

MOBILE-IP

—

THEORY AND IMPLEMENTATION

Richard Parry, C.C.P., P.E.

email address:
rparry@qualcomm.com

Home Web Page:
<http://people.qualcomm.com/rparry>

wireless packet address:
W9IF @ K6JCC.#SOCAL.CA.USA.NA

ABSTRACT

The explosion in demand for mobile computing, both wire and wireless based systems, places demands on the Internet Protocol (IP) that it was not originally intend to support. The protocol assumed that the point at which a computer is connected to the Internet is fixed. If a host is moved to a new network, the current routing protocol will be unable to route the datagrams to the correct destination. Mobile-IP is the new standard intended to address the mobility issue. The current standard developed by the Internet Engineering Task Force (IETF) is described in RFC 2002. This paper describes the Mobile-IP protocol in theory and an actual implementation. The theory portion of the paper describes the protocol in detail and reviews the present literature and work in the area. The implementation portion of the paper describes hands-on experience with the installation of a Mobile-IP system. The software is currently available for the Linux operating system and is freely released under the GNU Public License (GPL) agreement. The software was successfully demonstrated at ACM's First International Conference on Mobile Computing (MobiCom '95) held at Berkeley, CA and Nomadic '96 held in San Jose, CA.

March, 1997

Introduction

There is little doubt that the networking world's ultimate goal is *anywhere, anytime computing* with continuous connectivity. Mobility is becoming a requirement rather than an amenity. With the advent of the cellular phone, people have tasted the flexibility that roaming provides and the freedom to be untethered from the confines of a fixed network. The demand for laptop computers with connectivity to the Internet is yet another example. However, the ability to provide mobile connectivity has lagged far behind the demand. The authors of the IP protocol in the late 60s could not have imagined where their protocol, originally developed to allow a small number of scientists, engineers, and host computers to communicate, would lead. Certainly there was never a requirement for mobility or for automatic reconfiguration of the network when hosts were moved. In fact, IP implicitly assumes that a host is fixed and the IP address registered to the host is a direct function of the network that the host is connected to. This provides the mechanism for datagrams to be properly routed to the host using the IP address.

The present process of modifying IP addresses is cumbersome and tedious, even for a computer savvy user. The process requires both know-how and coordination with a system administrator. Ideally, we would like to be able to roam from network to network seamlessly. The Mobile-IP working group of the Internet Engineering Task Force (IETF) has developed a standard to do just that. The standard, now called Mobile-IP, is an enhancement to standard IP and is described in detail in RFC 2002. There are actually two mobility standards, one to support the IPv4 standard and another for the future Internet IPng (a.k.a. IPv6) standard. In this paper we will discuss the unique characteristics of Mobile-IP and an implementation developed for the Linux operating system. As we shall see, there are many problems to address in addition to just allowing a mobile host to move seamlessly from network to network. There are issues of security, authentication, routing, delays, and more. For example, what do you do when the mobile is in transit, it is not connected to either network. Mobile-IP when implemented for a wireless network is particularly unique since it is possible for the Mobile Host to be connected to two or more networks simultaneously.

This paper is broken into four sections, the first is the introduction. The second section is a description of standard TCP/IP theory for fixed networks. A reader with a firm grasp of TCP/IP essentials may skip this section, but the information provided is important for an understanding of Mobile-IP. The third section builds on fixed network theory and discusses mobility supporting networks and specifically the theory of Mobile-IP. The fourth section discusses one of two publicly available implementations of Mobile-IP. The paper ends with the results of the implementation and a conclusion. An appendix is provided which provides additional supporting and reference information.

Theory - Fixed Networks

TCP, UDP, and IP

Although the ISO model consists of seven layers, TCP/IP can be shown by the four layers in Figure 1. This layered approach provides a means to manage complexity, since functionality is divided among the layers. The Internet layer (IP layer), may interface to one of two Transport Layers, the Transport Control Protocol (TCP) and/or the User Datagram Protocol (UDP). TCP is a reliable protocol whereas IP and UDP are unreliable protocols. The term unreliable means that the protocol does not support methods for verifying that the data received is error free. There is no error detection or error recovery mechanism for these protocols and they therefore rely on upper layers to provide that functionality.

One might ask why UDP is used, since it has less functionality than TCP. The need for UDP lies in its ability to provide the *application program* direct access to the datagram. This direct access has the advantage of reducing overhead which translates to reduced packet sizes and therefore increased transmissions speeds.

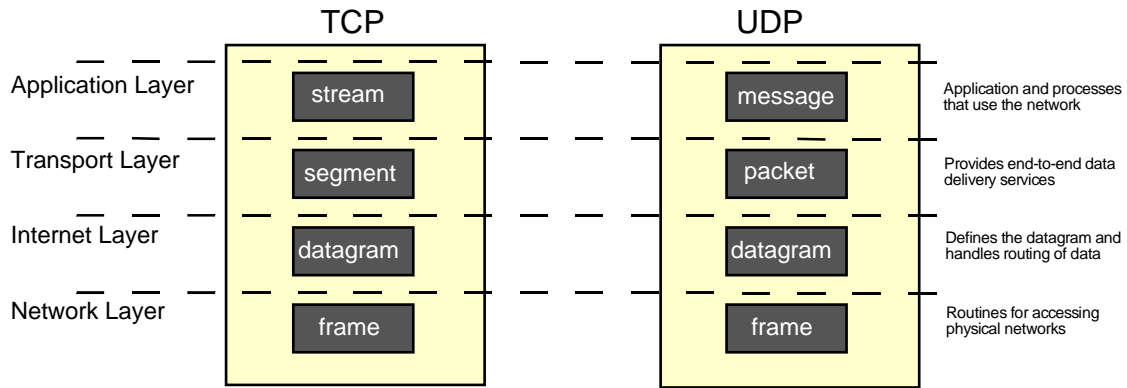


Figure 1 TCP/IP Model

The information that flows from layer to layer uses different terminology. For example, applications using TCP refer to the data as a *stream*, while UDP refers to the data as a *message*. When the information arrives at the transport layer, it is called a *segment* in TCP and a *packet* in UDP. The internet layer refers to the information as a *datagram* under both TCP and UDP terminology and the network layer calls the data *frames* for both protocols.

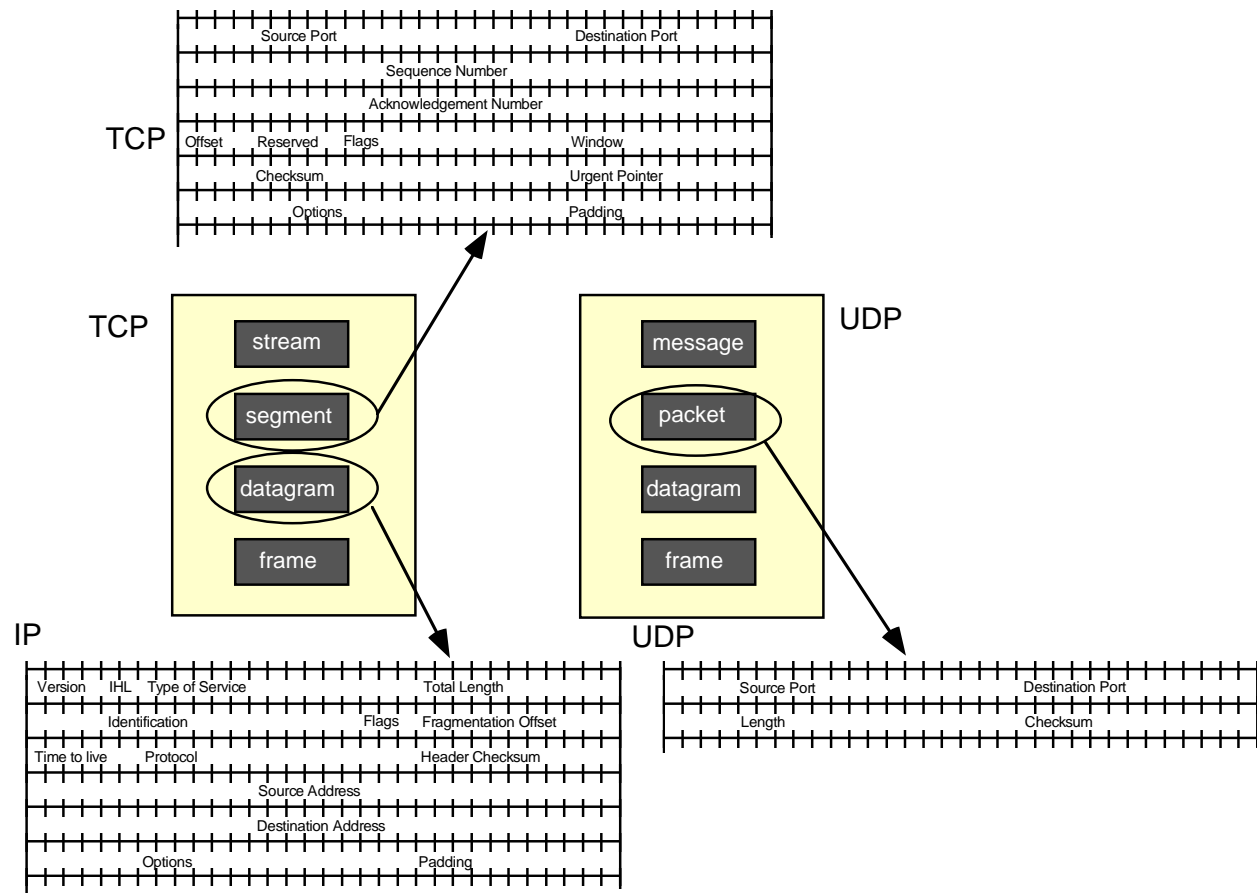


Figure 2 Protocol Headers

Routing Table

The ability of IP to move data from a host on one network to a host on a different network is referred to as data delivery. TCP/IP uses three methods for communication. *Addressing* occurs between hosts, *routing* occurs between networks, and *multiplexing* occurs between the layers.

