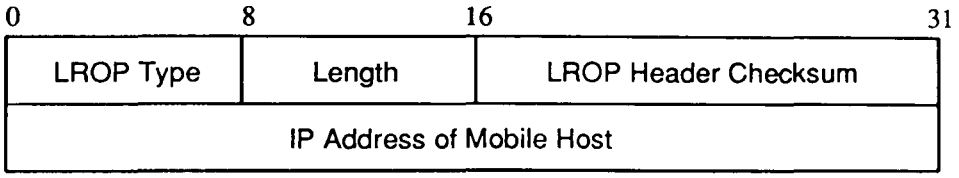
The original protocol field of the LROP header for encapsulation is no longer needed for registration, so it is used to specify the length of the LROP header. The length varies on where the registration was issued and what type of registration it is, i.e. move or patron. In the case of a move registration request, the LROP header has the IP addresses of the home agent and the redirection agent(s), and when it is just leaving its home agent, the previous foreign agent as well, so the length is usually 12 bytes or 16 bytes, or at most 24 bytes when crossing the local region. For the patron registration, the length depends on the number of patron hosts it wants to serve. The LROP header for a registration request from the current agent to registration destinations needs 8 bytes, regardless of whether it is a move or a patron registration. The registration reply header uses only 4 bytes. Each registration group has the following LROP type:

Registration requests from mobile host to the current agent:

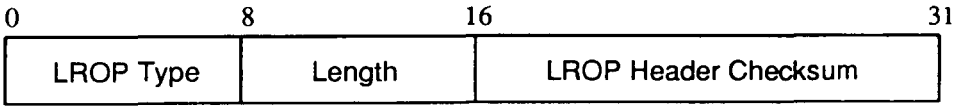
Move_Reg_Req : 50 Patron_List_Req : 30
Patron_Delete_Req : 31



(A) Registrations reguest from mobile host to the current agent



(B) Registrations request from the current agent to registry destinations



(C) Registration replies

Figure 5.4: The LROP Headers for Registration

Registration requests from the current agent to registry destinations:

Home_Agent_Req : 60
Previous_Agent_Req : 61
Redirect_Agent_Req : 62
Patron_List_Req : 30
Patron_Delete_Req : 31

Registration replies:

Move_Reg_Reply : 51
Home_Agent_Reply : 70
Previous_Agent_Reply : 71
Redirect_Agent_Reply : 72
Patron_List_Reply : 32
Patron_Delete_Reply : 33

Let us look at the registration process in detail. When a mobile host picks a prospective mobility agent to move to, a move event is issued[2]. It then builds up a registration request packet with the type *Move_Reg_Req*, and waits for the request's reply. The LROP header of the request packet includes a list of registration destinations. On receipt of the packet by the prospective agent, the agent saves some information such as who wants to be registered, and to whom it will be done. It first builds up a registration request packet by copying the first element (implicitly, if it is a home agent address of the host) of the destination list in LROP header received into the destination address field in new IP header. The source address of the new IP header is set to the prospective agent's address. The protocol number of the IP header is set to indicate the LROP, as for the encapsulation packet. After inserting the mobile host's address into the LROP header of the new request packet, the agent sends the packet with the type *Home_Agent_Req*, and waits its reply.

Similarly, the agent sends a registration request packet to the redirection agent, and the previous agent, if there was one. Each registry agent receiving the registration packet updates (or creates) the corresponding mobility bindings using the information of the IP header and the LROP header, and replies to the agent if the registration has succeeded. On receiving all replies from the registry agents, the prospective agent creates an entry on the visitor list for the host - now it becomes the current agent for the moving host. The current agent finally sends a registration reply, with the type *Move_Reg_Reply*, to the mobile host. Once the host receives the reply packet, it changes its state related moving history, that is, the previous and current local regions (including mobility agents). In a similar way, the patron service is carried out between the mobile host, the current agent and the patron hosts.

[2]In our implementation, on a move event, the host disconnects from the current agent, and sets network connectivity with the prospected agent in sequence. It is also possible to defer the disconnection until the registration sequence is finished – this is analogous to the host moving between overlapping wireless networks.

5.5 Implementation Dependent Issues

The LROP has been implemented with the protocol header encapsulation method, which requires minimal space for the protocol dependent information. The same packet format has been applied for registration, but only with a different type field. This approach does not need any extra processing by the in-between routers, and provides a simplified protocol definition. The following discusses some issues related to the implementation.

Fragmentation

The protocol header encapsulation approach for packet tunneling has a fundamental problem – fragmentation. When a gateway relays a datagram, if the datagram is larger than the MTU (Maximum Transfer Unit) of the outgoing data link, it is divided into several small fragments (fragmentation) and these fragments are reassembled into the original IP datagram at the destination (reassembly). If an encapsulated packet is fragmented by an intermediate router, the LROP header only exists in the first fragment because the LROP header is treated as data by the IP protocol. Once a datagram has been fragmented, the fragments travel as separate datagrams all the way to the ultimate destination where they must be reassembled. Even though IP reassembly is normally not done until a packet reaches its final destination, in the LROP design it is assumed that reassembly can be done by any gateway (RA or MA) along the way if necessary.

Let us consider the redirection agent's role, which is intended to support LROP tunneling for a local region. This would normally be done by the router that connects that network to the rest of the Internet, such that all packets from hosts that it is supporting must go through it. Thus, in order reliably to reassemble those packets, the redirection agent must have an absolute restriction on where it can be placed, such that no fragments can pass it on some other route (of course, this makes the redirection agent a single point of failure). If the fragments fail to be reassembled into the original IP datagram at the redirection agent, the fragments following (except for the first fragment) would be lost at the final destination because they do not have the LROP header. One possible remedy is simply not do any LROP tunneling of fragmented packets with such an intermediate router as redirection agent. However, on this issue, Steve Deering says in [22]:

"The fragmentation problem in the encapsulation method led me to use an IP option for multicast tunneling in the IP multicast code. A new option code approach was originally used, but that failed to work through some existing routers, so instead a Loose Source Route option was used but in a non-standard way. That turned out to be a big mistake because of the performance hit that IP packets with options suffer in most existing routers. As a result, we have had to undergo a major conversion in the MBone [3], to get everyone to switch to using encapsulating tunnels instead. I strongly suggest that you heed that experience and stay with an encapsulation approach. I don't think fragmentation will be a frequent occurrence anyway, because most IP hosts limit their packet size to 576 bytes when transmitting more than one hop, while most paths in the Internet can handle up to 1500 bytes without fragmentation."

ICMP Issues

The ICMP (Internet Control Message Protocol) mechanism is considered to be a required part of IP, in order to report error or control conditions to the source host of an IP datagram. Special attention should be paid to it in an environment with host mobility, where some hosts may be temporarily disconnected from the network (a rather regular feature) and datagrams may frequently expire the time-to-live counter due to routing (moving) loop or registration delay. These types of datagrams can be actively utilized by the mobility supporting system, or should be silently ignored by the ICMP software. Another thing to be considered is in the case of forwarding a datagram back through the same subnetwork from which it was received. In normal ICMP processing, an ICMP redirect packet is sent from the forwarding router to the source router (host), assuming the source is using a nonoptimal route. However, such return-back forwarding is common (but temporary) in the host moving environment. So it is reasonable to refrain from generating the ICMP redirect message for this case.

[3]MBone stands for the Virtual Internet Backbone for Multicast IP. IP Multicast is the class-D addressing scheme in IP implemented by Steve Deering at Xerox PARC. IP Multicast-based routing allows distributed applications to achieve time-critical (realtime) communications over wide area IP networks

Location update (and thus registration) packets can be defined with a new type of ICMP message, as [38] does. In the ICMP message delivery, there is an exception that ICMP messages are not generated for errors that result from datagrams carrying ICMP error messages [21]. A possible problem is that location update messages themselves may be lost or discarded. Here it is not possible to use the ICMP facility to detect or correct any message delivery errors; so additional work is needed to maintain reliability. Therefore, the protocol header encapsulation approach is preferable for the delivery of location update information.

Compatibility

From a practical point of view, LROP stresses two goals: backward compatibility and network and host scalability. To localize the impact of host mobility, we have tried to confine the local host movement to a designated area, the local region, in order to screen its effects from the outside. Only those source hosts most affected by a distant host movement are notified of the host's new location with need-based propagation. As a result, a mobility router is concerned with only the mobile hosts within its service boundary. The implementation of mobility routers would be limited to within the area in which the mobile hosts are actually attached. Moreover, a mobility router can be selectively implemented anywhere its influence is effective. Static hosts (even mobile hosts) could implement the patron service only. These hosts may be then capable of optimally communicating with most mobile hosts which are interested in contacting them. The other concern is related to communication between LROP entities and existing IP entities. When an IP entity receives LROP packets, that is, the protocol number of their IP header is LROP, it treats the LROP header as its data, therefore there is no action for LROP headers.

It is a matter of course that performance transparency is the primary concern of the LROP design. Most efforts are to optimize the trade-off between location overhead and routing efficiency in terms of the system performance. These stem from isolating local movement from the rest of the world and separating the location and routing roles for patrons from the network infrastructure. Also, it was worth trying to share the protocol processing burden between the mobile host and its current agent, to utilize the asymmetric communication nature of the wireless network. The performance details of LROP are discussed in the next chapter.

Chapter 6

Evaluation and Comparison

This chapter presents the results of the simulation study of the effectiveness of the local region and patron concepts in terms of system performance. Specifically, we devoted our attention to the operation costs imposed by each scheme, that is, the number (including network occupation time) of registration events, so as to show how large the location overhead is. We also look at the encapsulation details that result from the registration, in order to show how much of the location overhead is incorporated in routing efficiency, and the trade-off between them. Then, the data communication time and the number of direct routings are presented to show the eventual results of our location and routing optimization. These are actually the most significant outcomes because they reflect the system performance as well as the mobile user's satisfaction.

We first describe the host moving and packet calling scenarios used to conduct the evaluation, along with details of the simulation runs that were performed. Numerical results from the simulation are then summarized for each subject described above, according to the variance of the simulation parameters, such as the call to move ratio and the symmetric rate for directing the calling and moving scenario. A comparison with another protocol is provided with regard to location overhead and routing efficiency. Finally, a rationale is developed for certain features of our design, based on the simulation results.

6.1 Run Details

Simulation runs were done with a cross product of the 5 schemes, 10 calling and moving scenarios (see next section) and 39 CM_{ratio}s (ranging from 2 to 40[1]). In our experiments, it was found that these combinations exercised features of the various schemes simulated. For each simulation set-up, several random seed numbers were tried; they produced quite stable results due to the unformity of random numbers. Runs were performed with the same number of data packets for each scheme; a scheme with various parameters ran first, then the same number of data packets produced from the run was specified for any given runs of other schemes with corresponding parameters. Simulation was performed on a set of SUN SPARC workstations.

As a time sensitive parameter of simulation runs, the rate at which data packets are generated per unit time has to be properly adjusted in order to prevent the simulator from overloading; if the rate is too fast, the time taken to do a simulation run increases enormously due to the limitation of event (memory) management in the simulator, and that of computing power. In the experiments conducted for this simulation, it was appropriate that data packets were generated at the mean rate of one packet per 100 seconds in simulator time (thus, the time between the consecutive packets has an exponential distribution). The movement rate which brings about a set of registration packets depends upon the data packet generation rate multiplied by the CM_{ratio}.

The simulator works in an event-driven fashion. Components send event(s) to each other in order to communicate and to send packets through the network. The event manager provides a general facility for scheduling and sending events based on simulated time. The simulator time is managed in units of ticks, and one tick represents ten microseconds. The simulator maintains the time as an unsigned integer, so each run can last for about 12 hours of simulated time before overflow is reached (therefore the first run for each case was configured with 40000 seconds). Each simulation run processed about 24000 data packets from 60 mobile hosts; the number of registration packets was dependent on two simulation parameters, S_{rate} and CM_{ratio}s. Each entity in our simulation model calculates the time required

[1]In this experience, when CM_{ratio} is over 35, the simulation results were relatively stable.

to process a packet as it passes through, and records it in the packet. Each mobile host then gathers statistics concerning the packet's journey, such as the communication time required by a packet, the number of encapsulated packets, total network occupation time during the simulation and so on.

6.2 Moving and Calling Scenario

The most important aspect of the LROP design is the formalism of the locality property of the calling and movement pattern. In the internetwork communication environment, it is obvious that this approach is very general and covers most host moving and packet calling scenarios. Even though the moving and calling nature usually has some pattern in real life, it is difficult to characterize such things as how often a host will try to call, how long a host will stay with a mobility agent, etc. However, for a simulation study, it is necessary to define a set of typical scenarios which formalize the problem domain, and then to examine how each scheme proposed performs based on these scenarios.

To formalize these moving and calling patterns, we defined an important parameter – symmetric rate (S_{rate}). It is worth pointing out that calls and moves take place with the different destinations: calls for mobile hosts and moves for wireless networks. Therefore, the disciplines for moving and calling are directed with the different parameters as described below. For simplicity, a S_{rate} stands for each simulation discipline with the same parameter value. For host movement, S_{rate} is defined as the ratio between the time a mobile host stays at its home LR and the time it spends in others. It also reflects the ratio between the time a mobile host stays with its home agent and the time it spends with the others in the home LR. In the case of calling, the rate stands for the ratio between the calls bound for a host inside its home LR and those outside from its home LR.

In order to realize the moving symmetric rate, a moving reate, M_{rate}, is used for directing host moving destinations. Whenever a host decides to move, it will move to a mobility agent within the current LR with probability M_{rate}, and to those outside the current LR with $(1 - M_{rate})$. If a host is under a foreign agent within its home LR, it decides to move to another foreign agent in the home LR with probability M_{rate}, and $(1 - M_{rate})$ moves are to its home agent.

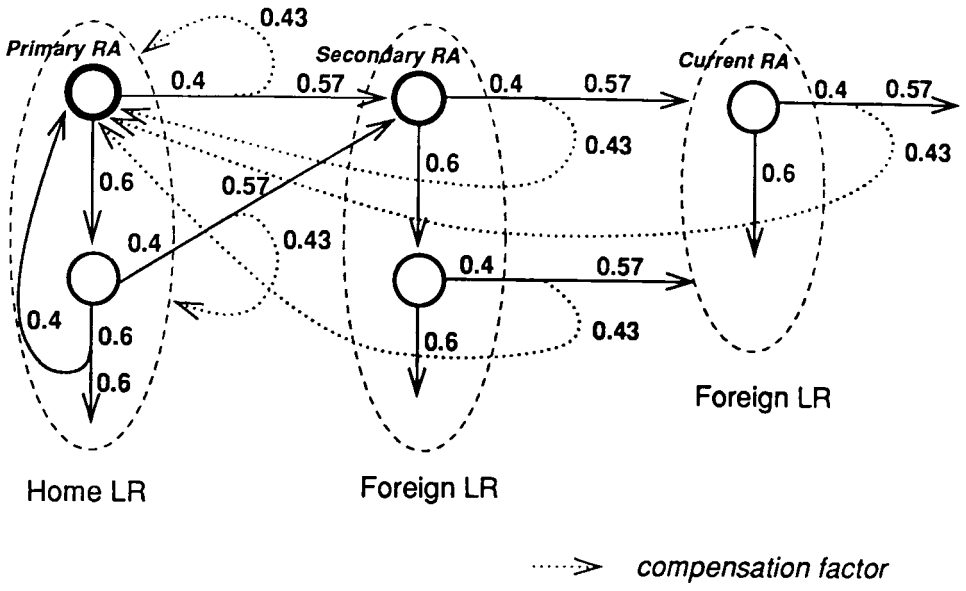


Figure 6.1: Moving Scenario for the Simulation (case S_{rate} 0.6)

However, the M_{rate} does not correctly direct the corresponding moving symmetric rate because some movements may still be done within the current LR (note that the moving symmetric ratio was defined between the time that a host stays at its home LR or agent and that of others, whilst the above moving directions are applied for each move event). We therefore define a compensation factor to balance the moving symmetric rate. This factor is defined as a rate that is applied to moves returning back to the home agent out of those moves which had decided to leave the current LR with M_{rate}. If a host moves away from its home LR, the factor stands for the rate with which the host returns back to an agent within the home LR, except the agent it has just left. After several runs with different factors, the factor was defined as 0.19, 0.30, 0.43, 0.56 for M_{rate} 0.4, 0.5, 0.6, 0.7 respectively.

Figure 6.1 shows a moving scenario with S_{rate} 0.6, so M_{rate} 0.6. When a host decided to move from its home agent, it moves to a mobility agent within the current LR with probability 0.6, and to those outside the current LR with 0.4. The outbound moves also goes to an agent within the home LR, except the agent it has just left, with probability 0.43. Similarly, if a host starts to move from an agent within the foreign LR, the host returns back to its home agent with 0.43.

