A Mobile Networking System based on Internet Protocol

Charles E. Perkins	Pravin Bhagwat
perk@watson.ibm.com	*pravin@cs.umd.edu*
IBM, T.J. Watson Research Center	*Computer Science Department*
Hawthorne, NY 10562	*University of Maryland*
	College Park, MD 20742

Abstract

Due to advances in wireless communication technology there is a growing demand for providing continuous network access to the users of portable computers, regardless of their location. Existing network protocols cannot meet this requirement since they were designed with the assumption of a static network topology where hosts do not change their location over time. We have developed a model which fits naturally into the usual framework for routing over Internet Protocol (IP) networks. Then, based on IP's Loose Source Route option, we have developed a scheme for implementing agents conforming to this model and providing transparent network access to mobile hosts. Our scheme is easy to implement, requires no changes to the existing set of hosts and routers, and achieves optimal routing in many cases. An outline of the proposed scheme is presented and a reference implementation is described.

1 Introduction

In the last two years, we have witnessed two major changes in computer technology. First, portable computers which are as powerful as some desktop workstations in terms of computing power, memory, display, and disk storage, are beginning to appear. Second, with the availability of wireless adapter cards, users of laptop computers are no longer required to remain confined within the wired LAN premises to get network access. Users of portable computers would like to carry their laptops with them whenever they move from one place to another and yet maintain transparent network access through the wireless link. By transparent network access we mean the ability of a mobile user to setup and maintain network connections despite migration from one network to another. This movement may possibly introduce a momentary pause in the operation, but it should not require reinitialization of network connections. The existing set of network protocols do not meet this requirement since they were designed with the assumption of a stationary network topology where hosts do not change their location over time.

The problem of providing continuous network connectivity to mobile computers has received considerable attention [3, 9, 10, 11], especially in the context of networks based on the TCP/IP [7, 8] suite of protocols. The proposed solutions either require changes to the existing network architecture [10] or introduce new encapsulation protocols [3, 11] to handle this problem. In contrast, our approach, which is based on the use of a natural model and an existing IP option, does not introduce any new protocol and achieves optimal routing. The solution is completely transparent to transport and higher layers and does not require any changes to stationary hosts and routers.

Our model is natural, because we coordinate a collection of mobile hosts as a new IP network. As with any IP network, we route packets to the mobile hosts by using a router. Our router is special because once it receives a packet, it does special things to insure the safe delivery to its destination (the mobile host). However, this special operation is invisible to existing hosts and existing routers, so that all the routing difference due to movement of the hosts can be hidden and effected by mechanisms under the control of our special entities. The other part of our model, which is a very natural part of a physical wireless data communications system, is the transceiver (the access point) which collects wireless packets from a mobile host for delivery to existing hosts along existing wired networks. This transceiver is required for wireless communications, and it provides the reference point by which the location of the mobile host is known.

We have implemented our scheme on a set of IBM PS/2 model 80s running AIX version 1.2. In this paper, we present an overview of our scheme and provide some details of our current implementation. The details provided in this article can serve as a guide for interested readers who would like to add mobile networking features to their network testbeds.

2 The mobility problem

The Internet is a large collection of networks which share the same address space and inter-operate using a common set of protocols, specifically IP, and TCP [7, 8], but including numerous others. It is desirable that the integration of mobile computers within the existing Internet be completely transparent to the transport and higher layers so that users of mobile computers can continue to run existing applications. Any acceptable solution for mobility should inter-operate with the existing infrastructure and not require any modifications to existing host or router software. This goal, however, is not easy to achieve in practice. The way in which the goal may be met depends in large part upon the precise nature of the assumptions made about existing hosts and existing protocol implementations.

An Internet address can be thought of as consisting of two parts, the network identifier and the host identifier. All hosts residing on a (sub)net are required to have the same (sub)net address. Within a (sub)net all attached hosts have a unique host id. The routing infrastructure uses the network part of the address to route the packet to the correct network. Historically, an Internet address served the purpose of a unique host identifier, but the location information was also embedded in it. When a host moves to a new network, it should acquire a new address. Since the transport layer and the network applications assume that network addresses do not change during the lifetime of a connection, the dynamic assignment of new addresses cannot be done without affecting them. To provide *application transparency* it is desirable to devise a method by which hosts retain their home addresses and continue to receive packets despite their migration from one network to another.

Over the last two years several proposals have been made to address this problem [3, 10, 11]. The scheme proposed by Ioannidis[3, 4] relies on a group of cooperating Mobile Support Routers, which advertise reachability to the same (sub)net. Each mobile host, regardless of its location within a campus, is always reachable via one of the Mobile Support Routers (MSR). When a host sends a packet to a mobile host, it first gets delivered to the MSR closest to the source host. This MSR encapsulates the packet and delivers it to the target MSR which strips the encapsulation header and relays the original packet to the mobile host. This approach is optimized to work within a campus environment and requires additional features before it can be extended to support wide area mobility.

In Sony's proposal [10], a mobile host is assigned a new temporary address when it is attached to a new network. The mapping between the home address and the temporary address of a mobile host is kept in an Address Mapping Table (AMT), which is maintained at the routers. Packets transmitted to the home address of the mobile host get intercepted by some router which holds an AMT entry for the mobile host. An address conversion is performed by the router before the packets are forwarded to the physical location of the mobile host. This method requires modifications to routers and host software and has problems inter-operating with the existing hosts unless so-called 'conversion gateways' are used.

Another proposal to support mobile hosts is from Matsushita[11]. This method is also based on the encapsulation approach. A mobile host is assigned a temporary address when it visits a new network. The packets destined to the home address of the mobile host are intercepted by a Packet Forwarding Server(PFS). The PFS encapsulates the packet and forwards it using the temporary address of the target mobile host. The problem with this method is that routing is always sub-optimal unless the software on all stationary hosts is modified.