IP moves datagrams from a 32 bit source address to a 32 bit destination address contained in word 4 and 5 of the IP header. This 32 bit address consists of a *network part* and a *host part*. The portion of the address that represents that network varies as a function of the IP *class*. (A, B, and C). This method of breaking an IP address into network classes allows easy distribution of IP addresses to organizations. For example, an organization given a Class B address (65,000 addresses) places less of a burden on a router than if many Class C addresses were provided since the router need keep only one route for that entire organization.

Routing is a network oriented operation, which makes decisions based on the network portion of the IP address. This operation can be fast since the router need only check the high order bits of the IP address to determine the address class (A, B, or C) which is then used to identify the network. If the address is not local, the route passes the data to another network for forwarding. If the address is on the local network, the local subnet mask is applied to the destination address by each of the local hosts.

It is important to note that the routing function is not limited to routers. All devices on a network must make routing decision, including hosts. For hosts, the decision is easy. If the destination host is on the local network, that destination host accepts the data as it own, otherwise it is ignored.

Below is an example of a routing table generated using the UNIX **netstat -nr** command.

```
[rparry@doc rparry]$ netstat -nr
Routing Table:
  Destination          Gateway                Flags  Ref    Use  Interface
-----
127.0.0.1              127.0.0.1             UH          0 01851562 lo0
129.46.0.0             129.46.156.96        U           3   6513 hme0
224.0.0.0             129.46.156.96        U           3     0 hme0
default               129.46.156.1         UG          0     852
```

A datagram transmitted by a host that contains a destination address not on the local network, is ignored by the local hosts. Actually, all local hosts receive the datagram and check it against their own routing table and decide the datagram is not intended for them. Only the router on the network understands that the datagram needs to be forwarded and accepts the datagram and passes it on to another network where it eventually arrives at the destination.

Conversely, a datagram passed to a local network is accepted by all hosts, but only one host will recognize it as its own. Again, the *IP module* (layer) in each of the local hosts does a lookup using the routing table (also called *forwarding table*) to determine if the information is addressed to them. If the datagram is intended for a host, it is accepted and passed up the TCP layer and eventually to the application layer. This forwarding or movement up the layers is referred as multiplexing and requires that the datagram contain sufficient information to assure it is passed to the correct application. The information required to perform this task, is embedded in the *protocol* and *port* number.

Protocol and Port Numbers

It is important to understand that routing also occurs within a host. This stems from the need to be able to support multiple users or a single user running multiple processes. For example, some data may be intended for one application (e.g., telnet) on a host and other data for another application (e.g., ftp). In fact, it can become a little more complex, since it is possible for one user to have many identical applications such as many telnet sessions. To assure that the data is provided to the correct telnet session on a host, additional information is needed in the IP header. This additional information takes the form of an 8 bit *protocol number*.

These protocol numbers have been standardized for many of the protocols in *Assigned Numbers* RFC 1700. Below is an example of a typical UNIX `/etc/protocols` file which contains the official protocol names. The first field contains the protocol name, the number of the protocol is in the second field, followed by an alias and comment in the third and fourth fields respectively. The appendix includes a more complete list of assigned names.

```
[rparry@doc rparry]$ more /etc/protocols
ip      0      IP      # internet protocol, pseudo protocol number
icmp    1      ICMP    # internet control message protocol
igmp    2      IGMP    # internet group multicast protocol
ggp     3      GGP     # gateway-gateway protocol
tcp     6      TCP     # transmission control protocol
pup     12     PUP     # PARC universal packet protocol
udp     17     UDP     # user datagram protocol
raw     255    RAW     # RAW IP interface
```

The protocol number provides us with a mechanism to correctly route the datagram from the IP layer to one of the transport layers above it (tcp, udp, etc.). In the example shown in Figure 3, the datagram is directed to the TCP transport layer, protocol number 6.

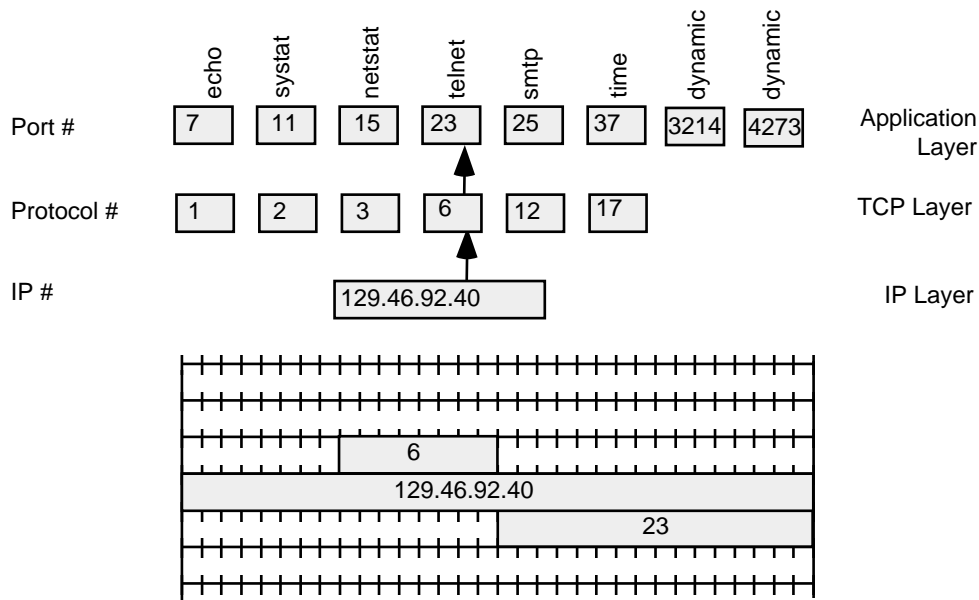


Figure 3 Protocol and Port Addressing

The task of forwarding is not complete until the data is passed up to the application layer. To route the data to the correct application, a port number is used. To simplify the connection process, *well-known* port assignments are used. Both systems agree in advance to use these standard ports. These *well-known* port assignments are fixed (static) and allow a computer to connect to a particular network *service*. A short list of services that one might find on a UNIX workstation is shown in the `/etc/services` file below.

```
[rparry@doc rparry]$ more /etc/services
tcpmux  1/tcp      # rfc-1078
echo    7/tcp
echo    7/udp
discard 9/tcp      sink null
discard 9/udp      sink null
systat  11/tcp     users
daytime 13/tcp
```

```

daytime      13/udp
netstat     15/tcp
gotd        17/tcp      quote
chargen     19/tcp      ttytst source
chargen     19/udp      ttytst source
ftp-data    20/tcp
ftp         21/tcp
telnet      23/tcp
smtp        25/tcp      mail
time        37/tcp      timserver
time        37/udp      timserver

```

We have previously mentioned that we must allow for multiple users, in other words, allow for multiple sessions of an application (e.g., many telnet sessions). In the case of several telnet sessions only one can use the well-known port number 23, the other sessions are given a unique temporary port number dynamically. The port number is a 16 bit value, which allows port number assignments to 65,535. Let's look at an example.

The first telnet session is given a source and destination port assignment of 23. If another telnet session needs to be started, it will be uniquely identified with a dynamically allocated port number and a well-known port number. Below is an example showing a host (sleepy.action.com) that has 6 telnet sessions. The first entry in the table shows the dynamically chosen source port number is 1160 and the destination (foreign) address as 23 (the number is displayed as the word *telnet*).

```

[rparry@sleepy rparry]$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp    0      0 sleepy.action.com:1160  wizard.qualcomm.:telnet ESTABLISHED
tcp    0      0 sleepy.action.com:1157  abura.qualcomm.c:telnet ESTABLISHED
tcp    0      0 sleepy.action.co:telnet sneezy.action.com:2049  ESTABLISHED
tcp    0      0 sleepy.action.co:telnet sneezy.action.com:2050  ESTABLISHED
tcp    0    124 sleepy.action.co:telnet sneezy.action.com:2052  ESTABLISHED
tcp    0      0 sleepy.action.co:telnet doc.action.com:1028     ESTABLISHED

```

The combination of a source and destination port is referred to as a *socket*. In fact, the term port number and socket are often used synonymously. In a similar manner, well-known port numbers are often referred to as well-known sockets. The term socket may also refer to the complete IP and port number. In the example, sleepy.action.com:1160 is the source socket, and wizard.qualcomm.com:telnet (telnet = 23) is the destination socket.