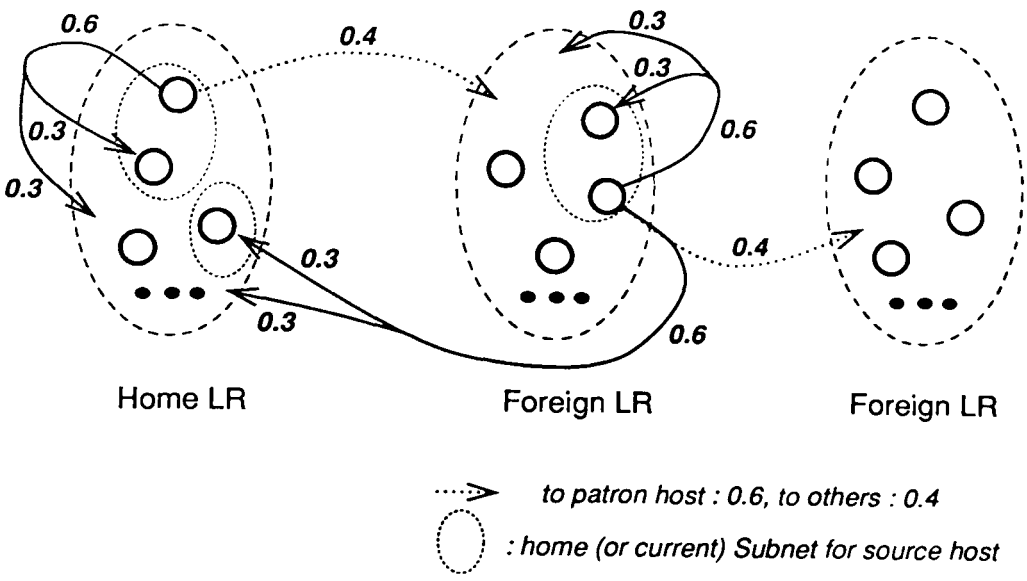


Figure 6.2: Calling Scenario for the Simulation (case S_{rate} 0.6)

For packet calling, a calling symmetric rate, C_{rate}, is defined for directing calling destinations. When a host decides to call, it will choose the destination host inside the current LR with probability C_{rate}, and to those outside the current LR with (1 - C_{rate}). In addition, half the inside calls go to hosts which have the same home (or current) agent as the source host; this reflects the locality of calling. To meet the corresponding calling symmetric rate, if a host is currently in a foreign LR, C_{rate} out-bound calls are sent to hosts in its home LR, and half of those calls go to hosts which share the home agent as the source host. Figure 6.2 shows a calling scenario with S_{rate} 0.6, so C_{rate} 0.6.

In addition, if a source host has a calling list (as a patron host) and it decides to make an out-bound call, it will send the packet to a (patron) host in the list with the probability C_{rate}, and to others with (1 - C_{rate}). In consequence, a host will receive nearly C_{rate} calls from the patron hosts; of course, in some cases, such as when the simulation has just started, a host may not have any entries in its calling list. Further, C_{rate} is used to define the patron rate, which is the number of patron hosts out of the total number of source hosts that visited a host. For each host, it defines the size of the patron list, which is managed in a LFU (Least Frequently Used) manner based on the patrons' calling frequency.

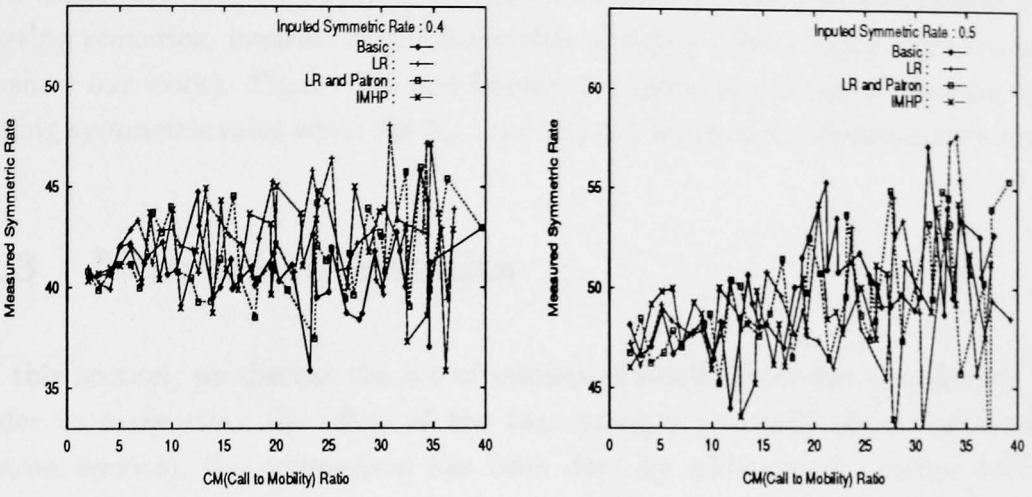


Figure 6.3: Measured Moving Symmetric Rate (cases S_{rate} 0.4 and 0.5)

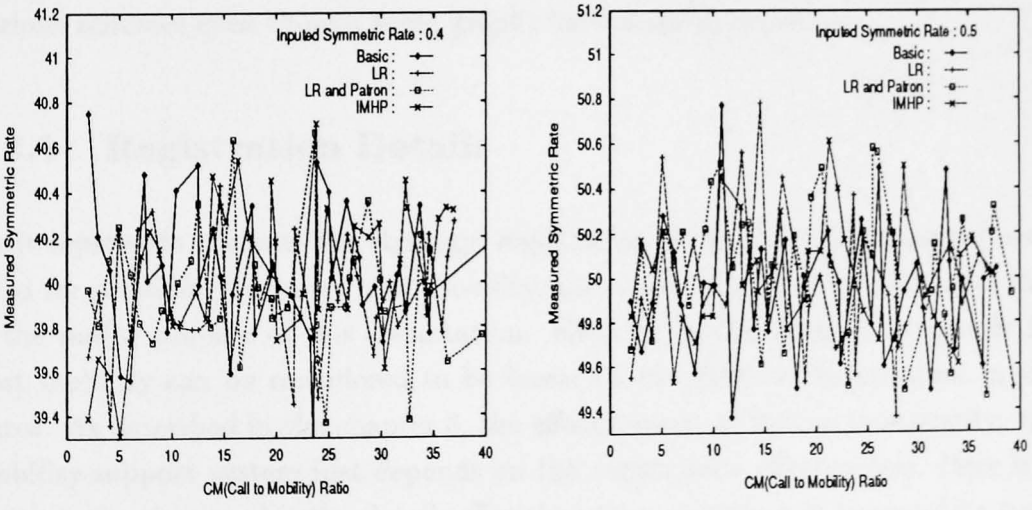


Figure 6.4: Measured Calling Symmetric Rate (cases S_{rate} 0.4 and 0.5)

The simulation carried out in terms of 0.1 units of S_{rate}. In this thesis, the results of 4 S_{rate}s (0.4, 0.5, 0.6 and 0.7) are shown to illustrate the different calling and moving scenarios, because of the limitation of space (also 4 S_{rate}s were enough to show our work). Figure 6.3 and Figure 6.4 shows the measured moving and calling symmetric rates when the S_{rate} 0.4 and 0.5 are used in our simulation runs.

6.3 Numerical Results

In this section, we discuss the set of numerical results from the simulation. In order to make clear the effect of the two concepts proposed (local region and patron service), the comparison has been done by adding each concept to the basic scheme in turn – basic + local region (LR scheme), basic + local region and patron (LR and patron scheme). When appropriate, we also use the results of a particular scheme to justify its corresponding concept. Although simulations were performed with several random seed numbers, only a result with 52763817 is described here. Again, the simulation was stable with different random seeds (but the same simulation parameters). With benefit of the uniformity of the random numbers, this is actually enough to illustrate general behavior of the various schemes even though some graphs have some fluctuation.

6.3.1 Registration Details

As it represents the location strategy, registration is the most fundamental overhead for providing seamless host mobility, and eventually routing efficiency; that is the major concern of this dissertation. Essentially, the different proposals for host mobility can be considered to be based on the different registration procedures. As described in the chapter 3, the effectiveness, as well as practicality, of a mobility support system just depends on the registration effectiveness. Here it is worth trying to examine the details of registration procedure in terms of its overhead. The following two figures show the operating costs imposed by registration process.

Figure 6.5 depicts the number of registration events for the three schemes. When

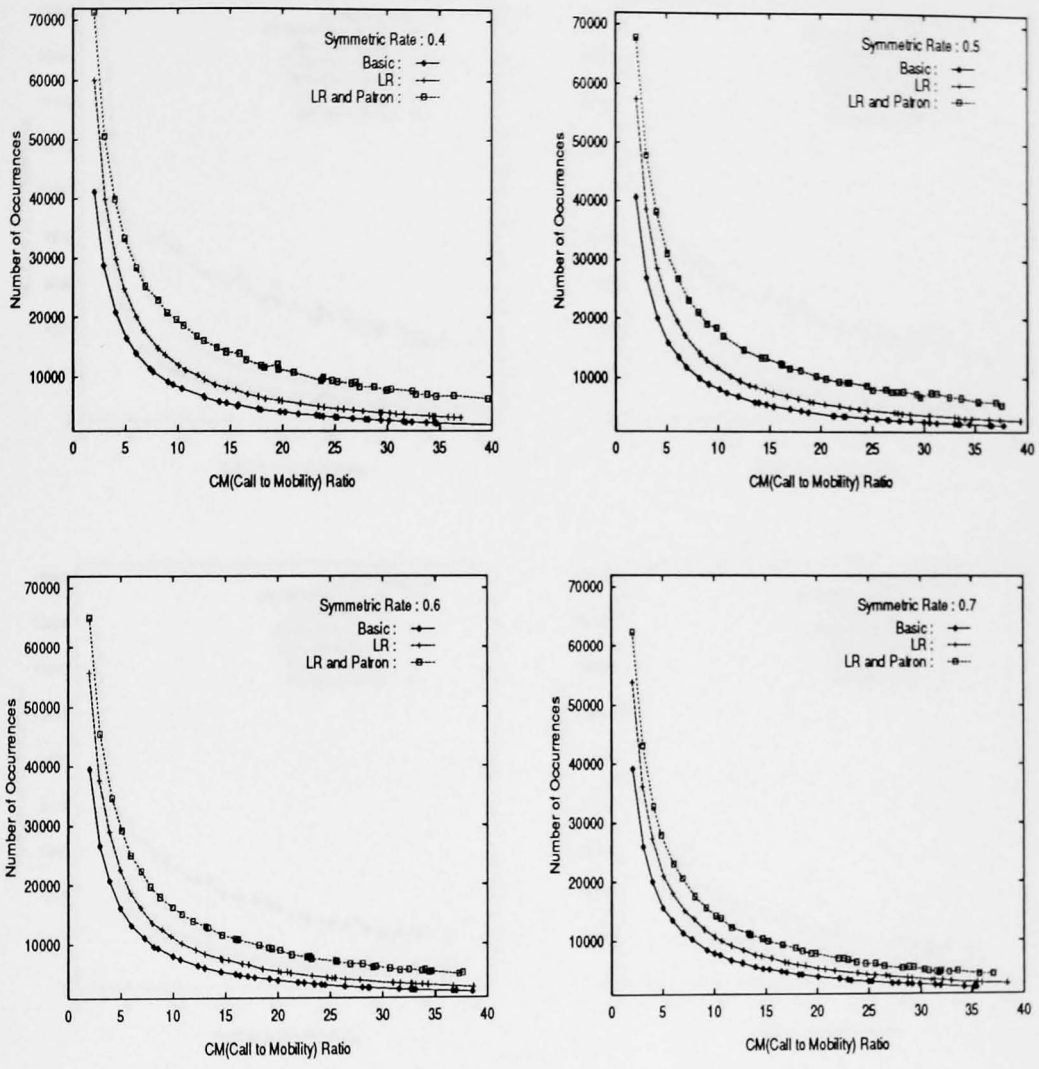


Figure 6.5: Number of Registration Event (Each Concept)

the CM_{ratio} is small (in this case, a mobile host moves frequently because the number of data packets for each simulation run is fixed), the number of registration events required in the LR scheme is relatively large, in comparison with the basic scheme. This is attributable to the extra registrations with the redirection agents – one for each move within its current LR, two or three (primary RA or secondary RA, or both, including current RA) for crossing the local region.

With the LR and patron scheme, the registration overhead is considerably more

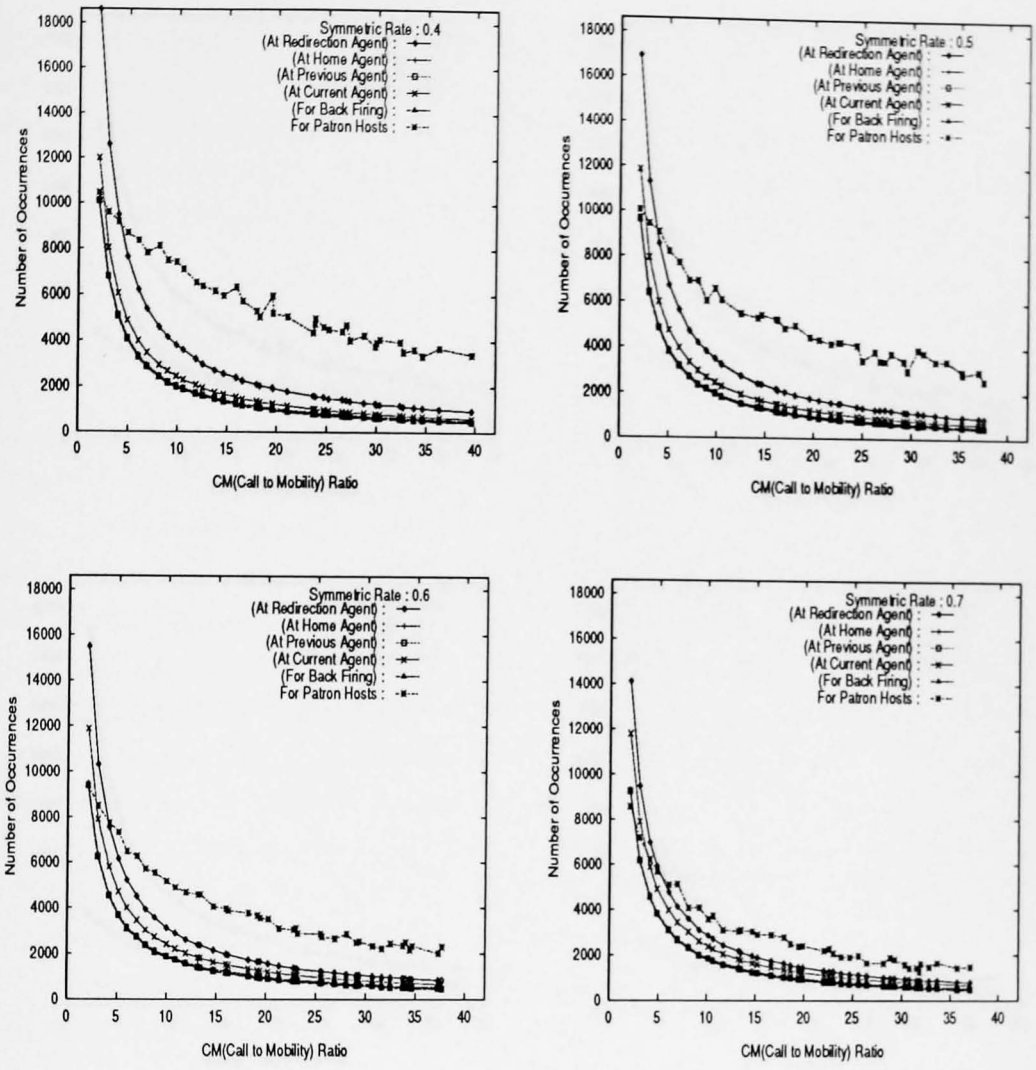


Figure 6.6: Registration Details (LR and Patron)

than that for the one for the LR scheme. Figure 6.6 shows the registration events imposed by each part in the LR and patron scheme. When the S_{rate} is low, the patron service contributes greatly to the overhead, and the number of RA registrations is large. As S_{rate} increases (in this case, a host is apt to move around, and calls are inclined to be for its home or current area), the registration overhead for the patron service dramatically decreases, and the extra registration to the RAs is getting close to the number of registrations to the current agent, that is, the number of host movements. The number of registration to the home

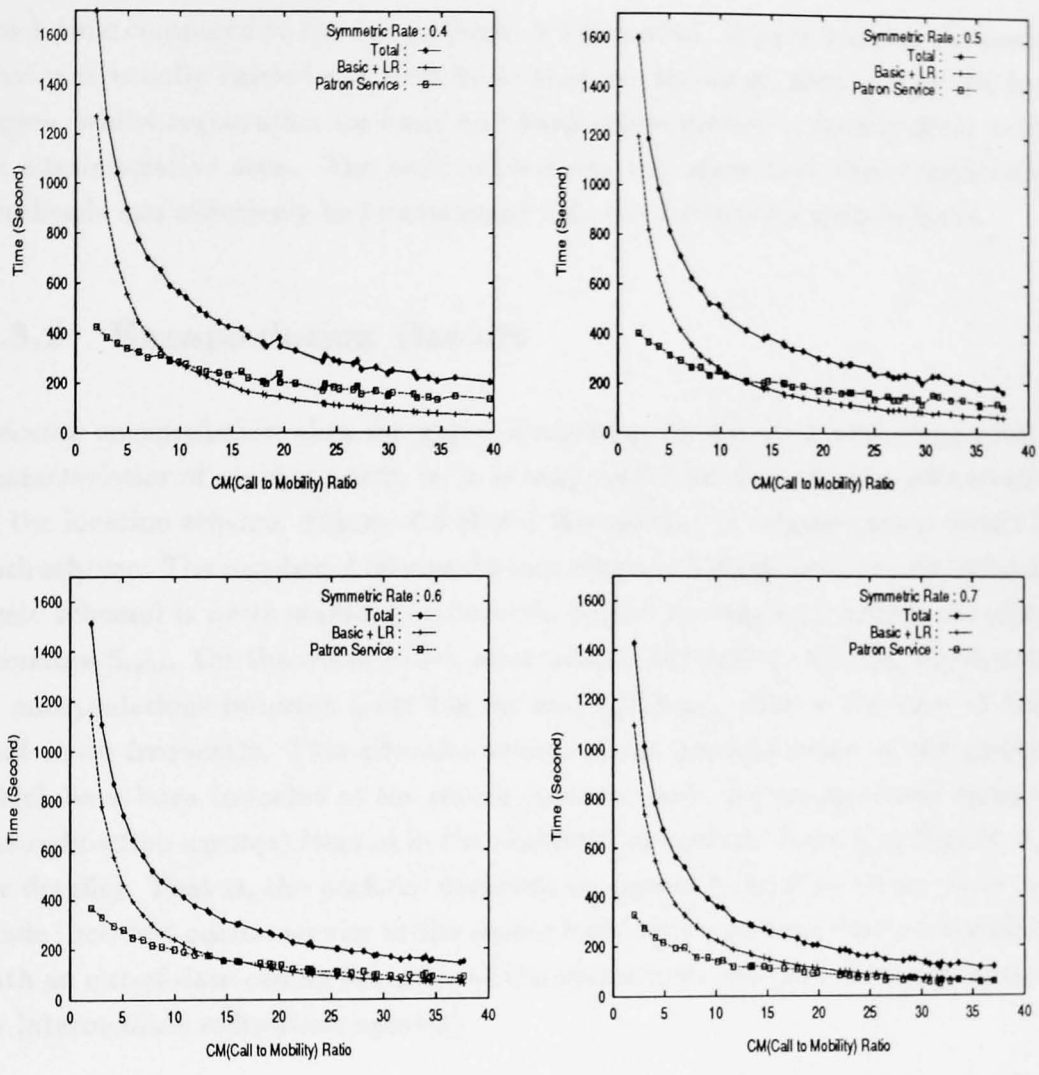


Figure 6.7: Network Occupation Time for Registration (LR and Patron)

agent and the previous agent (as well as for the back firing) is similar with the variation of S_{rate}.