We have abstracted out particular functions which are necessary to have a mobile networking solution, and built our system to use just those functions. Identifying the minimal set of features allowed us to work towards a solution with few encumbrances stemming from our model. Our model was naturally suggested by the idea of segregating the mobile hosts into their own distinct network. This new network is a logical network, or mobile network, not a network corresponding to a particular extent of wire. Once we decided to use this model, and thus agreed to create a router for the mobile network, we only needed to design the way packets are delivered from the router to the mobile host. This was done naturally enough by designing the mechanism for packets to find their way to the current location of the mobile host as defined by its connection to its current access point. Since any wireless mobile host has to have a transceiver to connect up with, we already had the last necessary functional piece of our model and then set about the task of making the necessary changes to the network protocol implementation in the access points, and the single other agent, which is the router for our new network.

Our approach is based on the use of an existing IP option and therefore, does not require any changes to the existing hosts and routers. The key idea is that each packet originating from a mobile host contains enough routing information that can be used by the remote host to send reply back to the source along an optimal path. In the rest of the paper, we first present an overview of our scheme and then describe our implementation.

3 System

In this section, we will describe the basic entities in our system, and point out why the problem solution becomes more natural when the model includes the appropriate entities, thus making implementation and administration of the overall system easier.

3.1 Model Definitions and Assumptions

Our system involves participation of three types of entities, viz., *Mobile Host (MH)*, *Mobile Access Station (MAS)* and *Mobile Router (MR)*. The networking architecture that we assume is that of a set of MASs connected through a wired backbone. An MAS supports at least one wireless interface and functions as a gateway between the wired and wireless side of the network. Due to the limited range of wireless transceivers, a mobile host can setup a direct link layer connection with an MAS only within a limited geographical region around it. This region is referred to as an MAS's *cell*. The geographical area covered by a cell is a function of the medium used for wireless communication. The range of infrared cells is typically limited to about 20 feet, while that of radio frequency cells could be significantly larger.

Within one campus or administrative domain there could be multiple (sub)networks reserved for mobile hosts. Each (sub)network has a separate router which is referred to as Mobile Router (MR). Unlike other routers, an MR is not required to have an interface corresponding to the wireless (sub)net it serves. If an MR has a wireless interface then it can also function as an MAS. The association between an MH and its current MAS is kept in a Location Directory (LD), which is maintained at the MR.

A mobile host retains its address regardless of which MAS cell it is in. It can start sessions with other hosts (both mobile and stationary) and move into other MAS cells without disrupting any active sessions. The movement of a mobile host is completely transparent to the running applications, except possibly for a momentary pause which may occur while the cell switch takes place. An MH can reside in the cell of only one MAS at any given time. Even if cells of two MASs spatially overlap, an MH routes its outgoing packets through only one of them. An MAS can have multiple MHs in its cell.

We use the term *Correspondent Host (CH)* to refer to the host communicating with an MH. In the following discussion, a stationary correspondent host is also referred to as *Stationary Host (SH)*.

3.2 Implications of our Model

As a result of the way we have framed the problem, solutions occur more naturally within the visible design space. We basically propose that the movement of mobile hosts can be enabled by solving a simply stated routing problem. Namely, we can achieve our goals by finding a way to route packets between the Mobile Router and the Mobile Access Station.

Since the mobile hosts are considered to be on their own network, we can provide for ever larger numbers of mobile hosts by adding more mobile networks. When a mobile router, say for capacity reasons, cannot route any more packets to the mobile hosts on its networks, a new mobile router can be placed into service for new mobile networks. There does not need to be any special relationship between the mobile routers for different mobile networks. Likewise, there does not need to be any relationship between the mobile routers and any Mobile Access Stations. As we will indicate, the Mobile Access Station can deliver packets to any mobile host within its cell regardless of what mobile network that host resides on. In fact, in our design, the mobile access stations are fairly passive devices, existing only to serve whichever mobile hosts come its way, and needing coordination with no other MASs in the system, or with any other entity. Thus, one can hope for an eventual implementation of MASs which costs very little in hardware – perhaps looking like a smoke detector mounted somewhere within a room.

As a result of the lack of interdependence between the model entities, our system is very easy to administer. MASs, MR, and certainly MHs can be added as needed. It is also possible to allow MHs to receive dynamically assigned IP addresses, which would eliminate even the last usual requirement for operation within an IP network. The means by which an IP address is allocated would also have to allow the Mobile Host to know which mobile network it is residing on.

3.3 Control Functions

Aside from the routing functions just mentioned, we have some other simply stated functions that are required. First, a mobile host needs to know when it has entered a cell. We model this process of cell discovery as a client/server interaction, with the MAS advertising its service just by announcing its IP address for use by the MH. The MH accepts service just by picking out one of the servers (MASs) in its area to send packets through.

A location update function is also needed, to allow the MR to know where all of its mobile hosts are. We model this as another simple client/server transaction, this time again with the Mobile Host as client. The client contacts its location directory server (conveniently located at the mobile router) with new updates as necessary – that is, whenever the mobile host decides to accept service from another MAS. Thus, the mobile host is largely in control of its fate. The MASs do not interact with the MH, nor each other, and do not provide any handoff services.

Lastly, we provide a simplistic method by which the MASs know which mobile hosts are in their cells. This could be done by a link-level interaction, but for our purposes it has been sufficient to allow the Mobile Router to provide location updates to whichever MASs are affected whenever a mobile host moves. If this information was not made available, a MAS would mistakenly transmit a packet into its wireless range whenever it received one, even if the desired mobile host had moved out to a new cell some time in the past. We instead prefer for the MAS to forward the packet back to the Mobile Router for further processing.

3.4 Simplest possible operation

Suppose for the moment that we had a working method by which packets could be delivered from a Mobile Router to the current location (MAS) of a Mobile Host. Suppose also that the control functions mentioned above were operational. Then with just these few elements we can already build a working system to allow Mobile Hosts free movement between MAS cells.