In this section we have followed data as it entered the network, was routed to a particular LAN, a specific host, and then up the layers to a unique application. With a firm understanding in fixed TCP/IP networks, we are now ready to discuss mobility networks and Mobile-IP in particular.

Mobile Network Theory

Design Requirements

IP was never intended for mobility. Nevertheless, we are faced with the need to provide this functionality without changing the existing protocol. This is not unlike the requirement for the telephone system to remain compatible with legacy pulse dialing phones. If we had the luxury of starting over again, hindsight would provide a wealth of wisdom in implementing mobility. If we could start with a "clean sheet of paper" we would have great design flexibility. However, given the installed base, our overriding design requirement is to remain compatible with standard IP. In addition, wireless communication is no longer a luxury, but a design requirement. Therefore the Mobile-IP design must be able to function in the harsh wireless world where high error rates are the norm and bandwidth is limited. Taking mobility a step further, the design must accommodate a mobile host in a heterogeneous

environment where a mobile can move between wireless LANs, as well as, between a wire and wireless based LAN.

The wireless medium leads to unique configurations. For example, consider that a mobile host may be simultaneously connected to more than one network, a situation that would never occur in a wired LAN. However, it happens everyday in a cellular phone network where a user moves between coverage areas that overlap. In the overlapping area, data is received by more than one base station. The cellular network must be able to adapt to this environment and so must the design for Mobile-IP.

Mobile communication often implies battery powered devices, therefore the design should include a protocol with as little overhead as possible to help reduce transmissions and therefore conserve battery power.

In summary, the design requirements for Mobile-IP are:

- Function in existing IP network
- Be able to support both wire and wireless LANs
- Support heterogeneous LANs (wire and wireless)
- Function in high error rate environments
- Limit bandwidth use
- Limit data size to conserve battery energy

A Mobile Supporting Network

To allow a Mobile Host (MH) to move between networks while keeping its fixed IP address, two support components are required, a Home Agent (HA) and a Foreign Agent (FA). A HA is a host on the network where the MH normally resides. In practice, it is quite possible that the MH is never physically connected to the Home Network (HN). A FA resides on each remote network where the MH wishes to visit. A foreign agent located on a remote network acts as coordinator when the mobile host is visiting. In Figure 4 the MH can move freely between any of the foreign networks that provide a FA.

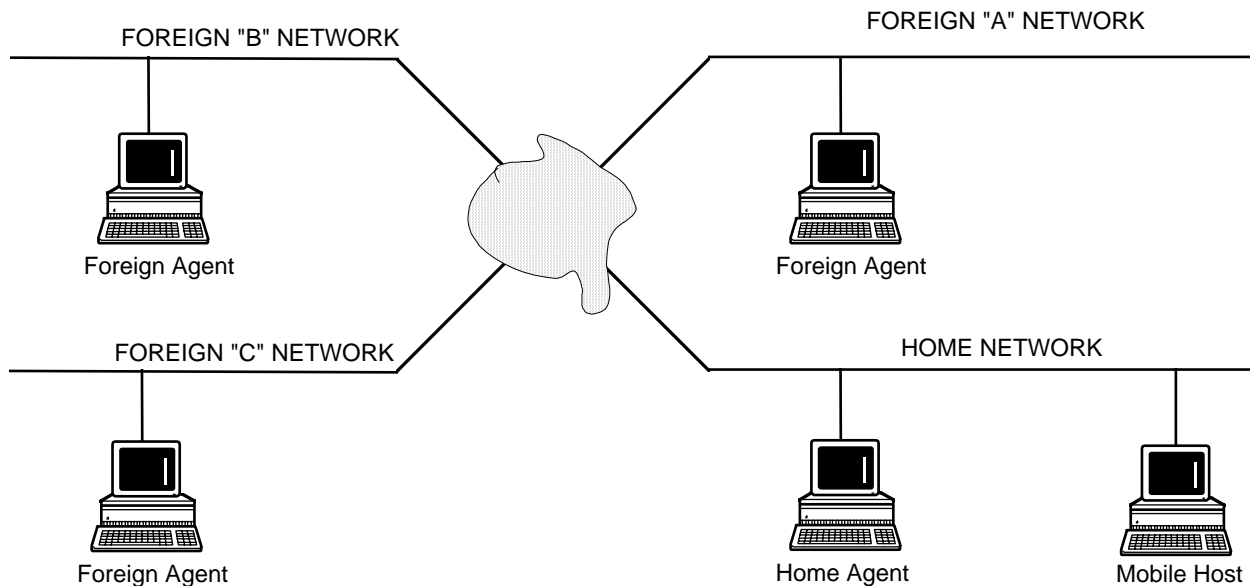


Figure 4 A Mobility Supporting Network

It is interesting to note that it is possible for a FA to exist on a network other than the FN where the MH is temporarily located. In a similar manner, it is not a requirement that the HA reside on the HN. As long as there is a link, the FA and HA may be located almost anywhere. Using *proxy* or *gratuitous ARP* mechanisms it is possible that the agents not be on their respective networks.

It is also interesting to note that Mobile-IP allows a MH to move to a FN **without** the support of a FA. This allows the mobile host greater freedom since it implies the mobile host can move to any network. When a MH is on a foreign network without the aid of a FA, it must perform the functions of the FA. We will discuss in a subsequent section the advantages and disadvantage of this mode of operation.

How it Works - An Overview

Before discussing Mobile-IP in detail, the following is provided as a quick overview. The scenario explains the simple case where the HA and FA are located on the HN and FN respectively.

- Mobile Agents (MA) may advertise their presence by sending *Agent Advertisement* messages
- A MH may solicit MAs by sending *Agent Solicitation* messages
- A MH uses the MAs advertisements to determine if it is on the HN or a FN
- When the MH is on the HN, it acts *independent* of the HA
- When a MH returns from a FN, it must de-register with the HA through *Registration Request* and *Registration Reply* messages
- When a MH finds it has moved to a new FN, it obtains a care-of-address from the FA or from other means such as DHCP
- When a MH on the FN obtains its care-of address, it registers the new care-of address with the HA using a *Registration Request* and *Registration Reply*.
- Datagrams sent to the HN are intercepted by the HA and encapsulated in a new datagram which contains the care-of address and sent to the FA or to the MH if it is acting without the aid of a FA
- Datagrams sent by the MH on the FN need not return to the HA, they may be sent directly to the original source host

Discovery, Advertisement, and Solicitation

Agent discovery is the method by which a MH ascertains whether it is connected to the foreign or host network. The discovery process relies on the ICMP (Internet Control Message Protocol). It also requires cooperation between the mobile agents and the mobile host. A Mobile Agent may advertise that it is available to provide mobility services or the MH may send a message to learn the availability of possible mobile agents, referred to as *Agent Solicitation*.. If the MH is connected to a FN, it is provided with a care-of address during the discovery process.

When the MH is away from the home network, it is the responsibility of the HA to accept IP packets normally intended for the MH and forwards them to the remote network. This requires the MH to keep the HA informed of its present location. This is accomplished when the MH first attaches to the remote network, it uses a special registration protocol to update the HA of its location. This is accomplished when the MH first attaches to the remote network, it asks the FA to act in its behalf. Through the FA, the MH sends a registration message back to the HA indicating its present location. The process is reminiscent of a child traveling to a friend's house and then calling home to inform the parent (e.g., HA) that he or she has arrived. In this way, the parent (HA) knows where the child (MH) can be contacted.