Let us now look at the registration overhead from the network infrastructure point of view. It can be determined from the network occupation time introduced by the registration packets. Figure 6.7 depicts the occupation time for the LR and patron scheme, which is accumulated during a simulation run of about 40000 seconds simulated time. With a low S_{rate}, the patron service requires relatively

much time compared to the one for basic + LR control. This is because the patron service is usually carried out with hosts that are far away; thus out of the local region, whilst registration for basic and local region control is mostly done within an administrative area. The next subsections will show how these registration overheads can effectively be incorporated into the routing for mobile hosts.

6.3.2 Encapsulation Details

Because encapsulation aims for packet tunneling, its details reveal some routing characteristics of moving hosts, so it is very useful for showing the effectiveness of the location scheme. Figure 6.8 shows the number of encapsulation events for each scheme. The number of encapsulations with the LR scheme (as well as for the basic scheme) is quite stable, but depends on the moving and calling paradigm, therefore S_{rate}. On the other hand, after adding the patron service, the number of encapsulations becomes quite big for small CM_{ratio}, that is the case of hosts that move frequently. This situation comes about because many of the packets, which have been tunneled at the source (patron) host, are encapsulated again by the redirection agent(s) located in the middle of the packet route (see Figure 6.10 for details). That is, the packets' destinations moved to another place since they made their last patron service to the source host, so the packets were encapsulated with an out-of-date calling list entry of the source host and need to be re-tunneled by intermediate redirection agent(s).

Figures 6.9 and 6.10 illustrate the number of encapsulations for each part of the LR scheme and the LR and patron scheme respectively. In particular, these figures show the shifting of tunneling roles among the mobility entities – the mobility agent, the redirection agent and the mobile host. With only the local region concept, Figure 6.9 shows that the packet tunneling is carried out more on the redirection agent than the mobility agent when S_{rate} is low. As S_{rate} increases, the redirection agent's role for packet tunneling decreases and the mobility agent takes over main encapsulation duties. However, the encapsulation rate for them is stable over changes in CM_{ratio}.

With the LR and patron scheme, Figure 6.10 shows an interesting characteristic in that the number of encapsulations at the mobile host is increasing, whilst the

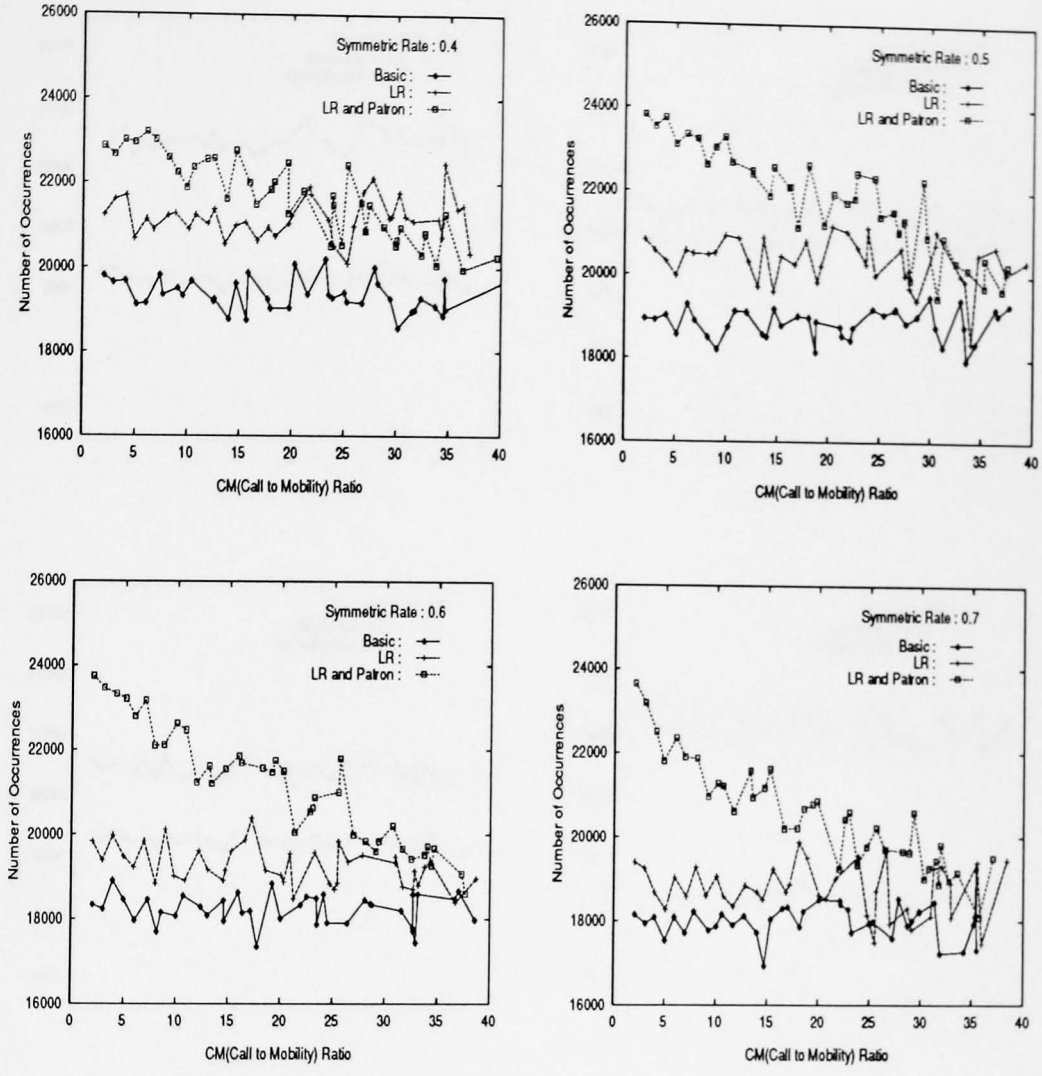


Figure 6.8: Number of Data Encapsulation (Each Concept)

number of registrations to the host (therefore the patron service) is decreasing (refer Figure 6.6), in proportion to the increase in CM_{ratio}. In addition, when S_{rate} increases, the number of patron services dramatically decreases, and becomes eventually below the number of encapsulation at the mobile host. This is caused by the fact that the patron service (thus the calling list) is used more for tunneling at the mobile host itself as the number of calls per move increases (see 6.19 for details). This encapsulation is significant because the packets would possibly be directly routed to the destination. Therefore, the patron service is more effec-

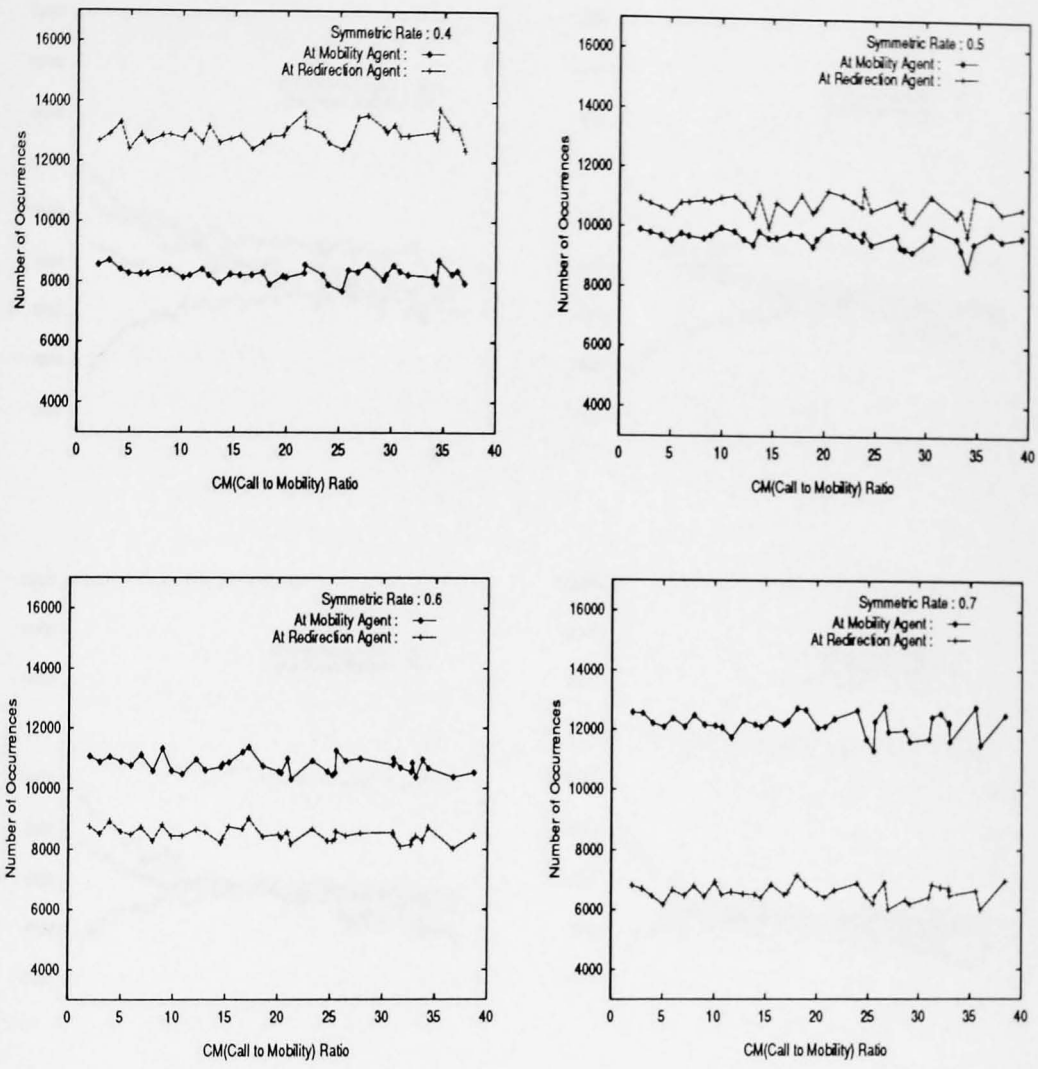


Figure 6.9: Encapsulation Details (LR)

tive from the location point of view when the CM_{ratio} is relatively big, i.e. by observation, greater than 10.

Changing the encapsulation role between the mobility agent and the redirection agent is similar to the one for the LR scheme above. However, when the CM_{ratio} is small and when the S_{rate} is relatively low, the number of encapsulations at the redirection agent is relatively large compared with the mobility agent. As explained above, packets directly tunneled by mobile hosts are re-encapsulated by

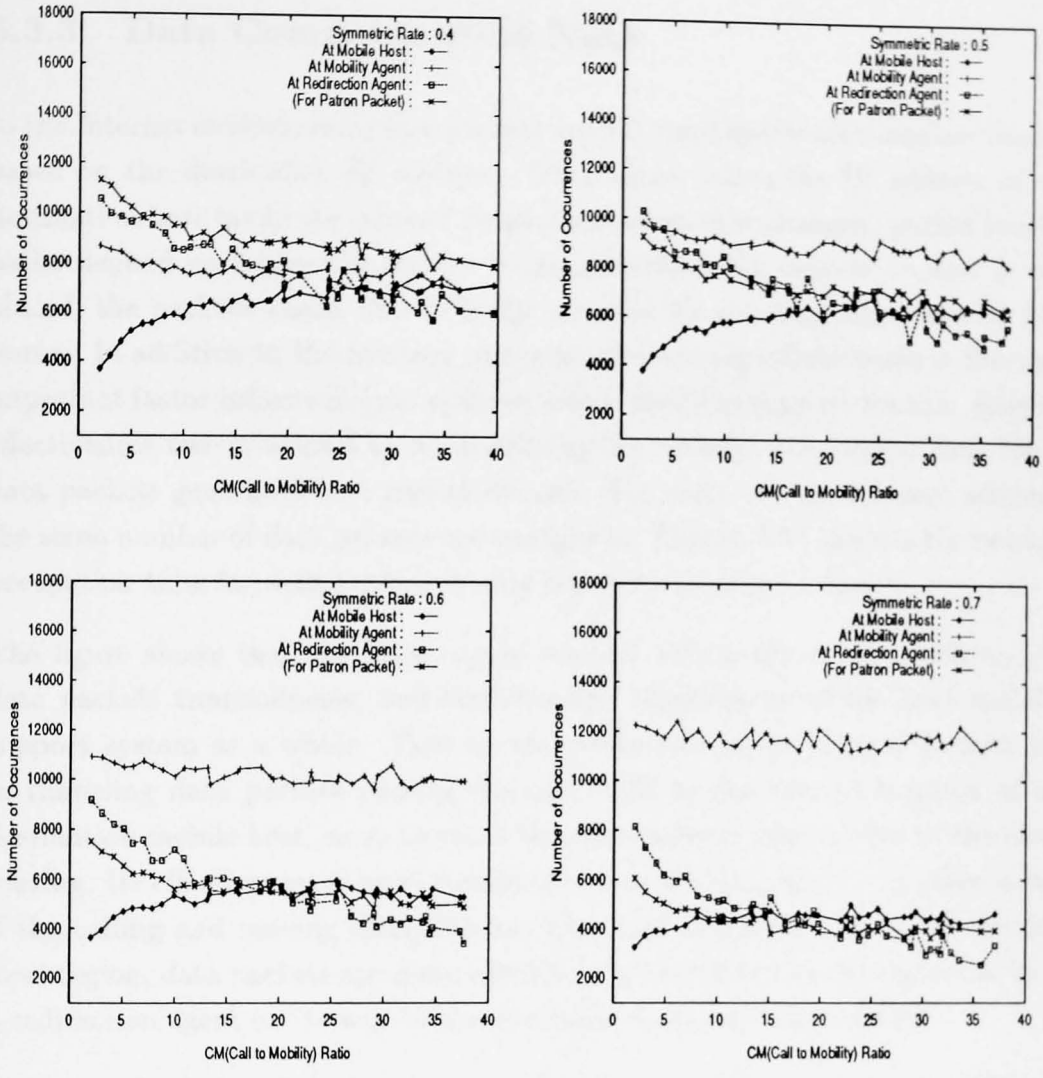


Figure 6.10: Encapsulation Details (LR and Patron)

the intermediate redirection agent(s), because the packet's destination frequently moves, so the calling list entries in the source hosts may easily become out-of-date. In the next subsection, we shall see how much these encapsulation could be resolved to improve routing effectiveness.

6.3.3 Data Communication Time

In the Internet environment, data packets are delivered by the intermediate routers based on the destination IP address. Once again, when the IP address of the destination host (as in the sense of its physical locator) is changed, packet routing paths depend on where the locator is maintained; if the current locator is well placed, the packets would be efficiently tunneled by avoiding unnecessarily long routes. In addition to the location overhead, the routing effectiveness is the most important factor influencing the performance of mobility support system. Routing effectiveness was measured by accumulating the network occupation time for all data packets generated in a simulation run. For each run for different schemes, the same number of data packets are configured. Figure 6.11 depicts the network occupation time for data packets during the given simulation time.

The figure shows that the local region concept affects the routing efficiency of data packets tremendously, and therefore the effectiveness of the host mobility support system as a whole. That is, the redirection agent plays a decisive role in tunneling data packets passing through itself to the current location of the destination mobile host, so as to route the data packets much closer to the direct routing. Its effectiveness is most significant when the S_{rate} is low. In other words, if the calling and moving discipline has dispersed outside its home (or current) local region, data packets are more effectively tunneled to the current location by a redirection agent on its way to the originally designated destination.

Moreover, it shows that the patron service considerably increases its effect on routing. Interestingly, its effect (relatively to the local region only) is stable over variance in the CM_{ratio}. However, as described in the previous subsection, when the CM_{ratio} is small and when the S_{rate} is low, the patron concept has considerable overhead for its registration activities as shown in Figure 6.6 (note that this situation is also shown in Figure 6.12). The figure therefore shows that the routing paths of data packets directed with the patron service are properly complemented by the local region control, to make them much closer to the optimal routing. This comes from the fact that the patrons are defined by considering the local region's spatial locality as well as the packet calling locality.

