COMPUTING SCIENCE

An Efficient Location and Routing Scheme for Mobile Computing Environments

G. Cho and L.F. Marshall

TECHNICAL REPORT SERIES

No. 508  April, 1995
An Efficient Location and Routing Scheme for Mobile Computing Environments

G. Cho and L.F. Marshall

Abstract

One of the most important issues affecting host mobility is the location and routing scheme that allows hosts to move seamlessly from one site to another. This paper presents a method that exploits the locality properties of a host's pattern of movement and access history. Two concepts, "local region" and "patron service", are introduced based on the locality features. 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. These are used to confine the effects of a host moving, so location updates are sent only to its local area, and to those source hosts which are most likely to call again. Our scheme has the advantages of limiting location updates, and providing optimal routing, whilst increasing network and host scalability.
Bibliographical details

CHO, Gihwan
An Efficient Location and Routing Scheme for Mobile Computing Environments
(University of Newcastle upon Tyne, Computing Science, Technical Report Series, no. 508)

Added entries
UNIVERSITY OF NEWCASTLE UPON TYNE.
MARSHALL, Lindsay Forsyth

Abstract
One of the most important issues affecting host mobility is the location and routing scheme that allows hosts to move seamlessly from one site to another. This paper presents a method that exploits the locality properties of a host's pattern of movement and access history. Two concepts, "local region" and "patron service", are introduced based on the locality features. 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. These are used to confine the effects of a host moving, so location updates are sent only to its local area, and to those source hosts which are most likely to call again. Our scheme has the advantages of limiting location updates, and providing optimal routing, whilst increasing network and host scalability.

About the author
G. Cho is currently a PhD student in the Department of Computing Science at the University of Newcastle upon Tyne. From 1987 to 1992 he worked at the ETRI (Electronics and Telecommunications Research Institute), Korea, where he implemented computer communication software.

Dr. L.F. Marshall is a Lecturer in the Department of Computing Science at the University of Newcastle upon Tyne.

Suggested keywords
BACKFIRING    BEACONING    ENCAPSULATION    INTERNET
LOCAL REGION LOCALITY MOBILE COMPUTING MOBILE HOST
MOBILITY AGENT MOBILITY BINDING MOBILITY ROUTER ORPHAN PACKET
PATRON REDIRECTION REGISTRATION TRIANGLE ROUTING
TUNNELLING

Suggested classmarks (primary classmark underlined)
Dewey (18th): 621.380413 384.5 621.38415
U.D.C. 621.391 654.165 658.284
An Efficient Location and Routing Scheme for Mobile Computing Environments

Gihwan Cho and Lindsay F. Marshall

Department of Computing Science
University of Newcastle
Newcastle upon tyne
NE1 7RU, England

Abstract

One of the most important issues affecting host mobility is the location and routing scheme that allows hosts to move seamlessly from one site to another. This paper presents a method that exploits the locality properties of a host’s pattern of movement and access history. Two concepts, “local region” and “patron service”, are introduced based on the locality features. 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. These are used to confine the effects of a host moving, so location updates are sent only to its local area, and to those source hosts which are most likely to call again. Our scheme has the advantages of limiting location updates, and providing optimal routing, whilst increasing network and host scalability.

Keywords: mobile computing, Internet, mobile host, triangle routing, tunneling, encapsulation, beaoning, registration, locality, redirection, mobility binding, mobility agent, mobility router, local region, patron, back firing, orphan packet
1 Introduction

Currently, in an Internet-based mobile computing system, mobile hosts can not 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 packets are routed based on the network number. So, if a host moves to a new network, the packets bound for it can not be delivered without extra redirection support. A number of proposals have been made for supporting host mobility within the Internet environment [9, 10, 11, 12, 13, 15]. The fundamental concepts underlying each proposal are similar: 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 location which is a forwarding (current) IP address, and a mechanism to forward packets to the mobile host’s current location. The major differences between these proposals are the way location information is propagated in order to trace a moving host and how packets are forwarded to the current location of a mobile host.

There are four common strategies proposed for finding the current location of a mobile host: broadcast [9], address consultant [1, 3], location cache [11, 14, 15, 16] and forwarding pointers [9, 10, 16]. Broadcast is too expensive for frequent use in a large network, but selective broadcast in a given area is workable. An address consultant is a directory system - central or distributed - that holds the current location of moving hosts. This brings with it efficiency and availability problems. A location cache requires costly mechanisms to ensure its consistency, but can partially be used in conjunction with other schemes. With the forwarding pointer method, each time a mobile host moves to another location, a reference to the new location is deposited somewhere, for instance at the old location. This method is among the fastest, and most useful in the IP environment, but it is prone to inefficient routing as described below.

In an IP-based network, there are two ways of forwarding packets to a moving host: source routing [12] using the IP option (loose source routing), and encapsulation [9, 10, 11, 13, 15, 16] using a protocol packet inside each IP packet. It has been observed that the source routing protocol is implemented to only a few hosts today, and routing would always be sub-optimal for UDP traffic [18]. In the research community, it is generally agreed that packet forwarding is best managed by some form of tunneling\footnote{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.} between the source and the destination, based on encapsulation.

Although much work has been done in this area, it has been concentrated on supporting the basic functionalities of host mobility. The problems that mobile computing now faces are how to distribute location information effectively, to keep it consistent and as up-to-date as possible, and then how to utilize the information efficiently to forward packets to mobile hosts. This paper describes a location and routing scheme exploiting the locality properties of a host's pattern of movement and access history. Our scheme
has the advantages of limiting location updates to a designated area and hosts and providing nearly optimal routing\(^2\) for most communication, whilst increasing network and host scalability.