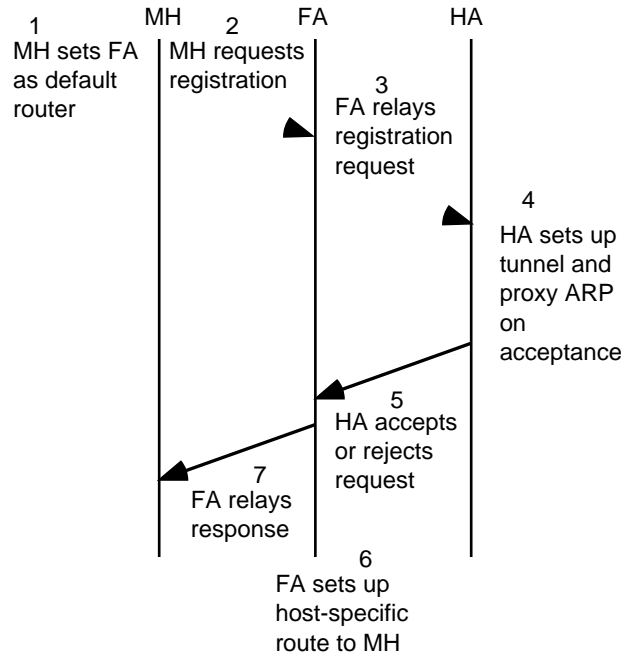


Figure 5 Registration

Multicasting

The observant reader will ask, how can an agent advertise its availability as a mobility agent efficiently? It is true that the mobility agents have a list of mobile hosts that it will support, however, this list could be long and to broadcast individual datagrams to all possible mobile hosts would be inefficient. To solve this problem IP supports a mechanism referred to as *multicasting*.

A brief digression is in order to help clarify multicasting and other similar terms. Let us begin with *broadcasting* which is a scheme to send from one to all hosts. This is the technique used by a radio broadcasting stations. At the other extreme is *unicast*, the traditional method of sending data from one host to one other host. Between these two extremes, is *multicasting*, a scheme where one host sends data to many other hosts. There are actually many ways of implementing multicasting, each with its own advantages and disadvantages. For example, under one multicasting scheme the sender has no knowledge or control of the number of hosts that are receiving the message.

Multicasting meets the needs of Mobile-IP perfectly, since it allows a mobility agent to multicast (advertise) its availability to the local network where it can support mobility services. In addition, the multicasting scheme can limit the broadcast to the subnet. This is exactly what we wish since we do not want to advertise availability off the subnet that we can provide service to.

Multicasting, as currently implemented in standard IPv4, falls into the Class E address range, 224.0.0.0 to 239.255.255.255. Messages sent in this range can be received by all machines with routing tables properly configured. However, for Mobile-IP we are interested in sending to only hosts on the local subnet. The range for this functionality is 224.0.0.0 to 224.0.0.255 and is reserved for maintenance protocols discussed in RFC 1060. The appendix includes a list of some multicast addresses generated by the UNIX "host -l mcast.net" command.

When the implementation of Mobile-IP is discussed later in this paper, the routing table and how it is configured will show clearly how multicasting is used to allow the mobility agents to send advertisements. The Mobile-IP RFC-2002 describes the specification for multicasting as follows:

As specified for ICMP Router Discovery, the IP destination address of an Agent Advertisement MUST be either the "all systems on this link" multicast address (224.0.0.1) or the "limited broadcast" address (255.255.255.255). The subnet-directed broadcast address of the form <prefix>.<-1> cannot be used since mobile nodes will not generally know the prefix of the foreign network

Tunneling (IP within IP)

When the registration process is complete, the HA has the care-of address of the MH. It is then prepared to intercept and forward packets intended for the MH on the home network, to the MH on the foreign network. The forwarding of packets is performed through a process known as *tunneling*. Tunneling is performed by *encapsulating* the original datagram in a new datagram. In this manner, routers that would normally direct the datagram to the address of the "inner" IP address, see only the "outer" IP address and correctly pass it to the foreign agent where the MH now resides. When the FA receives the packet intended for the MH, it *de-encapsulates* the datagram and passes the original datagram to the foreign network where only the MH receives it. The MH is oblivious to the tunneling process. It is the mobility agents that perform all the work and hide the true path that the datagram has taken.

From a routing standpoint, it should be clear that this process is not optimal. For example, all packets intended for the MH are first sent to the HA which results in delays. However, this is a small price to pay for mobility in most applications. In applications that require near real time information such as voice, the additional delay may make this service unusable. Note that the delay is in one direction only. The delay is for datagrams sent to the MH only, datagrams from the MH are not sent back to the HA, but are sent directly to their destination.

In the case where the MH is in transit and not connected to any network, the HA performs the necessary task of storing packets for forwarding until the MH connects to a foreign network or returns home. Note that a *care-of address* is the end of the tunnel, it may or may not be the FA.

Mobility without a FA

When a FA is available, it gives the HA its own IP address to use as the care-of address for the MH¹. This mode conserves IP addresses and places no limitations on the network. In fact, there can be any number of MHs on a FN since they all have a single care-of IP address, which is the foreign agent care-of address. This is the preferred mode of operation since it conserves IP addresses and requires no IP administration. In this mode the HA tunnels datagrams to the FA by encapsulating the true source IP address within an IP header that contains the IP address of the FA. When the datagram reaches the FA, it is de-encapsulated and sent to the MH.

A second mode supported by Mobile-IP provides the MH with a *co-located care-of address*.. One mechanism for dynamically dispensing the temporary *IP co-located care-of address* is DHCP. Under this mode of operation, there is no longer a need for the FA. This provides the MH with greater flexibility since the MH is not limited to networks that have a FA. The disadvantage is that each MH must have a unique IP address which ultimately limits the number of MHs on the network. In addition, de-encapsulation must be performed by the MH in the absence of the FA.

¹Don't confuse the care-of address with the IP address of the Mobile Host. The Mobile Host's IP address never changes, only the care-of address is changed to indicate to the Home Agent where the Mobile Host can be located.

Mobile Network Implementation

To gain a better understanding of Mobile-IP, the author investigated and installed a publicly available version of Mobile-IP.

Hardware

For the implementation, the home agent (doc.action.com) was a Pentium 90MHz CPU, with 64MB RAM, 1GB hard disk, and Ethernet card. The mobility host (sleepy.action.com) was a Sun Sparc 4C IPX, 32MB RAM, 720 MB hard drive, with built-in Ethernet support. Normally a laptop would be used but since none was available, the Sun Sparc had to meet this need. The foreign agent (daffy.qualcomm.com) used in the experiment was a Pentium 50 MHz CPU, with 8MB RAM, 250MB hard drive, and Ethernet card. Figure 6 shows the configuration of the networks.

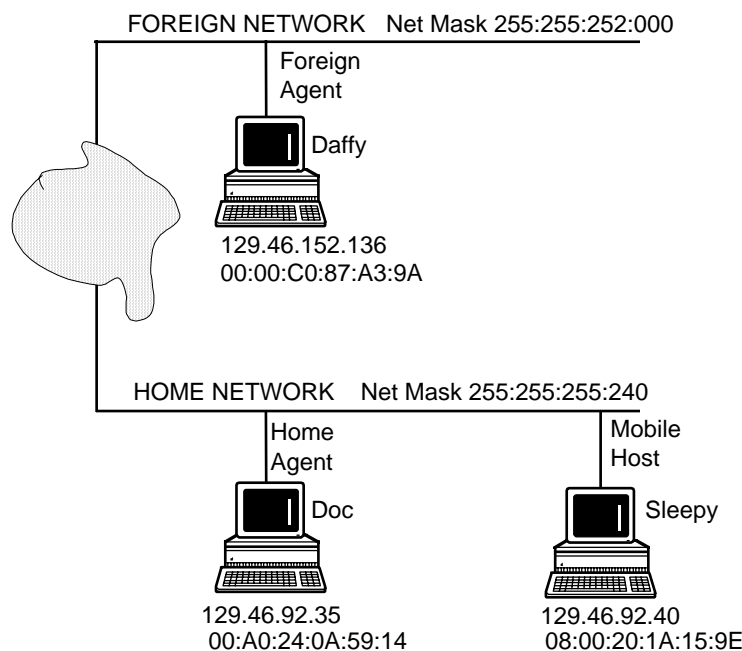


Figure 6 Implementation Configuration

The home network was literally the home network (author's home) and the foreign network was the author's corporate Intranet. The home network was connected to the Internet via an Ascend P50 Basic Rate Interface (BRI) ISDN router.

Software

There are currently only two publicly available implementations of Mobile-IP². This implementation used the Linux operating system version of Mobile-IP³ for all testing. The key requirement for an operating system to support mobility is: **IP-in-IP tunneling**, and **multicasting**. Each

²As of late 1996.

³Available at <http://anchor.cs.binghamton.edu/~mobileip>

of the Linux kernels used supported this functionality. The home agent and mobile host ran under version 2.0.27 of the Linux kernel and the foreign host under 1.3.68.

