Mobile IP

Charles E. Perkins

Abstract— **Mobile IP has been designed within the IETF to serve the needs of the burgeoning population of mobile computer users who wish to connect to the Internet and maintain communications as they move from place to place. The basic protocol is described, with details given on the three major component protocols:** *Agent Advertisement*, *Registration*, **and** *Tunneling*. **Then, route optimization procedures are outlined, and further topics of current interest are described.**

Keywords—**Mobile IP, mobile networking, IETF, route optimization**

I. Introduction

Recent years have seen an explosive growth both in the number of laptop and notebook computers sold, and in the number of nodes connected to the Internet and the World-Wide Web. The notebook computers are themselves ever more powerful, equal in processing capability to many systems sold as desktop workstations. In fact, the future growth of the Internet is likely to be fueled in large part by these very notebook computers, since they account for the part of the computer market that is growing the fastest.

Along with these trends, we also see the steady growth of the market for wireless communications devices. Such devices can only have the effect of increasing the options for making connections to the global Internet. Mobile customers can find a wide array of such wireless devices available. There are numerous varieties of radio attachments and infrared devices; of course, communications by way of the cellular telephone network is always an option for those willing to pay the fees.

A. Mobility vs. Portability

These trends are motivating a great deal of interest in making sure that mobile, wireless computers can attach to the Internet and remain attached to the Internet even as they move from place to place, establish new links, and move away from previously established links. Early on, it was apparent that solving the problem at the network layer (say, by modifying IP [27], the Internet Protocol, itself) would provide major benefits, including application transparency and the possibility of seamless roaming. Application transparency is almost required for all reasonable solutions, because it is unacceptable to force mobile users to buy all new mobile-aware applications. Seamless roaming, while not yet mandatory, is nonetheless expected to register very highly on the scale of user convenience factors once the physical wireless means for continued connectivity are widely deployed. Moreover, seamless roaming provides application transparency. Mobile IP is the only current means for offering seamless roaming to mobile computers in the Internet. It has recently progressed along the ladder to standardization within the IETF, and its specification is now available as RFC2002 [23]. Related specifications are available as RFCs 2003-2006.

This paper follows the logical outline indicated below. We first describe the problem that is solved by mobile IP in the next section. In section II, there is a list of terminology, and an overview of mobile IP is presented. In section III, the discovery mechanisms of mobile IP are described in detail. Following that, in section IV, the mechanisms are described by which a mobile computer is located. Next, in section V, the available tunneling mechanisms are shown, which the home agent uses to forward datagrams from the home network to the mobile computer.

Having covered the details of the base mobile IP specification, we then describe in section VI further protocol messages which help to decrease the inefficiency associated with inserting the home agent in the routing path of data destined for mobile computers. This *route optimization* is still a topic for further work within the IETF. Finally, we summarize and discuss the current problems facing mobile IP, as well as a few areas of active protocol development.

II. Mobile IP Overview

Mobile IP can be thought of as the cooperation of three major subsystems. First, there is a discovery mechanism defined so that mobile computers can determine their new attachment points (new IP addresses) as they move from place to place within the Internet. Second, once the mobile computer knows the IP address at its new attachment point, it *registers* with an agent representing it at its *home network*. Lastly, mobile IP defines simple mechanisms to deliver datagrams to the mobile node when it is away from its home network.

A. Why isn't Mobility Simple?

Consider how IP addresses are used to today in the Internet. In the first place, they are primarily used to identify a particular end system. In this respect, IP addresses are often thought of as being semantically equivalent to DNS's *Fully Qualified Domain Names* (FQDNs). In other words, one can (conceptually) use either an IP address or a FQDN to identify one particular node out of the tens of millions of computer nodes making up the Internet. Popular transport protocols such as TCP [25] keep track of their internal session state between the communicating endpoints by using the IP address of the two endpoints, stored along with the demultiplexing selectors for each session, that is, the *port* numbers.

C. Perkins is with Sun Microsystems, Mountain View, CA. 94303
E-mail: cperkins@finesse.corp.sun.com

However, IP addresses are also used to find a route between the endpoints. The route does not have to be the same in both directions. Modelling the session as a bidirectional byte stream, the IP destination address for datagrams going in one direction would be the same as the IP source address for datagrams going in the opposite direction. Typically, the route selected for a datagram depends only on the IP destination address, and not (for example) on the IP source address, time of day, or length of the payload. The only other factor usually influencing route selection is the current state of network congestion. In other words, a route that might usually be selected by an intermediate router for a particular destination may go out of favor if traffic along that direction is delayed or dropped because of congestion.

Putting these two uses together results in a situation fraught with contradiction for mobile computing. On the one hand, a mobile computer needs to have a stable IP address in order that it might be stably identifiable to other Internet computers. On the other hand, if the address is stable, then the routing to the mobile computer is stable, and the datagrams always go to essentially the same place – thus, no mobility. Mobile IP extends IP by allowing the mobile computer to effectively utilize two IP addresses – one for identification, the other for routing.

Some attempts have been made to manage the movement of Internet computers by less functional methods. For starters, it is certainly possible, given sufficient deployment of DHCP [1], [7], for a mobile node to get an IP address at every new point of attachment. This will work fine, until the mobile node moves somewhere else. Then the old address will no longer be of use, and the node will have to get another one. Unfortunately, this approach usually also means that every established IP client on the mobile node will stop working. So, the mobile node will have to restart its Internet subsystems. Many users will not be so selective, and just reboot their system. This isn't so bad if each new point of attachment is separated by some time during which the system is disconnected or turned off anyway. Many mobile computer users are satisfied with just that mode of operation, which we'll describe as *portability*.

Even with portable operation, however, there are other big difficulties. Most applications initially identify an Internet node by means of its FQDN, but subsequently make use only of the node's IP address. In order to contact the node, the application consults the appropriate DNS server to get an IP address. If the IP address is allocated dynamically, either the server will have it wrong, or the server will need to get updates (say, from the portable Internet node). Since DNS is typically at the administrative heart at most networked enterprises which are using the Internet, any protocols designed to alter the data are going to have to be extremely well designed, implemented, and administered. The more often that updates are applied to DNS records [32], and the more platforms that are involved in hosting the update protocol implementation, the more likely it is that things are going to go haywire in a big, expensive meltdown. At minimum, one can be confident that a lot more work is going to be necessary before system administrators learn to trust that thousands (or millions!) of mobile nodes can reliably reach into the guts of their enterprise operations and tweak a record or two here and there. Much of this work will involve precisely carrying out certain cryptographic techniques that are only now being being standardized for use with DNS [8].

B. Terminology

Before getting into more details, it's a good idea to frame the discussion by setting some terminology, adapted from the mobile IP specification [23]. Mobile IP introduces the following new functional entities:

- *Mobile Node* A host or router that changes its point of attachment from one network or subnetwork to another, without changing its IP address. A mobile node can continue to communicate with other Internet nodes at any location using its (constant) IP address.
- *Home Agent* A router on a mobile node's home network which delivers datagrams to departed mobile nodes, and maintains current location information for each.
- *Foreign Agent* A router on a mobile node's visited network which cooperates with the home agent to complete the delivery of datagrams to the mobile node while it is away from home.

A mobile node has a home address, which is a long-term IP address on its home network. When away from its home network, a *care-of address* is associated with the mobile node and reflects the mobile node's current point of attachment. The mobile node uses its home address as the source address of all IP datagrams that it sends, except where otherwise required for certain registration request datagrams (e.g., as in section IV-A).

The following terms are frequently used in connection with mobile IP:

Agent Advertisement
: Foreign agents advertise their presence by using a special message, which is constructed by attaching a special extension to a router advertisement [6], as described in the next section.