### 1.1 Infrastructure

Clearly, a static network infrastructure is inadequate for supporting host mobility. In our architecture, a mobile computing environment consists of a set of mobile hosts, mobility agents (base stations in a cellular network), and mobility routers. A mobile host may move from one network to another within the Internet whilst its connections remain unchanged. 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 mobile host connects to a subnetwork, it finds and then registers with a mobility agent - a router that will forward packets to the host. If the mobile host is initially registered with this agent we shall refer to it as the home agent, otherwise we shall call it a foreign agent. This agent also provides the wireless interface between mobile hosts and the rest of the network. It has the ability to tunnel packets to foreign agents 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 maintains a set of mobility bindings for the hosts currently under its control, and also much of the protocol related states for those hosts. A mobility binding is the association of a host's home agent with a current foreign agent. This is done by recording their IP addresses. The home agent maintains a home list identifying all mobile hosts it is configured to serve. Each foreign agent records the mobility bindings in force as a visitor list for each mobile host that it is currently serving. In addition, agents maintain a forwarding list that records the mobility bindings in force for each of its hosts that is away from home, or for each host that is not currently registered in its visitor list.

A mobility router is initially a normal router which is connected to the mobility agents directly or indirectly. A source host is any host with which a given mobile host is communicating. The following lists the basic functions\(^3\) of the mobility entities, without giving details of how to implement them. Based on these functions, we build an efficient location and routing scheme for mobile computing environments. However, our scheme will be described as independently of these functions as possible.

---

\(^2\)An optimal route is the route a packet would normally take between two stationary hosts using the conventional IP routing protocols.

\(^3\)Most of these features are drawn from the existing proposals [9, 10, 11, 17, 12]. A recommendation for the basic functions has been discussed in the IETF Mobile-IP group.
• 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 procedure 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 foreign agent, and previous foreign 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.

2 Motivation

The key service for providing seamless connectivity to mobile hosts is the creation and maintenance of a packet forwarding tunnel between a known mobility agent (the home agent) and the host’s current foreign agent. Obviously, a routing decision must be made based on the location information that is available. 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 the worst 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. Figure 1 shows this situation.

![Figure 1: Triangle routing](image)

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 to then 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.
Thus, packet routing efficiency depends critically on the propagation of location information into the network as a whole. Moreover, in devising a location information update procedure, it is important to keep in mind that excessive information propagation can be wasteful of network resources, whilst insufficient location information leads to non-optimal routing. This is especially important in the Internet environment, as host mobility demands frequent (and widespread) updates.

Even though the location and routing scheme is of such significance for handling host mobility, not much work has yet been devoted to improving its efficiency. \[13, 14, 15\] proposes a location scheme that does consider routing efficiency. This sends a location notification when one entity determines that another entity might have an incorrect location for a mobile host. However, its location cache size is proportional to the number of mobile hosts, and it seems to flood the network with the notifications.

Two other proposals address the location problem but do not consider implementation details. The work in \[19\] proposes a formal model for the tracking of mobile users which is based on a hierarchy of regional directories, that reflect spatial locality. However, it does not consider the impact of temporal locality on host moving and calling. The work from Rutgers University \[5, 6\], relies on the mobility profile of the mobile host. Partitions are defined depending on the user profile for host mobility, and are utilized to locate a mobile host by taking advantage of incomplete information\[4\]. Unfortunately, it does not consider the source of calls, only their frequency. Neither of these schemes deal with efficient routing.

As an orthogonal approach to the above, two systems have been proposed that consider routing efficiency. \[20\] presents a local mobility management scheme that limits the scope and frequency of location update for a moving host within local area boundaries. \[21\] proposes a concept, called a friend network, for building the access history into the routing scheme. Each of these provides only partial and conceptual solutions for the location and routing problem.

What we would like to achieve in the present work is a unified scheme that provides optimal routing (bypassing the default reliance on routes through the home agent), with as few location updates (hiding host mobility locally as far as possible, and, if need be, informing only those hosts that may need the location information). To do this let us first look at the locality characteristics of hosts' movement patterns and access histories.

\[4\] This approach exploits weakened consistency of location information to minimize its updates. It allows certain location information to become stale, but guarantees that the real location and the location which is known are always in the same partition.
3 Local mobility in mobile computing

When a host moves, it may change the mobility agent by which it is attached quite frequently, and may then generate a significant amount of location update. Moreover, 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 without warning and in the midst of data transfers.

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.

Intuitively, a local region can be defined for each mobile host by including those subnetworks between which the mobile host often moves 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 uses a hierarchical addressing scheme based on geography or network topology. As we shall see in the next section, it also may be useful to define a local region so as to include source hosts which frequently interact with the mobile host. Here, a local region naturally reflects the spatial locality by taking into account of temporal locality for the host moving pattern as well as for the calling nature of source hosts.

For example, Figure 2 shows the weekly routine of a professor who works mainly at the computing department, goes to the computing service for 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. Obviously, the computing department is a local region for the professor. It is desirable to use the local region to provide a means of maintaining a degree of knowledge about the professor’s whereabouts, so as to cut down the location and routing cost.

```
3 hours
Computing Service
```

```
4 days
Computing Department
```

```
1 day
Company
```

Figure 2: A host moving pattern

Each mobile host has a different local region depending on its interests. A local region 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. The local region is similar to the partition concept in [6] and the local mobility management in [20], but there
it is only used to limit the area of location update. In our scheme, the local region is employed to optimize packet routing, as well as to bound location updates.

A host tends to communicate with a limited number of source hosts which have an interest in contacting it. 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. For each mobile host, it seems useful to establish patron hosts which are the source hosts where the majority of traffic for the mobile host originated, and which are therefore highly likely to call again. Figure 3 shows a calling scenario from a sequence of source hosts to a mobile host. In this example, the host would certainly select A, B and F as its patron hosts.


