To appear in the proceedings of SIGCOMM '96. Made available on the authors' ftp server with permission of the ACM (www.acm.org)

Internet Mobility 4x4

Stuart Cheshire and Mary Baker
Computer Science Department, Stanford University
Stanford, California 94305, USA
{cheshire,mgbaker}@cs.stanford.edu

Abstract

Mobile IP protocols allow mobile hosts to send and receive packets addressed with their home network IP address, regardless of the IP address of their current point of attachment in the Internet.

While some recent work in Mobile IP focuses on a couple of specific routing optimizations for sending packets to and from mobile hosts [Joh96] [Mon96], we show that a variety of different optimizations are appropriate in different circumstances. The best choice, which may vary on a connection-by-connection or even on a packet-by-packet basis, depends on three factors: the characteristics the protocol should optimize, the permissiveness of the networks over which the packets travel, and the level of mobile-awareness of the hosts with which the mobile host corresponds.

Of the sixteen possible routing choices that we identify, we describe the seven that are most useful and discuss their benefits and limitations. These optimizations range from the most costly, which provides completely transparent mobility in all networks, to the most economical, which does not attempt to conceal location information. In particular, hosts should retain the option to communicate conventionally without using Mobile IP whenever appropriate.

Further, we show that all optimizations can be described using a 4x4 grid of packet characteristics. This makes it easier for a mobile host, through a series of tests, to determine which of the currently available optimizations is the best to use for any given correspondent host.

1. Introduction

The increasing number of portable computers, combined with the growth of wireless services, makes supporting Internet mobility important. Mobile hosts need to switch between networks in different administrative domains as they move around the network, and they need to switch between different types of networks (cellular telephone, packet radio, Ethernet, etc.) to achieve the best possible connectivity wherever they are located.

We believe that IP is the correct layer at which to implement basic mobility support, rather than lower or higher layers. The problem with link layer solutions is that they are limited to a single medium. For example, one of the current ways of providing mobile Internet access is to use a cellular telephone and modem to dial into a central PPP [RFC1661] server. While this method provides connectivity (albeit at a cost of about 40¢ per minute), it is limited to a single technology — cellular telephony. Since its inception in the early 1970s, one of the most important guiding principles of the Internet has been to seek general-purpose solutions that work for all network technologies, not special-purpose hardware-specific solutions [Cer74]. Similarly, mobility solutions that require widespread changes at layers above IP would be highly impractical.

The challenge for supporting mobility at the IP layer is handling address changes. Even if the IP address of a mobile host's network interface changes when it moves from one network to another, the mobile host should be able to continue corresponding with other machines in the Internet. Most mobile IP protocols [Bla94] [Gup96] [Hag93] [Ioa93] [Myl93] [Per94] [Per96a] [Tas94] [Ter94] address this problem by allowing a mobile host to use its home network IP address no matter where it currently resides. Of these protocols, the protocol specified by the Internet Engineering Task Force (IETF) [Per96a] has become the most popular. It is simple, compatible with existing applications and hosts, and places no special burdens on normal IP routers in the Internet.

When trying to implement the IETF protocol, however, we ran into several problems that led us to conclude that one size does not fit all. There are at least two areas warranting further investigation. Firstly, the basic protocol does not work in some security-conscious networks. Secondly, the route used for packets from other hosts to the mobile host is not the most efficient. We believe that different optimizations are appropriate in different circumstances. For instance, a mobile host corresponding with a host that is physically connected to the same Ethernet segment should not require every packet to travel via its home agent. Choosing the best way to send packets depends upon what characteristics should be optimized, the permissiveness of the networks through which the packets must travel, and whether or not the correspondent hosts have any awareness of mobility. The best choice may even vary on a packet-by-packet basis.

In the next section we briefly outline our Mobile IP protocol, which is based strongly on the Draft IETF Mobile IP proposal (soon to become an official RFC). In Section 3 we list our optimization goals and give examples of some of the difficulties we faced in trying to achieve them.

We then show how to implement the desired optimizations and describe the circumstances in which each is suitable. Section 4 describes the four ways that a mobile host can send packets to a correspondent host, and Section 5 describes the four ways that a correspondent host can send packets to a mobile host. We then describe in Section 6 how these two different choices influence each other. The choices are not completely independent, and the capabilities of the correspondent host constrain which combinations are applicable for the mobile host to use. In any particular situation, a mobile host should choose the best alternative that is available to it.

In Section 7 we give the current status of our implementation and propose some future work.

Although this paper assumes that a mobile host is communicating with a conventional non-mobile correspondent host, the same techniques and optimizations apply equally well if both hosts are mobile.

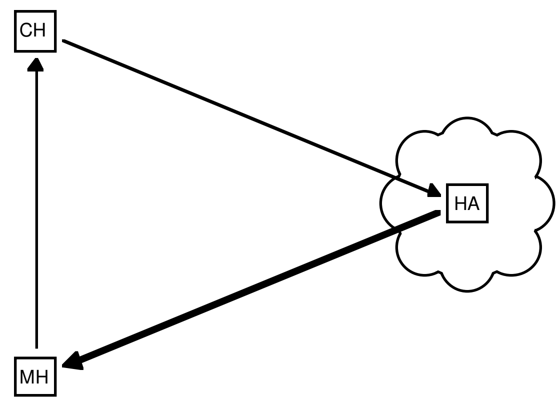

Figure 1. Basic Mobile IP.
This figure shows communication involving a mobile host (MH) away from its home network, its home agent (HA) and a correspondent host (CH). In these figures, squares represent Internet hosts, clouds represent administrative network domains, thin arrows represent normal IP packets being sent from one host to another, and thick arrows represent IP packets that carry an encapsulated IP packet. Here, the administrative domain represented by the cloud is the mobile host’s home domain.

2. Basic Mobile IP

The goal of Mobile IP is to allow a mobile host to send and receive packets addressed with its home IP address regardless of its current point of attachment to the Internet, and to maintain communication associations (such as TCP connections) even if the point of attachment changes during their lifetime. Users should not have to restart their applications whenever they change location, especially applications such as remote logins that build up significant state at the remote endpoint. It is important to note that computers do not need to be turned on continuously in order to maintain quiescent TCP connections; putting a laptop computer to sleep while moving it from place to place does not necessarily break connections. On our laptop computers running Linux we frequently have idle telnet connections that are preserved for hours, and sometimes even for days or weeks, while the laptop computer is sitting unused in ‘sleep’ mode.