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. There are 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 (as described in section IV-D), a secret (a shared key, or appropriate public/private key pair), and a style of replay protection in use.

Node A host or a router.

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

Security Parameters Index (SPI)
An index identifying a security context between a pair of nodes among the contexts available in the *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 agent, which decapsulates the datagram and then forwards it along the way to its ultimate destination.

Virtual Network
A network with no physical instantiation beyond its 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.

C. Protocol Overview

Mobile IP is a way of performing three related functions:

Agent Discovery
Mobility agents advertise their availability on each link for which they provide service.

Registration
When the mobile node is away from home, it *registers* its care-of address with its home agent.

Tunneling
In order for datagrams to be delivered to the mobile node when it is away from home, the home agent has to *tunnel* the datagrams to the care-of address.

The following will give a rough outline of operation of the mobile IP protocol, making use of the abovementioned operations. Figure 1 may be used to help envision the roles played by the entities.

- Mobility agents make themselves known by sending agent advertisement messages. An impatient mobile node may optionally solicit an agent advertisement message.
- After receiving an agent advertisement, a mobile node determines whether it is on its home network or a foreign network. A mobile node basically works like any other node on its home network when it is at home.
- When a mobile node moves away from its home network, it obtains a care-of address on the foreign network, for instance by soliciting or listening for agent advertisements, or contacting DHCP or PPP.
- While away from home, the mobile node registers each new care-of address with its home agent, possibly by way of a foreign agent.
- Datagrams sent to the mobile node's home address are intercepted by its home agent, *tunneled* by its home agent to the care-of address, received at the tunnel endpoint (either at a foreign agent or at the mobile node itself), and finally delivered to the mobile node.
- In the reverse direction, datagrams sent by the mobile node are generally delivered to their destination using standard IP routing mechanisms, not necessarily passing through the home agent (but see section VIII-B).

When the home agent tunnels a datagram to the care-of address, the inner IP header destination (i.e., the mobile node's home address) is effectively shielded from intervening routers between its home network and its current location. At the care-of address, the original datagram exits from the tunnel and is delivered to the mobile node.

It is the job of every home agent to attract and intercept datagrams that are destined to the home address of any of its registered mobile nodes. The home agent basically does this by using a minor variation on proxy ARP citerfc925, and to do so in the natural model it has to have a network interface on the link indicated by the mobile node's home address. However, the latter requirement is not part of the mobile IP specification. When foreign agents are in use, similarly, the natural model of operation suggests that the mobile node be able to establish a link its foreign agent. Other configurations are possible, however, using protocol operations not defined by (and invisible to) mo-

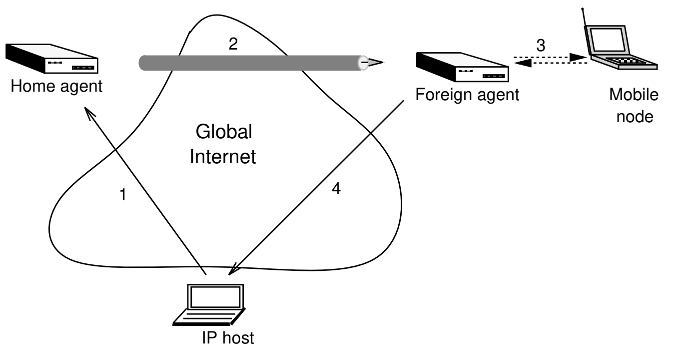

Fig. 1. Mobile IP Datagram Flow

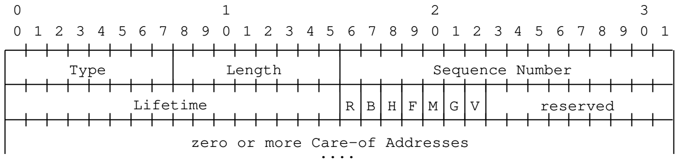

Fig. 2. Mobility Agent Extension Format

bile IP. Notice that, if the home agent is the only router advertising reachability to the home network, but there is no physical link instantiating the home network, then all datagrams transmitted to mobile nodes addressed on that home network will naturally reach the home agent without any special link operations.

Figure 1 illustrates the routing of datagrams to and from a mobile node away from home, once the mobile node has registered with its home agent. The mobile node is presumed to be using a care-of address provided by the foreign agent:

1. Datagram to mobile node arrives on home network via standard IP routing.
2. Datagram is intercepted by home agent and is *tunneled* to the care-of address, as depicted by showing an arrow going through the tube.
3. Datagram is detunneled and delivered to the mobile node.
4. For datagrams sent by the mobile node, standard IP routing delivers each to its destination. In the figure, the foreign agent is the mobile node's default router.

Now, we will go into more details about the various parts of the protocols outlined above.

III. Mobile Agent Discovery

The process of detecting a mobility agent is quite similar to that used by Internet nodes detect routers running ICMP Router Discovery (RFC 1256) [6]. The basic operation involves periodic broadcasts of advertisements by the routers onto their directly attached subnetworks. Noticing the similarity, the mobile IP working group decided to use RFC 1256 directly, and support the special additional needs of mobility agents by attaching special extensions to the standard ICMP [26] messages.

A. Agent Advertisement

By far the most important extension is the *mobility agent* extension, which is applied to ICMP Router Advertisement, and is illustrated in Figure 2.

The flags ('R', 'B', 'H', 'F', 'M', 'G', and 'V') inform mobile nodes regarding special features of the advertisement, and are described below. The type field allows mobile nodes to distinguish between the various kinds of extensions which may be applied by mobility agents to the ICMP Router Advertisements; the type for the mobility agent advertisement extension is 3. Other extensions may, of course, precede or succeed this extension; almost no other extensions are defined as of this writing. The length field is the length of this single extension, which really only depends on how many care-of addresses are being advertised. Furthermore, currently at most one care-of address will typically be advertised (but see section VIII-D). Home agents do not have to advertise care-of addresses, but they still need to broadcast mobility agent advertisements so that mobile nodes will know when they have returned to their home network. Indeed, mobility agents can advertise care-of addresses even when they do not offer any default router addresses as would be found in other ICMP Router Advertisements. No preferences apply to advertised care-of addresses.

The flags are defined as follows:

R Registration required. Registration with this foreign agent (or another foreign agent on this link) is required, even if using a co-located care-of address.
B The foreign agent is busy.
H The agent is a home agent.
F The agent is a foreign agent.
M Minimal encapsulation (RFC 2004 [20]).
G GRE encapsulation (RFC 1701 [11]).
V Van Jacobson header compression (RFC 1144 [13]).

Note that bits 'F' and 'H' are *not* mutually exclusive, and that 'B' cannot be set unless 'F' is also set. Note also that a foreign agent typically needs to continue sending advertisements out (with the 'B' bit set) even though it is too busy to provide service to new mobile nodes. Otherwise, the foreign agent's current customers might think the foreign agent had crashed, and move away unnecessarily.

The mobility agent generally increments the sequence number by one for each successive advertisement. Special rules enable a mobile node to distinguish between foreign agent crashes, and wraparound of the sequence number field.

B. Agent Solicitation

A mobile node is allowed to send ICMP Router Solicitation messages in order to elicit a mobility agent advertisement.

IV. Registration

Fig. 3. Mobile IP Registration Message Structure

There are two kinds of registration messages, the *registration request* and *registration reply*, both sent to UDP port 434. The overall data structure of the registration messages is shown in figure 3. The request message allows the mobile node to inform its home agent of its current care-of address, tells the home agent how long the mobile node wants to use the care-of address, and indicates special features that may be available from the foreign agent. The foreign agent is considered a passive agent in the registration procedure, and agrees to pass the request to the home agent, and subsequently to pass the reply from the home agent back to the mobile node.