Figure 3: A calling pattern

It is also reasonable to record only the patron hosts outside the local region. This condition reflects spatial locality based on frequency of calling. When a mobile host leaves its local region, it notifies its new location to the patron hosts, so they can use the location information for their next call. A host would determine its patron hosts by monitoring accesses to it. This concept is analogous to the friend network in [21], but makes use of the mobility pattern for each host to improve the utilization of the call history.

It is also important to point out that locality of mobility and calling varies over time: frequency and distance vary according to the user’s current interests. A location and routing scheme must take into account this tendency, in order to make it more useful. Also, location updates that sometimes result in sub-optimal packet delivery for infrequently visited source hosts may be acceptable. If a scheme has source hosts that tolerate the use of stale location information, the number of location updates would be significantly decreased. These features can easily be incorporated into the proposed scheme.

4 Local region and patron control

In this section, we show how the local region and patron concepts are used to realize a unified scheme for efficient host location as well as optimal routing. We first define a basic scheme, then describe the control structure for the two concepts based on it. The overall scheme is then explained with some examples.
4.1 Basic scheme

The basic functions described in section 1.1 define a home-based forwarding scheme. Packets destined to a mobile host are routed through that host's home agent, which then tunnels each packet to the host's current foreign agent. Each time a mobile host moves to other location, the mobile host invokes a registration procedure to create the mobility bindings, which include the forwarding list of the home agent and the foreign agent it has just left (if any). The previous foreign agent also resets its visitor list for the host. The foreign agent currently serving the mobile host records the mobile host's address in its visitor list. These agents are then all aware of the same address for the mobile host.

When a foreign agent receives a tunneled packet addressed to itself, the agent decapsulates it, if necessary, and consults its visitor list or forwarding list. The agent will deliver the packet directly to a mobile host named in its visitor list, or encapsulate the packet to send to the foreign agent named in the 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 tunnels the packet to the foreign agent currently serving the mobile host, which finally delivers it locally to the mobile host.

4.2 Local region control

When a mobile host first joins the network or its user's interest area changes, the 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. Here, a mobility router, like a mobility agent, has the ability to tunnel packets to a mobile host which is registered within its subnetworks. The root of the hierarchy is a mobility router which has the special duty of redirecting (tunneling) packets to or from a local region. We say that this root acts as a redirection agent. Even if a mobile host's users define their own local region, the redirection agent may be assigned by an authorized network manager who is familiar with the network topology of a given administration domain.

The redirection agent maintains a list - the redirection list - to preserve the location information for the hosts that have appointed this as their redirection agent. Each time a host moves, the mobile host must create an additional mobility binding in its redirection agent, including its home agent and previous foreign agent, as in the basic scheme. The redirection agent uses this redirection list to redirect any packets passing through itself into the current location of the host.

The location role of the redirection agent within a local region can be described using the redirection tree concept. If the source host is in the same subtree as the destination host and the destination's home agent is in the other subtree its redirection agent as a
centre, then the packets are intercepted and forwarded by the redirection agent, without going via the home agent (see the redirection tunneling path in Figure 5). Whilst our scheme has to maintain one more mobility binding than the basic scheme, the routing path would be much closer to optimal.

If a host moves within its local region and the packet comes from a source host outside its local region, its redirection agent will intercept the incoming packets, and forward to the mobility agent currently serving the host, bypassing the possibly lengthy route via its home agent. With the benefit of the redirection facility, a host does not need to declare its movements within 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 frame so that most packets are routed much close to their optimal paths, whilst movements within a local region need not be notified to the outside world. In the next section, the patron concept is used to improve further these benefits.

4.3 Patron control

If a mobile host moves outside its local region, the forwarding path via its home agent becomes much longer, at least for the packets forwarded by the redirection agent for its local region. By definition, the majority of calls to a host are still from what we called the patron hosts - those that are interested in the host. Each mobile host records the mobility binding, that is the current location, of those source hosts in its patron list from a receiver's standpoint. It records only hosts outside its local region, and only does this when located within its local region.

When a mobile host leaves or comes back to its local region and registers with a mobility agent, the host also sends registration packets to the redirection agent, home agent and previous foreign agent. By comparing the mobility binding and the registration information, the redirection agent can determine if the host has just crossed its local region, and, if so, will send a notice packet to the host. On receiving the (cross local region) notice packet, the mobile host sends its new location to the hosts on its patron list. We will refer to this as the patron service. The patrons then record the host's current location information in their calling lists, 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 standpoint, for those hosts with which they have most frequently interacted.

As a result, source hosts that access a host frequently, even if it is located far from them, will keep up-to-date location information about it. The traffic from the patron

---

5 In this case, packets always go via the redirection agent, so the routing path is longest in the local region. If the source or the destination and the destination's home agent are in the same subtree, the home agent may tunnel the packet.
hosts, which covers most communications, can always achieve optimal routing, whilst traffic from infrequently visited source hosts needs to be tunneled by the home agent, or possibly by the redirection agent.

It is important to pay attention to preserving consistency between the location caches, that is the calling lists in the case of patrons. If source hosts have different location information for a destination mobile host, the location scheme must be more complicated to tolerate the inconsistency. In our scheme, a mobile host uses the same patron list for the patron services both when it leaves its local region and when it returns. The list is also refreshed only while the mobile host is in its local region. As a result the calling lists of patron hosts are always consistent (except in the presence of network partitions).