The problem is that all packets destined for the Mobile Hosts have to go out of their way, because they need to visit the Mobile Router before they can be delivered to the current MAS. Packets from a mobile host to an existing host (CH) do not experience this problem. Thus, this routing phenomenon is called "triangle routing", with the MR, MAS, and CH labeling the vertices of the triangle. Thus, if we devise a method for delivering packets between the MR and the current MAS, our remaining problem will be to find ways of eliminating this "triangle routing", which can represent a big loss of routing efficiency. It turns out that this is possible to do in a fairly elegant way, as long as data packets can carry along current routing information with them.

4 Overview of Scheme

Our scheme is based on the use of IP's loose source route (LSR) option. The LSR option provides a means for the source host to supply partial routing information to be used by routers in forwarding the datagram to the destination. A source can specify a list of routers which are to be visited in the specified sequence before the datagram is delivered to the final destination. According to the host requirements document [1], return traffic to the originator of the loose source routed packet is also sent with the LSR option by reversing the route taken by the incoming packets. We use this technique to achieve optimal routing between an MH and a CH.

There are four possible communication scenarios depending on whether the CH is stationary or mobile and, if the CH is mobile, whether the MH and the CH are in the same cell or not. We consider each case separately and show how optimal routes are constructed in each scenario.

4.1 Mobile Host to Stationary Host

An MH, while communicating with an SH, issues packets with the LSR option which specifies that packets should be routed via the MAS serving the MH (see arcs 1 and 2 in Figure 1). The SH sends reply packets with the LSR option containing the reversed route. These packets are first delivered to the MAS which forwards them to the MH. Notice that if the LSR option is not used in the reply packets, then these packets would get delivered to the router (MR) for the MH's (sub)network (subsequently called the wireless subnet). The MR would eventually forward these packets to the MH; however, the complete path followed by the reply packets would not be optimal in this case.

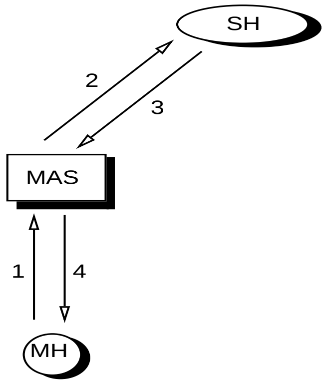

Figure 1: MH to SH

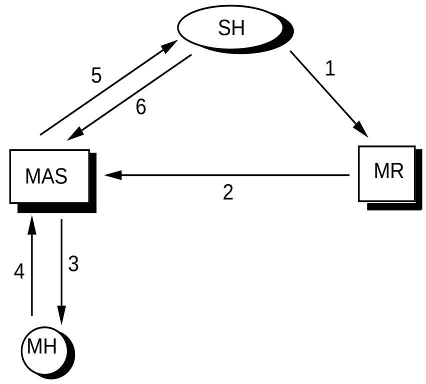

Figure 2: SH to MH

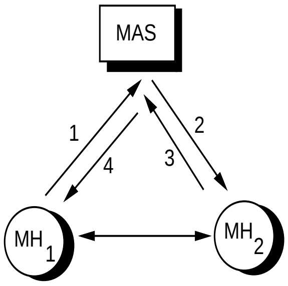

Figure 3: MH to MH (same cell)

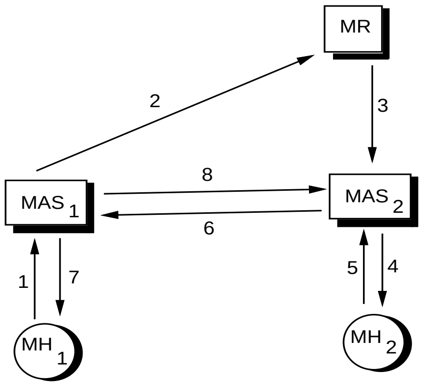

Figure 4: MH to MH (different cell)

4.2 Stationary Host to Mobile Host

An SH need not be aware of the current location of the MH when it initiates a session. If it is not aware, the packet sent from the SH (denoted by arc 1 in Figure 2) arrives at the MR, which advertizes reachability to the wireless subnet. The MR, using the information in its Location Directory, inserts the LSR option in this packet which causes this packet to be delivered to the MH via the current MAS serving the MH (arcs 2 and 3 in Figure 2).

When a reply to this packet is sent, the MH reverses the LSR and sends the packet back to the SH via the MAS (as denoted by arcs 4 and 5 in Figure 2). Once the SH receives a source routed packet, it can send subsequent packets to the MH along the optimal path by reversing the incoming loose source route.

4.3 Mobile Host to Mobile Host within the same cell

An MH does not keep track of other MHs residing in the current MAS's cell. It always uses the current MAS as its default gateway for all outgoing traffic. When an MH initiates a session with another MH, it sends all packets to the MAS just as it would do if it were to send those packets to an SH. Since MAS keeps a list of all MHs residing in its cell, it can forward those packets to the destination MH. If the wireless link layer technology supports direct MH to MH communication then the MAS can also send an ICMP[6] redirect message to the source MH so that it can directly communicate to the destination MH rather than source routing its traffic through the MAS.

4.4 Mobile Host to Mobile Host in different cells

An MH does not inspect the destination IP address to determine whether the destination host is a stationary host or a mobile host. Consequently, it always starts off by sending packets with the LSR option. By normal routing mechanisms, these packets are forwarded to the MR associated with the destination MH (see arcs 1 and 2 in Figure 4). The MR extends the existing LSR option by inserting the address of the the MAS presently serving the destination MH, and then forwards the packet. Normal routing procedure ensures that these packets get delivered to the MAS serving the destination MH, followed by the destination MH (as shown by arcs 3 and 4 in Figure 4). Notice that the LSR option list of the incoming packet contains addresses of two MASs – one serving the source MH, and one serving the destination MH. The reply packets are sent by reversing the incoming loose source route, which follow the optimal path as denote by arcs 5, 6 and 7 in Figure 4. Once the source MH receives a packet back from the destination MH, it can also send the subsequent packets along the optimal path.