In order to meet these goals of location transparency and connection durability, each mobile host has a permanent home IP address that does not change. This unchanging address enables conventional Internet hosts, which are unaware of mobility issues, to communicate with the mobile host. If the address changed each time the mobile host changed its point of connection to the Internet, then our requirement of connection durability would not be met, because TCP connections to other Internet hosts would break every time the mobile host moved.

When the mobile host is at home, it sends and receives packets using its home IP address and functions like a normal non-mobile Internet host.

When it is away from home, the mobile host obtains a temporary ‘guest’ connection to the Internet at the site it is visiting. This connection may be obtained by connecting to an Ethernet segment and asking a friendly network administrator to assign an IP address to the visiting host; it may be obtained by connecting to an Ethernet segment and having an address assigned automatically by DHCP [RFC1541], or the connection may be obtained via communication with an IETF ‘foreign agent’ that has been placed on the network expressly for the purpose of supporting visiting mobile hosts.

After the mobile host has connected to the visited network (directly, or via a foreign agent), it registers its new location with its home agent. The home agent is a machine on the mobile host’s home network that acts as a proxy on behalf of the mobile host for the duration of its absence. The home agent uses gratuitous proxy ARP [RFC1027] to capture all IP packets addressed to the mobile host. When packets addressed to the mobile host arrive on its home network, the home agent intercepts them and uses encapsulation (often called ‘tunneling’) to forward them to the mobile host’s current location, as shown in Figure 1.

If the mobile host moves again to a different point of attachment on the Internet, then it must again inform its home agent of its new location. During this transition period it may be possible to lose packets, but higher-level Internet protocols are already responsible for mechanisms to ensure reliable packet delivery where required, so it is not necessary to duplicate this functionality within Mobile IP.

As shown in Figure 1, delivering packets in the opposite direction, from the mobile host to the correspondent host, is simpler. The mobile host transmits its packets into the Internet, addressed directly to the correspondent host. Because the source address in the packets is the mobile host’s permanent home source address, when the correspondent host replies to those packets, the replies will find

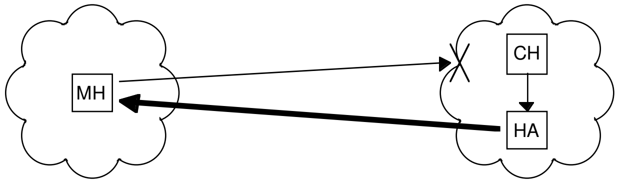

Figure 2. Problem with Source Address Filtering
The home agent encapsulates the correspondent host's packets and correctly forwards them to the mobile host. Unfortunately, the mobile host's replies are discarded by a security-conscious boundary router and never reach the correspondent host to which they are addressed.

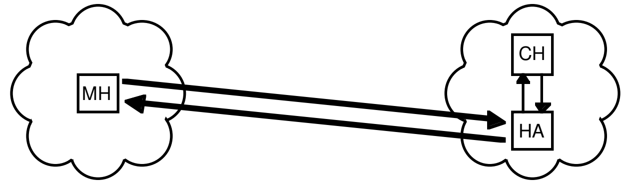

Figure 3. Bi-directional Tunneling
By tunneling all of its packets via the home agent, the mobile host avoids their being discarded by the routers at the boundary of its home domain.

the mobile host even if it moves; all replies will travel indirectly via the home agent. The latency and available bandwidth over the two different paths may be significantly different, but this is not unusual for IP. The IP specification makes no promises about the path that packets will take, and much of the current Internet backbone already routes packets going in different directions over different paths [Par96].

The IETF Mobile IP proposal says that mobile hosts may connect directly to the visited network or indirectly via a "foreign agent". When connecting via a foreign agent, the home agent tunnels packets to this foreign agent, which decapsulates them and delivers the enclosed packet to the mobile host.

Our implementation of the protocol emphasizes self-sufficiency for mobile hosts. They connect directly to the Internet and operate independently without requiring a foreign agent. It is impractical for mobile hosts to assume that foreign agent services will be available everywhere. Fortunately it is not difficult for mobile hosts to provide their own mobility support in the absence of foreign agents [Bak96]. Foreign agents may be able to provide useful services to mobile hosts, but they also restrict the freedom of the mobile host to choose from the full range of possible optimizations.

The most important of these optimizations, which foreign agents prevent, is the freedom to forgo the services of Mobile IP for communications that do not need them. A lot of work has been done to make protocols client-originated wherever possible. The trend towards using POP [RFC1725] to retrieve electronic mail is one such example. Mobile IP makes it possible to send packets addressed to a mobile host without knowing its current location and to be able to set up durable connections. These are important facilities, but they should not be provided at the expense of the ability to operate with the efficiency of a normal Internet host.

Some researchers [Tas94] have proposed a different approach to the mobility problem, by assigning a new unique permanent identifier to every host on the Internet. They propose rewriting transport protocols like TCP to identify connection endpoints using this new identifier instead of the location-dependent IP address. While this may appear to result in a more 'elegant' solution, it offers no real benefits over the more pragmatic solution proposed by the IETF. The information contained in this hypothetical enlarged TCP header with its unique endpoint identifiers is semantically identical to the information contained in the IETF's encapsulated IP and TCP headers taken together. Although adding an encapsulated IP header to the packet consumes slightly more space than a redesigned TCP header might, this overhead can be minimized by use of Generic Routing Encapsulation [RFC1702] or Minimal Encapsulation [Per95]. In addition, there is no need to invent a new namespace when the existing IP address namespace is already well understood, and we already have established mechanisms for allocating, assigning, and managing unique identifiers in that namespace.

3. Project Goals

In this section we describe the areas for optimization and improvement over the basic Mobile IP protocol, and the constraints to which we must adhere. The purpose of the optimizations is to achieve efficient delivery of packets, in terms of their size and the path they take through the network. These factors affect the delivery latency and the load on the shared resources in the Internet. There are two constraints. The first is that we may not assume any special support from routers, except for normal IP routing. This constraint is motivated by the end-to-end argument [Sal84], which states that we should not burden the network with functions that can be performed equally well, or better, at the endpoints. The second constraint is that packets be correctly deliverable to their destination: the choice of feasible optimizations is constrained by the permissiveness of the networks over which the packets travel and the level of mobility awareness of the correspondent host. (The term "correctly deliverable" is used in the normal context of a "best-effort" datagram network, meaning that with high probability packets are successfully delivered. Existing causes of packet loss still exist, even when using Mobile IP.)