<table>
<thead>
<tr>
<th>Mobility entities</th>
<th>Mobility bindings</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redirection agent</td>
<td>Redirection list</td>
<td>Local region control</td>
</tr>
<tr>
<td>Mobility agent</td>
<td>Home list, Forwarding list</td>
<td>Basic scheme</td>
</tr>
<tr>
<td>(Home agent)</td>
<td>Visitor list, Forwarding list</td>
<td></td>
</tr>
<tr>
<td>(Foreign agent)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mobile host</td>
<td>Calling list</td>
<td>Patron control</td>
</tr>
<tr>
<td>(Sender standpoint)</td>
<td>(Receiver standpoint)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Patron list</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4: Mobility entities and their mobility bindings

Figure 4 shows the relationship between mobility entities, mobility bindings and their usage for local region and patron control. Our scheme needs three additional location caches for the redirection list, calling list and patron list. The cache size of the redirection list is restricted to the total number of mobile hosts within its serving area, which is small in comparison with the scale of the Internet. In the case of the calling and patron list, the size is limited by the number of hosts that actually communicate with each other.

4.4 Routing examples

Let us illustrate the operation of our scheme by comparing it with the basic scheme. Firstly, we show the routing path within a local region using Figure 5. Initially, a mobile host, MH₁, is registered with a mobility agent MA₁, as its home agent. MH₁ has a redirection agent, RA₁, in accordance with the initial assignment of the local region, which consists of Subnetworks 1, 2 and 3.

Suppose MH₁ moves to MA₂ from MA₁. After identifying its new mobility agent, MA₂,
MH$_1$ registers its movement with its home agent, MA$_1$, and, in the case of our scheme, with the redirection agent, RA$_1$. MA$_2$ then adds MH$_1$ to its visitor list. Now suppose host MH$_2$ sends a packet to MH$_1$.

In the basic scheme, the packet arrives at MA$_1$, which is MH$_1$'s home address. Because MA$_1$ has a forwarding list entry for MH$_1$, indicating that MH$_1$ is currently served by MA$_2$, it tunnels the packet to MA$_2$. Then, MA$_2$ uses its visitor list entry for MH$_1$ to forward the packet locally to MH$_1$.

In our scheme, the source host sends a packet to MH$_1$'s home address, 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$_2$, which currently serves the host MH$_1$, thus bypassing the mobile host's home agent (see the redirection tunneling path). Then, MA$_2$ delivers the packet locally to MH$_1$. For the next movement from MA$_2$ to MA$_n$, on the right side of Figure 5, the packet delivery process is similar.

Now, let us describe the operation for a host moving in the Internet environment using the network configuration shown in Figure 6. A source host, SH$_1$, and Subnetwork$_p$ are located somewhere in the Internet. Suppose MH$_1$ moves from MA$_1$ to MA$_p$, then to MA$_q$, which is outside its local region. After identifying its new mobility agent, MA$_q$, MH$_1$ will try to register with its home agent, MA$_1$, its previous mobility agent, MA$_p$, and, in the case of our scheme, its redirection agent, RA$_1$.

On receiving the registration packet, RA$_1$ will realize that MH$_1$ has just crossed its local region, and tells MH$_1$. MH$_1$ sends a patron service packet to its patron hosts. Suppose again that the source host sends a packet to MH$_1$. 

Figure 5: Packet redirection within a local region
In the basic scheme, it always sends the packet to MA₁, which is MH₁'s home address. Because MA₁ has a forwarding list entry for MH₁, indicating that MH₁ is currently served by MA₂, it tunnels the packet to MA₂. Then, MA₂ forwards the packet locally, using its visitor list entry for MH₁.

In our scheme, it is possible to have two kinds of source host. One is a patron host for the destination mobile host, the other is not. If the source host is not a patron host, the packet for the mobile host is sent to MH₁'s home agent, MA₁. When the packet arrives at RA₁, it is intercepted by RA₁, which should have a redirection list entry for MH₁. RA₁ then tunnels the packet to MA₂ without visiting the mobile host’s home agent, MA₁ (see the redirection tunneling path). MA₂ forwards the packet locally to MH₁. If the source host is a patron host of MH₁, it should have the current location address for MH₁ because of the patron service described above. The source would tunnel the packet directly to MA₂ using its calling list entry, so optimal routing is achieved (see the patron tunneling path).

5 Informal protocol descriptions

In this description, we will not specify implementation details. For all the protocols specified below, if a mobility agent or host receives a packet and the destination of the packet belongs is itself, then the packet should be passed to higher layers for further processing. The normal routing mechanisms are used to forward the packet in all other
cases not specified in this description. We assume that the local region is deliberately defined by user. It is also assumed that a redirection agent can use the structure of IP addresses to determine whether a mobility agent is within its service boundary. Similarly, a host can decide whether a packet has came from a source host outside its local region or not.

Mobile hosts are connected to the rest of the network via a wireless link. The wireless medium inherently has a property that transmission of a message from a mobile host consumes more power than reception. To adapt to this asymmetric communication, we assume an architecture where much of the responsibility for information propagation resides with the mobility agent, not with the host itself. In the case of registration, a mobile host simply tells its current agent its home agent address, redirection 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.

5.1 Mobile host protocol

- If a host wants initially to register with a mobility agent, it sends a registration packet which includes the addresses of its new redirection agent to its current agent, and waits for a registration ack packet.

- If a host finds out it is under the control of a new mobility agent, it sends a registration packet which includes the addresses of its home agent, previous foreign agent(if had), and redirection agent, to its current agent, and waits for an ack packet.

- If a host receives a registration ack packet from its current agent, which says the registration procedure finished normally, it may try to send or receive data packets or there was a fault in the registration, it will retry.

- If a host receives a data packet from a source host outside its local region, it creates a patron list entry if it has no entry for the host, or it increases the reference count of the patron list entry for the source host if it already has an entry for it.

- If a host receives a patron service packet from a mobile host, it creates a calling list entry if it has no entry for that host, or it updates the calling list entry for the mobile host if it already has an entry.

- If a host receives a packet from its redirection agent, which says that it crossed its local region, it sends a patron service packet, which includes all addresses of patron hosts, to its current foreign agent.