5 Implementation

We have implemented the aforementioned scheme on a set of IBM PS/2 model 80s running AIX version 1.2. Each of these machines is equipped with an infrared (IR) wireless adapter card supporting up to 1Mb/s data transfer rate. The range of an IR transceiver is limited to about 20 feet. This adapter card uses an Ethernet chip. From the perspective of the device driver, a wireless interface behaves much like an Ethernet interface, since both use a carrier sense multiple access (CSMA) protocol. The only difference is that the wireless adapter card does not support collision detection (CD). This shortcoming, however, did not affect us much because most of our experiments were limited to a small cell population.

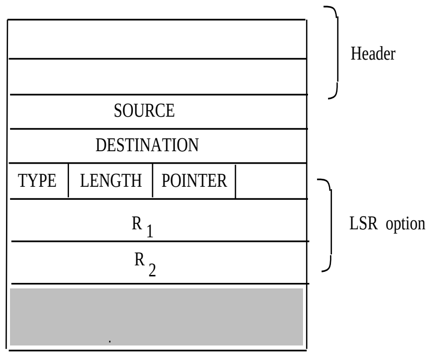

Figure 5: An IP header with LSR option

The basic idea of IP's Loose Source Routing (LSR) option is to enable any data packet to include routing information, so that a particular packet would follow a routing path possibly different than the path taken by normal data packets. This is done by including in IP's option data fields the necessary addresses of the desired intermediate routers, along with some ancillary fields to manage the consumption of the routing data (see Figure 5). These fields indicate, first, the number of intermediate routing nodes, and, second, the next desired intermediate router in the list. When one of the intermediate nodes is reached, the next intermediate routing point is taken out of the list and placed in the destination field of the IP header; then, the intermediate router copies the IP address of its own outgoing interface into the IP options data, and finally increments the pointer past its own address. This causes the final destination to have a natural reverse path through the intermediate routing agents.

There are actually two different kinds of IP source routing options. We use Loose Source Routing, because we only want to include the relevant MAS addresses in our source routes. The other variety of IP source routing option is called Strict Source Route. When using this option, every intermediate routing node must be included in the IP option data. These two varieties of source routing options are distinguished by the use of different IP option numbers.

Our existing implementation consists of approximately 800 lines of kernel code and 1500 lines of user code. It can be thought of as consisting of two parts, viz., the *packet routing part* and the *location information management part*. Actions related to packet routing are performed in the kernel. To avoid creating new data structures, location information is stored implicitly in the kernel routing table. This approach has some obvious advantages. First, minimal kernel modifications are needed to route packets to/from MHs. Secondly, with a little modification, the existing *route* command can be used to manipulate the location information. In the following, we first describe how packets are routed among various components and how location information is managed. Later, a description of the processing required at each component is included.

5.1 Packet Routing

For each MH, which has an address on the wireless (sub)net served by an MR, a host route is maintained by the MR. The current location information of the MH, i.e., the address of the MAS serving the MH is kept in the gateway field of the routing table entry. This routing table entry is distinguished from other entries by the presence of a new flag called `RTF_MOBILE`. Since the MR advertizes reachability to the range of addresses on the mobile (sub)net, an IP packet destined to an MH is first routed to the MR for further delivery. At the MR, one of the host routes with `RTF_MOBILE` flag is chosen to route this packet. The MR knows how to interpret `RTF_MOBILE` flag; that flag specifies that the MR should insert the LSR option in this packet (by using the MAS[1] as the intermediate hop) before forwarding it. Due to the inserted LSR option, this packet is delivered to the MAS currently serving the destination MH. The LSR option is processed here and, finally, the packet is delivered to the MH.

An MAS performs the forwarding function between the wired and the wireless interface. It keeps a host route corresponding to each MH residing in its cell. Thus, an IP packet which is destined to one of the MHs in its cell is delivered to the MH by the MAS.

Processing of packets originating from MHs is relatively simpler. An MH always keeps a default route to the MAS currently serving it. This routing table entry also has the `RTF_MOBILE` flag set, meaning that the LSR option should be used on all outgoing packets. Packets originating from an MH are source routed via the current MAS and are delivered to the final destination by the normal routing mechanism. The reply packets use the reverse route and are delivered back to the MH by the optimal route.

5.2 Location Information Propagation

We now describe how the location information, which is implicitly maintained by routing table entries at the MR, MAS and MH, is updated when an MH switches cells.

[1] The IP address of the MAS currently serving the MH is available from the gateway field of the routing table entry.

A server program (*beacon_snd*) running on the MAS periodically broadcasts beacons on the wireless interface. These beacon packets contain the IP address of the MAS. An MH, upon entering the MAS's cell, receives these beacons and sets up the MAS as its default router. At the same time the MH also sends an *mh2mr* message to its MR. This message contains the new location information of the MH and a timestamp. A server (*mr_serv*) running at the MR receives this message, uses the location information contained in it to update its routing table, and notifies the previous MAS serving the MH about its migration by sending it a *delete_host_route* message. The previous MAS, upon receiving this message, deletes the host route corresponding to the migrated MH.

All messages are exchanged using UDP packets. The message exchange protocol is very simple and completely implemented by user level processes running at MH, MR and MAS.

5.3 Processing at Mobile Host

The software running on a mobile host performs primarily two functions: cell discovery and insertion of LSR option in all outgoing IP packets.

The purpose of the cell discovery operation is to send a notification and supply the address of the MAS to the network layer whenever an MH performs a cell switch. It is desirable that the cell discovery feature be supported by the link layer. However, due to lack of such support, a user level process (*beac_rcv*) performs this operation at an MH. It continuously monitors beacon packets that are broadcast by MASs, and changes the default route at the MH whenever a cell switch occurs. If the cells of two or more MASs spatially overlap, the MH may potentially receive beacons from multiple MASs. The cell discovery mechanism should avoid rapid cell switching by the MH in this scenario. In our implementation, if beacons from the current MAS continue to arrive at regular intervals, then beacons from other MASs are ignored. The MH switches to another MAS cell, i.e., changes its default route, if it does not receive any beacon from the current MAS for 3 seconds. After an MH has determined the cell to which it belongs, it is required to notify its MR about its current location. This is accomplished by sending an *mh2mr* message to the MR.