3.1 Ensure Deliverability

An important goal is that all Mobile IP systems be able to work correctly in the current Internet and interoperate correctly with current hosts, but there are situations where even the IETF Mobile IP solution fails. For security or policy reasons, many networks will not deliver packets which are sent the way the IETF specification describes.

In most networks, the packets from the mobile host will never reach the correspondent host, for the reason illustrated in Figure 2. When the packet arrives at the boundary

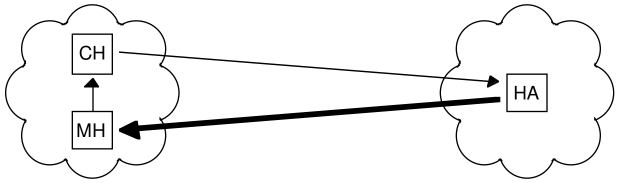

Figure 4. Behavior when CH is Close to MH
The correspondent host addresses packets to the mobile host's permanent home address. The Internet naturally delivers the packets to the mobile host's home network, where the home agent encapsulates them and sends them to the mobile host. Outgoing packets from the mobile host are delivered in the normal fashion, directly to the correspondent host.

Figure 5. A Smart Correspondent Host.
The correspondent host knows the mobile host's temporary care-of address, so it encapsulates the packet itself and sends it directly.

of the home institution, the boundary router will see a packet coming from outside the home network, with a source address claiming that the packet originates from a machine inside the home network. Most network administrators, concerned about security, will configure boundary routers to drop such packets. Many network services, including the majority of NFS servers, determine whether or not they can safely trust the host sending the packet solely based on the source address of the packet. If we allow machines *outside* our network to send in packets with source addresses claiming to originate from trusted machines *within* our network, we effectively allow any machine on the Internet to impersonate any machine in our organization. Although in most cases replies will not get back to the machine originating the attack, many kinds of attack can be performed without needing to see any replies.

Another reason that packets sent by the mobile host might be discarded is that most end-user networks have a policy forbidding *transit traffic*. (Transit traffic is traffic passing through an intermediate network on the way to its final destination.) Some network administrators enforce this policy by configuring routers to discard packets with source addresses that appear to be invalid. Most traffic on the Internet backbone is transit traffic, but tail circuits, such as a 100Base-T connection to a desktop computer, are typically not expected to carry transit traffic. Packets appearing on such a tail circuit with source addresses belonging to a foreign network normally indicate some inappropriate use of the network, and would be discarded by the router.

The solution to these problems, which has also been described in [Mon96], is shown in Figure 3. By having the mobile host encapsulate outgoing packets and send them via the home agent, the inner packets are protected from scrutiny by routers. The boundary router only sees packets coming from a machine at some other institution (the mobile host using a temporary care-of address belonging to that institution) going to a local machine (the home agent). These packets are able to travel through the router, and the home agent can send the enclosed packets on the local network on behalf of the mobile host. This lengthens the distance that the packets travel but meets the deliverability requirement.

Note that it is not just 'firewall' routers that will drop these packets. Even the most forgiving of boundary routers would be expected to perform the rudimentary source address checks described above. Firewall routers usually impose much stricter restrictions. In situations where a mobile user is communicating with home services protected by a firewall, we anticipate that the firewall itself would be set up to act as the mobile user's home agent, sitting as it does on the boundary between the untrusted outside world and the trusted world inside.

3.2 Minimize Latency

Subject to the constraint that the packet must be successfully deliverable to the destination, our next goal is to minimize the distance that it travels through the Internet. Packets delivered via the home agent typically travel further through the Internet than they would if they were delivered by the optimal unicast route. As well as increasing the round-trip delay observed by the communicating parties, this also affects other users by increasing the overall load on the shared resources of the Internet.

In Figures 2 and 3 the extra distance added by indirect delivery is small compared to the distance that the packets would travel anyway. Even if the mobile host had been communicating directly with the correspondent host, the packets would still have had to make the long journey across the Internet between the two sites.

Unfortunately in Figure 4 the extra distance is not small. When they travel indirectly via the home agent, packets sent by the correspondent host travel significantly further than is necessary. It would be more efficient if a correspondent host could discover that the mobile host is nearby, and send the packets directly to it. A correspondent host that is aware of mobility issues should be allowed to do this.

We and others [Joh96] have approached this problem by developing an optional routing optimization mechanism to avoid the overhead of indirect delivery via the home agent, as shown in Figure 5. A correspondent host with enhanced networking software can learn the mobile host's temporary care-of address, and then perform the encapsulation itself, sending the packet directly to the mobile host. This avoids the overhead of indirect delivery.

There are several ways that a smart correspondent host can learn that a host is mobile and learn its current temporary care-of address. We are implementing two mechanisms. The first is that when the home agent forwards a packet to the mobile host, it may also send an ICMP message back to the packet's source, informing it of the mobile host's current temporary care-of address. The second is an extension to the Domain Name Service [RFC1034], similar to the current MX records which provide alternative addresses for mail delivery [RFC974]. A mobile host that is away from home, but not currently changing location frequently, could register its care-of address with the extended DNS service. When a smart correspondent looks up a host name and sees that it has a temporary address record in addition to the normal permanent address record, it then knows that it has the option to send packets directly to that temporary address.

It has also been proposed [Per96b] that support for route optimization should be included in the base IPv6 specification [RFC1883] for all IPv6 hosts.

3.3 Minimize Size

In addition to the overhead that indirect delivery adds, encapsulation also adds overhead by increasing the size of the packets. Encapsulation typically adds 20 bytes to the size of the packet in IPv4, and more in IPv6. If the addition of the extra 20 bytes makes the packet exceed the IP maximum transmission unit (MTU) for a particular link, then the packet will be fragmented, doubling the packet count. To avoid this overhead, we should avoid encapsulation when possible.

4. Outgoing Packets

In this section we look at how to achieve our previously described optimization goals for outgoing packets from the mobile host. Although we could use loose source routing, this achieves little that can't be done equally well using an encapsulating header [Per96c]. Current IP routers typically handle packets with options much more slowly than they handle normal unadorned IP packets. In IPv6, source routing is performed exclusively using routing headers, which is equivalent to encapsulation.

Other than loose source routing the only way to influence the path that the packets take through the Internet is by the choice of source and destination addresses in the IP header. If we choose to encapsulate the packet, then we also have the freedom to choose the source and destination addresses in the encapsulating IP packet.