- If a host decides to change its redirection agent, it sends a registration packet, which includes the old and new addresses of its redirection agent, to its current agent.
5.2 Home agent protocol

- If a home agent receives an initial registration packet from a mobile host, it sends a registration packet to the redirection agent for the host. On receiving a registration ack packet from the agent, it records the mobile host in its home list, and sends a registration ack packet to the host.

- If a home agent receives a registration packet from a mobile host in its home list, it sends a registration packet to the redirection agent and to the previous foreign agent for the host. On receiving a registration ack packet from them, it deletes the forwarding list entry of the host, and sends a registration ack packet to the mobile host.

- If a home agent receives a registration packet, which includes the old and new addresses of its redirection agent, from a mobile host in its home list, it sends a registration packet to the old and new redirection agents. On receiving a registration ack packet from the redirection agents, it sends a registration ack packet to the mobile host.

- If a home agent receives a registration packet addressed to a mobile host in its home list from a foreign agent, it creates a forwarding list entry if it has no entry for that host, or it updates the forwarding list entry for the host if it already has an entry, and finally sends a registration ack packet to the foreign agent.

- If a home agent receives a data packet addressed to a mobile host in its home list and forwarding list, it uses the forwarding list entry to tunnel the packet to the foreign agent that currently serves the host.

- If a home agent receives a patron service packet from a mobile host in its home list, it sends a patron packet to each host listed in the patron service packet. On receiving all registration ack packets from the patron hosts, it sends a registration ack packet to the mobile host.

5.3 Foreign agent protocol

- If a foreign agent receives a registration packet from a mobile host, it sends a registration packet to the host's redirection agent, home agent and previous foreign agent (if had). On receiving their registration ack packets, it creates a visitor list entry for the mobile host, and sends a registration ack packet to the host.

- If a foreign agent receives a registration packet, which includes old and new addresses of redirection agents, from a mobile host in its visitor list, it sends each registration packet to the old and new redirection agents. On receiving their registration ack packets, it sends a registration ack packet to the mobile host.

- If a foreign agent receives a registration packet addressed to a mobile host in its forwarding list from another mobility agent, it creates a forwarding list entry,
deletes the visitor list entry for the mobile host, and finally sends a registration
ack packet to the foreign agent.

- If a foreign agent receives a data packet addressed to a mobile host in its forwarding
list, it uses the forwarding list entry to tunnel the packet to the foreign agent that
currently serves the mobile host.

- If a foreign agent receives a patron service packet from a mobile host in its visitor
list, it sends a patron packet to each host listed in the patron service packet. On
receiving all registration ack packets from the patron hosts, it sends a registration
ack packet to the mobile host.

5.4 Redirection agent protocol

- If a redirection agent receives a registration packet from a mobility agent in its
control, it updates the redirection list entry, and sends a registration ack packet
to the agent. If the received redirection address is the same as its home agent
address, it clears the entry so as to use normal routing mechanisms.

- If a redirection agent receives a registration packet from a foreign agent which
is out of its control boundary, it updates the redirection list entry corresponding
to the mobile host, and sends a registration ack packet to the agent, including a
notice that the host has just crossed its local region boundary.

- If a redirection agent receives a tunneled or normal packet addressed to a mobile
host in its redirection list, it uses the redirection list entry to tunnel the packet to
the mobility agent that currently serves the mobile host.

6 Other issues

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 assumes that routing efficiency is the responsibility of the network
infrastructure, but that locating can effectively be handled by the mobile host itself.
These approaches, which appear to conflict with each other somewhat, fit naturally
within the existing Internet, and are well accommodated with the locality properties in
which we are interested.

From a practical point of view, our scheme stresses two goals: backward compatibility
and performance transparency. For backward compatibility we attempt to limit the
number of mobility routers within the area in which the mobile hosts are actually at-
tached, and to separate the location service from the routing scheme as far as possible,
though these co-operate for optimal packet routing. To localize the performance impact
of host mobility, we try to minimize the location update traffic with need-based propagation, and to share the protocol processing burden between the mobile host and its current agent.

### 6.1 Operating mode

When a mobile host is out of its local region, there are two possible ways to control the host’s location and routing. A mobile host may have multiple local regions for several areas based on its user’s interests, and can take advantage of the redirection facility for each area. On the other hand, a host may want to define only one local region. Therefore, a host can have two different operating modes: redirection mode when working in a local region, and direct mode for the otherwise.

For a host under direct mode, forwarding pointers are used to trace the host from the foreign agent at the time the patron service was carried out, i.e. at the time the host was out of its local region. This is sometimes prone to inefficiency due to long chains of pointers. For a mobile host in redirection mode, a redirection agent can be assigned, and a patron list can be maintained, for each of the defined local regions. Whilst maintaining the additional mobility binding comes with a cost, as long as a host moves around its local regions, most traffic to the mobile host uses optimal routing.

### 6.2 Patron service

In redirection mode, the patron service takes place whenever a host movement breaks its local region boundary. If a mobile host has multiple local regions, as the patron services progress, some patron hosts which belong to different local region may have different calling list entries for the host. However, the packets from them are always forwarded to the current location by the redirection agents of each local region.

The patron service is still appropriate for a host which uses the direct mode. But it is difficult to establish when the service has to be carried out and when the patron list should be refreshed. One possible approach is to take into account movement and access statistics. The patron service is actually a location service, and may be done at a higher layer rather than the network layer. As a result, static hosts could chose to make use of the patron service, whilst a mobile host must be capable of communicating with existing hosts that do not implement our scheme.

### 6.3 Forwarding pointers

Forwarding pointers generally require the reclamation of old pointers which have been superseded. For a mobile host which works in redirection mode, our scheme can refresh
the inaccurate forwarding lists of previous foreign agents. The source host within its local region does not cache any location information, but rather forwards the packets from the source host to its current location via the home agent or the redirection agent. A source host outside its local region might or might not be a patron host, but its packets always pass through the redirection agent for the reason described above, and are redirected to the target’s current location.