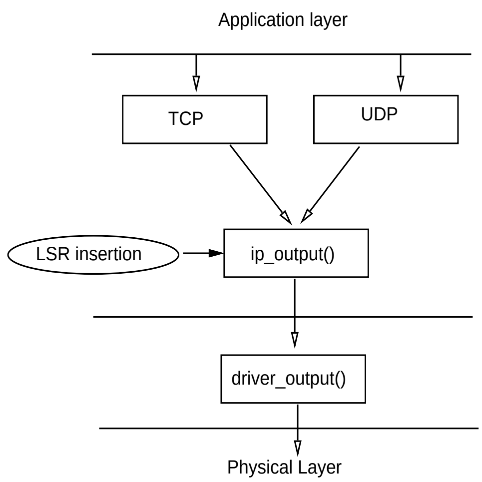

Figure 6: Kernel processing at Mobile Host

Unlike cell discovery which is performed by a user level process, LSR option insertion is performed in the MH kernel (see Figure 6). The transport layer (TCP, UDP) running at an MH passes packets to the IP layer for further delivery. The IP layer at the MH is required to modify this packet before passing it on to the network interface. If the outgoing

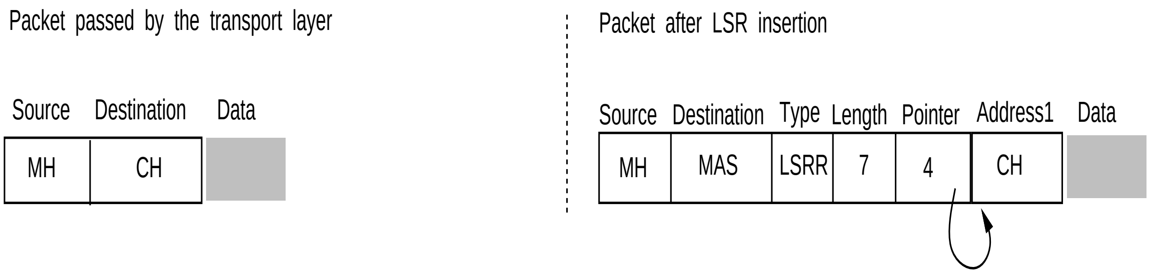

Figure 7: LSR insertion by an MH

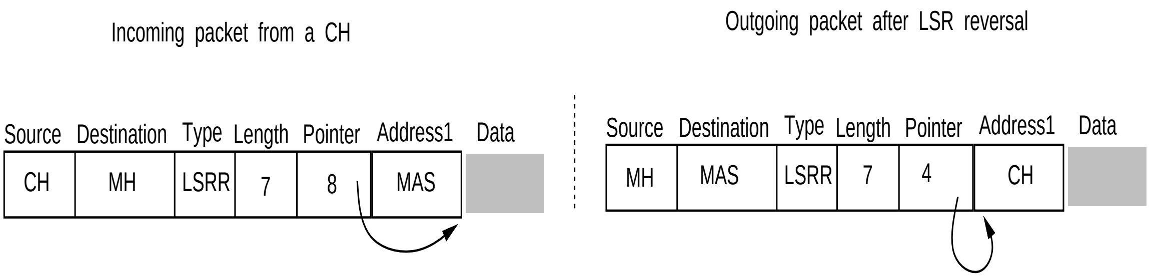

Figure 8: LSR reversal by an MH

packet does not contain LSR, then conceptually the MH must insert the LSR option and specify the MAS as the only intermediate router. In some cases, the packet passed by the transport layer already contains the LSR option. This option is formed by reversing the route contained in the LSR option of an incoming packet. An existing LSR option in an outgoing packet has already been extended by the MR to include the local MAS as a required intermediate router for the source route. This model of operation conforms to the description given in Comer[2]. However, since in most current implementations all routing is done on the basis of the destination address, the actual operation is somewhat different (see Figures 7 and 8). The only routine which needs modification is `ip_output()`. The changes are described below:

```
if ( pkt does not have LSR option) {
    insert LSR option in pkt;
    first address in option list = pkt.destination;
    pkt.destination = MAS address;
} else      /* option already exists by TCP route reversal */
    if ( pkt.destination != MAS address )
        pkt.destination = MAS address;
```

Suppose a packet from a CH arrives at the MH and thus has an LSR option. Suppose also that before a reply is sent, the MH switches to another MAS cell. The destination address contained in the reply packet will be that of the previous MAS. Since the MH has already left the previous MAS cell, it is necessary to modify the destination address of this packet to the address of the current MAS. After this packet reaches the CH, the subsequent packets originating from the CH will follow the reverse route and arrive at the MH via the current MAS. Thus, no interruption in the active transport layer session will be observed.

5.4 Processing at Mobile Access Station

The primary function of an MAS is to provide an access point through which an MH, regardless of its address, can attach itself to the network. To forward packets to MHs, an MAS should keep a host route corresponding to each MH that resides in its cell. The host route is installed by the MAS kernel as soon as the first packet (e.g, ARP packet) is received from the MH. Beyond this point, the existing `ip_forward()` code is able to forward packets to the MH via the wireless interface. When an MH moves out of the MAS cell, the corresponding host route at the MAS should be deleted, otherwise the MAS would continue to transmit packets to the MH which no longer exists in its cell. As mentioned earlier, a server process (*mas_serv*) running on the MAS performs the deletion of the host route when it receives a *delete_host_route* message from the MR.

After an MH moves out of the MAS cell, the packets addressed to it may still arrive at the MAS. These packets could be directly forwarded to the MAS currently serving the migrated MH. This, however, would require the MAS to maintain forwarding pointers for all MHs even after they migrate out of its cell, thereby, unnecessarily increasing the protocol complexity and memory overhead. We wanted to design a system where there was no need for any information exchange among MASs. Therefore, packets which cannot be delivered by the MAS are returned to the MR associated with the destination MH. Note that the MAS need not know the address of the MR for this purpose. The MAS only forwards this packet on to the wired side of the network. The normal routing mechanism automatically delivers this packet to the MR, since it is the router to the MH's home subnet. It is necessary to mark these returned packets from MAS so that MR can distinguish them from other packets. Since there is no extra space available in the packet where this marking can be incorporated, we do it implicitly by setting the last address in LSR equal to the packet destination address.