Linux MobileIP Implementation

Mobility brings flexibility and freedom, but it also brings with it additional concerns for security and authentication. For this reason, a mechanism has been built into the implementation to allow the HA to ascertain that it is communication with the MH and no other MH. If authentication were not provided, it would be fairly easy for a pirate MH from a foreign network to masquerade as the true MH. All that the pirate MH need do is to use register with the HA and provide the HA with the MH IP address. Therefore, the HA needs a way to assure it is communicating with the authorized MH. This implementation of Mobile-IP provides authentication by providing a shared “key” that is known only to the HA and the MH. Below is an example of a 16 byte key.⁴

```
259704 md5 16 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36
```

The key begins with a Security Parameter Index (SPI) number followed by the encryption algorithm, which in our case is md5 (message digest 5). Following the md5 field is the key length shared between the two machines, in this case 16 bytes. The remainder of the record is the 16 byte authentication key. This short file is called mip-ha.dat in the implementation and is stored in /etc/mip-ha.dat on the MH. The HA has a file called mip-mh.ok located in /etc/mip-mh.ok and contains the IP address and key for each MH that the HA is to support. The example below shows that the HA will support a single (1) MH whose fixed IP address is 126.46.92.40. The remainder of the record is the authentication key for that MH which we have already discussed. A HA can support any number of MHs by merely adding similar records for each additional MH.

```
1
126.46.92.40 259704 md5 16 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36
```

The FA needs to know the HA and the MH. That information is included in the file /etc/mip-vmh.ok. For the implementation, the information used by the FA is shown here.

```
1
129.46.92.40 129.46.92.35
```

This implementation of Mobile-IP requires that all parties, MH, HA, and FA be started with a short shell script. The script (mhscrip) to start the mobile host is shown below. An examination shows that the script adds a new route to the routing table and then starts the MH daemon.

```
IPADDR="129.46.92.40"      # IP address of MH "sleepy"
NETMASK="255.255.255.240" # Netmask of home network
HOMEAGENT="129.46.92.35" # IP address of HA "doc"
INTERFACE="eth0"         # Interface name of MH
LOGFILE="mh.log"         # Log file
#Join multicast group
/sbin/route add -net 224.0.0.0 netmask 240.0.0.0 dev ${INTERFACE};
# Start up the mh daemon
mh -a ${IPADDR} -m ${NETMASK} -g ${HOMEAGENT} 2> ${LOGFILE}
```

To start the HA daemon, the *agentscript.ha* script shown below is executed. It also sets up a new route in the HA routing table and starts the *agent* daemon.

```
IPADDR="129.46.92.35"      # Home Agent IP
NETMASK="255.255.255.240" # Home network netmask
HADDR="00:A0:24:0A:59:14" # Home Agent hardware address
INTERFACE="eth0"         # Home Agent interface name
LOGFILE="agent.ha.log"    # Home Agent log file
# The following line forces link level broadcasts (sent on 255.255.255.255)
# to go out on a specific interface.
```

⁴This implementation supports keys up to 63 bytes in length.

```
/sbin/route add -net 224.0.0.0 netmask 240.0.0.0 dev ${INTERFACE};
# Start up the agent daemon
agent -a ${IPADDR} -m ${NETMASK} -h ${HADDR} -i ${INTERFACE} 2> ${LOGFILE}
```

Lastly, the FA daemon is started by executing the shell script *agentscript.fa* shown below. Like the MH and HA, a new route is added and a daemon agent is started.

```
IPADDR="129.46.152.136" # Foreign Agent IP
NETMASK="255.255.252.0" # Foreign network netmask
HADDR="00:00:C0:87:A3:9A" # Foreign Agent hardware address
INTERFACE="eth0" # Foreign Agent interface name
LOGFILE="agent.fa.log" # Foreign Agent log file
# The following line forces link level broadcasts (sent on 255.255.255.255)
# to go out on a specific interface.
/sbin/route add -net 224.0.0.0 netmask 240.0.0.0 dev ${INTERFACE};
# Start up the agent daemon
agent -a ${IPADDR} -m ${NETMASK} -h ${HADDR} -i ${INTERFACE} 2> ${LOGFILE}
```

After each script is run, the routing table changes as shown⁵. Note that a new network has been added which will be used for multicasting.

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
129.46.92.32	*	255.255.255.240	U	0	0	8	eth0
127.0.0.0	*	255.0.0.0	U	0	0	1	lo
224.0.0.0	*	240.0.0.0	U	0	0	0	eth0
default	rparry-router.q	0.0.0.0	UG	0	0	4	eth0

Figure 7 shows the files that are major components to the operation of the mobility system. Each component has three types of files: a parameter file, which provides the information about the network, a short shell script to start the daemon process, and a log file used for administrative and/or debugging purposes.

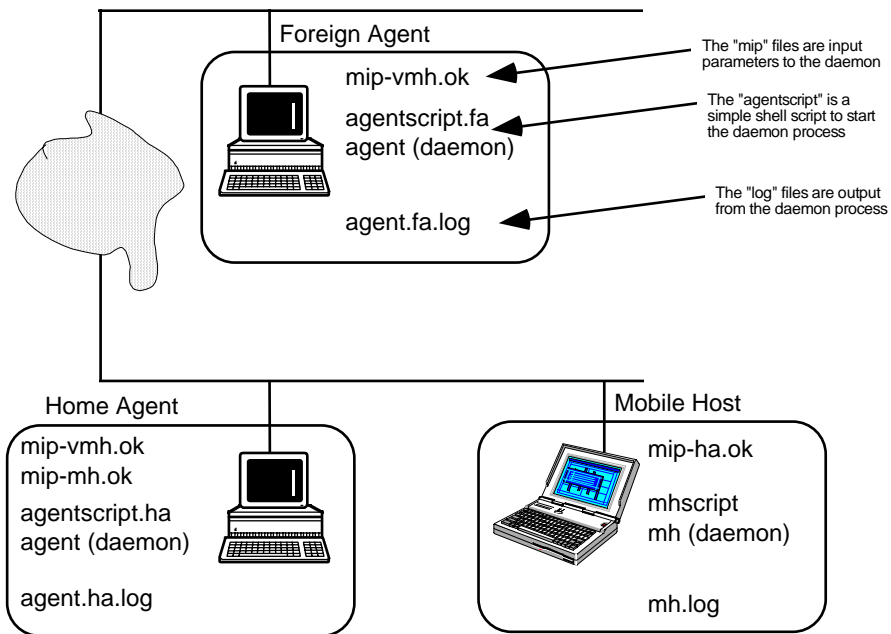


Figure 7 Program File Structure

⁵The routing tables for the HA, MH, and FA look virtually the same and therefore only one is shown. The difference between the tables is the IP address of the router. For the HA and MH it is 129.46.92.32 as shown, for the FA it is 129.46.152.0

Startup

Home Agent

When the HA is started it performs many tasks, a log file is generated which contains sequence of events. An example of the HA log file is shown below. In the interest of brevity a line by line description will be omitted, however, the right justified bold text emphasizes the key actions and milestones.

```
IPaddr: 812e5c23, Mask: ffffffff0HWaddr: 0:a0:24:a:59:14
Initialized sockets ...
IFACEreq 3 (DOWN=3,MKPT2PT=4,REDIR=5), tnum 0, haaddr 0, coaddr 0.
IFACEreq 3 (DOWN=3,MKPT2PT=4,REDIR=5), tnum 1, haaddr 0, coaddr 0.
ARPreq 0 (REM0=0,REM1=1,PROXY=2), addr 0, hwaddr 0:a0:24:a:59:14
lowARPreq called with zero ipaddr. Ignored.
ARPreq 0 (REM0=0,REM1=1,PROXY=2), addr 0, hwaddr 0:a0:24:a:59:14
lowARPreq called with zero ipaddr. Ignored.
ARPreq 0 (REM0=0,REM1=1,PROXY=2), addr 0, hwaddr 0:a0:24:a:59:14
lowARPreq called with zero ipaddr. Ignored.
ARPreq 0 (REM0=0,REM1=1,PROXY=2), addr 0, hwaddr 0:a0:24:a:59:14
lowARPreq called with zero ipaddr. Ignored.
ARPreq 0 (REM0=0,REM1=1,PROXY=2), addr 0, hwaddr 0:a0:24:a:59:14
lowARPreq called with zero ipaddr. Ignored.
ARPreq 0 (REM0=0,REM1=1,PROXY=2), addr 0, hwaddr 0:a0:24:a:59:14
lowARPreq called with zero ipaddr. Ignored.
ARPreq 0 (REM0=0,REM1=1,PROXY=2), addr 0, hwaddr 0:a0:24:a:59:14
lowARPreq called with zero ipaddr. Ignored.
ARPreq 0 (REM0=0,REM1=1,PROXY=2), addr 0, hwaddr 0:a0:24:a:59:14
lowARPreq called with zero ipaddr. Ignored.
ARPreq 0 (REM0=0,REM1=1,PROXY=2), addr 0, hwaddr 0:a0:24:a:59:14
lowARPreq called with zero ipaddr. Ignored.
ARPreq 0 (REM0=0,REM1=1,PROXY=2), addr 0, hwaddr 0:a0:24:a:59:14
lowARPreq called with zero ipaddr. Ignored.
ARPreq 0 (REM0=0,REM1=1,PROXY=2), addr 0, hwaddr 0:a0:24:a:59:14
lowARPreq called with zero ipaddr. Ignored.
Done cleaning up ...
Wed Feb 12 18:53:37 1997
```

Read config file for MHs that I am HA for.