So, a mobile host in redirection mode need only preserve the most recent forwarding lists from the initiator. To do this, we introduce the concept of back firing, where whenever a hosts previous agent creates an element of the forwarding list, the agent will delete the forwarding list entry of its previous mobility agent for the host, if it had one.

6.4 Orphan packets

A mobile host can cross cell boundaries in the midst of a data transfer. In this case some packets would be delivered to the previous agent rather than the current agent until a redirection or mobility agent starts to forward them. These packets which lose their route are called orphan packets. They are 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 still arrive at the previous agent.

The systems need to forward the orphan packets to the new mobility agent, and therefore eventually to the mobile host. In contrast with a host state hand-off, this is an application hand-off and depends on the quality of service of the application. Some applications may ignore the orphan packets; some may not permit the orphan packet and may need to send them to the mobility agent currently serving the target host. By cooperating with the buffer, the previous agent may tunnel the orphan packets to the current agent when the forwarding list entry for the host is created. This is why the previous forwarding list is meaningful within a local region even though a source host never uses the previous agent address due to the consistent location cacheing of our scheme.

6.5 Security and fault tolerance

A mobility support scheme must not introduce any additional security weakness to the current Internet environment, excluding of course the inherent weaknesses of wireless communication. In our scheme, one possible security hole is a fake 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 problem is that malicious forwarding entities with location caches, such as the home agent, redirection agent or patron host, could deliberately divert any data packet. To protect against these and other security problems, we are now considering three basic ideas; 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 [23]. Each entity then would authenticate itself with its home agent whenever it needed to.

In company with the security issue, fault tolerance is of great concern to the designers of mobile computing systems. The two main problems faced are the loss of registration packets and crashes of entities that maintain location caches. When a mobility entity restarts from a crash, it must minimally recover the mobility binding for the hosts under its control. It is assumed that each mobility entity has the facility to retrieve mobility bindings from another entity, in order to make become consistent. In our scheme, 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, without missing at any mobility entities. This also protects against agent crashes during the registration procedure.

Mobile hosts are often disconnected from the rest of the network, sometimes for long periods. Disconnection in a mobile environment is distinct from failure as it is a voluntary act. A mobile host can inform the system of an impending disconnection prior to its occurrence, and must then execute a disconnection protocol. So, our scheme assumes the existence of a function for responding to disconnection requests, possibly in the home agent. This will provide buffering for incoming packets during disconnection. When the host connects again, the protocol will hand off the stored states to the new current agent.

7 Analysis

In this section, we provide an analysis of our scheme, by comparing it with others. Below are given a set of network parameters and their definitions taken from [22] that characterize the network environment shown in Figures 5 and 6. Thus, network latencies, $L_{wl}$ and $L_w$, include data link and network layer processing time. $T_{prot}$ and $T_{reg}$ comprise the update time for the mobility binding entry for a registration. $T_{forw}$ is taken from [9], and includes the time for looking up a mobility binding and for encapsulation, and if need, for decapsulation. $T_{Int}$ has an estimated value which is drawn from similar environments.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$BW_{wl}$</td>
<td>Bandwidth of the wireless link</td>
<td>1 Mbps</td>
</tr>
<tr>
<td>$BW_w$</td>
<td>Bandwidth of the wired backbone network</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>$L_{wl}$</td>
<td>Latency of the wireless link</td>
<td>7 ms</td>
</tr>
<tr>
<td>$L_w$</td>
<td>Latency of a link in the wired backbone</td>
<td>0.5 ms</td>
</tr>
<tr>
<td>$S_{reg}$</td>
<td>Upper bound on the size of a registration packet</td>
<td>50 bytes</td>
</tr>
<tr>
<td>$S_{data}$</td>
<td>Maximum size of a data packet</td>
<td>8192 bytes</td>
</tr>
</tbody>
</table>
\( T_{acq} \) Time for a mobile host to acquire a wireless channel 20 ms
\( T_{prot} \) Protocol processing time for the registration packets 3 ms
\( T_{reg} \) Time to generate the registration packets from the current agent, including protocol processing time 5 ms
\( T_{forw} \) Protocol processing time to tunnel a data packet 7 ms
\( T_{Int} \) Packet delivery time between two hosts through Internet 300 ms

We assume that our mobile computing environment consists of \( N_{MA} \) mobility agents in a local region, and \( N_{source} \) source hosts. It is difficult to predict how often a mobile host will move and how often a source host will access the mobile host. Let \( M_{rate} \) be the mobility rate, expressed in number of movements per hour. Also, let \( CM_{ratio} \) be the call to mobility ratio which was introduced in [5]. It is the average number of calls to the mobile host per move. Because overall communication time is the sum of registration time for each host movement and data transmission time, the \( CM_{ratio} \) is most significant in our analysis, so it is used as a variable parameter. We shall use \( N_{MA} = 30 \), \( N_{source} = 200 \) and \( M_{rate} = 2 \).

Other assumptions are made for \( PC_{ratio} \), the calling ratio from the patron hosts to the mobile host. It is related to the patron rate, \( P_{rate} \), the number of patron hosts divided by the number of source hosts. We assume the patron rate is 20%, so \( PC_{ratio} = 0.8 \). As in Figures 5 and 6, the packet routing path depends on the network topology in a local region. A mobile host may mostly move around the environs of its home agent. \( S_{ratio} \) is the symmetric ratio which captures the movement pattern of the host and is between the time a mobile host stays in the home agent side subtree and the time it spends in the other, with its redirection agent as a centre. It is also used to characterize the calling discipline between calls originated from a host in the local region and those from outside the local region, and the balance between moves inside the local region and outside. We shall use \( S_{ratio} = 0.7 \).