If we choose not to encapsulate IP packets, then the mobile host sends out normal IP packets exactly as a conventional non-mobile host does. The source address of such a packet identifies the entity with which the correspondent host is communicating. If the mobile host uses its home IP address as the source address, then its mobility remains transparent to the correspondent host. As described in Section 3.1, some networks will discard such packets if they are sent directly to the correspondent host. If the mobile host instead uses its temporary care-of address, then the packets will not be discarded by the network, but transparent mobility is lost and TCP connections will be unceremoniously broken when the mobile host moves. Both of these addressing techniques are appropriate in some situations and not in others.

To achieve transparent mobility *and* successful delivery in security conscious networks, we use encapsulation. Encapsulation increases the size of the packet, but it has advantages. It allows us to use different source addresses in the inner and outer headers. We use the home IP address as the source address of the inner packet, to preserve location transparency. We use the temporary care-of address as the source address of the outer packet, so that security conscious routers will not discard it.

When using encapsulation, we have to choose which host will perform the decapsulation. The most conservative choice is to send the packets back to the home agent for decapsulation and subsequent delivery to the correspondent host. We know that we can rely on our own home agent to perform decapsulation for us, but the packets may travel significantly further through the network than is necessary. The most aggressive choice is to send the packets directly to the correspondent host. This avoids indirect delivery via the home agent but can only be used if the correspondent host is able to process encapsulated packets. As before, both of these techniques have situations where they are appropriate, and situations where they are not.

Figures 6 and 7 show the choices the mobile host must make. For packets sent unencapsulated, it has a choice of two possible source addresses for the packet: the permanent home address or the temporary care-of address. For packets sent encapsulated, it must decide whether to send the encapsulating packet to its home agent or directly to the correspondent host. Below we summarize these four delivery choices available to the mobile host and give examples of situations where each is useful. In our notation 'S' denotes the source address, and 'D' denotes the destination address. In the cases where the packet is encapsulated, 's' and 'd' denote the source and destination addresses in the encapsulating (outer) header.

Out-IE: Outgoing, Indirect, Encapsulated (Conservative mode)

s = From temporary care-of address
d = To home agent
S = From permanent home address
D = To correspondent host

Advantages: 1. Avoids the risk of an intervening router discarding the packet because it appears to have an invalid source address for the network from which it originates. 2. The correspondent host is unaware that the packet originated on a mobile host, and needs no special software to receive the packet.

Disadvantages: 1. Indirect delivery. 2. Encapsulation overhead.

Motivation: All mobile hosts must support tunneling through the home agent, since this is the only method that can be relied upon to work in all situations. For example, a mobile host in a network with source address filtering, communicating with a correspondent host that is not mobile-aware, has no choice but to use Out-IE.

Privacy concerns provide another motivation for tunneling through the home agent. In some situations, mobile users may not wish to reveal their current location to the corre-

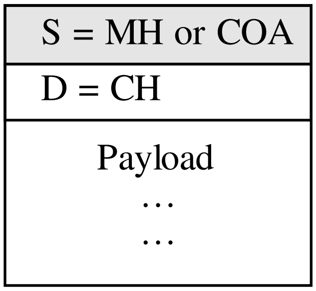

Figure 6. Outgoing Packet Sent Unencapsulated
The source address (shaded) may be either the mobile host's permanent home address (MH) or its temporary care-of address (COA). The destination address (D) is the address of the correspondent host (CH).

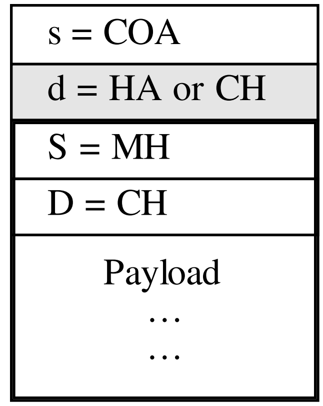

Figure 7. Outgoing Packet Sent Encapsulated
For the inner (encapsulated) packet, the source address (S) is the mobile host's permanent home address (MH), and the destination address (D) is the address of the correspondent host (CH). For the outer packet, the source address (s) is always the mobile host's temporary care-of address (COA), but the destination address (shaded) is either the address of the home agent (HA), or the address of the correspondent host (CH), depending on which host is to decapsulate the packet.

spondent host. In these cases, sending all outgoing packets indirectly via the home agent may be the method the user wants, even when other more efficient alternatives are also available.

Out-DE: Outgoing, Direct, Encapsulated
(Decapsulation-capable correspondent host)

s = From temporary care-of address
d = To correspondent host
S = From permanent home address
D = To correspondent host

Advantages: 1. Direct delivery. 2. Avoids the risk of an intervening router discarding the packet because it appears to have an invalid source address for the network from which it originates.

Disadvantages: 1. Encapsulation overhead. 2. The correspondent host must have the capability of decapsulating encapsulated IP packets.

Motivation: Out-DE is the best choice for a mobile host in a network with source address filtering, communicating with a correspondent host that is able to process encapsulated packets.

Out-DH: Outgoing, Direct, Home Address
(No source address filtering)

S = From permanent home address
D = To correspondent host

Advantages: 1. Direct delivery. 2. No encapsulation overhead. 3. The correspondent host is unaware that the packet originated on a mobile host, and it needs no special software to receive the packet.

Disadvantages: 1. An intervening router may discard the packet because it appears to have an invalid source address for the network from which it originates.

Motivation: Out-DH is the best choice when none of the routers on the path from the mobile host to the correspondent host performs source address filtering.

Out-DT: Outgoing, Direct, Temporary Address
(No Mobile IP)

S = From temporary care-of address
D = To correspondent host

Advantages: 1. Direct delivery. 2. No encapsulation overhead. 3. Avoids the risk of an intervening router discarding the packet because it appears to have an invalid source address for the network from which it originates.

Disadvantages: 1. The permanent home address is not used at all. Hence, packets sent this way forgo the benefits of Mobile IP — if the mobile host moves to a new location then reply packets addressed to the old temporary address will be lost.

Motivation: Out-DT is the best choice when transparent mobility is not required. For example, HTTP connections are frequently very short lived, and if the host does move during the brief life of the connection, causing it to break, the user has the option of clicking the Web browser's 'reload' button. In many cases the user may prefer the small risk of an occasional incomplete image, rather than the large cost of slowing down all Web browsing with the overhead of using Mobile IP for every connection. Connectionless datagram transactions, such as DNS name lookups, may also be usefully performed this way.