Figure 6.12 is representative of the total network possession time for data packet

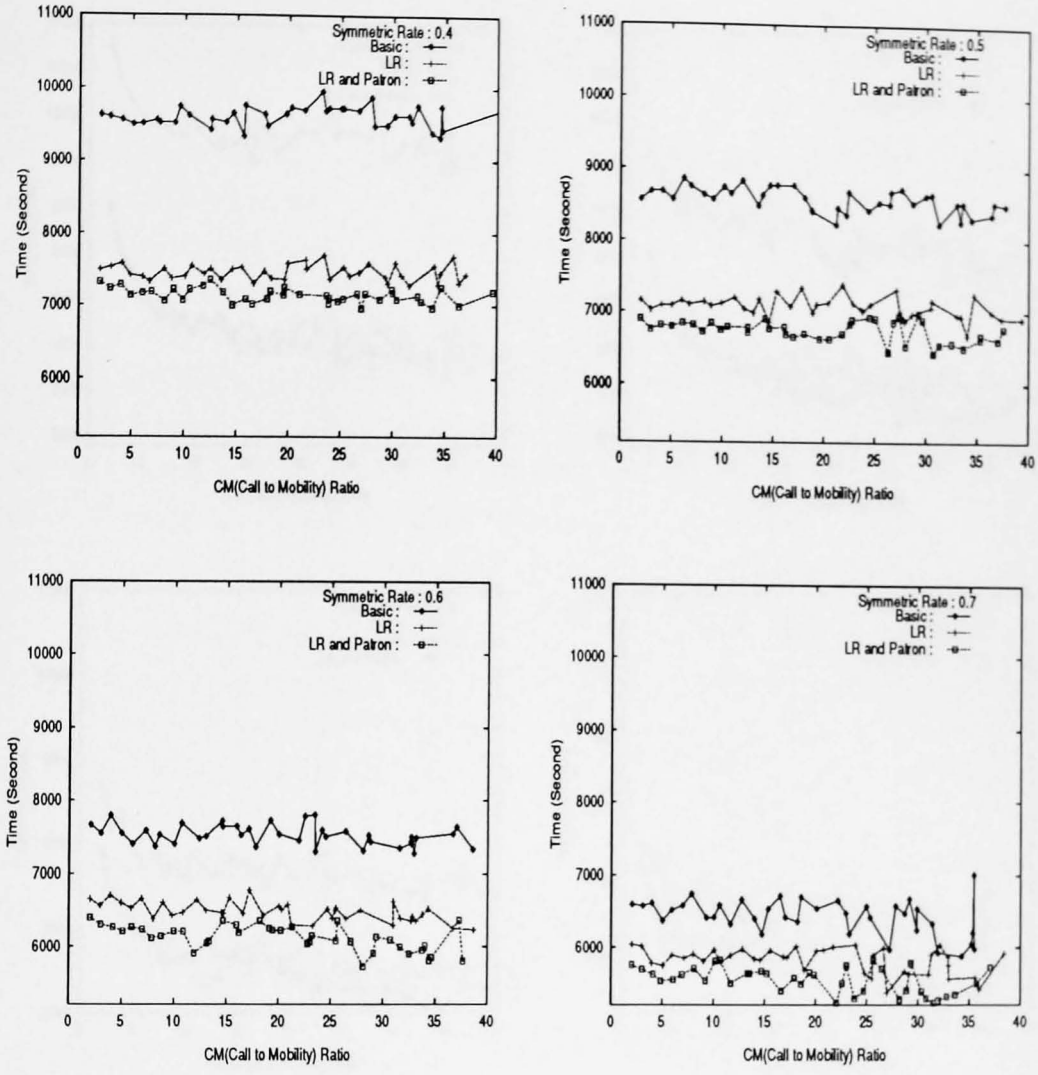


Figure 6.11: Network Occupation Time with Data Packets (Each Concept)

and registration packets. It shows how much the location and routing optimization effort can improve the network infrastructure's communication overhead as a whole. The extra registration (that is, location) overhead for the proposed concepts, shown in Figure 6.7, is mostly incorporated in decreasing the overall network occupation time with the benefit of providing nearly direct routing for most communications. This situation is greater when S_{rate} is relatively low. The figure shows that the redirection agent's role is the most clear due to its relatively small registration overhead, but with great effect on packet routing to moving

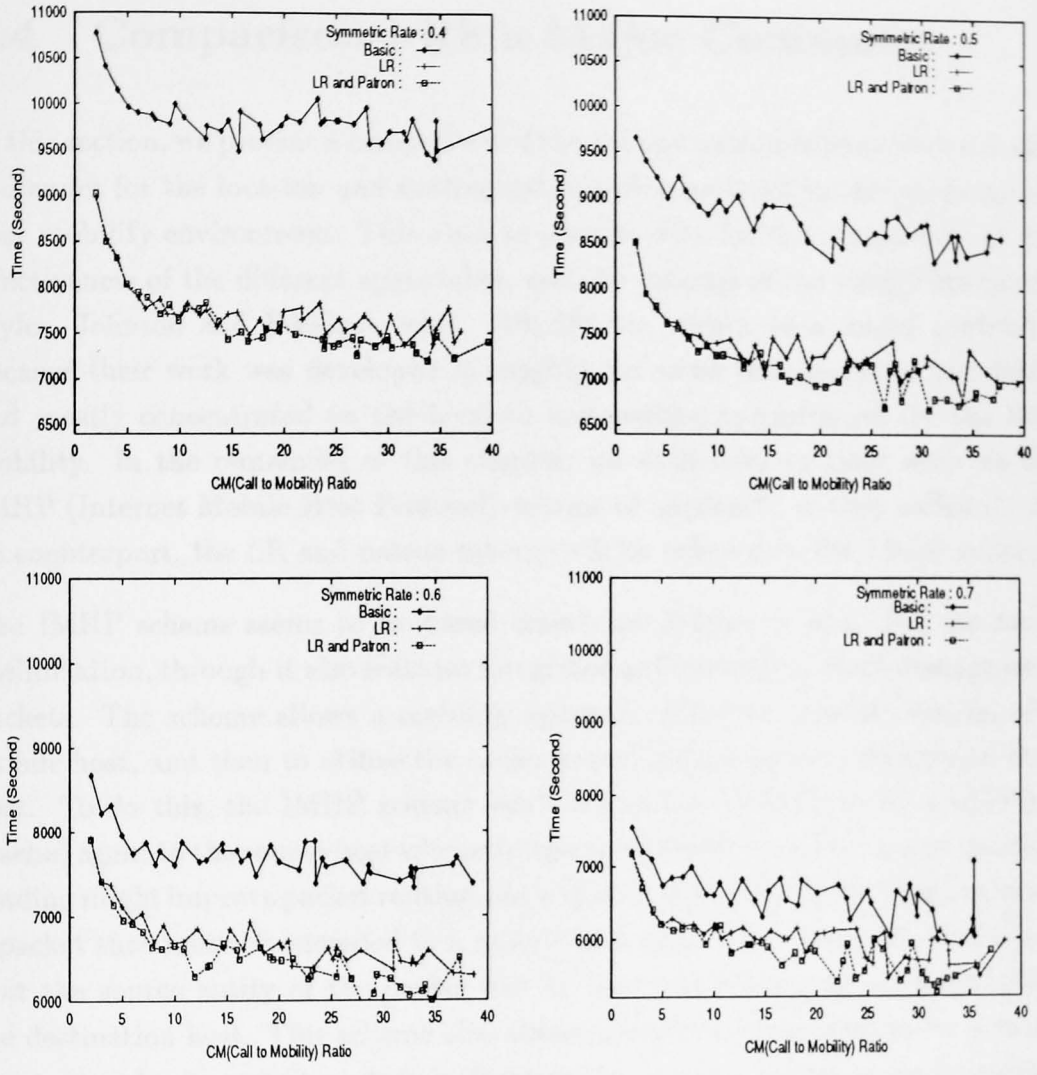


Figure 6.12: Total Network Occupation Time (Data and Registration)

hosts. When the CM_{ratio} is small, the patron service contributes little to improve the system performance due to its large registration overhead. Nevertheless, it is very useful for providing direct routing between mobile hosts, especially when the CM_{ratio} is relatively large (see subsection 6.4.5). We can now conclude that the approach presented is greatly effective for optimizing the location and routing tradeoff from the system performance point of view.

6.4 Comparison with a Major Contender

In this section, we present a comparison of the LR and patron scheme with a major contender for the location and routing optimization protocol in the internetwork host mobility environment. This aims to provide data for the comparison of the effectiveness of the different approaches, and the rational of our design approach. Myles, Johnson and Perkins's work [40, 53] was chosen as a major contender because their work was developed in roughly the same time frame as our work, and mostly concentrated on the location and routing optimization for the host mobility. In the remainder of this chapter, we shall refer to their work as the IMHP (Internet Mobile Host Protocol) scheme or approach, as they called it. As its counterpart, the LR and patron scheme will be referred to the LROP scheme.

The IMHP scheme seems to be based mainly on Johnson's idea [39] for route optimization, through it also features integrated authentication of all management packets. The scheme allows a mobility agent to cache the current location of a mobile host, and then to utilize the cache to send future packets directly to that host. To do this, the IMHP scheme sends a location notification by a mobility (cache) agent to the source host whenever the agent determines that a new location binding might improve packet routing. As a typical case, if the home agent receives a packet that must be tunneled to a mobile host away from the agent, it is likely that the source entity of the packet has an incorrect binding or no binding for the destination host. This scheme also allows the previous agent(s) to be a cache agent, in order to permit packets in flight to the previous agents to be forwarded to the new location. In either case, this entity may send a binding notification to the source node of the packet, and the corresponding host now acts as a cache agent.

The most recent work [53] extended the cache agent functions: IMHP permits any intermediate agents, even in the middle of a packet's route, to function as a cache agent. If necessary, the intermediate agents issue a location notification message. When an intermediate cache agent snoops on the notification, this cache agent can use the notification as a trigger to acquire a binding for the mobile host. If a packet passes through the intermediate cache agent which has a location cache entry for this packet's destination, then the cache agent will tunnel the packet to the mobile host's current location.

Each notification message indicates the maximum lifetime for any location cache entry created from the message. An old cache entry, especially on the previous agent(s), is eventually deleted after the expiration of the lifetime period established. A mobility entity wanting to provide continued service with a particular location cache entry may attempt to reconfirm that mobility binding before the expiration of this lifetime period. IMHP also defines a special tunneling, which tunnels packets to the destination host's home agent, to resolve a routing loop or a lost-route packet (that is, the cache agent has no cache entry for the destination, possibly expiring its timeout); in this case, the tunneling agent may not send a notification to the source.

6.4.1 IMHP Implementation

The IMHP scheme can be regarded as an extension of the IETF Mobile-IP working group's recommendation [56] (see subsection 2.2). Once again, the basic scheme used for LROP is also mainly drawn from the IETF work. Both schemes use forwarding pointers as the location strategy for moving hosts. The only difference is the way they maintain the forwarding pointer; generally, the forwarding pointer scheme requires other facilities to reclaim useless pointers, that is, on the predecessors of the previous agent (refer the back firing in the subsection 4.2). The LROP basic scheme uses the back firing concept to clear these old cache entries, and therefore just the previous agent has the location cache entry for a mobile host. On the other hand, the IMHP work uses a timeout concept to delete the old cache entry of the previous agents, so the previous agents may have a cache entry for a mobile host until the cache entry expires; some of these agents (the predecessors of the previous agent) may be out-of-date. First of all, the back firing of the LROP basic scheme was changed into that for the timeout-based one for the IMHP basic scheme. Interestingly, in our experiences, these two basic schemes produced quite similar simulation results from the various aspects, so the number of encapsulations and the network occupation time with data packets.

For the IMHP implementation, the description in [40] is used due to its availability when we tried to implement it. That is, only previous agents serve as the intermediate cache agents. However, it is difficult to provide any criteria for a cache agent to determine if it will issue a location notification to the source, if

the new location binding might improve packet routing in the current formulation. An encapsulation event is the most obvious way to issue a notification. In this implementation, the home and previous agents are cache agents for a mobile host. Some code was added into the IMHP basic scheme for manipulating the notification packet, and for processing special tunneling[2]. Also the encapsulation functionality is implemented on the mobile host component. It is assumed that the cache size for preserving location information is not limited in this simulation. Also, we are interested only in the location and routing aspects, so the authentication details of IMHP [40] are not considered in this implementation.

All system parameters for the IMHP simulation, such as component parameters, simulation parameters and the network model as well, are the same as for LROP as specified in the previous chapter. A new system parameter is the lifetime of each cache entry; if it is too short, the tunneling effect will be decreased due to the inaccessibility of location information, On the other hand, if it is too long, packets are tunneled with possibly long paths due to out-of-date cache entries. In the real world, the lifetime of each location cache entry would be different depending on the application running on the mobile host. For simplicity in our simulation, the same lifetime is defined for location caches for the notifications. After several tries with different lifetimes, we found that 5 seconds of simulator time produced the effective results. In order for a fair comparison to be made, the simulation running details and moving and calling scenarios for IMHP simulation used are the same ones as used for LROP scheme.

For the sake of understanding the way that the IMHP scheme works, Figure 6.13 shows a snapshot of an IMHP simulation run. This is captured with the parameters, seed number : 52763817, S_{rate} : 70, CM_{ratio} : 7, and the datagram number is 7799. The cache entry is represented by a pair, where the first element denotes the destination and the second one stands for the current address of the destination. The snapshot shows that the source host, MH 211, is currently connected with MA 28, and the destination, MH 212, has moved from somewhere to MA 11, MA 18, MA 21, and MA 24 (which is now its current agent) in sequence. Two hosts

[2]In our experience, a host moves with a routine which possibly makes a loop. This also seems to be true in the real world. However, it is well known that detecting a routing loop is not easy. So, if the same tunneling takes place over 6 times, before a packet gets delivered to the final destination, we assume the tunneling route is a loop, then special tunneling is used for delivery.

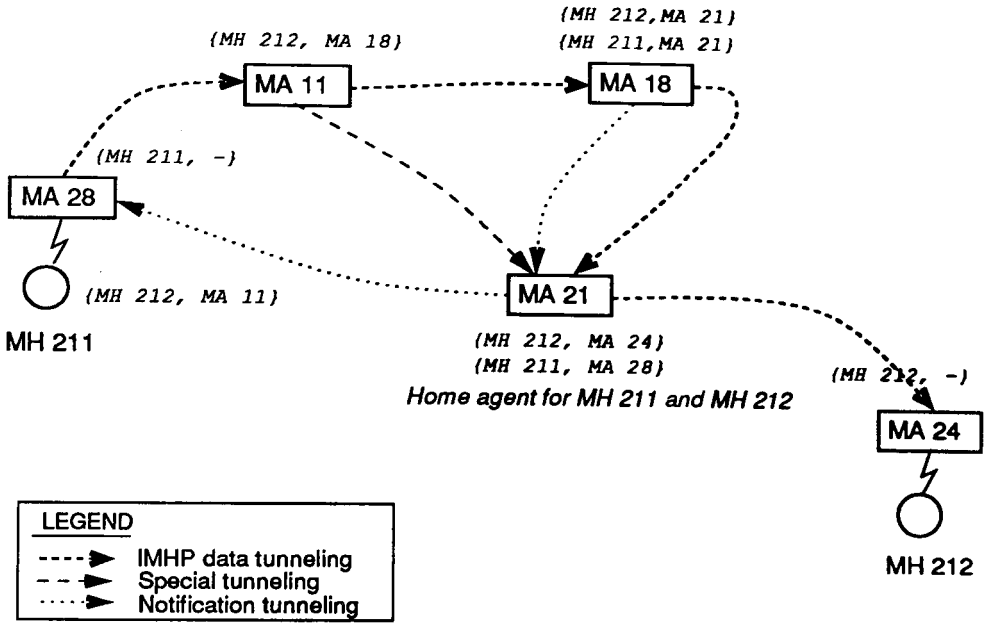


Figure 6.13: Snapshot in an IMHP Simulation Run

have the same home agent, MA 21.

When MH 211 tries to send a datagram to MH 212, it encapsulates the datagram and sends it to MA 11, because MH 211 has a cache entry for the datagram destination, {MH 212, MA 11}[3]. After MA 11 decapsulates the datagram, it again finds a cache entry {MH 212, MA 18}, so it encapsulates the datagram and sends it to MA 18. In addition, MA 11 determines that the datagram source has the wrong location cache for its destination, it then sends a notification, which includes {MH 212, MA 18}, to the source, MH 211. At this time, MA 11 has no cache entry for the notification destination, MH 211, so it uses special tunneling; the datagrams are now bound for the notification destination's home agent, MA 21. MA 21 should have a cache entry as a home agent for MH 211, and will tunnel the notification to the current agent, MA 28. The notification would eventually be delivered to MH 211; then MH 211 will change the corresponding cache entry for MH 212, to {MH 212, MA 18}. Likewise, when the datagram arrives on MA 18

[3]This situation can be explained as that MH 211 received the last notification indicating that MH 212 connected with MA 11. Therefore, after MH 212 left MA 11, there was no communication from MH 211 to MH 212

and MA 21, the same procedures take place as on MA 11. Finally, the datagram will be forwarded to MA 24 and delivered to MH 212. The source, MH 211, now has a cache entry for MH 212, {MH 212, MA 24}. The next datagram from MH 211 to MH 212 would be directly forwarded to MA 24, as long as MH 212 does not move before then.