A. Registration Request

The registration process is almost the same whether the mobile node has obtained its care-of address from a foreign agent, or alternatively has acquired it from another independent service such as DHCP. In the former case, the mobile node basically sends the request (with fields filled in as described below) to the foreign agent, which then relays the request to the home agent. In the latter case, the mobile node sends its request directly to the home agent, using its *co-located* care-of address as the source IP address of the request.

After the IP and UDP headers, the registration request has the structure illustrated in Figure 4.

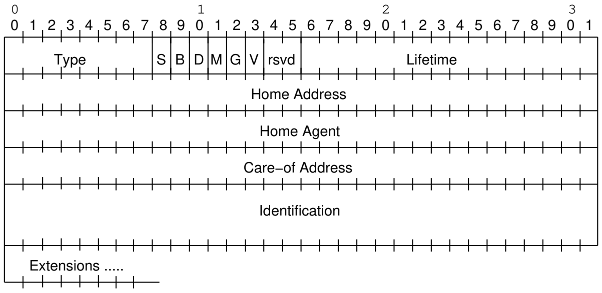

Fig. 4. Registration Request Format

Given the discussion about the bit fields in the agent advertisement extension in section III-A, the need for most of the fields is clear. The 'V' bit in the request serves to inform the foreign agent whether Van Jacobson compression is desired. The 'M' and 'G' bits tell the home agent which additional encapsulation methods can be used. The 'B' bit is used to tell the home agent to encapsulate broadcast datagrams from the home network for delivery to the care-of address (and from there to the mobile node). The 'D' bit describes whether or not the mobile node is co-located with its care-of address, and is mainly useful for determining how to deliver broadcast and multicast datagrams to the mobile node.

Also included are the home address and the proposed care-of address. The identification field is used for replay protection as described below when security is discussed. The most important extension is the *mobile-home* authentication extension, described in section IV-D, which is required in every registration in order to allow the home agent to prevent fraudulent *remote redirects*.

B. Registration Reply

The registration reply has the structure illustrated in Figure 5.

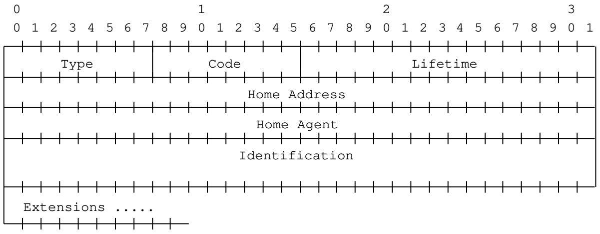

Fig. 5. Registration Reply Format

The lifetime field tells the mobile node how long the registration will be honored by the home agent. It can be shorter than requested, but never longer. The code field describes the status of the registration. If the registration succeeds, well and good. If the registration fails, the code field offers details about what went wrong.

Typical values include:

0 registration accepted

Registration denied by the foreign agent:

66 insufficient resources
69 lifetime request > advertised limit
70 poorly formed request
71 poorly formed reply
88 home agent unreachable

Registration denied by the home agent:

130 insufficient resources
131 mobile node failed authentication
133 registration identification mismatch
134 poorly formed request
136 unknown home agent address

Receiving code number 133 usually indicates the need for resynchronization between the home agent and the mobile node. This synchronization can either be time-based, or based on the exchange of randomly generated *nonce* values. Note that error code number 130 should be, effectively, impossible. The home agent should not be configured to accept the mobile node if it does not have the needed resources.

Up-to-date values of the code field are specified in the most recent *Assigned Numbers* (e.g., [28]).

C. Dynamic Home Agent Discovery

Rejection code 136 forms the basis for allowing the mobile node to find the address of a home agent when needed. If the registration reply is addresed to the *directed broadcast* address, then every home agent on the home network should receive it, and reject it. However, the registration reply containing the rejection also contains the home agent's address, so that the mobile node can try again and succeed.

D. Securing the Registration Procedure

Registration in mobile IP must be made secure so that fraudulent registrations can be detected and rejected. Otherwise, any malicious user in the Internet could disrupt communications between the home agent and the mobile node, by the simple expedient of supplying a registration request containing a bogus care-of address (perhaps the IP address of the malicious user). This would effectively disrupt all traffic destined for the mobile node.

The method specified to protect against such malicious users involves the inclusion of an unforgeable value along with the registration that changes for every new registration. In order to make each one different, a timestamp or newly generated random number (a *nonce*) is inserted into the *identification* field. The home agent and the mobile node have to agree on reasonable values for the timestamp or the nonce, and the protocol allows for resynchronization, as described earlier, by use of reply code 133.

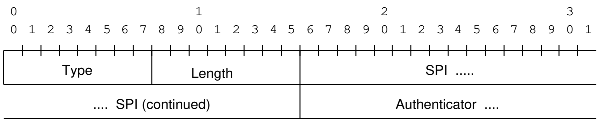

Fig. 6. Mobile IP Authentication Extensions

There are three authentication extensions defined for use with mobile IP, as follows:

- the mobile-home authentication extension,
- the mobile-foreign authentication extension, and
- the foreign-home authentication extension.

As illustrated in figure 6, they all have similar formats, distinguishable only by having different type numbers. The *mobile-home* authentication extension is required in all registration requests and replies. The Security Parameters Index (SPI) within any of the authentication extensions defines the security context which is used to compute (and to check) the authenticator. In particular, the SPI selects the authentication algorithm and modem, and secret (a shared key, or appropriate public/private key pair) used in computing the authenticator. A mobile node has to be able to associate arbitrary SPI values with any authentication algorithm and mode which it implements. SPI values 0 through 255 are reserved and not allowed to be used in any mobility security association.

The default authentication algorithm uses keyed-MD5 [29] in *prefix+suffix* mode to compute a 128-bit *message digest* of the registration message. The default authenticator is a 128-bit value computed as the MD5 checksum over the the following stream of bytes:

- the shared secret defined by the mobility security association between the nodes and by SPI value specified in the authentication extension, followed by
- the protected fields from the registration message, in the order specified above, followed by
- the shared secret again.

The authenticator itself and the UDP header are *not* included in the computation of the default authenticator value. All implementations of mobile IP are required to implement the default authentication algorithm just described.

V. Routing and Tunneling

The home agent, after a successful registration, will begin to attract datagrams destined for the mobile node and *tunnel* each one to the mobile node at its care-of address. The tunneling can be done by one of several encapsulation algorithms, but the default algorithm that must always be supported is simple IP-within-IP encapsulation, as described in RFC2003 [19]. Encapsulation is a very general technique, in use for many different reasons including multicast, multiprotocol operations, authentication, privacy, defeating traffic analysis, and general policy routing.

Pictorially, figure 7 shows how an IP datagram is encapsulated by preceding it with a new IP header (the *tunnel* header). In the case of mobile IP, the values of the fields in the new header are selected naturally, with the care-of address used as the destination IP address in the tunnel header. The encapsulating IP header indicates the presence of the encapsulated IP datagram by using the value 4 in the outer *protocol* field. The inner header is not modified except to decrement the TTL.