Reading configuration file...

```
addr:812e5c28 SPI:259704 authtype:md5 keylength:16 key: 31 32 33 34 35 36 37 38 39 30
31 32 33 34 35 36
Wed Feb 12 18:53:37 1997
```

Read config file for MHs that I am FA for.

Reading configuration file for vmh...

Wed Feb 12 18:53:37 1997

Reading recovery log...

```
status:0 time:0 tunnel:0 coaddr:0 id:[135600cb:d926157a]
```

Start sending advertisements at 2 second intervals

```
Sending a URhere message to 224.0.0.1
Data on WhereAmIsid.
Sending a URhere message to 224.0.0.1
Data on WhereAmIsid.
Sending a URhere message to 224.0.0.1
Data on WhereAmIsid.
```

A Mobile Host wants to know where it is!

```
-- WHEREAMI from 129.46.92.40
Sending a URhere message to 129.46.92.40
Data on WhereAmIsid.
Data on WhereAmIsid.
```

```
-- WHEREAMI from 129.46.92.40
Sending a URhere message to 129.46.92.40
Data on RegisterMesid.
<<<<<< Entering processRegisterMe
```

```
=====
Wed Feb 12 18:56:13 1997
```

DEREGISTER THE MOBILE HOST

```
-- DEREGISTERME from 129.46.92.40 Port 3332
[ 0:323e7c17] Type 1 Flags 0 Lifetime 0
Homeaddr: 812e5c28, Homeagent: 812e5c23, Careof: 812e5c28
Extension: 32 Length: 20
```

```
-----
 1 | 0 | 0 | 0 | 81 | 2e | 5c | 28 | 81 | 2e | 5c | 23 | 81 | 2e | 5c | 28 | 0 | 0 | 0 | 0 | 17 | 7c
| 3e | 32 | 20 | 14 | 0 | 3 | f6 | 78 | df | 1 | 95 | 6f | f1 | e9 | 73 | 87 | 95 | df | 72 | e | f7
| 6e | ba | e3 |
-----
```

```
=====
Acting as home agent...
```

```
=====
Wed Feb 12 18:56:13 1997
```

```
-- REPLY to 129.46.92.40 at port 3332 (Rejection)
[2c629878:323e7c17] Type 3 Code 133 Lifetime 0
Homeaddr: 812e5c28, Homeagent: 812e5c23
Extension: 32 Length: 20
```

```
-----
 3 | 85 | 0 | 0 | 81 | 2e | 5c | 28 | 81 | 2e | 5c | 23 | 78 | 98 | 62 | 2c | 17 | 7c | 3e | 32 | 20 | 14
| 0 | 3 | f6 | 78 | ae | 9d | 75 | 34 | 4f | 92 | 46 | a5 | b1 | ac | 19 | ac | b7 | d0 | 3e | 53 |
-----
```

```
=====
```

REFUSAL SENT (see results section for explanation)

```
Refusal sent.
In main finished processing RegisterMe.
Sending a URhere message to 224.0.0.1
Data on WhereAmIsid.
Sending a URhere message to 224.0.0.1
Data on WhereAmIsid.
```

Mobile Host

When the MH is started, it also creates a log file as shown below.

```
IPaddr: 812e5c28, Mask: ffffffff, HomeAgent: 812e5c23
Initialized sockets ...
```

```
Authtype is md5 SPI is 259704 Key (length = 16) is:
31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36
```

```
-- RouterSol to 224.0.0.1.
The current state is INIT
```

```
-- RouterSol to 224.0.0.1.
Decision Maker with doit = 0
Default interface address 812e5c28 Interface name eth0 Gateway : 812e5c21
..Rt. discovery addrnum = 1, flags = 3000 seqno = 76 lifetime = 300
Address 812e5c23
.Rt. discovery addrnum = 1, flags = 3000 seqno = 77 lifetime = 300
Address 812e5c23
.Doer processing HOMEACTION
129.46.92.35
```

```

Deleting arp entry...
129.46.92.35
Deleting arp entry...
129.46.92.35 (129.46.92.35) -- no entry
DOWNiface dummy.
lowifacedown failed SIOCSIFFLAGS: No such device
ROUTEset dest      0 dev  Type(6:DELRT 7:ADDRT)  6.
ROUTEset dest      0 dev  Type(6:DELRT 7:ADDRT)  6.
lowrtreq DELRT failed SIOCDELRT: No such process
Setting new routes...
ROUTEset dest net 812e5c20 dev eth0
                        Type(6:DELRT 7:ADDRT)  7
Finished adding/del route for net 812e5c20
ROUTEdefault gw 812e5c23.
lowhwaddrreq called on eth0

=====
Wed Feb 12 18:57:03 1997

-- DEREGISTERME to 129.46.92.35 at port 434 (attempt 0).
[      0:177c3e32] Type 1 Flags 0 Lifetime      0
Homeaddr: 812e5c28, Homeagent: 812e5c23, Careof: 812e5c28
-----

Extension: 32 Length: 20 1 | 0 | 0 | 0 | 81 | 2e | 5c | 28 | 81 | 2e | 5c | 23 | 81 | 2e | 5c
| 28 | 0 | 0 | 0 | 0 | 17 | 7c | 3e | 32 | 20 | 14 | 0 | 3 | f6 | 78 | df | 1 | 95 | 6f | f1 | e9
| 73 | 87 | 95 | df | 72 | e | f7 | 6e | ba | e3 |
-----

=====
Done with HOMEACTION

=====
Wed Feb 12 18:57:04 1997

-- REPLY from 129.46.92.35 (Rejection)
[7898622c:177c3e32] Type 3 Code 133 Lifetime      0
Homeaddr: 812e5c28, Homeagent: 812e5c23
-----

Extension: 32 Length: 20 3 | 85 | 0 | 0 | 81 | 2e | 5c | 28 | 81 | 2e | 5c | 23 | 78 | 98 | 62
| 2c | 17 | 7c | 3e | 32 | 20 | 14 | 0 | 3 | f6 | 78 | ae | 9d | 75 | 34 | 4f | 92 | 46 | a5 | b1 | ac
| 19 | ac | b7 | d0 | 3e | 53 |
-----

=====

```

Implementation Results

The goal of this paper was to study the current literature in the area of Mobile-IP and to provide a summary. A secondary goal was to use an actual Mobile-IP implementation which was also a success. However, the final step of taking the MH on the road was not realized due primarily to the inability to make the FA fully operational. The FA required an OS that supported tunneling and multicasting. These are features must be built into the kernel of the OS. Compiling a new kernel on the FA to meet this functionality was not possible.

Conclusion

As the size and cost of computer systems continues to plummet, the interest and demand for mobility continues to soar. The use of the Internet as an information resource and vital form of communication will continue to attract new users and new applications placing further demands to support functions that the original Internet Protocol was never intended to support. The authors of the

protocol could not have imagined where their idea would eventually lead. It is equally amazing that the protocol has withstood the test of time and provided sufficient flexibility so as to support an incredibly wide range of applications.

Mobile-IP is still in its infancy, RFC 2002 from the IETF is barely a few months old. Various implementations are being developed of which one was discussed in detail in this paper. Although there is still additional development work that needs to be done, it appears we are well on our way to supporting the myriad uses including many that the new wireless age will bring. The ultimate goal or anywhere, anytime, connectionless computing is one step close to reality with the implementation Mobile-IP.

Bibliography

- Atkinson, R. "Security Architecture for the Internet Protocol," RFC 1825. Available at <ftp://ds.intenetic.net/rfc/rfc1825.txt>.
- Badrinath, B.R., et. al, "Handling Mobile Clients: A Case for Indirect Interaction," 4th workshop on Workstation Operating Systems, pp. 91-97. Also available from <ftp:paul.rutgers.edu/pub/badri/wwos4>.
- Balakrishnan, Hari, et. al., "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," ACM SIGCOMM '96, Stanford, CA August 1996.
- Caceres, Ramon, "Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments," IEEE Journal on Selected Areas in Communications, Vol. 13, No. 5, June 1995 pp. 850-857.
- Dixit, Abhijit, Vipul Gupta, "Mobile-IP for Linux (ver 1.00)," Documentation for the Linux Mobile-IP software. Available in postscript format via <http://anchor.cs.binghamton.edu/~mobile/MIPv100/Doc/mip-doc.v100.ps>.
- Droms, R. "Dynamic Host Configuration Protocol," RFC 1541. Available at <ftp://ds.intenetic.net/rfc/rfc1541.txt>.
- Hunt, Craig, "TCP/IP Network Administration," O'Reilly & Associates, Sebastopol, CA, May 1994.
- Internet Engineering Task Force. Home page is <http://www.ietf.cnri.reston.ca.us/home.html>.
- Jacobson, Va, "Congestion Avoidance and Control," 1988 ACM 0-89791-279-9/88/008/0314.
- Johnson, David B. and Charles Perkins, "Mobility Support in IPv6," draft-ietf-mobileip-ipv6-02.txt. November 26, 1996. Available from <http://www.ietf.org/ids.by.wg/mobileip.html>.
- Johnson, David B., "Route Optimization in Mobile IP," draft-ietf-mobileip-optim-05.txt. November 26, 1996. Available from <http://www.ietf.org/ids.by.wg/mobileip.html>.
- Lancki, Ben, et. Al., "Mobile-IP: Transparent Host Migration on the Internet," Linux Journal, August 1996, pp. 10-11, 65.
- Lantinga, S. *README.tunnel* - Documentation for the Linux IPIP tunnel interface. Distributed with kernel source code in directory /usr/src/linux/drivers/net. Documentation is provided in the same directory as README.tunnel, Feb. 1995.
- Mobile-IP for Linux Home page is <http://anchor.cs.binghamton.edu/~mobileip>. This is the source for the software described in this paper. It is provided by the authors of Linux Mobile-IP.
- Myles, Andrew, et. Al. "A Mobile Host Protocol Supporting Route Optimization and Authentication," IEEE Journal on Selected Areas in Communications, Vol. 13, No. 5, June 1995, pp. 839-849.