6.4.2 Registration Details

In the IMHP scheme, the location notification is a passive reaction to the data packet encapsulation by the home or previous agents. This is characterized as lazy notification to spread the binding changes of a mobile host; the location update is delayed until a host possibly needs to use the destination's current binding. If a host does not have the most recent cache entry for a destination, that is, the destination has changed its mobility binding since the source updated the corresponding cache entry, the first packet from the source has always to be tunneled by the destination's home or previous agents. Each tunneling agent will send a notification packet to the source. With the timeout method, the predecessors of the previous agent may have out-of-date cache entry for a destination, until the cache entry gets expired after the lifetime period. It is therefore very difficult to formalize the registration behavior.

Now, let us look at the simulation results for the IMHP's registration. Figure 6.14 depicts the number of registration events for the two schemes, namely LROP and IMHP, with just their basic schemes. The basic schemes, LROP basic and IMHP basic, register with the same objects on each host moving: home, and previous agents. However, LROP basic also needs to register with the predecessor of the previous agent so as to clear the out-of-date binding entry (back firing); this may be a little less than the number of registrations at the previous agent (if the predecessor of the previous agent is the home agent or the current agent, the reset is not necessary). With the location extension, LROP has to register with the redirection agent(s) every move, and the patron hosts per crossing of a local region. On the other hand, IMHP has to register with the source host whenever the cache agent(s) determines that the source has no (or an incorrect) location binding for a destination. The result appears to be that the number of IMHP registrations varies little with increasing the CM_{ratio}, but it becomes higher than

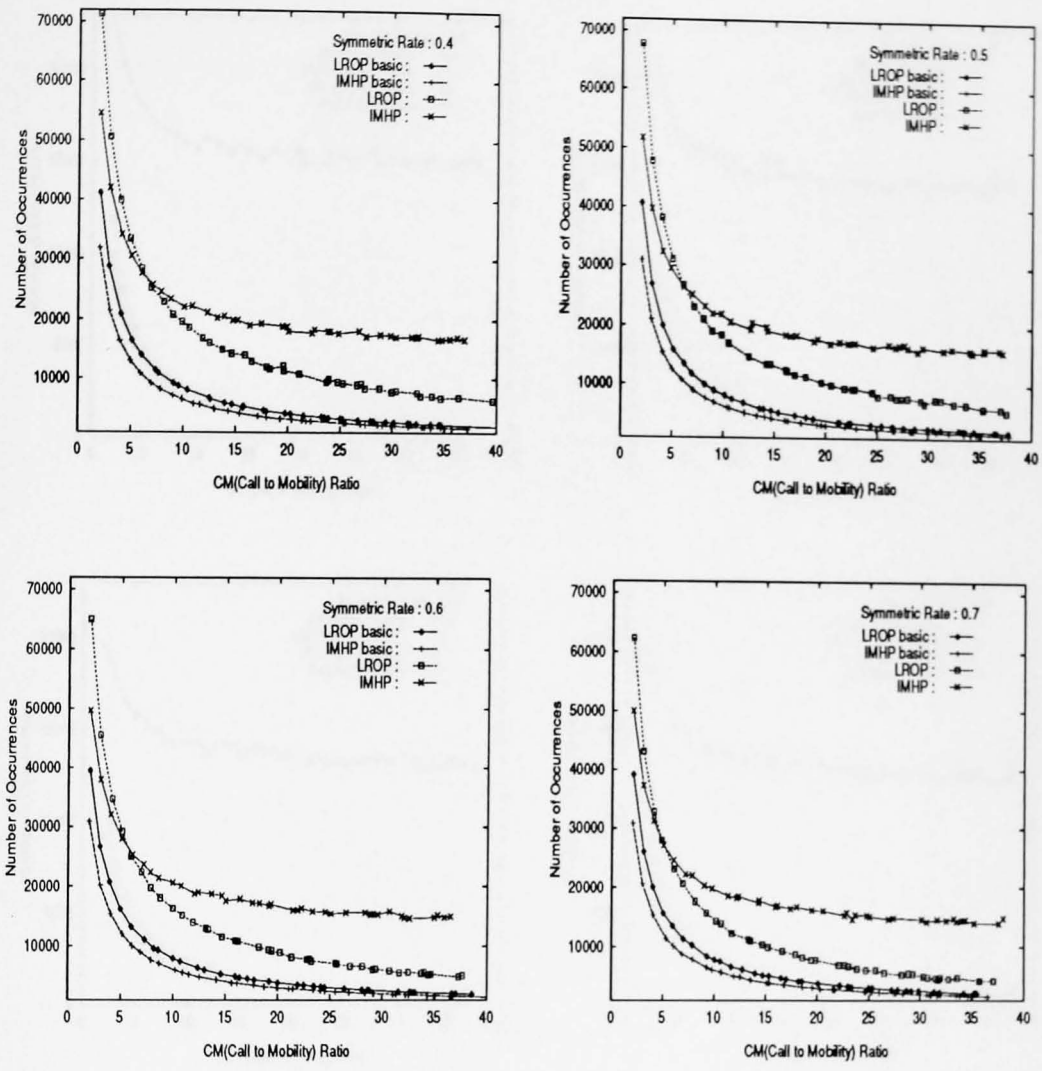


Figure 6.14: Number of Registration Event (Each Scheme)

the one for LROP when the CM_{ratio} is over about 6.

Figure 6.15 depicts the IMHP registration details. It shows that the highest registration overhead for IMHP is due to the notification. Unlike the patron service, IMHP does not consider the locality properties. Even though there are relatively many notifications when the hosts move frequently (low CM_{ratio}), the number of notifications is stable over the variance of CM_{ratio}. Also, the number of registrations varies little with S_{rate}; that is, it hardly depends on the symmetric rate for

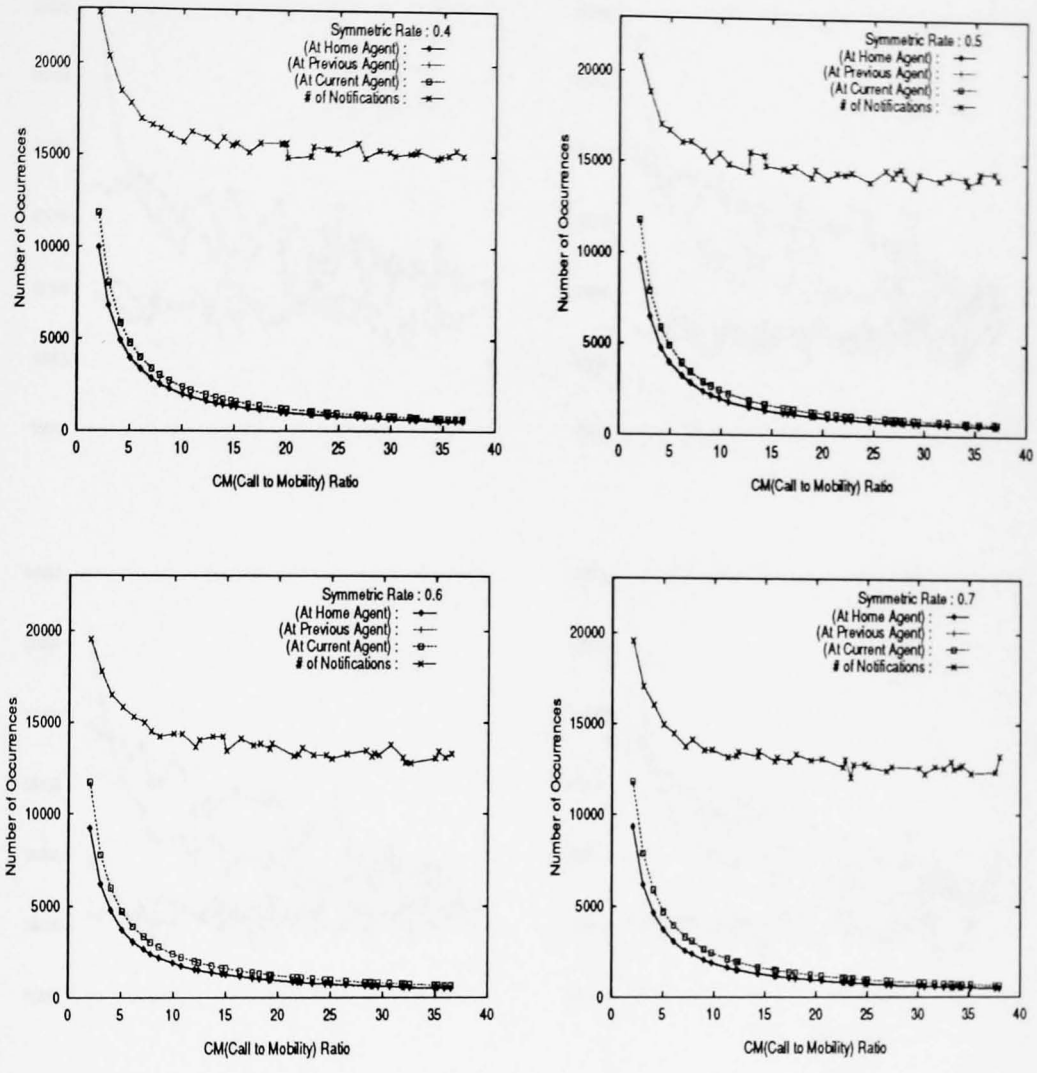


Figure 6.15: Registration Details (IMHP)

the calling and moving discipline, but would be rather subject to the (relative) temporal discipline between the source and the destination. If the calls and moves can be well clustered in time, the number of registrations (i.e. notifications) may decrease; but in this implementation the calling and moving events are generated randomly, so it is stable over the variance of CM_{ratio} and S_{rate}. In the next subsection, the effectiveness of this registration is shown with encapsulation details and the data communication time.

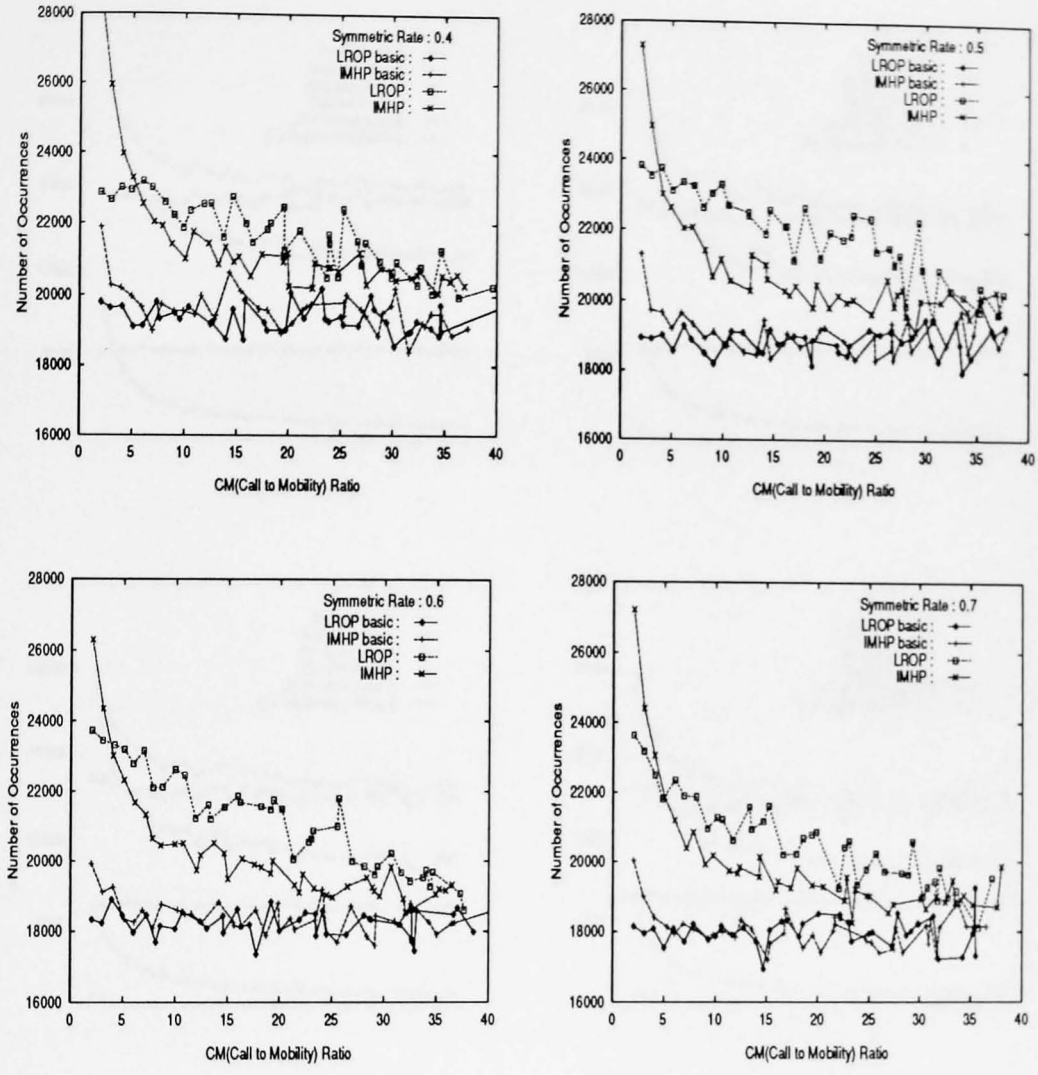


Figure 6.16: Number of Data Encapsulation (Each Scheme)

6.4.3 Encapsulation Details

Figure 6.16 is representative of the number of encapsulation events for the two schemes. For small CM_{ratio}, below about 5, the IMHP approaches encapsulate a great number of datagrams (many of these come from the encapsulation of the notification packets and on the previous agents as will be shown in Figure 6.17). However, as a whole, packets with the LROP scheme are more frequently tunneled when compared with the IMHP scheme. This means that the location strategy of

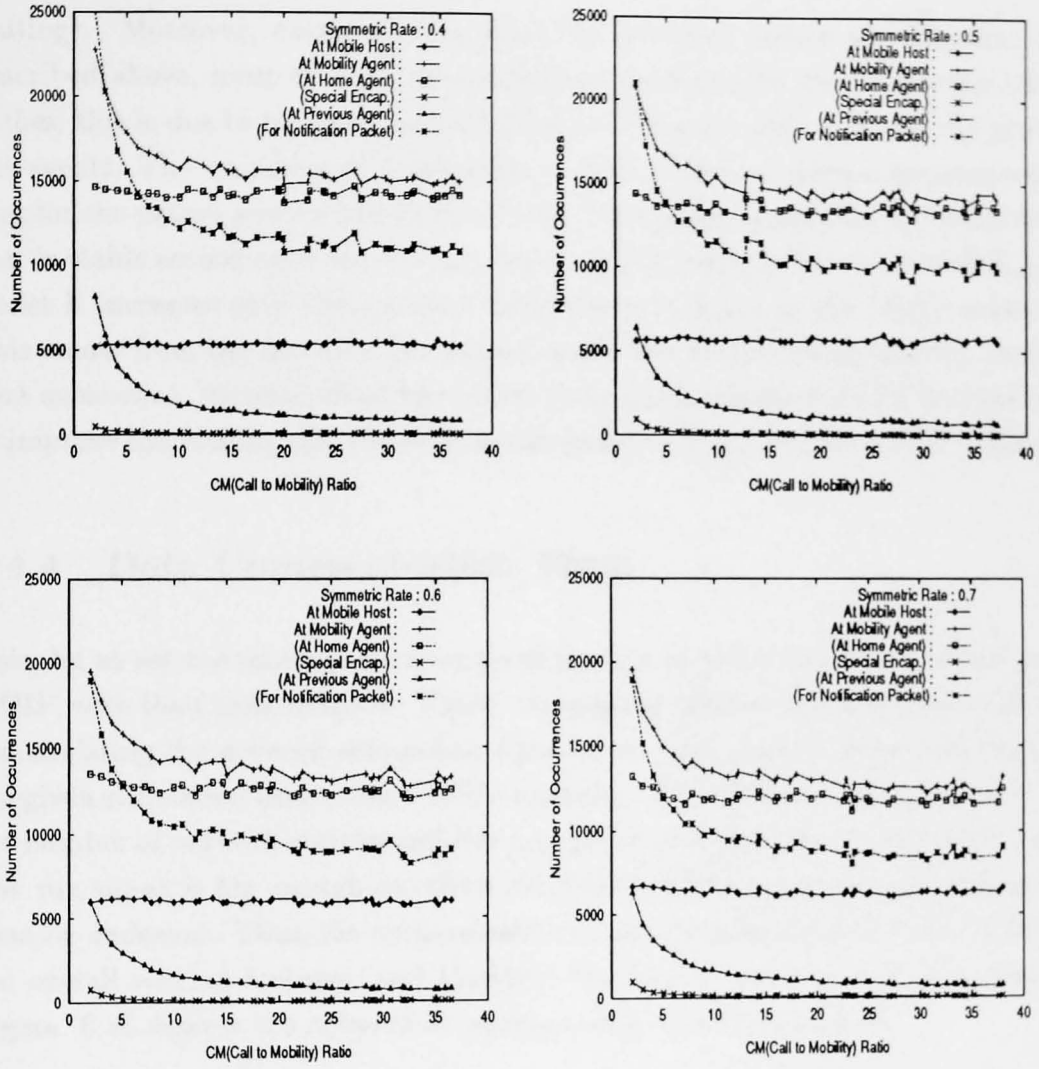


Figure 6.17: Encapsulation Details (IMHP)

the LROP scheme is more heavily used for packet tunneling than the one in the IMHP scheme. The two basic scheme are similar to each other in the number of encapsulations, as with registrations.