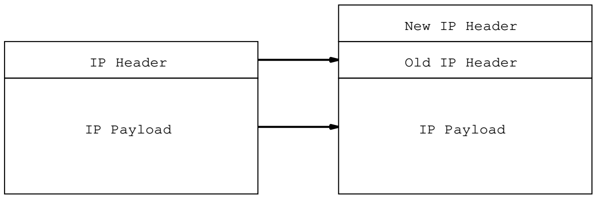

Fig. 7. IP-within-IP encapsulation

Alternatively, *minimal encapsulation* [20] can be used as long as the mobile node, home agent, and foreign agent (if present) all agree to do so. IP-within-IP uses a few more bytes per datagram compared to minimal encapsulation, but allows fragmentation at the home agent when needed to deal with tunnels with smaller path MTUs.

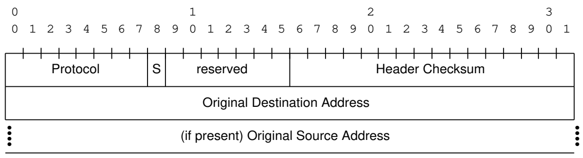

Fig. 8. Minimal Encapsulation Format

The minimal encapsulation header fits in the same relative location within the encapsulated payload, as indicated by the `Old IP header` in figure 7. The presence of the minimal encapsulation header is indicated by using protocol number 55 in the encapsulating IP header *protocol* field. Figure 8 shows the fields of the minimal encapsulation header, which are described below. The length of the minimal header is either 12 or 8, depending on whether the original source IP address is present.

Protocol
Copied from the Protocol field in the original IP header.

Original Source Address Present (S)
If one, the *Original Source Address* field (below) is present, otherwise it is not.

reserved Sent as zero; ignored on reception.

Header Checksum
The 16-bit one's complement of the one's complement sum of all 16-bit words in the minimal forwarding header. For purposes of computing the checksum, the value of the checksum field is 0. The IP header and IP payload (after the minimal forwarding header) are not included in this checksum computation.

Original Destination Address
Copied from the Destination Address field in the original IP header.

Original Source Address
Copied from the Source Address field in the original IP header. This field is present only if the Original Source Address Present (S) bit is set.

A. Soft Tunnel State

One unfortunate aspect of ICMP error messages is that they are only required by protocol to incorporate 8 bytes of the offending datagram. Therefore, when delivery of a datagram tunneled to a care-of address fails, the ICMP error returned to the home agent may not contain the IP address of the original source of the tunneled datagram.

Naturally, it makes sense for the home agent to try to notify the correspondent host (the source of the datagram which could not be delivered) in this situation. If the home agent keeps track of which datagrams have been tunneled to which care-of addresses (including the IP sequence number), then the ICMP error return can be used by the home agent to indicate which datagram caused the problem. If that determination is made, then the ICMP error return can be *relayed* by the home agent to the correspondent node which sent the offending datagram.

When a correspondent node sends the datagram to the home network, and the datagram did arrives at the home network, it seems inappropriate for the home agent to relay ICMP `network unreachable` messages without any change. In fact, from the point of view of the correspondent node, the tunnel *should* be invisible, almost as if it were an extension of the home link. So, when the home agent can determine which correspondent node should receive the error, it makes sense for the home agent to transform the `network unreachable` message into a `host unreachable` message.

When the home agent is about to tunnel a datagram to a care-of address which has just failed, it is quite feasible for the home agent to remember that the tunnel is broken. The home agent can then inform the correspondent host directly, using an ICMP `host unreachable` message. In fact, the home agent can keep track of other interesting tunnel parameters, especially including the path MTU for the tunnel, and the necessary TTL for encapsulated datagrams using that tunnel. This collection of tunnel parameters is called the *soft state* of the tunnel. The IP-within-IP encapsulation specification, RFC 2003 [19], recommends maintenance of soft state, and gives specific rules for relaying ICMP messages.

B. Home network configurations

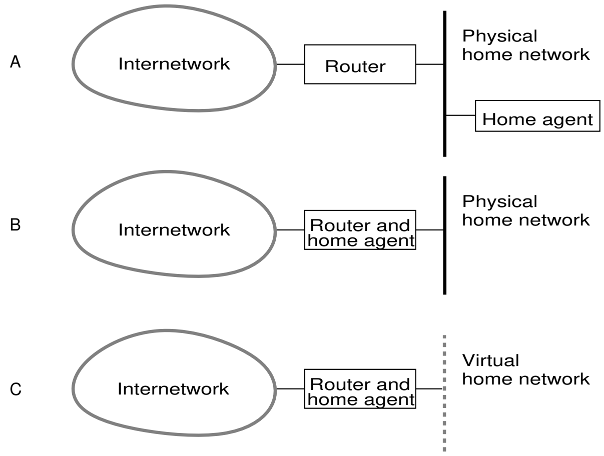

Fig. 9. Home Network Configurations

There are three basic configurations for home networks. The first is a standard physical network connected by way of a router, and with another node on the network acting as a home agent. The configuration shown in figure 9(a) will be very popular, especially for enterprises starting to use mobile IP. If the home agent is also an enterprise router, then the physical home network layout can be conceptually simpler as illustrated in part (b). In either case, wireless devices can be configured with IP addresses on existing physical (say, ethernet) networks with the help of bridging devices that cause the wireless packets to be bridged onto the physical network.

On the other extreme, it is possible to manage a home network that has no physical realization, as shown in part (c) called a *virtual network*. The home agent appears to the rest of the Internet as the router for the home network, but whenever datagrams arrive at the home agent, they are never forwarded. Instead, the home agent encapsulates them and sends them to a known care-of address.

C. Proxy and Gratuitous ARP

In either configuration (a) or (b) of Figure 9, the home agent has to perform proxy ARP for the mobile node. Otherwise, existing Internet hosts on the home network would not be able to contact the mobile node after it has moved to some new care-of address.

In fact, hosts remaining on the home network which communicate with the mobile node while it is at home are likely to have ARP [24] cache entries for the mobile node that become stale the instant the mobile node moves away. For this reason, the home agent is required to broadcast *gratuitous ARPs* as soon as the mobile node moves away from its home network and registers a new care-of address. The gratuitous ARPs are supposed to have the effect of updating the ARP caches of every node physically attached to the home network, so that they resolve the IP home address of the mobile node into the link-layer address of the home agent. Similarly, when the mobile node returns to its home network, it broadcasts gratuitous ARPs so that its home address is again associated to its own link-layer address by the other nodes on the home network. Networks on which nodes are attached, which do not work with gratuitous ARP, should not be administered as home networks.

Because of the danger of irreparably creating stale ARP caches, mobile nodes must *never* broadcast an ARP request or ARP reply packet on any visited network. If, for instance, a wireless mobile node broadcast an ARP request to find the link-layer address of the foreign agent broadcasting a care-of address, any other wireless stations within range could possible create ARP cache entries for that mobile node. Those entries would make it hard to contact the mobile node after it moves away.

VI. Route Optimization

As noted above, datagrams going to the mobile node have to travel through the home agent when the mobile node is away from home, but datagrams from the mobile node to other stationary Internet nodes can instead be routed directly to their destinations (see Figure 10). This asymmetric routing, called *triangle routing* is generally far from optimal, especially in cases when the correspondent node is very close to the mobile node.