5. Incoming Packets

In this section we look at how to achieve our previously described optimization goals for incoming packets sent to

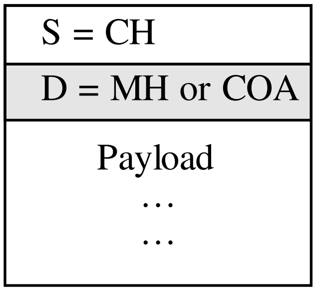

Figure 8. Incoming Packet Sent Unencapsulated
The source address (S) is the address of the correspondent host (CH). The destination address (D) may be either the mobile host's permanent home address (MH) or its temporary care-of address (COA).

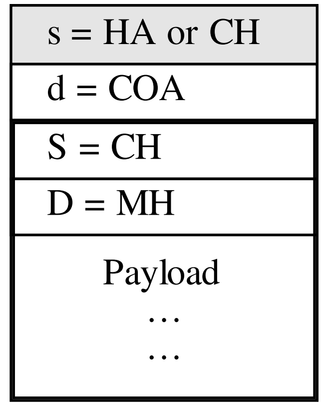

Figure 9. Incoming Packet Sent Encapsulated
For the inner (encapsulated) packet, the source address (S) is the address of the correspondent host (CH), and the destination address (D) is the mobile host's permanent home address (MH). For the outer packet, the destination address (d) is the mobile host's temporary care-of address (COA). When the mobile host receives the packet, the source address (shaded) may be either the home agent's address (HA), or the address of the correspondent host (CH), depending on who performed the encapsulation.

the mobile host from a correspondent host. Normally, packets sent by a correspondent host will be straightforward IP packets addressed to the mobile host's permanent home address, since today's correspondent hosts run conventional IP networking software that is unaware of mobility issues. However, over time, hosts will be enhanced so that their networking software is aware of mobility issues. As with the mobile host, the only way a correspondent host can influence the path that the packets take through the Internet is through its choice of source and destination addresses in the IP header. If it chooses to encapsulate the packet, then it also has freedom to choose the source and destination addresses in the encapsulating IP packet.

The destination address identifies the entity with which the correspondent host is communicating. If the correspondent host uses the mobile host's home IP address as the destination address, then the packets will be successfully deliverable regardless of where in the Internet the mobile host is connected, but they may not travel by the most efficient route. If the correspondent host instead uses the mobile host's current temporary care-of address, then the packets will be delivered efficiently, but transparent mobility is lost and TCP connections will be broken without warning when the mobile host moves. Both of these addressing techniques have situations where they are appropriate and situations where they are not.

To achieve transparent mobility *and* efficient delivery, we use encapsulation. Encapsulation increases the size of the packet, but it allows us to use different destination addresses in the inner and outer headers. We use the home IP address as the destination address of the inner packet, to preserve location transparency, and we use the temporary care-of address as the destination address of the outer packet, so that it will be routed directly to the destination.

When both the mobile host and the correspondent host are physically connected to the same link-layer network segment, there is a better alternative than any of the three choices listed above. In this situation, the IP packet need not pass through any Internet routers at all. It can be delivered directly to the mobile host in a link-layer packet, without invoking IP-layer routing mechanisms. In this case, the IP packet that the correspondent host sends looks exactly the same as the packet that a host with no mobility awareness would send. The only difference is in the link-layer destination to which the packet is addressed.

Figures 8 and 9 show the possible kinds of packets that can arrive at the mobile host. For packets sent unencapsulated, directly from the correspondent host, the destination address will either be the temporary care-of address or, in the special case of two hosts on the same link-layer segment, it may be the home address. For packets sent encapsulated, the source address will depend on whether the packet was encapsulated at the correspondent host or whether it was first sent to the home network and encapsulated there by the home agent.

Below we summarize the four ways that a correspondent host can send packets to a mobile host, and give examples of situations where each is useful. Note that these four ways are *not* the same as the four options for outgoing packets, although when the choices for incoming and outgoing packets are compared, some symmetry does emerge.

In-IE: Incoming, Indirect, Encapsulated
(Correspondent unaware that host is mobile)

S = From Correspondent host
D = To Permanent home address

On arrival at the home agent, the packet is encapsulated to make:

s = From home agent
d = To temporary care-of address
S = From correspondent host
D = To permanent home address

Advantages: 1. The correspondent host is unaware of the special status of the mobile host, and needs no special software to send the packet.

Disadvantages: 1. Indirect delivery. 2. Encapsulation overhead.

Motivation: All Mobile IP systems must support tunneling through the home agent, since this is the only method that can be relied upon to work in all situations. For example, current Internet hosts will simply address packets to the mobile-host's home IP address, so the home agent must be present and able to forward those packets to the mobile host.

In-DE: Incoming, Direct, Encapsulated (Mobile-aware correspondent host)

s = From correspondent host
d = To temporary care-of address
S = From correspondent host
D = To permanent home address

Advantages: 1. Direct delivery.

Disadvantages: 1. Encapsulation overhead. 2. The correspondent host needs to be aware of the special status of the mobile host, and needs special software to look up the temporary address and perform the encapsulation.

Motivation: In any situation where the correspondent host is mobile-aware and knows the mobile host's current care-of address, sending the packets directly is preferable to sending them via the home agent.

In-DH: Incoming, Direct, Home Address (Same physical network segment)

S = From correspondent host
D = To permanent home address

Advantages: 1. Direct delivery. 2. No encapsulation overhead.

Disadvantages: 1. Only applicable when the correspondent host and the mobile host are connected to the same network segment.

Motivation: In-DH is the best choice when visiting another institution and connecting to their network to access data or services on that network. As well as being a fairly common case, the benefit of avoiding communicating through the home agent can be significant, especially if the visited institution is in Japan and the home agent is at MIT.

In another context, this delivery technique is already used when a mobile host operates using a separate foreign agent. The foreign agent uses this delivery technique to deliver the packet over the final hop to the mobile host.

In-DT: Incoming, Direct, Temporary Address (No Mobile IP)

S = From correspondent host
D = To temporary care-of address

Advantages: 1. Direct delivery. 2. No encapsulation overhead.

Disadvantages: 1. The permanent home address is not used at all. Hence, packets sent this way forgo the benefits of Mobile IP — if the mobile host moves to a new location then packets addressed to the old temporary address will be lost.

Motivation: As described in Section 4, this may be useful for short-lived connections and short connectionless datagram exchanges. Also, when a mobile host chooses to initiate a direct communication using its temporary care-of address, replies from the correspondent host will implicitly be sent back to that temporary address without it ever being aware that any mobility issues are involved.