In addition, Figure 6.17 shows the encapsulation details processed by each part in the IMHP scheme. Firstly, most encapsulations take place on the home agent – this situation shows that the IMHP notification (location) would not be appropriately incorporated for solving the problem of the basic scheme (i.e. triangle

routing). Moreover, encapsulations from the previous agents are frequent: as described above, most of these encapsulations are based on out-of-date location caches; this is due to the timeout method of resetting the old caches on the previous agents. The tunneling of notification packets is greater when compared with that for the patron service (see Figure 6.10). Finally, encapsulation by the mobile host is stable according to the CM_{ratio} variances (but slightly increases with S_{rate}), whilst it increases with CM_{ratio} (but decreases with S_{rate}) in the LROP scheme. This comes from the fact that the patron makes use of the calling locality. In the next subsection, we shall show how much these encapsulations would be resolved to improve the routing effectiveness, as compared with the one for LROP scheme.

6.4.4 Data Communication Time

Now, let us see the routing effectiveness of the two location strategies, LROP and IMHP, with their basic schemes. Again, the routing effectiveness was measured by accumulating the network occupation time for all data packets generated during the given simulation time (about 40000 seconds). The simulation domain, such as the number of network entities and the number of data packets (including simulation run time), is big enough to reflect the routing effectiveness for each scheme's location endeavor. Thus, the accumulated network time should eventually exhibit the overall routing features, and therefore the system performance as a whole. Figure 6.18 depicts the network occupation time with data packets.

This figure shows that the LROP scheme requires much less time than IMHP for delivering data packets; that is, the former is greatly effective with respect to routing efficiency (therefore, high system performance), whilst it needs relatively less location overhead. Its effectiveness is much more significant for low S_{rate}; that is, when the calling and moving pattern has an out-bounded tendency (calls and moves mostly go outside the current local region), the datagrams are more effectively tunneled by an intermediate redirection agent on the way they pass through. Evidently, this results mostly from the local region concept; for which the hierarchical structure of a fixed network is utilized for location and routing of the moving hosts, whilst examining the locality property of host moving and packet calling.

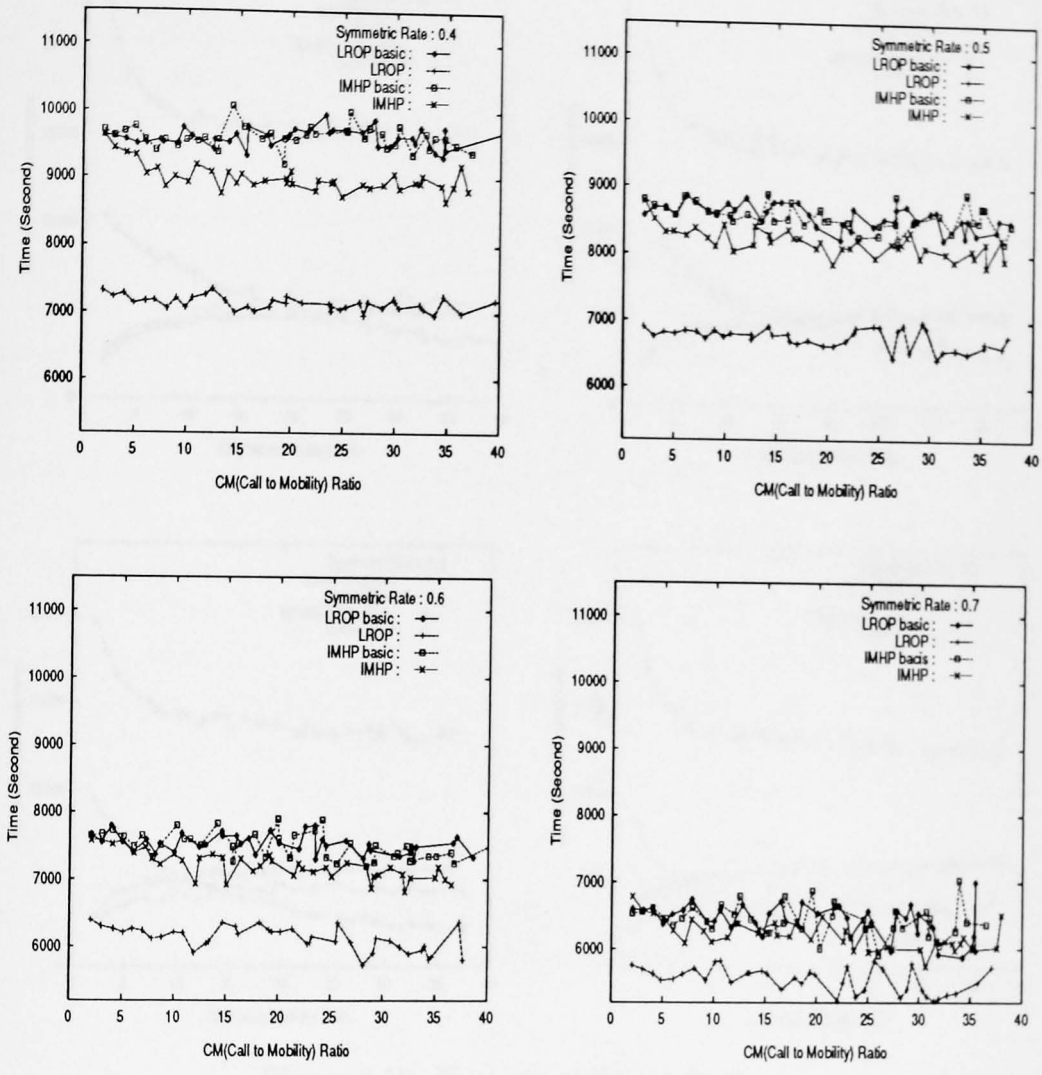


Figure 6.18: Network Occupation Time with Data Packets (Each Scheme)

The two basic schemes appear quite similar to each other, as in the cases of their registration and encapsulation. The IMHP scheme itself is not effective for low S_{rate}, as compared to its notification (location) overhead which has been shown Figure 6.14. This means that many of the notifications issued by IMHP scheme are barely used for real location purposes, and eventually for effective routing. In particular, when CM_{ratio} (i.e. the number of calls per move) is small, the routing effect is worse when compared to its huge number of encapsulations (see Figure 6.16). This is mainly caused by the lazy notification; most data packets are re-

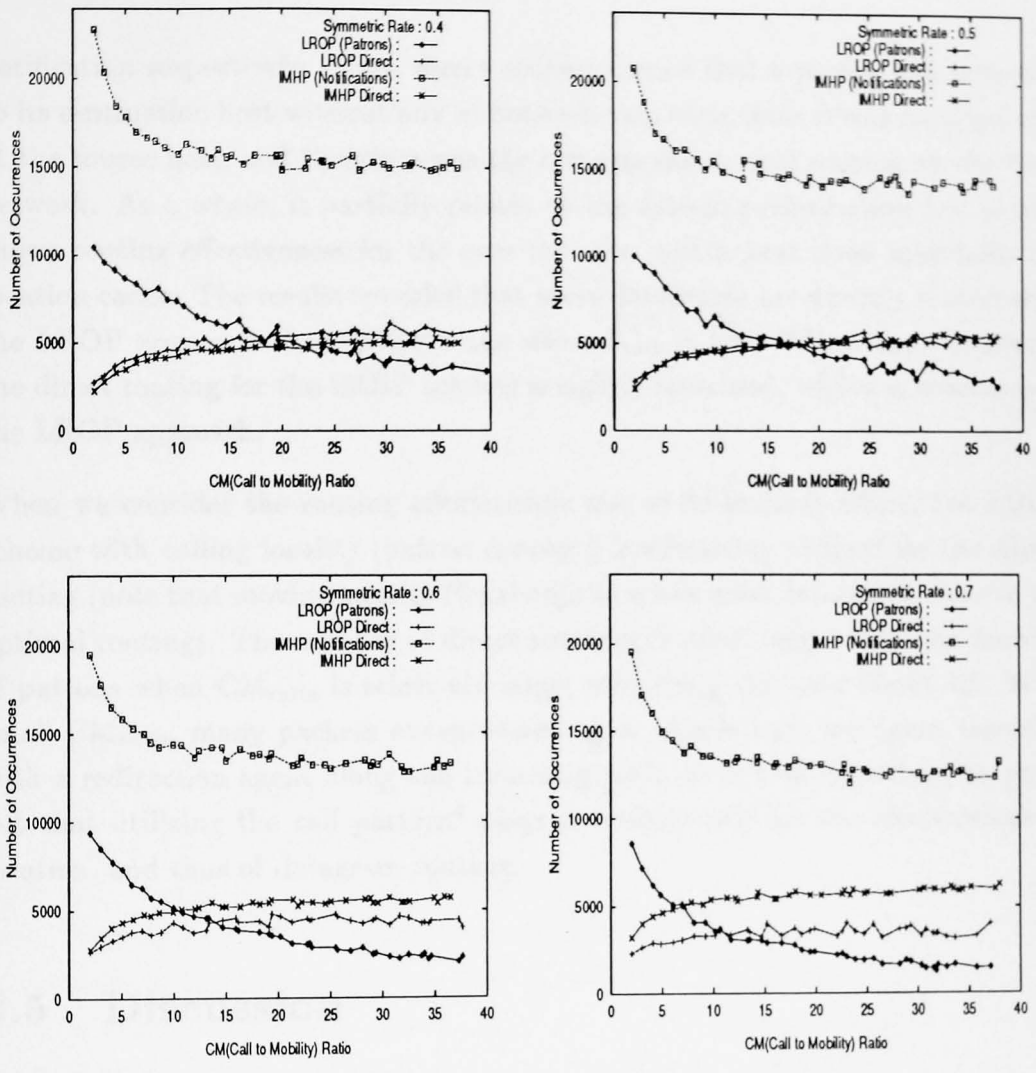


Figure 6.19: Number of Direct Routing

encapsulated by the previous agent(s) or the home agent because the location caches easily get out-of-date with frequent host movement. We conclude that LROP approach is better, and overall discussion is given in section 6.5.

6.4.5 Direct Routing

Finally, Figure 6.19 shows the number of direct routings for the two approaches, against the location effort on the mobile host such as patron service and location

notification respectively. Thus, direct routing means that a packet was delivered to its destination host without any in-between tunneling since it was encapsulated at the source host, and therefore was the same as the normal routing on the fixed network. As a whole, it partially relates to the system performance; i.e. it only shows routing effectiveness for the case that the mobile host itself maintains the location cache. The results revealed that more datagrams are directly routed with the LROP approach than IMHP's one when S_{rate} is low. When S_{rate} increases, the direct routing for the IMHP scheme is sightly increased, whilst it decreases in the LROP approach.

When we consider the routing effectiveness out of its location effort, the LROP scheme with calling locality (patron concept) is effectively utilized for the direct routing (note that moving locality (local region) sends most datagrams close to the optimal routing). The number of direct routings is much larger than the number of patrons when CM_{ratio} is relatively large, something else over about 12. With small CM_{ratio}, many packets encapsulated by a mobile host are again tunneled with a redirection agent along the tunneling path. It is now important to point out that utilizing the call pattern[4] plays a decisive role for the effectiveness of location, and thus of datagram routing.

6.5 Discussion

During this simulation study, we tried to show that internetwork host mobility is most effectively supported by exploiting the locality properties of host movement and packet calling. The basic approach for accomplishing these is to move up some mobility support functionalities into the lower level autonomous area of the fixed Internet structure. With more elaboration, the mobility pattern seems to be the most important, and so should provide a basis for the system design. In addition, the calling pattern plays an important role. What we are pursuing with this approach is a unified framework for achieving optimal routing for most traffic whilst limiting location caches and/or updates as far as possible. As a result, the simulation concentrated on showing in great detail the location and routing

[4]Note that this call pattern is formalized based on host moving pattern in the LROP scheme, so the patron service includes only those hosts outside its current local region.

efficiency from the system performance point of view. The simulation results were stable across a variety of the simulation parameters, such as CM_{ratio} and S_{rate}.

Most of all, the redirection agent plays a decisive role in tunneling data packets passing through it to the current location of the destination, so as to route the data packets much closer to optimal routing. It also carries out the important duty of hiding all local movements outside its service area, and thus permits incomplete location information for sources residing outside the local region. The simulation shows that the routing effectiveness of the redirection agent is large compared with its location overhead. Most datagrams which have an incorrect destination address are delivered with a close optimal routing, as long as the destination moves around within its local region. The number of registrations with the redirection agent(s) is only proportional to the number of host moves, which is directly related to the local region control. The registration time with a redirection agent is also relatively short (that is, it is confined by the small size administration area).

The simulation also shows that the patron service considerably increases the routing effects, even though it requires relatively higher registration cost than the one for the local region (because it usually takes place through the Internet). Those hosts that frequently visit a mobile host take advantage of direct routing to the host. In particular, when there is more calling than moving, the patron service is advantageous for improving the system performance when compared with its own location overhead. Moreover, most datagrams tunneled from the patron hosts but with incorrect location information due to the destination's moving since the last update of their mobility binding, are correctly tunneled again by an intermediate redirection agent, so those should be nearly close to optimal routings.

With the comparison of the local region and patron approach and IMHP, we showed that our work is greatly effective from the point of view of location (registration) and routing efficiency (the encapsulation details and the network occupation time). It therefore results in greater performance transparency to the internetwork host mobility support system as a whole. With the lazy notification scheme, the location overhead is too much when compared with its contribution to overall routing efficiency. Most datagrams still pass through the destination's home agent, or the previous agent(s) which the destination had visited; these tunnelings usually have needlessly long routes.

These differences mainly result from the facts that the location update takes place in a lazy fashion, and the out-of-date location caches (mostly previous agents) are spread about the system according to the progress of the host, even when a timeout method helps to reset them. The lazy notification makes location information for most first calls to a moving host potentially unavailable. In the worst case, a datagram which is correctly tunneling can be wrongly re-tunneled with the out-of-date mobility binding of a previous agent(s) along the route. This comparison also reveals that the move-initiated location update is more appropriate than the need-initiated lazy notification from the location and routing effectiveness point of view. In addition, when the mobile host must cache location information, the calling locality greatly effects the location and routing efficiency; to cache the location for infrequently visited destinations brings about useless notifications and lengthy routes to them because of the destination's mobility. Also, the notification approach could give rise to caching overhead to the mobile host for preserving the location information for all potential (visited) destinations. Finally, the simulation has also shown that the out-of-date mobility binding on the previous agents (except the immediately previous one to protect orphan packets) has to be reset before it is falsely used, in order to prevent fruitless datagram forwarding.

Chapter 7

Conclusions and Future Work

In an internetwork host mobility environment, there is generally a tradeoff between the two key issues, that is, locating mobile hosts' physical locale and routing datagrams to and from them. If the system initially puts effort into location, routing overheads caused by host mobility should be reduced. This thesis has shown that the routing costs can be dramatically lower if we make use of the locality properties of moving and calling and the hierarchical addressing and routing, including structural, nature of the Internet to accommodate an efficient location framework. This chapter concludes the design issues, approach and performance of the schemes discussed in this thesis and indicates some of the possible areas for further research.

7.1 Conclusions

As compared with the case of its fixed counterpart, routing paths with host mobility just depend on the location information that is available for them. These bring about a great variance with the relative locales among the source host, the destination host and their location holder. With a poor location strategy, most packets may be tunneled with a default route such as the destination's home agent, which makes for unnecessarily long routes. This situation is particularly significant when host mobility is spread Internet-wide, and when the application applied has any time-constraints, such as real-time multimedia. This motivation defines the

goal of this thesis to be the development of an optimized scheme that can provide nearly close optimal routes for most traffic as if it were done on an ordinary static network whilst still limiting costly location updates as much as possible.

LROP adds some elaboration of the location and routing optimization to a basic scheme which is based on the IETF work. The basic scheme makes use of a home-based forwarding strategy for supporting host mobility. Mobile hosts can connect and move anywhere in the existing Internet, even while retaining their network connections. A mobile host maintains a constant (home) address regardless of its physical location, whilst the address of the mobility agent, which currently serves the host, is used to represent the physical locale. Each time a mobile host moves, the host passes its physical location to the home agent, and the previous agent it has just left, if it is different with the home agent. Packets destined for a mobile host are always routed through that host's home agent. In order to reset the out-dated forwarding pointers on previous agents, whenever the current agent updates an element of the forwarding list of the host's previous agent, the latter agent clears the forwarding list entry for the host on the previous agent, if it had one.

The two concepts proposed, local region and patron, were added into the basic scheme in turn. When a mobile host first joins the network or its user's interest area changes, the mobile user will be asked to define a local region. A hierarchical relationship is assumed between the mobility agents and mobility routers which make up the local region. Thus, the root router, called the redirection agent, has the special duty of redirecting packets passing through itself to the host's current location. To do this, the redirection agent maintains an additional mobility binding that preserves current location information for the hosts that have appointed this as their redirection agent. As a matter of course, the mobile host must update the corresponding mobility binding on its redirection agent whenever it moves. A mobile user would define multiple local regions based on her moving interests. When a host moves around within its primary (home) LR, all packets may be correctly redirected by the primary RA. If the host crosses out from the primary LR and defines a new local region (a secondary LR), it then updates its redirection entry of primary LR and the secondary LR. During the host movement around within the secondary LR, it updates only the secondary RA. Most packets are correctly redirected by the primary and secondary RA. Similarly, when the host

again moves out of its secondary LR, and tries to define the other local region (a current LR); additional location updates are done on the primary, secondary and current RA, and packets are still correctly redirected among the RAs. When the host moves around within the current LR, it updates only the current RA.