We now measure the communication time to give some idea of the motivation of our work and the effectiveness of our scheme. Firstly, let us compare three schemes, namely broadcasting, basic and ours, when a mobile host only moves around within its local region. Amongst the proposed schemes, the broadcasting and basic schemes show two extremes in the degree of location information propagation and data communication efficiency. With those for comparison, we will show the power of the redirection agent in our scheme. We will assume a mobile host works 8 hours per day, and 5 days per week. The communication time is calculated for 1 week.

When a host moves, it identifies a new mobility agent and sends a registration packet to the mobility agent using the wireless channel. The time is \( T_1 = T_{acq} + (S_{reg} / BW_{wl}) + L_{wl} + T_{reg} \), and the data transmission time in a wireless channel is \( T_2 = (S_{data} / BW_{wl}) + L_{wl} \). In the wired backbone, the time for a registration packet is \( T_3 = ((S_{reg} / BW_w) + L_w) \times \text{number_of_hops} + T_{prot} \), and data transmission time, including tunneling time, is \( T_4 = (S_{data} / BW_w) + L_w \times \text{number_of_hops} + T_{forw} \). The number_of_hops means the number of in-between entities. In Figure 5, there are three number_of_hops, long = 4, short =1 and redirect = 2.

In our architecture, the mobility agent which currently serves a mobile host has the
responsibility for delivering the registration packets to the corresponding agents which will maintain its location information. The registration time accompanied by each movement is the sum of the time to acquire a wireless channel, the transmission time of a registration packet on the wireless link, and the propagation time of the packets to corresponding agents by the current mobility agent using the wired backbone. In the basic scheme, the current mobility agent propagates the registration packet to the home agent and the previous foreign agent, and, in the case of our scheme, the redirection agent as well. In the broadcasting scheme, the registration packets are sent to all mobility agents, except itself. We will not consider the registration acknowledgement packet. For each scheme:

\[ T_{\text{reg,bas}} = T_1 + (T_3(\text{long}) \times (1 - S_{\text{ratio}}) + T_3(\text{short}) \times S_{\text{ratio}}) \times 2 \]
\[ T_{\text{reg,our}} = T_1 + (T_3(\text{long}) \times (1 - S_{\text{ratio}}) + T_3(\text{short}) \times S_{\text{ratio}}) \times 2 + T_3(\text{redirect}) \]
\[ T_{\text{reg,broad}} = T_1 + (T_3(\text{long}) \times (1 - S_{\text{ratio}}) + T_3(\text{short}) \times S_{\text{ratio}}) \times (NMA - 1) \]

There are two kinds of call originators: a source host inside the local region and one outside. The data transmission time for our scheme is depicted as \( T_{\text{data from inside}} \) and \( T_{\text{data from outside}} \) for each call originator. It consists of the time in a wireless channel for sender and receiver and the time in a wired link for which the average distance is related to the symmetric ratio of the destination. The metrics for the other schemes can be derived in a similar way.

\[ T_{\text{data from inside}} = T_2 \times 2 + T_4(\text{long}) \times (1 - S_{\text{ratio}}) + T_4(\text{short}) \times S_{\text{ratio}} \]
\[ T_{\text{data from outside}} = T_{\text{Int}} + T_4(\text{redirect}) + T_2 \]

Let \( N_{\text{move}} \) be the number of host movements, and \( N_{\text{calls}} \) be the number of calls received for given period. Since \( M_{\text{rate}} \) was assumed to be 2, \( N_{\text{move}} \) is 80, and \( N_{\text{calls}} \) is the same as multiplying \( N_{\text{move}} \) by \( CM_{\text{ratio}} \). In the case of our scheme, the total communication time is computed as below, under the scenario described above.

\[ T_{\text{total comm time}} = T_{\text{reg,our}} \times N_{\text{move}} + T_{\text{data from inside}} \times N_{\text{calls}} \times S_{\text{ratio}} \]
\[ + T_{\text{data from outside}} \times N_{\text{calls}} \times (1 - S_{\text{ratio}}) \]

Figure 7 shows the communication time over the call to mobility ratio, \( CM_{\text{ratio}} \), for the three schemes. For the registration time per move, each scheme - broadcasting, basic and ours - has 147 ms, 40 ms and 44 ms respectively. It is proportional to the number of mobility bindings which each scheme has to preserve. On the other hand, the data transmission time is the opposite to the registration time. Broadcasting has the fastest data delivery time for packets from inside the local region, 146.1 ms, 153.1 ms and 154.2 ms respectively. Our scheme has the fastest time for incoming packets from outside the local region due to the role of the redirection agent, 382.4 ms, 382.4 ms and 380.6 ms respectively. When we consider the total communication time, broadcasting is inefficient for a lower \( CM_{\text{ratio}} \) because of the higher registration time, whilst the basic scheme is poor for high \( CM_{\text{ratio}} \) because every packet has to go via the home agent. Our scheme is the most efficient throughout the range of the call to mobility ratio. Even though the difference between the three schemes is small, it shows that the efficiency of mobile computing depends on how effectively the location information is distributed as a whole.

Now, let us compute the communication time for Figure 6 in section 4.4. We compare
the basic scheme and our scheme in order to show the effectiveness of the local region and patron concept. Let us assume that a host works within its local region for one and a half days, then moves out of the its local region. The host works outside of its local region for one and a half days, and returns and works 2 days within its local region. This scenario is based on the symmetric ratio defined above.