6. 4x4 Choices

The choices presented in Sections 4 and 5 are not independent. For some communication mechanisms, such as dissemination of information via unreliable multicast streams, one-way packet delivery may be sufficient. However, the majority of protocols require two-way communication in order to operate. An NFS request requires a response, and a TCP data segment requires an acknowledgement. This means that for any conversation between two hosts, two decisions must be made: How packets *from* the mobile host are to be sent, and how packets *to* the mobile host are to be sent. Because these decisions are not independent, not all of the sixteen possible combinations are useful.

Figure 10 shows the possible combinations. Below we describe the useful combinations starting with the first row, which is the most conservative and most location transparent, and ending with the last row, which is the least transparent. We then describe why the darkly shaded options in the fourth row and fourth column are not useful.

6.1.1 Row A (Communication with Conventional Correspondent Host)

The first row of the chart shows combinations that are useful when the mobile host is communicating with a conventional Internet host that is not aware of mobility issues. All packets the correspondent host sends addressed to the mobile host's permanent address will be routed naïvely to the home agent, hence all incoming packets will be delivered by the indirect, encapsulated method.

However, the mobile host still has a choice about how it sends outgoing packets back to the correspondent host:

In-IE/Out-IE. The mobile host can send outgoing packets, encapsulated, to the home agent. This is the most conservative approach to Mobile IP. All that is required is for the mobile host to be able to send and receive packets from a single other host on the Internet — its home agent. If it cannot do even this, then it can be reasonably claimed that

		Out-IE **Out**going **I**ndirect, **E**ncapsulated	Out-DE **Out**going **D**irect, **E**ncapsulated	Out-DH **Out**going **D**irect, **H**ome Address	Out-DT **Out**going **D**irect, **T**emp. Address
Conventional Correspondent Host	Row A: In-IE **In**coming **I**ndirect, **E**ncapsulated	Most conservative: most reliable, least efficient	Requires only decapsulation capability of the correspondent host	Requires there to be no security-conscious routers on the path	
Mobile-Aware Correspondent Host	Row B: In-DE **In**coming **D**irect, **E**ncapsulated		Requires fully mobile-aware correspondent host	Requires there to be no security-conscious routers on the path	
Both Hosts on Same Network Segment	Row C: In-DH **In**coming **D**irect, **H**ome Address			Requires both hosts to be on same network segment	
Forgoing Mobility Support	Row D: In-DT **In**coming **D**irect, **T**emp. Address				Most efficient, but forgoes benefits of Mobile IP

Figure 10. Internet Mobility 4x4

This figure shows useful ways that a mobile host can communicate with a correspondent host. The different rows show routing options for incoming packets to the mobile host, and the different columns show options for outgoing packets from the mobile host. Each box lists key attributes of that particular communication mode. Lightly shaded boxes indicate combinations that would work correctly with current protocols such as TCP, but for other reasons would not normally be used. Darkly shaded boxes indicate combinations that would not work correctly with current protocols such as TCP. Note that a single host may have many different conversations in progress at the same time, choosing for each of them the communication mode that is most appropriate.

the mobile host is not in any meaningful sense connected to the Internet at all.

In-IE/Out-DE. The mobile host can send outgoing packets directly to the correspondent host, while still encapsulating them to shield the home source address from examination by routers along the path. For this method to work, the correspondent host does not need to be fully mobile-aware, but it does need to be able to decapsulate encapsulated IP packets. Some operating systems, such as recent versions of Linux, have this capability built-in. However, automatic decapsulation is a feature that should be used with caution. Hosts that perform automatic decapsulation lose some degree of firewall protection — automatic decapsulation makes it easy to spoof packet source addresses — so automatic decapsulation should only be done on hosts that use strong authentication mechanisms instead of simply trusting the packet addresses.

In-IE/Out-DH. The most efficient choice for the mobile host is to send outgoing packets directly to the correspondent host, unencapsulated. This does not require any special capabilities on the part of the correspondent host, but some routers may discard such packets.

6.1.2 Row B (Mobile-Aware Correspondent Host)

If the correspondent host is mobile-aware and knows the mobile host’s current temporary care-of address, then choices from row B become available. The correspondent host can bypass the step of sending the packet to the home agent for encapsulation, and instead it can encapsulate the packet itself and send it directly to the mobile host.

The mobile host may choose to reply directly with an encapsulated packet (In-DE/Out-DE), or to avoid encapsulation overhead, it may choose to reply directly with an unencapsulated packet (In-DE/Out-DH). The first category (In-DE/Out-IE) is also valid, but is unlikely to be used. If the correspondent host is able to send packets directly to the mobile host, then the mobile host should also send its replies directly.

6.1.3 Row C (Both Hosts on Same Network Segment)

When the correspondent host and the mobile host are connected to the same network segment, routers need not be involved with the communication at all. The correspondent host can simply generate the IP packet, and then send it directly to the mobile host, even though naïve examination of the destination IP address would suggest that it does not belong on this network segment.

When a mobile host receives a packet this way, it should reply the same way, using (In-DH/Out-DH). The first two categories, (In-DH/Out-IE) and (In-DH/Out-DE), are also valid, but are unlikely to be used. If the correspondent host is able to send packets directly to the mobile host in a single link-layer hop, then the mobile host should reply the same way.

6.1.4 Row D (Forgoing Mobility Support)

We believe that the most important option, and the least emphasized in the current Mobile IP literature, is that mobile-aware applications should be able to specify when they *do not* want the services of Mobile IP. They should be allowed to send and receive normal, non-mobile IP packets. The last element of the table (In-DT/Out-DT) represents this option. In effect this is the way that most people currently connect their portable computers to the network when they visit some other institution. In the absence of Mobile IP, they have no other choice.

However, even when a host is capable of using Mobile IP, there are many cases where it might choose not to. Communicating directly incurs no Mobile IP overhead and can be used beneficially in some situations without requiring any mobile-awareness on the part of the correspondent host. Some applications may not need to have connections maintained when the mobile host moves, especially if connections are short, moves are rare, and the application has its own higher-level recovery mechanism. A simple example is viewing Web pages. HTTP connections are typically very short-lived, and if the connection is broken then the Web browser handles it by displaying a broken icon in place of the missing picture. The user can choose to either accept the broken icon, or to click the 'Reload' button to try again.

Our Mobile IP support software itself communicates using the temporary address when registering with the home agent. It has no choice, since until it has registered with the home agent the other Mobile IP delivery services are not available.