If a host moves outside its primary LR, i.e. the physical locale become different from the home area's addresses, the redirection role is only slightly effective for packets sent by a source host outside the LR because the packet will travel via the primary LR. The mobile host now takes most responsibility for providing an effective location for these packets. Each host keeps track of patron hosts, which are frequently visited and therefore highly expected to visit the host again. When a host crosses from the primary LR to the secondary LR, it then carries out a patron service based on the calling statistics on the primary LR, in order to pass its new address to the patrons. When the host moves around within the secondary LR, packets from the patrons are delivered with a direct routing (sometimes through the redirection of the secondary LR). The host may again cross its secondary LR; these further crossing between the current LRs will be done without any patron services because of the high cost of its management. When the host moves around within the current LR, packets from the patrons are redirected by the redirection agent of secondary LR. If the host tries to rejoin its primary LR, it needs to re-execute the patron service in order to reset the mobility binding which the host sent when it crossed outside the home LR.

The simulation results revealed many important things for our design approach. Most of all, we have been successful in showing that exploiting the locality property of moving and calling and the hierarchical nature of the Internet is crucial for effective location and then efficient routing. The key finding of these experiments is that the local region plays a decisive role for hiding all local movements, and permitting incomplete location information, to the outer world. Datagrams passing through the redirection agent were always forwarded to the current location of the destination, so their routes were nearly close to direct routing. The routing efficiency due to the redirection agent is tremendous when compared with its location overhead. The results also show that the patron service considerably increases the routing effectiveness, even if it requires relatively costly location updates than the ones for local regions. However, it is very effective for direct routing when compared with its location efforts when the call to mobility ratio is relatively big.

A comparison between our approach and the IMHP scheme was carried out with the same simulation framework to provide some rational of the design choices. IMHP is based on lazy location notification, so is need-initiated whilst LROP is based on move-initiated locating. IMHP reclaims the out-of-date mobility bindings with timeouts whilst, on the other hand, LROP makes use of back-firing. The simulation shows that the location updates for IMHP are much higher than the one for LROP, even in comparison with its contribution to the routing efficiency. With the IMHP scheme, most datagrams are still delivered by a possibly lengthy route via the destination's home agent, or via the predecessor(s) of the previous agent which might has an out-of-date mobility binding for the destination. The comparison shows that the LROP approach is better than IMHP in terms of both location overhead and routing efficiency, so it results in more effective performance of the internetwork host mobility support system as a whole. The results clearly support our design decision that the move-initiated location update is most appropriate for supporting host mobility, and that the out-of-date mobility binding has to be reset before it is used for unnecessary tunneling.

The location and routing optimization approach could be applied to most mobile computing infrastructures. Host mobility can result in a harsh environment for mobile computing; moreover, it has now become important to support multimedia applications which may sometimes have real-time constraints. In the presence of frequent and/or extensive host movements, an optimized location and routing solution can provide uninterrupted high-quality service for the applications. The key concepts exploited in this thesis could be mostly applied the other communication protocols, such as CLNP, but not necessarily IP.

The conclusions drawn from the work conducted in this thesis can be summarized as below:

- By considering the host's moving locality and the Internet's hierarchical nature, moving some mobility support duties to some part of the fixed network is very useful for location and routing optimization.

- Exploiting the locality property of packet calling greatly helps the location endeavor from the mobile host itself.

- The combination of location and routing duties between the network infrastructure and the moving host produce a scalable and efficient system.

- Move-initiated is the right approach for updating location caches, it reduces unnecessary location updates and helps effective routing.
- Location caches must be reset at the time they are out-of-date, otherwise packets are uselessly forwarded by them.

7.2 Areas for Future Research

The LROP experience offers not only a good solution to location and routing optimization in the internetwork host mobility context, but also gives valuable lessons on providing effective framework to support Internet-wide host mobility. However, several areas of future work have become apparent throughout the work in this thesis. The first area is that LROP should be implemented and tested on a real network configuration. Even though the simulation tried its best to consider most of mobile computing environment, there would be more practical problems which have to be solved, and requirements for realistic evaluations to give more insights into the issues proposed. Many existing protocol facilities, e.g. ICMP messages and routing table, and useful algorithms such as location cache management could take part in the real implementation.

An extension of the LROP scheme stems from the fact that the locality of movement and calling varies over time: frequency and distance vary according to the user's current interests, so it is a temporal locality. Although local region and patron concepts generally absorbs this tendency, they cannot reflect the dynamic changes of calling sources (and moving destinations) and their frequencies over time; only the total frequency when a host crosses its local region is considered in this work. In addition, if the host mobility pattern could not permit the definition of a local region or the mobile user does not want to set a local region, the patron service should still be applicable for effective location and routing. However, in this case, it is difficult to establish when the service has to be carried out and when the patron list should be refreshed (note that the patron service takes place using the calling history gathered in the previous local region the host has already left). As a result, a sophisticated method, which can sort out more likely patrons, would come out of exploiting the temporal locality of calling pattern.

The patron service has simply been managed as an IP extension in this thesis. It is, however, only a location method. From the layering viewpoint, the connectionless network layer protocol should not have an address list of corresponding end points; this should be done at a higher layer above the network layer. A real implementation of the patron service on a higher layer would need careful study. One possible way is to make use of some form of the network daemon process, such as *routed* or *gated*, to monitor patrons, and the same table that it uses already to handle the existing host specific ICMP redirect message type, but with a different type field on the table entry, to maintain related mobility bindings. Static hosts could then selectively chose to make use of the patron service, and they may be capable of optimally communicating with mobile hosts.

Another area of future research comes from the assumption that a redirection agent has only one connection path outside of the local region. In other words, the redirection agent is a single point of failure for the network entities within its local service area, usually an administration domain. In practice, subnetworks maintained by an organization sometimes may have multiple network connections to the backbone network for the purpose of fault protection, or to the other different subnetworks owned by other organizations. For these cases, special schemes are required; one possible way would be to make all routers connected outward to act as multiple redirection agents for a local region, and make use of a replication management method to keep them consistent. Investigating the traffic passing through the different redirection agents for a host may be helpful for this.

The advent of mobile computing has prompted new security requirements in contrast to the traditional fixed network. In many cases, mobile hosts will be connected to the Internet via wireless links. The first problem is due to the usual nature of the mobile links, that is, it is easier to eavesdrop. This can be solved by an encryption scheme. A mobility support system must be able to authenticate the source host of a received packet because a mobile host can easily masquerade as another host. However, the most fundamental area for future research must be on enabling secure interchange of the registration packets (and thus location updates) transacted by the mobile host and the mobility agents. One possible security hole in the work described in this thesis is an entity trying to spoof the registration procedure, by using another host's address, or by emulating a mobility agent. The same kind of attack may take place during the patron service.

Another one is that malicious forwarding entities with location caches, such as the home agent, previous agent or redirection agent, could deliberately divert any data packet. These would break the functionality of host mobility itself.

To protect against these, a three stage solution has been widely suggested in the research community; sharing a secret key between home agent and mobile host, a randomly chosen challenge number for sending a packet, and using a signature for the challenge along with the significant address using MD5 [62]. Each mobility entity would then authenticate itself with its home agent whenever it needed to. In addition, two vertical issues are involved in the application framework; privacy and anonymity. Accessing any information related to the mobile user's location data without his consent, could bring a serious or unexpected violation of his privacy. Moreover, it is necessary that the mobile user's real identity is not revealed to unintended parties. These problem should be carefully considered in the authentication procedure, possibly based on using public key encryption.

This thesis focused on network-layer solution for location and routing optimization in the system performance point of view. As was discussed in section 3.1.2, host mobility has an effect on not only the internet protocol but also TCP and higher-layer protocols. Some time constrained applications sometimes cannot cope with the fluctuations of network bandwidth or latency due to the presence of a wireless link and host movement. Mobile hosts are more likely to introduce errors which cause packets to be dropped. The current TCP implementation hardly discriminates these from network congestion [11, 45]; some part of the protocol should be made aware of host mobility. Therefore, providing performance transparency to mobile hosts involves understanding mechanisms the transport protocol, and then adapting the protocol to the mobile computing environment [12, 41, 74]. This area also needs further research.

Bibliography

[1] B. Awerbuch and D. Peleg, "Concurrent online tracking of mobile users," in *Proc. ACM SIGCOMM 91*, Nov. 1991, pp. 221-233.

[2] B. Awerbuch, M. Luby, A. V. Goldberg and S. A. Plotkin, "Network Decomposition and Locality in Distributed Computation," in *Proc. 30th IEEE Symposium on Foundations of Computer Science*, 1989. pp. 364-369.

[3] A. Aziz, "A Scalable and Efficient Intra-Domain Tunneling Mobile-IP Scheme," *ACM SIGCOMM Computer Communication Review*, Vol. 24, No. 1, Jan. 1994

[4] B. R. Badrinath, T. Imielinski, and A. Virmani, "Locating strategies for personal communication networks," in *Proc. IEEE Globecom 92*, Dec. 1992.

[5] B. R. Badrinath, Arup Acharya and Tomasz Imielinski, "Impact of mobility on distributed computations," *ACM SIGOPS Review*, Apr. 1993.

[6] B. R. Badrinath, Arup Acharya and Tomasz Imielinski, "Structuring Distributed Algorithms for Mobile Hosts," Rutgers University, Technical Report *DCS-TR-298/WINLAB TR-55*, June 1993.

[7] P. Bhagawat and C. E. Perkins, "A Mobile Networking System based on Internet Protocol(IP)," in *Proc. USENIX 93: Mobile and Location Independent Computing*, Cambridge, MA, Aug. 1993, pp. 69-82.

[8] T. Blackwell et al., "Secure Short-Cut Routing for Mobile IP," in *Proc. USENIX Summer 1994 Technical conference*, Boston, MA, June 1994.

[9] R. T. Braden, "RFC 1122: Requirements for Internet Hosts - Communication Layers," Oct. 1992.

[10] R. Caceres, P. B. Danzig, S. Jamin and D. J. Mitzel, "Characteristics of Wide-Area TCP/IP Conversations," *Computer Communication Review*, Vol. 21, No. 4, Sept. 1991.

[11] R. Caceres and L. Iftode, "The Effects of Mobility on Reliable Transport Protocols," in *Proc. 14th Int'l conf. on Distributed Computing Systems*, June 1994.

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

[13] K. G. Carlberg, "A Routing Architecture that Supports Mobile End Systems," in *Proc. MILCOM'92*, 1992.

[14] D. R. Cheriton, "Dissemination-Oriented Communication Systems," in *Proc. of the 14th ACM Symposium on Operating Systems Principles*, Dec. 1993.

[15] G. H. Cho, "Issues and Solutions for Mobile Computing," *Unpublished Manuscript*, University of Newcastle, May 1993.

[16] G. H. Cho and L. F. Marshall, "A Multicast Service for Mobile Computing," in *Proc. 6th IEEE Workshop on Local and Metropolitan Area Networks*, San Diego, CA, Oct. 1993.

[17] G. H. Cho and L. F. Marshall, "An Efficient Location and Routing Scheme for Mobile Computing Environments," *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 5, June 1995.

[18] G. H. Cho and L. F. Marshall, "Location and Routing Optimization in Support of Internet Host Mobility," Submitted to *ACM Wireless Network*, Jan. 1996.

[19] K. C. Claffy, H. W. Braun and G. C. Polyzos "Tracking Long-Term Growth of the NSFNET," *Communications of the ACM*, Vol. 37, No. 8, Aug. 1994.

[20] D. Cohen, J. B. Postel, and R. Rom, "IP Addressing and Routing in a Local Wireless Network," *Unpublished Manuscript*, University of Southern California, July 1991.

[21] D. E. Comer, *Internetworking with TCP/IP*, Vol 1, Englewood Cliffs, NJ: Prentice Hall, 1991.

[22] S. Deering, A response to the Comments for Internet Draft "Transparent Internet Routing for IP Mobile Hosts," sent to the *mobile-ip* mailing list, Aug. 1993.

[23] D. Duchamp, S. Feiner, and G. Maguire, "Software Technology for Wireless Mobile Computing," *IEEE Network Magazine*, pp. 12-18, Nov. 1991.

[24] P. Dupont, "Local Mobility Management in IP Networks," *Unpublished Manuscript* Motorola Wireless Data Group, Sept. 1993.

[25] G. H. Forman and J. Zahorjan, "The Challenges of Mobile Computing," University of Washington, Technical Report *TR 93-11-03*, Dec. 1993.

[26] S. Ginn, "Personal Communication Services: Expanding the Freedom to Communicate," *IEEE Communications Magazine*, Vol. 29, Feb. 1991.

[27] D. Goodman, "Trends in Cellular and Cordless Communications," *IEEE Communications Magazine*, Vol. 29, No. 6, pp. 31-40, June 1991.

[28] S. C. Hedrick, "RFC 1058: Routing Information Protocol," June. 1988.

[29] A. Heybey, "The Network Simulator Version 2.1," MIT LCS Advanced Network Architecture group, Sept. 1990.

[30] H. R. Hinden and A. Sheltzer, "RFC 823: DARPA Internet gateway," Sept. 1982.

[31] T. Imielinski and B. R. Badrinath, "Mobile Wireless Computing : Solutions and Challenges in Data Management," Rutgers University, Technical Report *DCS-TR-296/WINLAB TR-49*, Feb. 1993.

[32] T. Imielinski and B. R. Badrinath, "Querying in highly distributed environments," in *Proc. 18th Int'l Conf. on VLDB*, Vancouver, Canada, Aug. 1992, pp. 41-52.

[33] J. Ioannidis, and G. Maguire Jr., "The Design and Implementation of a Mobile Internetworking Architecture," in *Proc. 1993 Winter USENIX*, San Diego, CA, Jan. 1993, pp. 491-502.

[34] J. Ioannidis, "Protocols for Mobile Internetworking," Columbia University, Ph.D Thesis, 1993.

[35] J. Ioannidis and M. Blaze, "The Architecture and Implementation of Network-Layer Security under Unix," *1993 USENIX Security*, 1993.

[36] ISO, "ISO 7498: Information processing systems - Open Systems Interconnection - Basic Reference Model," 1984.

[37] D. B. Johnson, "Mobile Host Internetworking Using IP Loose Source Routing," Carnegie Mellon University, Technical Report *CMU-CS-93-128*, Feb. 1993.

[38] D. B. Johnson, "Ubiquitous Mobile Host Internetworking," in *Proc. 4th Workshop on Workstation Operating Systems*, Oct. 1993.

[39] D. B. Johnson, "Scalable and Robust Internetwork Routing for Mobile Hosts," in *Proc. 14th Int'l conf. on Distributed Computing Systems*, June 1994.

[40] D. B. Johnson, A. Myles, and C. Perkins, "Route Optimalization in Mobile IP," Mobile-IP Working Group of IETF, Internet Draft *draft-ietf-mobileip-optim-01.txt*, Carnegie Mellon University, Nov., 1994, Working Draft; Expires Sept. 1995.

[41] K. Keeton, B. A. Mah, S. Seshan, R. H. Katz, and D. Ferrari, "Providing Connection-oriented Network Services to Mobile Hosts," in *Proc. USENIX 93: Mobile and Location Independent Computing*, Cambridge, MA, Aug. 1993, pp. 83-102.

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

[43] P. Krishna, M Chatterjee, N. H. Vaidya and D. K. Pradhan, "A Cluster-based Approach for Routing in Ad-Hoc Networks," in *Proc. USENIX 95: Mobile and Location Independent Computing*, Ann Arbor, Michigan, Apr. 1995, pp. 1-10.

[44] B. M. Leiner, "Internet Technology," *Communications of the ACM*, Vol. 37, No. 8, Aug. 1994.

[45] P. Manzoni, D. Ghosal and G. Serazzi, "Impact of Mobility on TCP/IP: An Integrated Performance Study," *IEEE Journal on Selected Area in Communications*, Vol. 13, No. 5, June 1995.

[46] B. Marsh, F. Douglis, and R. Caceres "Systems Issues in Mobile Computing," Matsushita Laboratory, Technical Report *MITL-TR-50-93*, Feb. 1993.

[47] J. P. Mello Jr. and P. Wayner, "Wireless Mobile Communications," BYTE, Feb. 1993.

[48] D. Mills, "RFC 904: Exterior Gateway Protocol formal specification," Apr. 1984.

[49] J. Mogul and J. Postel, "RFC 950: Internet Standard subnetting procedure," Aug. 1985.

[50] P. Mockapetris, "RFC 1034: Domain Names - Concepts and Facilities," Nov. 1987.

[51] A. Mukherjee and D. P. Siewiorek, "Mobility: A Medium for Computation, Communication, and Control," in *Proc. Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, Dec. 1994, pp. 8-11.

[52] A. Myles and D. Skellern, "Comparison of Mobile Host Protocols for IP," *Computer Networks and ISDN Systems*, Vol. 26, pp. 349-355, 1993.

[53] A. Myles, D. B. Johnson, and C. Perkins, "A Mobile Host Protocol Supporting Route Optimization and Authentication," *IEEE Journal on Selected Area in Communications*, Vol. 13, No. 5, June 1995.

[54] R. Perlman, *Interconnections: Bridges and Routers*, Addison-Wesley, 1992.

[55] C. Perkins, "Providing Continuous Network Access to Mobile Hosts using TCP/IP," *Computer Networks and ISDN Systems*, Vol. 26, pp. 357-369, 1993.

[56] C. Perkins, "IP Mobility Support," Mobile-IP Working Group of IETF, Internet Draft *draft-ietf-mobileip-protocol-10.txt*, IBM Corporation, May 1995, Working Draft; Expires November 1995.

[57] D. C. Plummer, "RFC 826: Ethernet Address Resolution Protocol," Nov. 1982.