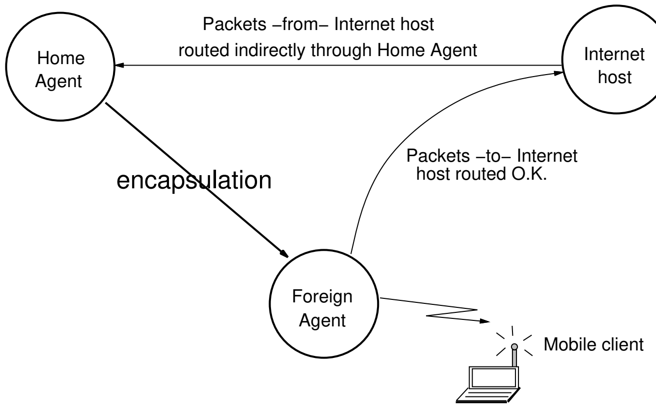

Fig. 10. Triangle routing

In this section, we will describe in some detail the necessary protocol operations (called *route optimization*) to eliminate the triangle routing problem. The current protocol definition may be found in the Internet draft [15], and there are additional details in an earlier paper on the subject [21]. The advantages of route optimization are clear. The disadvantage is that, for the first time, and in major distinction to the base mobile IP protocol, changes are required in the correspondent nodes.

A. Route Optimization Overview

The basic idea underlying route optimization is that the routes to mobile nodes from their correspondent nodes can be improved if the correspondent node has an up-to-date *mobility binding* (see section II-B) for the mobile node in its routing table. Most of the proposed protocol described below is geared towards providing such an updated mobility binding (usually shortened to just *binding*) to correspondent nodes that need them. With an updated binding, the correspondent node will be able to send encapsulated datagrams directly to the mobile node's care-of address instead of relying on a possibly distant home agent to do so.

Every aspect of the design is influenced by the need to allow the correspondent nodes to be sure of the authenticity of the updates. Mobile computer users would not be very satisfied if their traffic were easily hijacked, and their very mobility increases the likelihood that aspects of network security at their point of attachment may be inadequate. We also have to keep in mind that a majority of such nodes, today, will not be able to understand the protocol.

The current unsatisfactory state of security within the Internet, and especially the lack of key distribution protocols, has determined several further aspects of the design of the route optimization protocols. In particular, we believe that, for the near future while security protocols are still in the early stages of development and deployment, correspondent nodes are more likely to maintain security relationships with home agents than with individual mobile nodes. Observe that mobile nodes usually spend time connected to nodes either within their home domain, or near to their current point of attachment.

For instance, suppose an employee from one enterprise, say *Home Domains, Inc.* (company H), wishes to use mobile IP while roaming the premises of another enterprise, say *Fly Away With Us, Inc.*(company F). We expect that the employee would, first of all, make sure that the administrator of the home domain would set up a security association with the administrator of the foreign domain at company F. If the enterprises communicate frequently for business purposes (a likely circumstance given the employee's need to roam there), such a security association might already exist and be ready for use. Then, we further hope that any relevant correspondent node could get the necessary security association needed for communication with company H's home agent, perhaps by browsing an administrative panel and requesting the necessary information encrypted by its own local security transform.

Following this speculative model of the future, we have designed the protocol so that the home agent is responsible for providing binding updates to any concerned correspondent nodes at foreign enterprises. Briefly, the protocol operates in as many as four steps:

- a *binding warning* control message may be sent to the home agent, indicating a correspondent node that seems unaware of the mobile node's care-of address.
- the correspondent node may send a *binding request*
- the home agent (typically) may send an authenticated *binding update* containing the mobile node's current

care-of address

- for smooth handoffs (section VI-F), the mobile node transmits a binding update and has to be certain that the update was received. Thus, it can request a *binding acknowledgement* from the recipient.

In the next sections, a brief description of the above message types will be presented. Note that, particularly with the binding warning and binding update messages, the sending agent *must* be careful not to blindly send the messages without regard to past history. If the message has been sent recently, and seemingly has had no effect, the natural conclusion can be drawn that the intended recipient does not understand route optimization protocol messages. Therefore, the sender is obligated to send those messages less frequently in the future, or perhaps not at all. The protocol specifies a random exponential backoff mechanism for retransmitting these messages. Also note that all reserved fields are ignored on reception and must be set to zero upon transmission. Later, in section VI-G, is presented a brief description of the security architecture currently planned to make the above transactions secure. All messages are transmitted by way of UDP. As with the basic mobile IP protocol, there is no need for the additional features of TCP.

B. Binding Warning

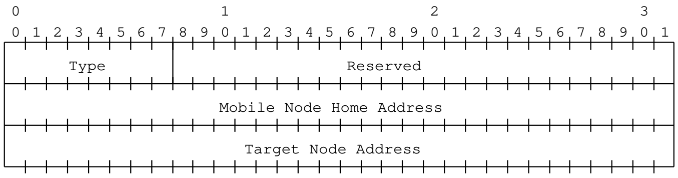

Fig. 11. Binding Warning Message Format

A *binding warning* message informs the recipient that the target node could benefit from obtaining a fresh binding for the mobile node. Usually, the recipient is the home agent, which is likely to be known to the sender because the sender obtained *its* binding from the home agent in the first place.

C. Binding Request

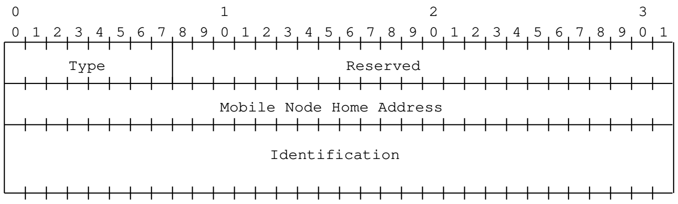

Fig. 12. Binding Request Message Format

Any time a correspondent node determines that its binding is stale, or is going stale, it can issue a *binding request* message to the home agent. The correspondent node sends a 64-bit number (the *identification*) to the home agent for use in protecting against replay attacks, and also to help match pending requests with subsequent updates.

D. Binding Updates

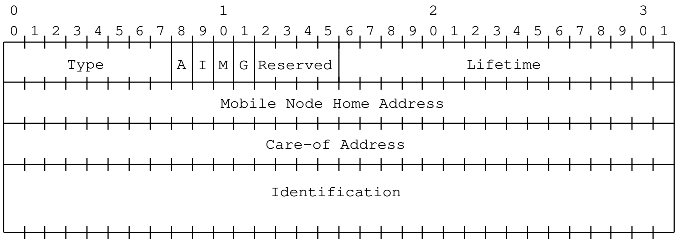

Fig. 13. Binding Update Message Format

The home agent (typically) sends a *binding update* message to those correspondent nodes that need them. This often happens because the home agent has received a datagram addressed to a mobile node from the correspondent node, which subsequently has to be tunneled by the home agent to the mobile node's current care-of address. If the home agent has a security relationship with the correspondent node, it can send a binding update straightaway without waiting for any binding warning or binding request. As with any binding, the binding included in the update must contain an associated lifetime, after which the binding is to be purged by the recipient.

Notice that the correspondent node may be willing to use minimal encapsulation or GRE to tunnel datagrams to the mobile node. The home agent sets the appropriate bits ('M' or 'G') to notify the correspondent node that the respective encapsulation protocols may be used if desired. The 'A' bit is used to request an acknowledgement, and the 'I' bit is set if the identification field is present. Cases involving *smooth handoff* require acknowledgements. On the other hand, the home agent usually finds out if the correspondent node has not gotten the update yet, just by the fact that it still has to encapsulate datagrams from that correspondent node sent to the mobile node.

The binding update must be accompanied by the *route optimization* authentication extension, similar to the *mobile-home* authentication extension.