Another case where applications should be given the option to bypass Mobile IP services is when using IP multicast [RFC1112]. One of the goals of IP multicast is to reduce unnecessary replication of network traffic. Tunneling multicast packets from the home network to the visited network is therefore a little self-defeating. It would be better if the multicast application were able to join the multicast group through its real physical interface on the current local network, rather than through its virtual interface on its distant home network.

6.1.5 Inapplicable Combinations

The other entries in the fourth row and fourth column of the 4x4 table (shaded dark grey) are not especially useful. The choices in the fourth column denote cases where the mobile host sends packets using the temporary care-of address, not simply as the source address of an outer encapsulating header, but as the sole means of determining whence the packet originates. If the mobile host sends packets using only its temporary care-of address to identify their source, then the correspondent host would almost certainly reply to those packets using that same address. The networking software on the correspondent host would not be expected to have any way of even knowing that the host it's communicating with has other addresses.

The choices in the fourth row denote cases where the correspondent host sends packets addressed to the mobile host's temporary care-of address. If the mobile host receives packets addressed to its temporary care-of address, it ought to reply using that as its source address, or the correspondent host will have no way to associate the reply with the packet that caused it. For these reasons, the use of the temporary care-of address for communication in one direction effectively mandates the use of the same address for the corresponding return communication. Except in contrived circumstances, trying to mix temporary care-of addresses with permanent addresses as communication endpoints is not of any use.

7. Implementation

We have implemented our Mobile IP protocol in Linux. We override the IP route lookup routine and replace it with a routine that consults a mobility policy table before the usual route table. This allows us to control, on a packet by packet basis, whether a packet should use Mobile IP, and if so which interface to use. For the unencapsulated options, the interface is a physical interface. If the packet is to be encapsulated, then the routine directs IP to send the packet to our virtual interface, which encapsulates the packet and resubmits it to IP. This framework allows us to use all of the alternatives that we have described.

The choice of source IP address, and whether or not to encapsulate, needs to be made not only when sending a packet, but also at certain other times. For instance, this decision must also be made when TCP decides what address to use as the endpoint identifier for a TCP connection. Overriding the IP route lookup routine (instead of modifying the IP send packet routine) allows us to capture all of these crucial decision points automatically, without any extra special-case work.

Having provided the framework that allows us to control how packets are sent, we are now experimenting with various ways to make the actual decision about which method to use in each case. Below we describe the choices to be made by the mobile host, and the choices to be made by the correspondent host. We will be making our software freely available at http://mosquitonet.stanford.edu.

7.1 Mobile Host Choices

For the mobile host, there are two decisions to make. The first is whether to use the home address or the temporary address. If using the home address, then the second decision is which of the three home address methods to use.

7.1.1 Temporary Address or Home Address?

There are two ways to make the decision of whether to use the home address or the temporary address. One way is for a mobile-aware application to make the decision explicitly, and the other is for the host's networking software to make the decision based on heuristics.

In our Linux implementation, mobile-aware applications indicate their preferences to the networking software by binding their sockets to specific addresses. If the application binds its socket to the source address of (any of) the machine's physical interface(s), then the packets sent through that socket are sent directly through that interface using Out-DT, honoring the application's desired source address. If a socket is not bound to a particular address, or is bound to the host's permanent home address, then that is taken as an indication that the application is not mobile-aware, and our Mobile IP software should use its heuristics to decide which kind of source address to use. One of the heuristics we are experimenting with for TCP is to make the decision based on port numbers. For example, connections to port 80 are likely to be HTTP requests and can safely use Out-DT. Similarly, UDP packets addressed to UDP port 53 are likely to be DNS requests and can also safely use Out-DT.

7.1.2 Which Home Address Method to use

If the mobile host has decided to use its permanent home address, then it must decide which of the three home address methods to use. The mobile host keeps a cache of the currently selected delivery method associated with each target IP address. This saves it from having to make the decision afresh for every packet and allows it to build up a history, for each correspondent host, of which communication methods have proven to be successful and which have not.

One way the mobile host can choose which home address delivery method to use is to start with the most conservative (Out-IE), and then over the lifetime of the conversation tentatively try each of the more aggressive options (Out-DE and Out-DH), at each stage being prepared to return to the conservative method if the more aggressive method fails [Fox96]. Unfortunately, this can be wasteful, because in many cases either one or both of Out-DH and Out-DE will work fine, and having every conversation start out overly conservative is wasteful.

Another way for the mobile host to choose which home address delivery method to use is to start with the most aggressive (Out-DH). If this fails it can then try the more conservative options (Out-DE and then Out-IE) until one succeeds. Unfortunately, this can also be wasteful because in some easily identifiable circumstances, such as connecting to resources behind a protective gateway at the home institution, Out-DH is known to fail every time.

One solution to the question of which delivery method to start with is to allow the user, as part of the configuration of a Mobile IP machine, to specify rules stating which addresses Mobile IP should begin using in an optimistic mode and which addresses it should begin using in a pessimistic mode. These rules could be specified similarly to the way routing table entries are currently specified, as an address and a mask value. This would allow a single rule to identify, for example, the entire home network as a region where Out-IE should always be used.

In the discussion above, we tacitly assume that the IP layer has some way to tell whether delivery is 'succeeding' or 'failing', but in current operating systems this information is not readily available. This is not a new problem. The Ethernet Address Resolution Protocol (ARP) Specification [RFC826], written fourteen years ago, mentions the problem of stale ARP cache information. It suggests that when transport-level protocol software suspects that packets are not being delivered correctly it should indicate this to the lower layer software, but it also says that "implementation of these is outside the scope of this protocol." We propose that the required behavior could be obtained by a simple addition to the IP programming interface: all IP clients (e.g. TCP) could indicate, for every IP packet they send and receive, whether the packet is an 'original' packet or a retransmission. If the IP layer sees repeated retransmissions to a particular address, then this suggests that the currently selected delivery method may not be working. Similarly, if the IP layer sees repeated retransmissions *from* a particular address, then that suggests that acknowledgements are not getting through, which also indicates that the currently selected delivery method is not working. We have not yet implemented this.

7.2 Correspondent Host Choices

For the correspondent host, the choices are relatively simple. If the correspondent host is not mobile-aware then it will simply send normal IP packets, which means it is using the In-IE method. The same is true of a correspondent host that is mobile-aware, but is not yet aware that the host with which it is communicating is a mobile host.