[58] J. Postel, "RFC 791: Internet Protocol," Sept. 1981.

[59] J. Postel, "RFC 792: Internet Control Message Protocol," Sept. 1981.

[60] T. Rappaport, "The Wireless Revolution," *IEEE Communications Magazine*, Vol. 29, pp 52-71, Nov. 1991.

[61] Y. Rekhter and C. Perkins, "Optimal routing for mobile hosts using IP's Loose Source Route option," Mobile IP Working Group of IETF, Internet Draft, IBM Corporation, Oct. 1992, Working Draft; Expires Jan. 1993.

[62] R. L. Rivest, "RFC 1321: The MD5 Message-Digest Algorithm," Apr. 1992.

[63] M. Santifaller, *TCP/IP and NFS: Internetworking in a UNIX Environment*, Addison-Wesley, 1991.

[64] M. Satyanarayanan, "Report on the IEEE Mobile Computing Systems and Applications Workshop," ;*login:*, Vol. 20, No. 2, Apr. 1995.

[65] B. Schilit and D. Duchamp, "Adaptive Remote Paging for Mobile Computers," Columbia University, Technical Report *TR CUCS-004-91*, Feb. 1991.

[66] W. D. Sincoskie and C. J. Cotton, "Extended Bridge Algorithms for Large Networks," *IEEE Network*, Vol. 2, No. 1, Jan. 1988.

[67] C. Tait and D. Duchamp, "Service Interface and Replica Management Algorithm for Mobile File System Clients," in *Proc. 1st Int'l Conf. Parallel and Distributed Information System*, Dec. 1991.

[68] M. P. Tasman, "Protocols and Caching Strategies in Support of Internetwork Mobility," University of Wisconsin-Madison, Ph.D Thesis, 1994.

[69] F. Teraoka, K. Claffy, and M. Tokoro, "Design, Implementation, and Evaluation of Virtual Internet Protocol," in *Proc. 12th Int'l Conf. on Distributed Computing Systems*, Los Alamitos, CA, June 1992, pp. 170-177.

[70] F. Teraoka, M. Tokoro, "Host Migration Transparency in IP Networks: The VIP Approach," *ACM Computer Communication Review*, Vol. 23, No. 1, Jan. 1993.

[71] F. Teraoka, "A Study on Host Mobility in Wide Area Computer Networks," Ph.D Thesis, Jan. 1993.

[72] K. Uehara, F. Teraoka, H. Sunahara and J. Murai, "Enhancement of VIP and its Evaluation," in *Proc. INET'93*, Aug. 1993.

[73] H. Wada, T. Yozawa, T. Ohnishi, and Y. Tanaka, "Mobile Computing Environment Based on Internet Packet Forwarding," in *Proc. 1993 Winter USENIX*, San Diego, CA, Jan. 1993, pp. 503-517.

[74] R. Yavatkar, N. Bhagawat, "Improving Enad-to-End Performance of TCP over Mobile Internetworks," in *Proc. Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, Dec. 1994, pp. 146-152.

[75] R. Yuan, "An Adaptive Routing Scheme for Wireless Mobile Computing," *Unpublished Manuscript* NEC Systems Laboratory, Sept. 1993.

Appendix A

Configuration Input

A.1 Component Definition

A.1.1 Internet

```
component 'INTERNET' '192.127.110.99' INTERNET 320 390
param 'INTERNET'     # 192.127.110.99
param 192.127.110.99 # INTERNET
param 128            # Link Speed (KBit/sec): 128
param 1000           # Latency (usec): 1000
```

A.1.2 Mobile Host

```
component 'MH 101' '192.165.141.1' MHOST 215 815
param 'MH 101'       # 192.165.141.1
param 192.165.141.1  # MH 101
param 500            # Mean packet processing time (uSec): 500
param 100            # Packet processing time variation (uSec): 100
param 0.0000001      # Poisson input (pkts/usec) : 0.0000001

component 'MH 102' '192.165.141.2' MHOST 235 820
```

```
param 'MH 102'        # 192.165.141.2
param 192.165.141.2   # MH 102
param 500             # Mean packet processing time (uSec): 500
param 100             # Packet processing time variation (uSec): 100
param 0.0000001       # Poisson input (pkts/usec) : 0.0000001

component 'MH 111' '192.165.142.1' MHOST 305 815
param 'MH 111'        # 192.165.142.1
param 192.165.142.1   # MH 111
param 500             # Mean packet processing time (uSec): 500
param 100             # Packet processing time variation (uSec): 100
param 0.0000001       # Poisson input (pkts/usec) : 0.0000001

...
```

A.1.3 Wirelessnet

```
component 'WIRELESS 10' '192.165.141.99' WIRELESS 225 720
param 'WIRELESS 10'   # 192.165.141.99
param 192.165.141.99  # WIRELESS 10
param 1000            # Link Speed (KBit/sec): 1040
param 10000           # Latency (uSec): 10000

component 'WIRELESS 11' '192.165.142.99' WIRELESS 305 720
param 'WIRELESS 11'   # 192.165.142.99
param 192.165.142.99  # WIRELESS 11
param 1000            # Link Speed (KBit/sec): 1000
param 10000           # Latency (uSec): 10000

component 'WIRELESS 12' '192.165.151.99' WIRELESS 395 720
param 'WIRELESS 12'   # 192.165.151.99
param 192.165.151.99  # WIRELESS 12
param 1000            # Link Speed (KBit/sec): 1000
param 10000           # Latency (uSec): 10000
```

```
component 'WIRELESS 13' '192.165.152.99' WIRELESS 475 720
param 'WIRELESS 13'    # 192.165.152.99
param 192.165.152.99   # WIRELESS 13
param 1000             # Link Speed (KBit/sec): 1000
param 10000            # Latency (uSec): 10000

...
```

A.1.4 Ethernet

```
component 'ETLINK 10' '192.165.100.99' ETLINK 570 520
param 'ETLINK 10'      # 192.165.100.99
param 192.165.100.99   # ETLINK 10
param 100000           # Link Speed (KBit/sec): 100000
param 500              # Latency (uSec): 500

component 'ETLINK 11' '192.165.110.99' ETLINK 310 590
param 'ETLINK 11'      # 192.165.110.99
param 192.165.110.99   # ETLINK 11
param 100000           # Link Speed (KBit/sec): 100000
param 500              # Latency (uSec): 500

component 'ETLINK 12' '192.165.120.99' ETLINK 570 590
param 'ETLINK 12'      # 192.165.120.99
param 192.165.120.99   # ETLINK 12
param 100000           # Link Speed (KBit/sec): 100000
param 500              # Latency (uSec): 500

...
```

A.1.5 Mobility Agent

```
component 'MA 10' '192.165.140.101' MAGENT 240 690
```

```
param 'MA 10'          # 192.165.140.101
param 192.165.140.101 # MA 10
param 2000             # Delay to process a packet (uSec): 2000
param 1                # Speed of magent (uSec/kbyte): 1
param -1               # Max output queue size (-1 = inf): -1
param -1               # Max input queue size (-1 = inf): -1

component 'MA 11' '192.165.140.102' MAGENT 315 690
param 'MA 11'          # 192.165.140.102
param 192.165.140.102 # MA 11
param 2000             # Delay to process a packet (uSec): 2000
param 1                # Speed of magent (uSec/kbyte): 1
param -1               # Max output queue size (-1 = inf): -1
param -1               # Max input queue size (-1 = inf): -1

component 'MA 12' '192.165.150.101' MAGENT 405 690
param 'MA 12'          # 192.165.150.101
param 192.165.150.101 # MA 12
param 2000             # Delay to process a packet (uSec): 2000
param 1                # Speed of magent (uSec/kbyte): 1
param -1               # Max output queue size (-1 = inf): -1
param -1               # Max input queue size (-1 = inf): -1

...
```

A.1.6 Mobility Router

```
component 'MR 10' '192.127.110.101' MROUTER 605 480
param 'MR 10'          # 192.127.110.101
param 192.127.110.101 # MR 10
param 2000             # Delay to process a packet (uSec): 2000
param 1                # Speed of mroutor (uSec/kbyte): 1
param -1               # Max output queue size (-1 = inf): -1
param -1               # Max input queue size (-1 = inf): -1
```

```
component 'MR 11' '192.165.100.101' MROUTER 345 550
param 'MR 11'          # 192.165.100.101
param 192.165.100.101 # MR 11
param 2000             # Delay to process a packet (uSec): 2000
param 1                # Speed of mroutor (uSec/kbyte): 1
param -1               # Max output queue size (-1 = inf): -1
param -1               # Max input queue size (-1 = inf): -1

component 'MR 12' '192.165.100.102' MROUTER 605 550
param 'MR 12'          # 192.165.100.102
param 192.165.100.102 # MR 12
param 2000             # Delay to process a packet (uSec): 2000
param 1                # Speed of mroutor (uSec/kbyte): 1
param -1               # Max output queue size (-1 = inf): -1
param -1               # Max input queue size (-1 = inf): -1

...
```

A.2 Neighbor Definition

```
neighbor 'MH 101' 'WIRELESS 10'
neighbor 'MH 102' 'WIRELESS 10'
neighbor 'MH 111' 'WIRELESS 11'
...

neighbor 'MA 10' 'WIRELESS 10'
neighbor 'MA 11' 'WIRELESS 11'
neighbor 'MA 12' 'WIRELESS 12'
...

neighbor 'ETLINK 10' 'MR 10'
neighbor 'ETLINK 10' 'MR 11'
neighbor 'ETLINK 10' 'MR 12'
```

```
neighbor 'ETLINK 10' 'MR 13'
neighbor 'ETLINK 11' 'MR 11'
neighbor 'ETLINK 11' 'MR 14'
neighbor 'ETLINK 11' 'MR 15'
neighbor 'ETLINK 12' 'MR 12'
neighbor 'ETLINK 12' 'MR 16'
neighbor 'ETLINK 13' 'MR 13'
neighbor 'ETLINK 13' 'MR 17'
...

neighbor 'INTERNET' 'MR 10'
```

A.3 Local Region Definition

```
# Local Region (LR 1)
#
lr_hosts.1 'MH 101' 'MH 102' 'MH 111' 'MH 112' 'MH 121' 'MH 122'
lr_hosts.1 'MH 123' 'MH 131' 'MH 132' 'MH 141' 'MH 142' 'MH 151'
lr_hosts.1 'MH 152' 'MH 161' 'MH 162' 'MH 171' 'MH 172' 'MH 181'
lr_hosts.1 'MH 182' 'MH 191' 'MH 192' 'MH 193'

# Local Region (LR 2)
#
lr_hosts.2 'MH 201' 'MH 202' 'MH 211' 'MH 212' 'MH 221' 'MH 222'
lr_hosts.2 'MH 223' 'MH 231' 'MH 232' 'MH 241' 'MH 242' 'MH 251'
lr_hosts.2 'MH 252' 'MH 261' 'MH 262' 'MH 271' 'MH 272' 'MH 281'
lr_hosts.2 'MH 282' 'MH 291' 'MH 292' 'MH 293'

# Local Region (LR 3)
#
lr_hosts.3 'MH 301' 'MH 302' 'MH 303' 'MH 311' 'MH 312' 'MH 321'
lr_hosts.3 'MH 322' 'MH 323'

# Local Region (LR 4)
```

```
#
lr_hosts.4 'MH 401' 'MH 402' 'MH 403' 'MH 411' 'MH 412' 'MH 413'
lr_hosts.4 'MH 414' 'MH 415'
```

A.4 Mobile Host Routing Definition

```
m_route 'MH 101' 'WIRELESS 10' 'MA 10' 'MR 10'
m_route 'MH 102' 'WIRELESS 10' 'MA 10' 'MR 10'
m_route 'MH 111' 'WIRELESS 11' 'MA 11' 'MR 10'
m_route 'MH 112' 'WIRELESS 11' 'MA 11' 'MR 10'
m_route 'MH 121' 'WIRELESS 12' 'MA 12' 'MR 10'
...

m_route 'MH 201' 'WIRELESS 20' 'MA 20' 'MR 20'
m_route 'MH 202' 'WIRELESS 20' 'MA 20' 'MR 20'
m_route 'MH 211' 'WIRELESS 21' 'MA 21' 'MR 20'
m_route 'MH 212' 'WIRELESS 21' 'MA 21' 'MR 20'
m_route 'MH 221' 'WIRELESS 22' 'MA 22' 'MR 20'
...
```

A.5 Internetwork Routing Definition

```
i_route 'MA 10' 'WIRELESS 10' 'MH 101'
i_route 'MA 10' 'WIRELESS 10' 'MH 102'
i_route 'MA 10' 'ETLINK 14' 'MA 11' 'WIRELESS 11'
i_route 'MA 10' 'ETLINK 14' 'MR 14' 'Default'

i_route 'MA 11' 'WIRELESS 11' 'MH 111'
i_route 'MA 11' 'WIRELESS 11' 'MH 112'
i_route 'MA 11' 'ETLINK 14' 'MA 10' 'WIRELESS 10'
i_route 'MA 11' 'ETLINK 14' 'MR 14' 'Default'
```

```
...

i_route 'MR 11' 'ETLINK 10' 'MR 12' 'MA 14'
i_route 'MR 11' 'ETLINK 10' 'MR 12' 'MA 15'
i_route 'MR 11' 'ETLINK 10' 'MR 13' 'MA 16'
i_route 'MR 11' 'ETLINK 10' 'MR 13' 'MA 17'
i_route 'MR 11' 'ETLINK 10' 'MR 13' 'MA 18'
i_route 'MR 11' 'ETLINK 10' 'MR 13' 'MA 19'
i_route 'MR 11' 'ETLINK 10' 'MR 12' 'WIRELESS 14'
i_route 'MR 11' 'ETLINK 10' 'MR 12' 'WIRELESS 15'
i_route 'MR 11' 'ETLINK 10' 'MR 13' 'WIRELESS 16'
i_route 'MR 11' 'ETLINK 10' 'MR 13' 'WIRELESS 17'
i_route 'MR 11' 'ETLINK 10' 'MR 13' 'WIRELESS 18'
i_route 'MR 11' 'ETLINK 10' 'MR 13' 'WIRELESS 19'
i_route 'MR 11' 'ETLINK 11' 'MR 14' 'MA 10'
i_route 'MR 11' 'ETLINK 11' 'MR 14' 'MA 11'
i_route 'MR 11' 'ETLINK 11' 'MR 15' 'MA 12'
i_route 'MR 11' 'ETLINK 11' 'MR 15' 'MA 13'
i_route 'MR 11' 'ETLINK 11' 'MR 14' 'WIRELESS 10'
i_route 'MR 11' 'ETLINK 11' 'MR 14' 'WIRELESS 11'
i_route 'MR 11' 'ETLINK 11' 'MR 15' 'WIRELESS 12'
i_route 'MR 11' 'ETLINK 11' 'MR 15' 'WIRELESS 13'
i_route 'MR 11' 'ETLINK 10' 'MR 10' 'Default'

...

i_route 'MR 10' 'ETLINK 10' 'MR 11' 'MA 10'
i_route 'MR 10' 'ETLINK 10' 'MR 11' 'MA 11'
i_route 'MR 10' 'ETLINK 10' 'MR 11' 'MA 12'
i_route 'MR 10' 'ETLINK 10' 'MR 11' 'MA 13'
i_route 'MR 10' 'ETLINK 10' 'MR 12' 'MA 14'
i_route 'MR 10' 'ETLINK 10' 'MR 12' 'MA 15'
i_route 'MR 10' 'ETLINK 10' 'MR 13' 'MA 16'
i_route 'MR 10' 'ETLINK 10' 'MR 13' 'MA 17'
i_route 'MR 10' 'ETLINK 10' 'MR 13' 'MA 18'
```

```
i_route 'MR 10' 'ETLINK 10' 'MR 13' 'MA 19'
i_route 'MR 10' 'ETLINK 10' 'MR 11' 'WIRELESS 10'
i_route 'MR 10' 'ETLINK 10' 'MR 11' 'WIRELESS 11'
i_route 'MR 10' 'ETLINK 10' 'MR 11' 'WIRELESS 12'
i_route 'MR 10' 'ETLINK 10' 'MR 11' 'WIRELESS 13'
i_route 'MR 10' 'ETLINK 10' 'MR 12' 'WIRELESS 14'
i_route 'MR 10' 'ETLINK 10' 'MR 12' 'WIRELESS 15'
i_route 'MR 10' 'ETLINK 10' 'MR 13' 'WIRELESS 16'
i_route 'MR 10' 'ETLINK 10' 'MR 13' 'WIRELESS 17'
i_route 'MR 10' 'ETLINK 10' 'MR 13' 'WIRELESS 18'
i_route 'MR 10' 'ETLINK 10' 'MR 13' 'WIRELESS 19'

i_route 'MR 10' 'INTERNET' 'MR 30' 'MA 30'
i_route 'MR 10' 'INTERNET' 'MR 30' 'MA 31'
i_route 'MR 10' 'INTERNET' 'MR 30' 'MA 32'
i_route 'MR 10' 'INTERNET' 'MR 30' 'WIRELESS 30'
i_route 'MR 10' 'INTERNET' 'MR 30' 'WIRELESS 31'
i_route 'MR 10' 'INTERNET' 'MR 30' 'WIRELESS 32'

i_route 'MR 10' 'INTERNET' 'MR 40' 'MA 40'
i_route 'MR 10' 'INTERNET' 'MR 40' 'MA 41'
i_route 'MR 10' 'INTERNET' 'MR 40' 'WIRELESS 40'
i_route 'MR 10' 'INTERNET' 'MR 40' 'WIRELESS 41'
i_route 'MR 10' 'INTERNET' 'MR 20' 'Default'
```