E. Binding Acknowledgement

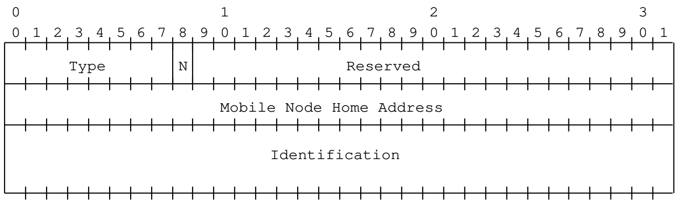

Fig. 14. Binding Acknowledgement Message Format

The *binding acknowledgement* message is used to acknowledge the reception of binding update messages. The 64-bit *identification* field, again, protects against replays and allow the acknowledgement to be associated with a pending binding update. The 'N' bit allows the recipient of the binding update to satisfy the 'A' bit of the binding update, while informing the updating agent that the update was not acceptable.

F. Smooth Handoffs

As mobile nodes move from one point of attachment to the next within the Internet, it would be nice if the transitions (called *handoffs*) were as smooth as possible. This could be a problem if datagrams heading towards one point of attachment were dropped because the mobile node had just left to attach somewhere else nearby. With route optimization, such problems will almost certainly arise, because there is no way that all correspondent nodes can instantaneously receive updated bindings reflecting the node's movement. Moreover, studies have shown [3] that because of the way TCP works, the distraction caused by dropping datagrams is magnified (by about a factor of two).

Thus, it is important to deliver datagrams correctly even though they may arrive at the "wrong" care-of address. Route optimization enables the solution to this problem, by allowing previous foreign agents to maintain a binding for their former mobile visitors, showing a current care-of address for each. With such information, a previous foreign agent can re-encapsulate a datagram with the right care-of address and send it along to the mobile node.

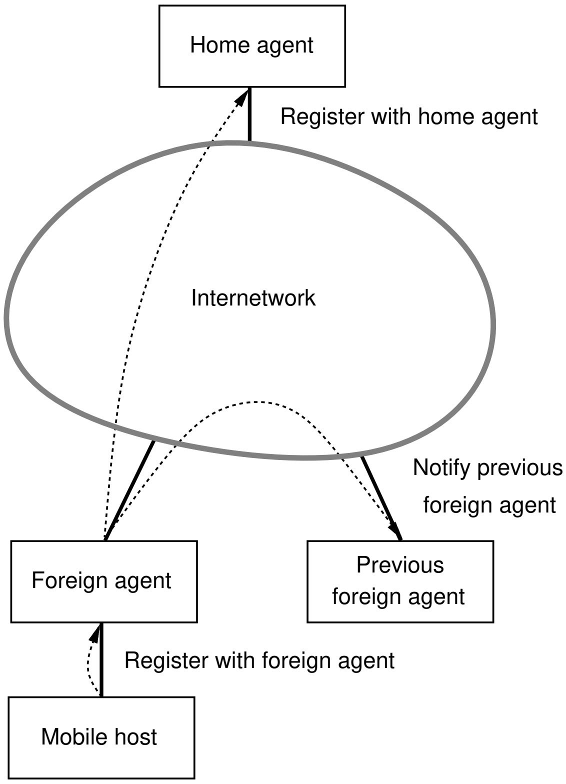

Fig. 15. Smooth Handoff during Registration

In order to obtain the maximum benefit from using route optimization to effect smooth handoffs from one foreign agent to the next, it would be best if the home agent were not involved. In fact, the handoff is targeted towards handling datagrams in flight without dropping them, but the home agent is often too far away to respond in time. If datagrams are being dropped for the hundreds of milliseconds that it would take for a distant home agent to respond, megabits of data could be dropped. Recognizing this problem, we have designed a method by which cooperating foreign agents can, by authority of the mobile node, agree to perform smooth handoffs before the new registration has completed. See figure 15 for an illustration of the process. Essentially, when the mobile node moves to a new point of attachment, it instructs its new foreign agent to send a binding update to its previous foreign agent.

If the previous foreign agent has no fresh binding for the mobile node, it can deliver the datagram to the home agent for further handling. This might conceivably be done the simple expedient of decapsulating the datagram and sending it out for normal IP routing. The datagram would then be routed to the home agent again. Such action, however, would probably cause routing loops whenever the home agent encapsulates datagrams for delivery to a foreign agent which has lost track of one of a visiting mobile nodes.

Instead, route optimization defines a way to use *special tunnels*, which indicate to the home agent the need for special handling. When a foreign agent wants to send a datagram back to the home agent (because the home address in the decapsulated datagram is not available), it instead encapsulates the datagram to be sent to the home agent. The newly encapsulated datagram uses the foreign agent's care-of address as the source IP address. Upon reception of the newly encapsulated datagram, the home agent compares the source IP address with the care-of address known in the binding created from the last registration. If the two addresses match, the home agent *must not* tunnel the datagram back to the care-of address. Otherwise, the home agent is allowed to re-tunnel the decapsulated result to the current care-of address known from the registration.

G. Securing the Binding Updates

Whenever a binding update is transmitted, it has to be accompanied by an authentication extension. However, doing so is more challenging in the case of smooth handoffs. It is important to note that, again, foreign agents are considered to be anonymous entities that are not trusted by the mobile node to do anything except follow protocol, and whose identity cannot necessarily be verified. The implication follows that the mobile node and foreign agent might share no special secret which can be used to build a security association. Even without a secret, however, the mobile node needs to persuade its previous foreign agent that the binding update (sent for the purpose of effecting a smooth handoff) has not been forged. The process of offering this persuasive evidence has been a challenging problem for designing the smooth handoff mechanism. The persuasive evidence possessed by the mobile node is called a *registration key*, and obtaining the registration key is accomplished by one of several means.

In the interest of keeping the description to an appropriate size, the precise details of managing security between the mobile node and foreign agent will be largely omitted. However, the overall procedure is as follows.

1. The foreign agent uses agent advertisement flags and extensions to provide information about the style of

security it is prepared to offer the mobile node.

2. The mobile node selects one of a menu of possible actions, depending upon availability.
3. The foreign agent responds to the mobile node's request, and if necessary cooperates with the mobile node to provide smooth handoff operation and to obtain a registration key from the home agent.

Our design of the smooth handoff procedure, using the binding update message as shown above in section VI-D, relies mostly on the mobile node to observe available methods and initiate their execution. The mobile node will know whether or not the foreign agent is willing to take part in the smooth handoff procedure by inspecting the advertised flags. In addition, the mobile node, when it first detects the foreign agent, will know immediately whether a mobility security association is available with that agent. In that case, the mobile node can establish a registration key by the simple expedient of picking a good random number and encoding it for the foreign agent, using their shared secret. Recall that, in this case, the registration has to include a *mobile-foreign authentication* extension.

However, the appropriate security association is a luxury not likely to be encountered, in our estimation. Therefore, the mobile node may instead rely on the home agent to pick out a registration key for use by the mobile node and foreign agent. This, again, can be done in one of two ways. If the foreign agent and home agent share a security association, then the foreign agent can request that the home agent encrypt a diligently selected registration key using that security association and transmit the result back to the foreign agent as part of the registration reply. The home agent informs the mobile node of the registration key value by using the mobility security association which is always known to exist between the two nodes.

If, on the other hand, the foreign agent does not have a security association with the home agent, but it *does* instead have a public key, it can send the public key to the home agent along with the registration, and accomplish much the same result as outlined in the last paragraph. Lastly, if the foreign agent does not have a public key, and has security associations with neither the home agent nor the mobile node, then there is still the possibility for a *Diffie-Hellman* key exchange [30].

Performing smooth handoffs is complicated by the need for creating a registration key in the absence of well-defined, standardized, and widely deployed security protocols. Nevertheless, it is hoped that the complication of the latter operation will not obscure the basic simplicity of the protocol, and that providing the protocol definition for each of a variety of feasible scenarios will broaden the appeal of smooth handoffs rather than clouding its future.