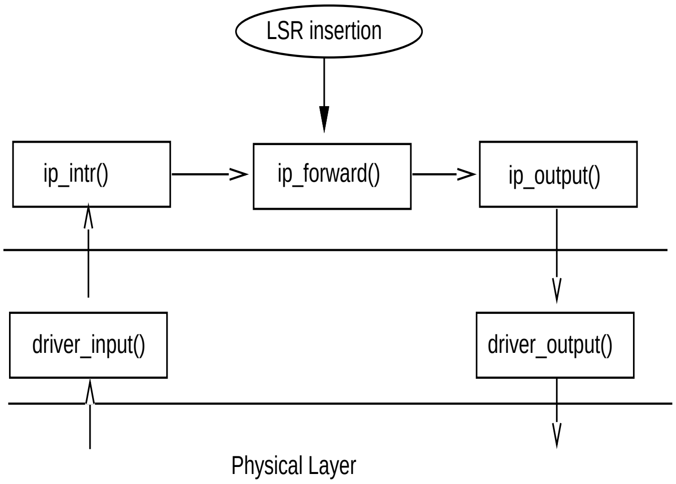

Figure 9: Kernel processing at Mobile Router

5.5 Processing at Mobile Router

An MR performs the following two functions:

1. Maintains location information of all MHs on its (sub)net.

2. Forwards packets to MHs (via an MAS) by inserting/updating LSR option.

Location information is nothing but the necessary routing information which is maintained at an MR. Therefore, the most convenient place for storing it is the kernel routing table. Corresponding to each MH, a host route is maintained at the MR. The gateway field of the route table entry stores the address of the MAS which servers the MH. Each time a MH switches to another cell, it sends a location update message (*mh2mr*) to the MR. A user level process (*mr_serv*) receives this message and updates the routing table by executing a *route* command.

Actions related to LSR option insertion are performed in the kernel. The only routine that needs modification is `ip_forward()`. Packets arriving at the MR, which are destined to some MH, can be classified into four categories depending on the contents of their LSR option. For each case, Figure 10 shows the format of incoming and outgoing packets. The steps taken by the MR are outlined below.

1. Packets originating from Stationary Host : The packet originating from a stationary host, and destined to some MH, usually not contain any LSR option until it reaches the MR associated with the destination MH. The MR modifies this packet (as shown in case 1 of Figure 10) by inserting an LSR option in it. Notice that the original packet destination address is moved in LSR option list and the new packet destination is set equal to the address of the MAS presently serving the MH. This modification causes this packet to be forwarded to the MAS, where the LSR option is processed, and finally the packet is delivered to the MH.

2. Packets originating from Mobile Host : An MR can also receive packets which already contain LSR option. This can happen, for example, when an MH (say MH1) starts up a session with another MH (say MH2). Suppose, in addition, that MH1 and MH2 reside in the cells of MAS1 and MAS2, respectively. When the packet originated from MH1 arrives at the MR, it appears as shown in case 2 of Figure 10. The destination address in the packet header contains the IP address of MH2. The LSR option list contains the address of MAS1 with the pointer pointing beyond the end of the address list.

The MR takes the destination address and appends it to the end of the option list and places the IP address of MAS2 into the destination field of the packet header. Normal forwarding mechanism first delivers the packet to MAS2 and the packet is finally delivered to MH2.

3. Packets returned from Mobile Access Station : A packet arriving at the MR could also be from an MAS which failed to deliver this packet to the target MH. As in the previous case, this packet also contains the LSR option with one address and pointer pointing beyond the end of the address list (see case 3 in Figure 10). However, unlike the previous case, the MR should forward this packet to the current MAS serving the MH without extending the LSR option list. To do that, the MR should be able to distinguish this packet from the packets mentioned in case 2. The MR compares the destination address of this packet against the address contained in the LSR option list. If it is a packet returned from an MAS, then these two addresses are same. In this case, the MR sets the destination address field in the packet to the address of the current MAS serving the MH, resets the pointer in the LSR option to the beginning of the address list, and forwards the packet.

4. Incorrectly processed packets from Stationary Host : When an MH starts a conversation with an SH, the packets arriving at the SH contain LSR option. If the SH correctly processes LSR option, the reply packets will automatically follow the reverse path without any involvement of the MR. Unfortunately, several IP implementations, such as in

4.3 BSD and SUN OS 4.1, do not correctly process LSR option. Even though the incoming source route is reversed and included in the reply packet, the destination address is set equal to the original source and pointer is made to point to the end of the option list. This has the effect of sending the packet straight back to the source, rather than sending it along the reverse path. Since the destination of this packet is an MH, the normal routing mechanism delivers this packet to the MR associated with the MH. A sample of such a packet is shown in case 4 of Figure 10.

The MR checks the last address in the LSR option list. If this address is of the MAS associated with the destination MH, then the packet is coming from an SH which does not correctly process LSR options. The MR swaps the addresses stored in the destination address field and LSR option field, resets the pointer to the beginning of the address list, and forwards the packet to the network interface output routine.