If a mobile-aware correspondent host knows that the host with which it is communicating is a mobile host, and it knows the current care-of address, then it can encapsulate the packets and send them directly to that address. In this case it is using the In-DE method.

If the correspondent host knows that the mobile host is on the same Ethernet segment then it should also reply directly, using the In-DH method.

Finally, if the mobile host has chosen to initiate communication using its temporary care-of address, then the correspondent host, whether or not it is mobile-aware, will necessarily reply using that address, which means it is using the In-DT method.

8. Conclusions

One size does not fit all. Different situations call for different solutions, and our Mobile IP protocol gives mobile hosts the freedom to use the best solution for each situation. We are able to optimize for latency, packet size and Internet resource utilization. The best choice for each individual packet or conversation depends on what characteristics the protocol should optimize, the permissiveness of the networks over which the packets must travel, and the level of mobile-awareness of the hosts with which the mobile host corresponds.

Most communication does not need to use Mobile IP. We believe that all hosts should retain the ability to communicate using normal IP when that is appropriate. Mobile IP provides useful services, but these facilities should not be provided at the expense of losing the ability to operate as a normal Internet host.

Nevertheless, with the growing use of mobile computers and wireless networking, it is increasingly important that IP evolve to support mobile connections. Even though telnet connections may generate much less traffic than Web browsing, they are still important, and in a future world of ubiquitous mobile computing it is vital that long-lived connections be supported as well as short-lived communications.

9. Acknowledgements

We are grateful for the very helpful comments on this paper from Armando Fox, Hugh Holbrook, Nick McKeown, Venkat Padmanabhan, Craig Partridge, Charles Perkins, Elliot Poger, Xinhua Zhao, the anonymous SIGCOMM reviewers, and the anonymous student reviewers in Craig Partridge's Stanford CS341 networking class.

We are also very grateful to Jonathan Stone for his contributions to the ideas in this paper.

This work was supported by an NSF Faculty Career Development Award, a Robert N. Noyce Family Junior Faculty Chair, and the Center for Telecommunications at Stanford University.

10. References

[Bak96] Mary Baker, Xinhua Zhao, Stuart Cheshire & Jonathan Stone. Supporting Mobility in MosquitoNet. 1995 Winter USENIX, January 1996.

[Bla94] Trevor Blackwell et al. Secure Short-Cut Routing for Mobile IP. 1994 Summer USENIX, June 1994.

[Cer74] V.G. Cerf and R.E. Kahn. A Protocol for Packet Network Interconnection. IEEE Trans. on Communications, Vol 22, No. 5, May 1974, pp. 637-648.

[Gup96] Vipul Gupta and Abhijit Dixit. The Design and Deployment of a Mobility Supporting Network. To appear in the International Symposium on Parallel Architectures, Algorithms, and Networks, June 1996.

[Fox96] Armando Fox, Personal communication, April 1996.

[Hag93] R. Hager, A. Klemets, G. Maguire, M. Smith, F. Reichert. MINT - A Mobile Internet Router. 43rd IEEE Vehicular Technology Conference, New Jersey, USA, May 93.

[Ioa93] John Ioannidis and Gerald Q. Maguire Jr. The Design and Implementation of a Mobile Internetworking Architecture. 1993 Winter USENIX, January 1993.

[Joh96] David B. Johnson and Charles E. Perkins. Route Optimization in Mobile IP. draft-ietf-mobileip-optim-04.txt— work in progress, February 1996.

[Mon96] G. Montenegro. Bi-directional Tunneling for Mobile IP. draft-montenegro-tunneling-00.txt — work in progress, January 1996.

[Myl93] Andrew Myles and David Skellern. Comparing Four IP Based Mobile Host Protocols. Computer Networks and ISDN Systems, vol. 26, pp. 349-355, 1993. Also Proceedings of 4th Joint European Networking Conference, Trondheim, Norway, pp. 191-196, 10-13 May 1993.

[Par96] Craig Partridge, CS341, Stanford University, Spring 1996.

[Per94] Charles E. Perkins, Andrew Myles, and David B. Johnson. The Internet Mobile Host Protocol (IMHP). Proceedings of INET '94, June 1994.

[Per95] Charles E. Perkins. Minimal Encapsulation within IP. draft-ietf-mobileip-minenc-01.txt — work in progress, 25 October 1995.

[Per96a] Charles E. Perkins. IP Mobility Support. draft-ietf-mobileip-protocol-16.txt — work in progress, 22 April 1996.

[Per96b] Charles E. Perkins and David B. Johnson. Mobility Support in IPv6. draft-ietf-mobileip-ipv6-00.txt — work in progress, 26 January 1996.

[Per96c] Charles E. Perkins. IP Encapsulation within IP. draft-ietf-mobileip-ip4inip4-02.txt — work in progress, May 1996.

[RFC826] David C. Plummer. An Ethernet Address Resolution Protocol. RFC 826, November 1982.

[RFC974] Craig Partridge. Mail Routing and The Domain System. RFC 974, January 1986.

[RFC1027] Smoot Carl-Mitchell. Using ARP to Implement Transparent Subnet Gateways. RFC 1027, October 1987.

[RFC1034] P. Mockapetris. Domain Names — Concepts and Facilities. RFC 1034, November 1987.

[RFC1112] S. Deering. Host Extensions for IP Multicasting. RFC 1112, August 1989

[RFC1541] Ralph Droms. Dynamic Host Configuration Protocol. RFC 1541, October 1993.

[RFC1661] William Simpson. The Point-to-Point Protocol (PPP). RFC 1661, July 1994.

[RFC1702] Stan Hanks, Tony Li, Dino Farinacci and Paul Traina. Generic Routing Encapsulation over IPv4 networks. RFC 1702, October 1994.

[RFC1725] John G. Myers and Marshall T. Rose. Post Office Protocol - Version 3. RFC 1725, November 1994.

[RFC1883] Steve Deering and Bob Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 1883, December 1995.

[Sal84] J. H. Saltzer, D. P. Reed and D. D. Clark, End-To-End Arguments in System Design. ACM Transactions on Computer Systems, Vol.2, No.4, November 1984, pp. 277-288.

[Tas94] Mitchell Tasman. Protocols and Caching Strategies in Support of Internetwork Mobility. Ph.D Thesis, University of Wisconsin, October 1994.

[Ter94] Fumio Teraoka, Keisuke Uehara, Hideki Sunahara and Jun Murai. VIP: A Protocol Providing Host Mobility. Communications of the ACM, August 1994.