VII. IETF

In this section, we describe the pertinent details of the status of mobile IP in the standardization process, and describe interesting details about working groups and the standardization process itself.

The Internet Engineering Task Force (IETF) is a somewhat loose confederation of numerous (over 60, at last count) working groups that meets three times a year. At the three meetings, each working group may meet once or several times, or not at all. The working groups are divided into *areas*, each of which is administered by an *area director*. For instance, the mobile IP working group is part of the *routing* area. The area director for each area has to review the proposals from each working group before they can be submitted for further consideration by the IETF at large. The area directors, taken together, also constitute another group called the Internet Engineering Steering Group (IESG). The IESG, upon recommendation of the particular area director sponsoring a protocol document, tries to assure a high degree of protocol quality, and to assure that standardized protocols work well with each other. To put it mildly, this is a huge job, and getting bigger all the time with the growth of the Internet. Complicating an already complex problem is the fact that Internet protocols suddenly represent big business, and a false step on the part of an area director, or working group chair, could easily result in an expensive lawsuit.

The mobile IP working group itself has had a long and at times contentious history. A succession of eminently competent working group chairs has fortunately managed to bring the process to a somewhat successful milestone, with the recent publication of the base mobile IP protocol documents as proposed standards, and RFCs 2002-2006. A good place to look for such documents is on the IETF web page, `http://www.ietf.org`. After some further consensus has been achieved, and additional operational experience gained, mobile IP may progress to *draft standard*. This step should also be accompanied by a large increase in the number of deployed mobile IP systems in the Internet. For various reasons, mobile IP has not up until now enjoyed its full potential.

Route optimization, and the other protocol efforts described in the next section, are in a far more fluid state. These are still Internet drafts, not yet proposed standards.

VIII. Current Topics

A. IP version 6 (IPv6)

Although space does not permit a full exposition of the details of the proposed mobility protocols for IPv6, some overall discussion is certainly in order. The current Internet draft [14], and a recent paper on the subject [22], should be consulted for full details.

The IPv6 protocol [5], [12] and its attendant address configuration protocols (*Neighbor Discovery* [31] and *Stateless Address Autoconfiguration* [17] form an almost perfect protocol basis for mobile networking. The basic idea, that a mobile node is reachable by sending packets to its home network, and that the home agent sends packets from a home network to the mobile node's current care-of address, remains the same. Also, similarly to the method used before (for IPv4, as described in section V), the home agent encapsulates packets for delivery from the home network to the care-of address.

What has changed is that the mobile node now has an assured capability to obtain a care-of address by using the abovementioned address configuration protocols. Thus, there is a greatly reduced need for foreign agents, and they have been eliminated from the mobility support protocol. Moreover, the ideas from route optimization, of supplying binding updates to correspondent nodes, are able to be integrated nicely into IPv6 by using the newly defined *destination options*. Since destination options are inspected only by the destination, there is no performance penalty at intermediate routers for using them. Since such options can be placed into any IPv6 packet, there is far less overhead involved in sending binding updates to correspondent nodes. The binding update can be included in any normal data packet that the mobile node would be sending to the correspondent node anyway. If a packet ever arrives at the home network, it will be encapsulated and sent to the mobile node. Thus, when a mobile node receives such an encapsulated IPv6 packet, it can infer that the originator of the decapsulated packet should receive a binding update (in a destination option) sent along with the very next packet transmitted to the originator.

Just as with IPv4, binding updates need to be authenticated. What is different, however, is the expectation that every IPv6 node will be able to establish and maintain security relationships as needed. In order to comply with the IPv6 specification, each node is required to implement IPv6 *authentication header* [2] processing. Thus the mobile node can assume that, by using security protocols already specified, its binding updates will be confidently received by the correspondent nodes that need them. In IPv6, the mobile node is the only node authorized to supply binding updates to its correspondent nodes, and typically does so at the earliest reasonable time after moving to a new point of attachment to the IPv6 Internet.

B. Firewalls and Packet Filtering Problems

One of the biggest problems facing the deployment of mobile IP in today's Internet, is that mobile nodes roaming in foreign enterprises look like interlopers, and the firewalls and border routers administered at the foreign domain are usually configured to interrupt traffic to and from interloper nodes. This is a reaction to the growing danger of protocol attacks and the desire to eliminate as many as possible of the hiding places favored by malicious users.

So, for instance, a recent Internet draft [9] exhorts systems administrators to perform *ingress filtering*, by which is meant the action of disallowing datagrams entry into the Internet from any leaf domain, unless those datagrams conform to expectations about their source IP address. By doing so, the Internet is considered to be better protected from domains harboring malicious users, because users sending datagrams from the domain will not be able to impersonate users from the ingress-filtering domains.

This, of course, is anathema for mobile IP. Any mobile node in a foreign domain is going to have a source IP address which doesn't "look right" to such ingress-filtering border routers. One idea is to allow the mobile nodes to issue encapsulated datagrams using their care-of address as the outer source IP address. Note that using the care-of address as the source IP address of the original datagram is typically a losing proposition, since the correspondent node is keeping track of its sessions by way of the mobile node's home address, not its care-of address.

The downside of this encapsulation approach is that IPv4 correspondent nodes are unlikely to be able to decapsulate such datagrams. So, the mobile node has to find another likely target for the encapsulated datagrams, and there aren't many commonly available today. One possible target would be the mobile node's home agent, which is pretty much guaranteed to be able to perform decapsulation. Obviously, this introduces yet another inefficiency in the routing of datagrams from mobile nodes, and there is work actively in progress to try to find other solutions to this problem.

An associated difficulty is the problem of allowing the mobile node to send datagrams into its home domain. The border routers protecting the home domain are likely to disallow any datagrams which seem to have a source IP address belonging to an *internal* subnet of the home domain. This problem is probably amenable to solution, by way of some protocol which informs the (probably specialized) border routers about those source IP addresses which are allowed to externally originate datagrams into the home domain. It is also feasible for border routers to encapsulate such datagrams for delivery to an enterprise home agent [16], [10].

As a matter of administrative convenience, it is likely that the firewalls will be configured to allow all datagrams in as long as they are addressed to a home agent, protocol UDP, port 434. This will at least enable mobile IP to get the registrations in from the global Internet to the home agents. From the considerations in the previous paragraphs, it is also reasonable to expect that the local network administrator will demand a very high degree of reliability and code quality from the home agent.

C. Simultaneous Bindings

One feature of mobile IP which has not been stressed in this article is the use of multiple, simultaneous registrations. The base specification permits a mobile node to register more than one care-of address at the same time, and to deregister a specific one of the care-of addresses as necessary, by setting the 'S' bit in the *registration request* message. When there is more than one care-of address active for a mobile node, the home agent is instructed to send a duplicated encapsulated datagram to each care-of address. Presumably then the mobile node will receive the decapsulated result at each of the several care-of addresses.

This unusual behavior still does technically conform to router and host requirements for IP, because the IP specification allows for datagrams to be duplicated. There are times when such behavior is justified, for certain classes of links. Moreover, it is easier from a network-layer protocol standpoint not to require that network nodes enforce any policy ensuring that datagrams are not duplicated. Removing duplicates is typically done by transport or application layer protocols whenever it makes a difference. In the case of mobile IP, the original justification for simultaneous registrations was that many wireless links are error-prone, and it is certainly the case that receiving noisy signals from multiple sources can often allow a target to reconstruct the original signal more accurately.

Simultaneous registrations, while still holding promise for the improved handling of IP wireless connectivity, have not been available in any implementation known to the author. Thus, this optional feature should be considered as a future possible benefit. The unavailability of simultaneous registration is probably mostly due to the slow dissemination of wireless LAN technology into the marketplace, considering that wireless connectivity was the motivating factor for the inclusion of the feature in the first place.