We shall describe the derivation of our scheme only. The registration process is different depending on where the mobile host tries to connect. Registration within a local region is the same as above, $T_{reg\_out}$. If a mobile host leaves its local region and joins a mobility agent outside its local region, the host sends a registration packet to the current mobility agent. The agent sends three registration packets, one each to the home agent, previous foreign agent and redirection agent respectively. By comparing the mobility binding and the registration packet, the redirection agent then finds that the host has just crossed its local region, and sends a notice packet to the host. On receiving the packet, the mobile host sends the patron packets to the patron hosts using the same channel. In this case, the patron service time and the registration time are specified as $T_{patron}$ and $T_{reg\_cross\_out}$:

$$T_{patron} = T_2 + (T_{prot} + T_{Int}) \times (N_{source} - N_{MA}) \times (1 - P_{rate})$$
$$T_{reg\_cross\_out} = T_1 + T_{Int} + T_{reg} + (T_{Int} + T_3(redirect)) \times 2 + T_{prot} + T_{Int} + T_{patron}$$

For each movement outside the local region, the current mobility agent sends registration packets to the redirection agent and the home agent within its local region, and the previous foreign agent, so $T_{reg\_outside}$. Thereafter, if the mobile host returns to its local region, after the registration packets are sent, the redirection agent will send a notice packet to the host, just as if it had moved out its local region. The mobile host issues the patron service to the patron hosts which are the same hosts as in the previous one, so the time is $T_{reg\_cross\_in}$:

$$T_{reg\_outside} = T_1 + T_{reg} + (T_{Int} + T_3(redirect)) \times 2$$
$$T_{reg\_cross\_in} = T_1 + T_{Int} + T_3(long) \times (1 - S_{ratio}) + T_3(short) \times S_{ratio} + T_3(redirect) \times 2 + T_2 \times 2 + T_{patron}$$

There are four different cases for which to compute the time for data packet transmission, depending on where the call originates and where the mobile host is located. The first subscript shows the caller's location, and the second shows the mobile host's location. So, $T_{data\_in\_in}$ stands for when the caller and the mobile host are located in the host's local region.

$$T_{data\_in\_in} = T_2 \times 2 + T_4(long) \times (1 - S_{ratio}) + T_4(short) \times S_{ratio}$$
$$T_{data\_out\_in} = T_{Int} + T_4(redirect) + T_2$$
$$T_{data\_in\_out} = T_2 + T_4(redirect) \times (1 - S_{ratio}) + T_4(short+redirect) \times S_{ratio} + T_{Int}$$
$$T_{data\_out\_out} = (2 \times T_{Int} + T_{forw}) \times (1 - P_{rate}) + (T_{Int} + T_{forw}) \times P_{rate} + T_2$$

Let $N_{move\_in}$ be the number of movements within the local region, and $N_{move\_out}$ be the number of movements outside the local region. According to the defined scenario, we shall use $N_{move\_in} = 55$ and $N_{move\_out} = 23$. $N_{calls\_in}$ and $N_{calls\_out}$ are defined similarly, and are proportional to the $CM_{ratio}$. Thus, the total registration time and the total data
transmission time are depicted as $T_{total\_reg\_time}$ and $T_{total\_data\_time}$ respectively.

$$
T_{total\_reg\_time} = T_{reg\_our} \times N_{move\_in} + T_{reg\_cross\_out} + T_{reg\_outside} \times N_{move\_out} + T_{reg\_cross\_in} \\
T_{total\_data\_time} = (T_{data\_in\_in} + T_{data\_in\_out}) \times S_{ratio} + (T_{data\_out\_in} + T_{data\_out\_out}) \times (1 - S_{ratio})
$$

Our scheme requires an additional registration overhead to maintain the mobility bindings of the redirection agent and patron hosts. For the basic scheme and the our scheme, the registration times outside the local region are, 341 ms and 647 ms respectively. Also, the patron service takes about 10 sec whenever a mobile host crosses its local region boundary. On the other hand, when a mobile host stays in a local region, the delivery time for a data packet is 232 ms and 221 ms respectively. When a host moves around outside its local region, it takes 523 ms and 397 ms respectively.

Figure 8 shows the total data transmission time for the given scenario. Practically, the data transmission time is the most significant because it reflects the mobile user’s satisfaction. When the local region and the patron concept combine with the basic scheme so that most traffic takes advantage of the direct routing, their effect is significant in proportion to the CM$_{ratio}$. The total communication time is the sum of $T_{total\_reg\_time}$ and $T_{total\_data\_time}$. Figure 9 shows that our scheme has less total communication time than the basic scheme, when the CM$_{ratio}$ is over 12, i.e. when a mobile host gets an average $\geq$ 12 calls per move. So, our scheme is attractive in the situation where there are high calling rates per movement, and there are frequent outward local region movements. Even if their difference is small, this analysis shows that our scheme is most efficient amongst generally proposed scheme, and we expect the effect of our scheme to be better in practice.

## 8 Conclusion

In our scheme, the mobility agents and the mobility routers are required to become active entities to receive and tunnel packets. This does not fit standard networking models which have no active capacity to forward their own packets [24]. However, it is unavoidable if we wish to resolve the host mobility problem whilst preserving backward compatibility. This mismatch seems to limit its benefits to the area which implements the scheme, even if it is possible to communicate between a mobile host and a existing static host. Our work is significant for the following reasons:

- The two concepts of local region and patron host have been developed to utilize the locality properties of hosts’ movements and access histories to provide efficient location and routing.

- By keeping location updates localized into a “home” area, fewer update events are propagated throughout the Internet whilst still achieving optimal routing by using the redirection facility.
• The location update can be efficiently managed by notifying only those hosts that actually want to communicate with the mobile host only when the mobile user crosses its local region boundary.

• Although our scheme requires some caching overhead for the mobility bindings, the cache size of the redirection list is not dependent on the scale of the Internet, but limited to the total number of mobile hosts within its serving area. In the case of the calling and patron list, it is confined to the hosts that actually communicate each other.

• Our scheme offers improved network and host scalability by isolating local movement from the rest of the world and by separating the location and routing roles, and provides a convenient point at which to perform administration and security functions.

References


Figure 7: Comparison of three schemes
Figure 8: Data transmission time
Figure 9: Total communication time