- Perkins, C. "IP Mobility Support," RFC 2002. October 1996. Available at <ftp://ds.intenic.net/rfc/rfc2002.txt>.
- Perkins, Charles E, Pravin Bhagwat, "A Mobile Networking System based on Internet Protocol", IEEE Personal Communications, First Quarter 1994, 99. 32-41.
- Perkins, Charles, "Providing continuous network access to mobile hosts using TCP/IP," Computer Networks and ISDN Systems 26 (1993) pp. 357-369.
- Perkins, Charles, et. Al., "IMHP: A mobile host protocol for the Internet," Computer Networks and ISDN Systems 27 (1994) pp. 479-491.
- Reynolds, J. "Assigned Numbers," RFC 1700. Available at <ftp://ds.intenic.net/rfc/rfc1700.txt>.
- Rivest, R. "The MD5 Message-Digest Algorithm," RFC 1321. Available at <ftp://ds.intenic.net/rfc/rfc1321.txt>.
- Teraoka, Fumio, et. Al, "IP: A Protocol Providing Host Mobility," Communications of the ACM, August 1994, Vol. 37. No. 8, pp. 67-75.
- Woodward, R. "A Scheme for an Internet Encapsulation Protocol", RFC 1241. Available at <ftp://ds.intenic.net/rfc/rfc1241.txt>.

Mobile-IP and Related Web Sites

Mobile-IP:

- <http://anchor.cs.binghamton.edu/~mobile/>

Mbone:

- <http://andrew.triumf.ca/pub/mbone/mcast-hosts.txt>
- <ftp://ftp.isi.edu/mbone/faq.txt>
- <http://www.research.att.com/mbone-faq.html>
- <ftp://ee.lbl.gov/conferencing>
- <http://www.unige.ch/seinf/mbone.html>
- <http://andrew.triumf.ca/pub/mbone>

Biography



Richard Parry, holds a BS in Electrical Engineering from the University of Illinois, (Urbana, Illinois), an MBA from Northern Illinois University, (DeKalb, Illinois) and a MSCS from North Central College (Naperville, Illinois). He is currently attending the University of California San Diego where he is studying computer science. He is a licensed Professional Engineer (Texas) and has authored papers in various areas including: Wireless Packet Networks, Oxygen Monitoring Systems, Programmable Electronic Safety Systems, Computerized Security Systems, speech synthesis and recognition, management tools, and amateur radioteletype. He is a big fan of the Linux Operating System where he spends entirely too much time. Other interests include amateur radio and more recently satellite communication.

APPENDIX A Mobile-IP Terminology

This document frequently uses the terms shown below. The original source is RFC 2002, IP Mobility Support, Oct. 1996:

Agent Advertisement

An advertisement message constructed by attaching a special Extension to a router advertisement [4] message.

Care-of Address

The termination point of a tunnel toward a mobile node, for datagrams forwarded to the mobile node while it is away from home. The protocol can use two different types of care-of address: a "foreign agent care-of address" is an address of a foreign agent with which the mobile node is registered, and a "co-located care-of address" is an externally obtained local address which the mobile node has associated with one of its own network interfaces.

Correspondent Node

A peer with which a mobile node is communicating. A correspondent node may be either mobile or stationary.

Foreign Network

Any network other than the mobile node's Home Network.

Home Address

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

Home Network

A network, possibly virtual, having a network prefix matching that of a mobile node's home address. Note that standard IP routing mechanisms will deliver datagrams destined to a mobile node's Home Address to the mobile node's Home Network.

Link

A facility or medium over which nodes can communicate at the link layer. A link underlies the network layer.

Link-Layer Address

The address used to identify an endpoint of some communication over a physical link. Typically, the Link-Layer address is an interface's Media Access Control

(MAC) address.

Mobility Agent

Either a home agent or a foreign agent.

Mobility Binding

The association of a home address with a care-of address, along with the remaining lifetime of that association.

Mobility Security Association

A collection of security contexts, between a pair of nodes, which may be applied to Mobile IP protocol messages exchanged between them. Each context indicates an authentication algorithm and mode (Section 5.1), a secret (a shared key, or appropriate public/private key pair), and a style of replay protection in use (Section 5.6).

Node

A host or a router.

Nonce

A randomly chosen value, different from previous choices, inserted in a message to protect against replays.

Security Parameter Index (SPI)

An index identifying a security context between a pair of nodes among the contexts available in the Mobility Security Association. SPI values 0 through 255 are reserved and MUST NOT be used in any Mobility Security Association.

Tunnel

The path followed by a datagram while it is encapsulated. The model is that, while it is encapsulated, a datagram is routed to a knowledgeable decapsulating agent, which decapsulates the datagram and then correctly delivers it to its ultimate destination.

Virtual Network

A network with no physical instantiation beyond a router (with a physical network interface on another network). The router (e.g., a home agent) generally advertises reachability to the virtual network using conventional routing protocols.

Visited Network

A network other than a mobile node's Home Network, to

which the mobile node is currently connected.

Visitor List

The list of mobile nodes visiting a foreign agent.

APPENDIX B Assigned Numbers

In the Internet Protocol (IP) [DDN], [RFC791] there is a field, called Protocol, to identify the next level protocol. This is an 8 bit field.

Assigned Internet Protocol Numbers

Decimal	Keyword	Protocol	References
-----	-----	-----	-----
0		Reserved	[JBP]
1	ICMP	Internet Control Message	[RFC792,JBP]
2	IGMP	Internet Group Management	[RFC1112,JBP]
3	GGP	Gateway-to-Gateway	[RFC823,MB]
4	IP	IP in IP (encapsulation)	[JBP]
5	ST	Stream	[RFC1190,IEN119,JWF]
6	TCP	Transmission Control	[RFC793,JBP]
7	UCL	UCL	[PK]
8	EGP	Exterior Gateway Protocol	[RFC888,DLM1]
9	IGP	any private interior gateway	[JBP]
10	BBN-RCC-MON	BBN RCC Monitoring	[SGC]
11	NVP-II	Network Voice Protocol	[RFC741,SC3]
12	PUP	PUP	[PUP,XEROX]
13	ARGUS	ARGUS	[RWS4]
14	EMCON	EMCON	[BN7]
15	XNET	Cross Net Debugger	[IEN158,JFH2]
16	CHAOS	Chaos	[NC3]
17	UDP	User Datagram	[RFC768,JBP]
18	MUX	Multiplexing	[IEN90,JBP]
19	DCN-MEAS	DCN Measurement Subsystems	[DLM1]
20	HMP	Host Monitoring	[RFC869,RH6]
21	PRM	Packet Radio Measurement	[ZSU]
22	XNS-IDP	XEROX NS IDP	[ETHERNET,XEROX]
23	TRUNK-1	Trunk-1	[BWB6]
24	TRUNK-2	Trunk-2	[BWB6]
25	LEAF-1	Leaf-1	[BWB6]
26	LEAF-2	Leaf-2	[BWB6]
27	RDP	Reliable Data Protocol	[RFC908,RH6]
28	IRTP	Internet Reliable Transaction	[RFC938,TXM]
29	ISO-TP4	ISO Transport Protocol Class 4	[RFC905,RC77]
30	NETBLT	Bulk Data Transfer Protocol	[RFC969,DDC1]
31	MFE-NSP	MFE Network Services Protocol	[MFENET,BCH2]
32	MERIT-INP	MERIT Internodal Protocol	[HWB]
33	SEP	Sequential Exchange Protocol	[JC120]
34	3PC	Third Party Connect Protocol	[SAF3]
35	IDPR	Inter-Domain Policy Routing Protocol	[MXS1]
36	XTP	XTP	[GXC]
37	DDP	Datagram Delivery Protocol	[WXC]
38	IDPR-CMTP	IDPR Control Message Transport Proto	[MXS1]
39	TP++	TP++ Transport Protocol	[DXF]
40	IL	IL Transport Protocol	[DXP2]
41	SIP	Simple Internet Protocol	[SXD]
42	SDRP	Source Demand Routing Protocol	[DXE1]
43	SIP-SR	SIP Source Route	[SXD]
44	SIP-FRAG	SIP Fragment	[SXD]
45	IDRP	Inter-Domain Routing Protocol	[Sue Hares]
46	RSVP	Reservation Protocol	[Bob Braden]
47	GRE	General Routing Encapsulation	[Tony Li]
48	MHRP	Mobile Host Routing Protocol	[David Johnson]
49	BNA	BNA	[Gary Salamon]
50	SIPP-ESP	SIPP Encap Security Payload	[Steve Deering]
51	SIPP-AH	SIPP Authentication Header	[Steve Deering]