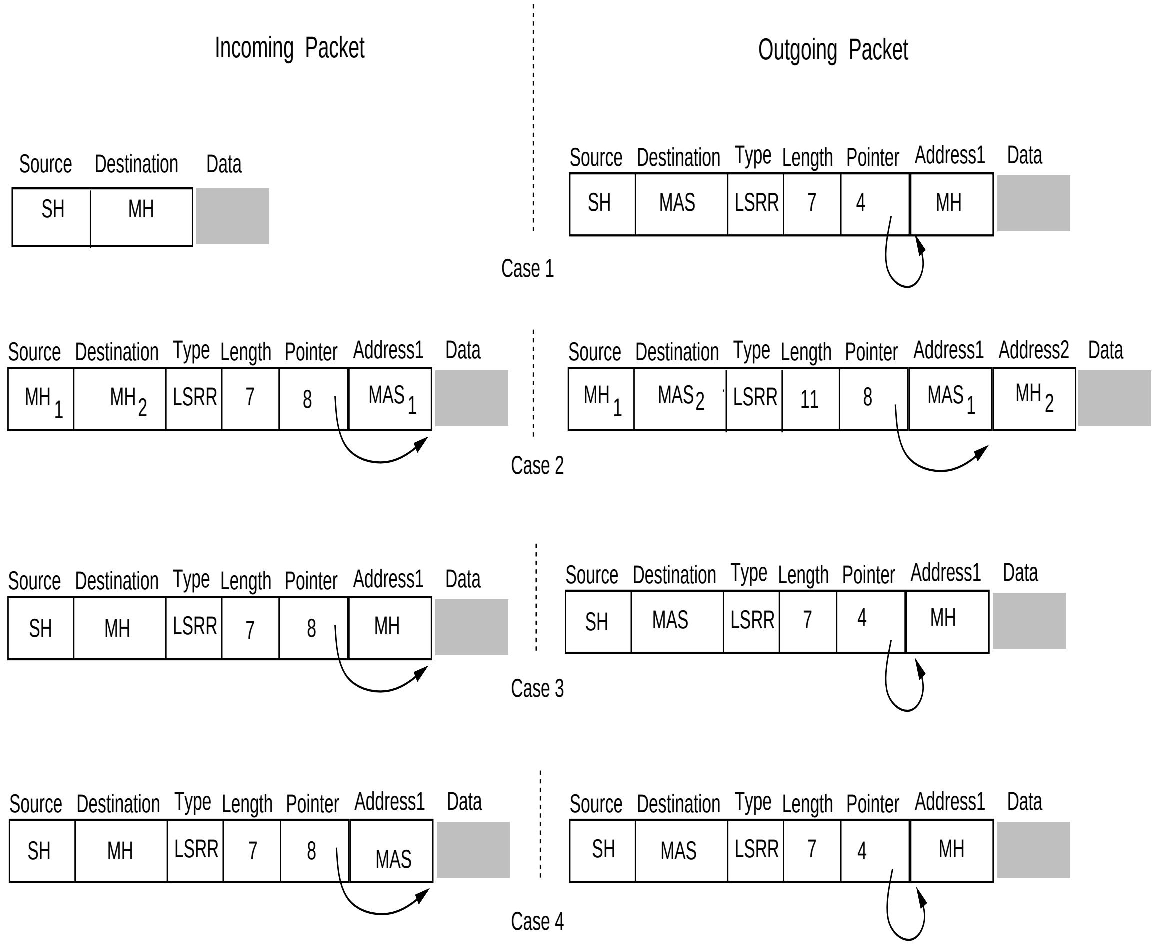

Figure 10: Packet processing at MR

5.6 Overhead

Processing and memory requirements of our scheme are very moderate. Overhead of storing LSR option is 8 bytes per packet (and 12 bytes in case of MH to MH communication, when they reside in different cells). Each host route requires 128 bytes of memory as it is stored in an *mbuf*[2]. At an MAS, the number of host routes is equal to the cell population. If we

[2]mbuf is a data structure that is used for dynamic memory allocation in Unix

assume 20 as an upper bound on the cell population, only 2K bytes would be required to keep the routing information. At an MR, there will be as many host routes as the number of mobiles which are assigned addresses on the wireless subnet. Suppose an MR serves 256 mobile hosts, then it will require about 32K bytes to store the location information. Given that we only require one MR per wireless subnet, this is a very reasonable figure. It is worth mentioning that it is not necessary to keep the location information in the kernel routing table. We did it only to simplify our implementation effort. A further reduction in data memory requirement can be achieved if a separate table is used to keep the location information, although more processing would be required.

6 Future Work

We currently have in operation the ability to let the MAS do the LSR insertion instead of the MH. With this capability, one could cause the MAS to understand more about which correspondent hosts do correct LSR processing, and limit the LSR insertion to those hosts which correctly process it.

Currently, our location updates are propagated by a special "protocol" called *mh2mr*. This protocol could be easily replaced by a facility whereby the MH pings the MR with the record-route option. Then when the MR receives the ping, it would notice that the sender was a MH within its mobile subnet, and then the MR could infer that the first stop along the way is the MAS currently serving the MH. Besides eliminating unnecessary protocol, this method has the advantage that the MR has a measure of the time delay associated with the current location of the MH. This information could trigger the instantiation of another MR closer to the MH, which would reduce the delay associated with sub-optimal routes arising from imperfect LSR implementations. In order to protect against accidental location updates with this scheme, we would need to include some special information within the data field of the ICMP packet.

Some measures will have to be taken to avoid the sub-optimal routing currently to be expected from all UDP applications, especially because of the prevalence of NFS. We intend to perform this in conjunction with the previously mentioned capability for providing LSR insertion at the MAS.

7 Concluding Remarks

Alternative solutions for supporting mobility are based on the encapsulation approach. This approach relies on the presence of a packet forwarding agent which can perform a packet encapsulation/decapsulation function. A packet whose destination address is a mobile host is intercepted by a packet forwarding agent. Intercepted packets are encapsulated and forwarded using the address of another agent that is located close to the mobile host. The destination agent strips the encapsulation header and relays the original packet to the mobile host.

The major problem with this approach is that routing is always sub-optimal, unless the packet forwarding agent is located close to the source. Another problem is the fragmentation and reassembly required at the packet forwarding agent. It has been observed that the majority of packets in the network are either very short or very large in size. Packet fragmentation may occur at the packet forwarding agent if the extra bytes added for encapsulation cause a packet to exceed its maximum transmission size. This may also happen at an MR when it inserts the LSR option. However, the risk of fragmentation is higher in an encapsulation based approach because the size of an encapsulated packet is larger than a packet with LSR option.