52	I-NLSP	Integrated Net Layer Security	TUBA [GLENN]
53	SWIPE	IP with Encryption	[JI6]
54	NHRP	NBMA Next Hop Resolution Protocol	
55-60		Unassigned	[JBP]
61		any host internal protocol	[JBP]
62	CFTP	CFTP	[CFTP,HCF2]
63		any local network	[JBP]
64	SAT-EXPAK	SATNET and Backroom EXPAK	[SHB]
65	KRYPTOLAN	Kryptolan	[PXL1]
66	RVD	MIT Remote Virtual Disk Protocol	[MBG]
67	IPPC	Internet Pluribus Packet Core	[SHB]
68		any distributed file system	[JBP]
69	SAT-MON	SATNET Monitoring	[SHB]
70	VISA	VISA Protocol	[GXT1]
71	IPCV	Internet Packet Core Utility	[SHB]
72	CPNX	Computer Protocol Network Executive	[DXM2]
73	CPHB	Computer Protocol Heart Beat	[DXM2]
74	WSN	Wang Span Network	[VXD]
75	PVP	Packet Video Protocol	[SC3]
76	BR-SAT-MON	Backroom SATNET Monitoring	[SHB]
77	SUN-ND	SUN ND PROTOCOL-Temporary	[WM3]
78	WB-MON	WIDEBAND Monitoring	[SHB]
79	WB-EXPAK	WIDEBAND EXPAK	[SHB]
80	ISO-IP	ISO Internet Protocol	[MTR]
81	VMTP	VMTP	[DRC3]
82	SECURE-VMTP	SECURE-VMTP	[DRC3]
83	VINES	VINES	[BXH]
84	TTP	TTP	[JXS]
85	NSFNET-IGP	NSFNET-IGP	[HWB]
86	DGP	Dissimilar Gateway Protocol	[DGP,ML109]
87	TCF	TCF	[GAL5]
88	IGRP	IGRP	[CISCO,GXS]
89	OSPFIGP	OSPFIGP	[RFC1583,JTM4]
90	Sprite-RPC	Sprite RPC Protocol	[SPRITE,BXW]
91	LARP	Locus Address Resolution Protocol	[BXH]
92	MTP	Multicast Transport Protocol	[SXA]
93	AX.25	AX.25 Frames	[BK29]
94	IPIP	IP-within-IP Encapsulation Protocol	[JI6]
95	MICP	Mobile Internetworking Control Pro.	[JI6]
96	SCC-SP	Semaphore Communications Sec. Pro.	[HXH]
97	ETHERIP	Ethernet-within-IP Encapsulation	[RXH1]
98	ENCAP	Encapsulation Header	[RFC1241,RXB3]
99		any private encryption scheme	[JBP]
100	GMTP	GMTP	[RXB5]
101-254		Unassigned	[JBP]
255		Reserved	[JBP]

APPENDIX C Multicast IP Numbers

Multicast reserved address from UNIX command "host -l mcast.net"

```

rparry@abura:/usr2/rparry 26 > host -l mcast.net
mcast.net          NS      VENERA.ISI.EDU
mcast.net          NS      NS.ISI.EDU
mcast.net          NS      SGI.COM
mcast.net          NS      NIC.NEAR.NET
EXPERIMENT.mcast.net  A      224.0.1.20
NWN-ADAPTOR.mcast.net  A      224.0.1.44
RSVP-ENCAPSULATION.mcast.net  A      224.0.0.14
SUB-NIS.mcast.net    A      224.0.1.8
RLN-SERVER.mcast.net  A      224.0.1.36
ISMA-1.mcast.net     A      224.0.1.45
CISCO-RP-DISCOVERY.mcast.net  A      224.0.1.40
DANTZ.mcast.net     A      224.0.1.38
ISMA-2.mcast.net     A      224.0.1.46
MTRACE.mcast.net    A      224.0.1.32
SVRLOC-DA.mcast.net  A      224.0.1.35
AMPR-INFO.mcast.net  A      224.0.1.31
NWN-DISCOVERY.mcast.net  A      224.0.1.43
LMSC-CALREN-1.mcast.net  A      224.0.1.27
SEANET-IMAGE.mcast.net  A      224.0.1.18
LMSC-CALREN-2.mcast.net  A      224.0.1.28
LMSC-CALREN-3.mcast.net  A      224.0.1.29
LMSC-CALREN-4.mcast.net  A      224.0.1.30
IBERIAGAMES.mcast.net  A      224.0.1.42
AUDIONEWS.mcast.net  A      224.0.1.7
IETF-2-VIDEO.mcast.net  A      224.0.1.15
NBC-PFN.mcast.net    A      224.0.1.26
MOBILE-AGENTS.mcast.net  A      224.0.0.11
TELERATE.mcast.net   A      224.0.1.47
BASE-ADDRESS.mcast.net  A      224.0.0.0
RWHO.mcast.net       A      224.0.2.1
PIM-ROUTERS.mcast.net  A      224.0.0.13
RWHOD.mcast.net      A      224.0.1.3
MUSIC-SERVICE.mcast.net  A      224.0.1.16
CIENA.mcast.net      A      224.0.1.48
IETF-1-VIDEO.mcast.net  A      224.0.1.12
GATEKEEPER.mcast.net  A      224.0.1.41
MICROSOFT-DS.mcast.net  A      224.0.1.24
ALL-CBT-ROUTERS.mcast.net  A      224.0.0.15
NSS.mcast.net        A      224.0.1.6
DVMRP.mcast.net      A      224.0.0.4
NTP.mcast.net        A      224.0.1.1
DART-VIDEO.mcast.net  A      224.2.0.2
SAIC-MDD.mcast.net   A      224.0.2.64
DCAP-SERVERS.mcast.net  A      224.0.1.49
DVMRP-MOSPF.mcast.net  A      224.0.1.21
RIP2-ROUTERS.mcast.net  A      224.0.0.9
SVRLOC.mcast.net     A      224.0.1.22
SEANET-TELEMETRY.mcast.net  A      224.0.1.17
VNP.mcast.net        A      224.0.1.4
MLOADD.mcast.net     A      224.0.1.19
IGRP-ROUTERS.mcast.net  A      224.0.0.10
DCAP-CLIENTS.mcast.net  A      224.0.1.50
SUN-RPC.mcast.net    A      224.0.2.2
IETF-2-AUDIO.mcast.net  A      224.0.1.14
ST-HOSTS.mcast.net   A      224.0.0.8
SGI-DOG.mcast.net    A      224.0.1.2
ALL-SYSTEMS.mcast.net  A      224.0.0.1

```

DHCP-AGENTS.mcast.net	A	224.0.0.12
VMTP-MGR.mcast.net	A	224.0.1.0
AVIATOR.mcast.net	A	224.0.1.5
RSVP-ENCAP-1.mcast.net	A	224.0.1.33
IETF-1-LOW-AUDIO.mcast.net	A	224.0.1.10
RSVP-ENCAP-2.mcast.net	A	224.0.1.34
XINGTV.mcast.net	A	224.0.1.23
IETF-1-AUDIO.mcast.net	A	224.0.1.11
NCB-PRO.mcast.net	A	224.0.1.25
MCNTP-DIRECTORY.mcast.net	A	224.0.1.51
IETF-2-LOW-AUDIO.mcast.net	A	224.0.1.13
CISCO-RP-ANNOUNCE.mcast.net	A	224.0.1.39
PROSHARE-MC.mcast.net	A	224.0.1.37
ST-ROUTERS.mcast.net	A	224.0.0.7
ALL-ROUTERS.mcast.net	A	224.0.0.2
MTP.mcast.net	A	224.0.1.9
DART-AUDIO.mcast.net	A	224.1.0.200
OSPF-ALL.mcast.net	A	224.0.0.5
SAP.mcast.net	A	224.2.127.254
OSPF-DSIG.mcast.net	A	224.0.0.6