The LSR approach also has shortcomings. The first problem is that UDP[5] agents do not perform route reversal. As a result, UDP packets originating from a stationary host are always routed sub-optimally via a mobile router. LSR route reversal is actually specified to occur in layers above IP, so that it is improper for IP to do the route reversal itself. This makes sense for the cases when source routing is to be selected for policy reasons by a particular application. However, UDP is specifically specified to be a very simple protocol, and thus it is considered inappropriate for UDP itself to perform any route reversal. The ultimate responsibility for LSR option route reversal in UDP applications would reside with the application itself, and there are few if any UDP applications which pay attention to this protocol requirement. Notice that if there is no encapsulating agent close to the stationary host, any scheme based on encapsulation approach will also suffer from this problem. One approach to get around this problem is to distribute the MR functionality among multiple cooperating MR agents and place them close to the UDP sources. This also improves the robustness of the system.

Another problem with the LSR option is that there is a potential security risk associated with its use. The authentication mechanisms which are based on the source address lookup can easily be broken with the use of the LSR option. For this reason, many routers disable this. Note that encapsulation has exactly same problem unless each MSR [3] knows every other MSR. Although this is a genuine problem, we should not overlook the benefits of source routing only on the basis of this limitation. IP itself has a lot of security holes. Besides, mobile networks, due to wireless links, are quite prone to security attacks. Better methods for data and address encryption are required to make them tolerant against impersonation attempts. In future, some of those techniques can be used to make the LSR usage more secure.

Another concern that has been raised is related to the performance aspect of LSR. Compared to normal IP packets, more processing time is required to process packets which contain options. Existing routers manufactured by router vendors have specialized code to rapidly forward IP packets that do not contain any IP options. It is, however, not difficult to optimize this, since a similar performance enhancement could be made for packets containing precisely the IP LSR option with option data of the expected length(s). In the normal case, routers would not have to do any further processing, but would only have to forward the IP packet unchanged.

The following is a quick summary of the comparison of the two approaches.

Criteria	LSR	Encapsulation
Existing Implementation	Usually Poor	Nonexistent
Impersonation	Easy spoof	Easy spoof
Router Efficiency	Not optimized	Optimized
Additional Fragmentation	Less	More
ICMP error propagation	Correct	Incorrect
Fragmentation/Reassembly	End-to-End	Done by encapsulating agent

In conclusion, we have shown that LSR provides a fast and elegant solution to the problem of providing seamless networking for mobile hosts using the Internet Protocol. The use of LSR also enables optimal routing in most cases, without incurring the expense of agents interceding for the destination hosts, as long as those hosts implement IP as it has been specified now for a decade. The additional protocol requirements are minimal, and with a few changes to our system could be eliminated almost entirely.

Besides using LSR, our approach demonstrates a modular construction with good separation between layer 2 (MAC/link layer) and layer 3 (IP) functions. The beaconing, cell discovery, and cell switch mechanisms are all quite distinct from location update and routing mechanisms. Either feature could be replaced without substantially affecting the other. Indeed, new cell discovery mechanisms are likely to become commercially available at protocol layer 2, and our system will still work in almost exactly the same way.

Our construction also eliminates any addressing relationship between the MAS and the MH. The MH needs to discover the wired IP address of the MAS, but that IP address need have no relationship to its own IP address. The MH does not ever have to know anything about the MAS's wireless IP address – if indeed the MAS even has a wireless IP address. The MAS is quite passive in our system, and there is no requirement for intercommunication between any two MASs. This will be very helpful to those intending to design secure wireless systems, insofar as MASs become more like dumb transceivers and do not have to be trusted agents.

Lastly, our system has shown itself to be quite practical. We have had test systems running in our laboratory for almost a year with the current implementation (previous systems ran mostly as a simulation). The mobile hosts are quite usable, and all new kernels and other programs have been installed by wireless file transfers of various sorts. The application transparency has been just about perfect; X11, NFS, and all the normal networking software has required absolutely no change. The only change above the IP network layer has been to eliminate the route-caching feature built into TCP for performance reasons. We have not noticed any significant slowdown in operation of the wireless network except for bulk data transfers, and that is due to the relatively slow speed of our wireless adapter hardware.

8 Acknowledgements

We wish to gratefully acknowledge Yakov Rekhter for his contribution to the design, implementation, and debugging of the system described in this paper.

References

[1] R. Braden. Requirements for Internet Hosts – Communication Layers. RFC 1122, Oct. 1989.

[2] Douglas E. Comer. *Internetworking with TCP/IP.* Prentice Hall, 1991.

[3] John Ioannidis, Dan Duchamp, and Gerald Q. Maguire Jr. IP-based Protocols for Mobile Internetworking. In *Proceedings of ACM SIGCOMM*, pages 235–245, 1991.

[4] John Ioannidis and Gerald Q. Maguire Jr. The Design and Implementation of a Mobile Internetworking Architecture. In *Proceedings of Winter USENIX*, pages 491–502, San Diego, CA, Jan 1993.

[5] J. Postel. User Datagram Protocol. RFC 768, Aug 1980.

[6] J. Postel. Internet Control Message Protocol. RFC 792, Sep 1981.

[7] J. Postel. Internet Protocol. RFC 791, Sep 1981.

[8] J. Postel. Transmission Control Protocol. RFC 793, Sep 1981.

[9] Fumio Teraoka and Mario Tokoro. Host Migration Transparency in IP Networks. *Computer Communication Review*, pages 45–65, Jan 1993.

[10] Fumio Teraoka, Yasuhiko Yokote, and Mario Tokoro. A Network Architecture Providing Host Migration Transparency. In *Proceeding of ACM SIGCOMM*, Sept 1991.

[11] Hiromi Wada, Takashi Yozawa, Tatsuya Ohnishi, and Yasunori Tanaka. Mobile Computing Environment Based on Internet Packet Forwarding. In *proceeding of Winter USENIX*, pages 503–517, San Diego, CA, Jan 1993.