D. Regionalized Registration

The concern has been raised that, for highly mobile computers, too much traffic between the visited and home networks would be generated by the registration process. Given the current state of the protocol, several counterarguments are able to be made against that objection:

- Unless route optimization is enabled, the normal traffic of encapsulated datagrams from the home agent will make the control traffic from the registration seem negligible.
- The mobile IP specification technically allows registrations to be issued no more often than once per second per mobile node. That should not present too much network traffic.

So, the problem of frequent registration is probably not terribly important until route optimization is more fully deployed. However, there are other factors that must be considered. First, with some diligent management of the local connectivity available to the mobile node and buffing of datagrams that have to be delivered, one can get some of the benefit of smooth handoffs without implementing route optimization in the foreign agents (for instance, see [4]).

In fact, it is also possible to have a collection of foreign agents joined together in a multicast group, and then subsequently to allow the mobile node to use the multicast IP address as its care-of address. In either case, work is necessary to cause each foreign agent to buffer each datagram, at least momentarily, in case the mobile node decides to depart the previous foreign agent from which the datagram was expected to be transmitted to the mobile node. Also, notably, any such approach requires new protocol to be operated by the foreign agents, and the schemes are really intended to only be used in a two-level hierarchy. It is an open question whether doing the buffering is better in conjunction with the abovementioned methods, or in conjunction with route optimization techniques.

Another alternative [18] establishes a hierarchy of foreign agents, and advertises multiple foreign agents in the agent advertisement. Then, registrations can be localized to the foreign agent which is the lowest common ancestor of the care-of addresses at the two points of attachment

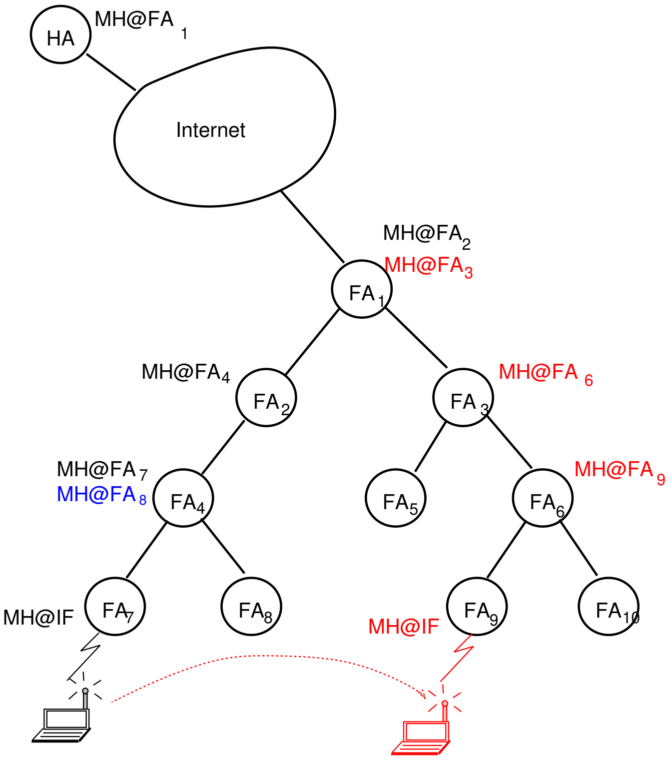

Fig. 16. Hierarchical Foreign Agents

of interest. To enable this, the mobile node has to figure out how high up in the tree its new registration has to go, and then arrange for the transmission of the registration to each level of the hierarchy between itself and the closest common ancestor between its new care-of address and its previous care-of address.

Consider the illustration in figure 16. While it was using services of foreign agent FA_7, the mobile node was receiving agent advertisements describing the hierarchical lineage $< FA_7, FA_4, FA_2, FA_1 >$, and had caused a registration, now specialized for this purpose, to be transmitted to each of those foreign agents as well as its home agent. Its home agent believes the mobile node is located at care-of address FA_1, foreign agent FA_1 believes the mobile node is located at foreign agent FA_2, and so on until foreign agent FA_7 actually knows the whereabouts of the mobile node. When the mobile node moves to foreign agent FA_8, it only has to cause the new hierarchical registration to propagate as far as FA_4. When the mobile node moves to foreign agent FA_9, it receives advertisements indicating the lineage $< FA_9, FA_6, FA_3, FA_1 >$. By comparing the previous and current lineage, the mobile node determines that it has to cause the registration to propagate up the hierarchy to FA_1, but the registration still does not have to reach the home agent. The home agent can, in this scenario, be considered to be the "ultimate" care-of address of the mobile node. Note also that, as a result of the differing views of the hierarchical agents about the mobile node's care-of address, the original datagram has to be relayed to a number of intermediate nodes in the hierarchy, each of which then is charged with the responsibility of retunneling the datagram if necessary to the next lower level of the hierarchy.

IX. Summary

In this article, we have explored most of the technical details of mobile IP, an extension to IP which allows mobile nodes to roam transparently from place to place within the Internet, usually with no discernible disruption of service. Mobile IP affects the routing of datagrams within the Internet, effectively by allowing the home agent to create a tunnel, using encapsulation, between the mobile node's home network and whatever care-of address happens to identify its current point of attachment. The advertisement and registration protocols are described in detail, and variations on the tunneling protocols shown.

Tunneling introduces additional routing links into the communication paths between mobile nodes and their correspondent nodes. This suboptimal routing can be cured, with the cooperation of the correspondent nodes, by allowing the dissemination of binding updates to each active correspondent, using the route optimization protocols. Binding updates allow the correspondents to tunnel datagrams directly to the mobile node's care-of address instead of relying on the home agent for this function. With virtually the same route optimization techniques, foreign agents can cooperate with the mobile node to effect smooth handoffs, being careful not to drop any datagrams even when the mobile node has moved away from the care-of address receiving the datagrams.

Mobile IP and route optimization both must be subject to rigid requirements for authentication of the claimed care-of addresses, because otherwise malicious hosts could disrupt or completely usurp communications with the mobile node. These new requirements have fostered the inclusion of simple yet relatively new techniques into these protocols, to assure that the care-of address information has been sent by an authorized entity.

Aspects of the standardization process within the IETF, which have had a major impact on the development of mobile IP, have been described. Finally, we describe some areas of current and supplemental interest related to mobile IP. The problems facing mobile IP in the realm of secure enterprise computing are detailed, especially regarding ingress filtering and firewalls. Mobility support for IP version 6 is outlined in its gross aspect. The possible future benefits of simultaneous registrations are briefly explained, and several ways to localize registration requests are described.

X. Final Words

We hope this brief introduction to mobile IP will engender interest in the solution to the remaining problems which continue to challenge the deployment of the protocol, particularly in the areas involving existing enterprise security facilities using firewalls and recent packet filtering techniques. Participation on the mobile IP mailing list is encouraged; the mailing list can be joined by sending mail to majordomo@Smallworks.COM, including the line 'subscribe mobile-ip' in the body of the message. One can keep up with general events within the IETF, by selecting the appropriate links on the Web page `http://www.ietf.org`. The author will also gladly answer electronic mail sent to `cperkins@finesse.corp.sun.com`. Acknowledgement is due to Vipul Gupta, without whom this paper could never have been finished even in the time it took to do so, and to the many people who have contributed greatly to the effort of producing and improving the mobile IP specifications.

References

[1] S. Alexander and R. Droms. DHCP Options and BOOTP Vendor Extensions. RFC 1533, October 1993.

[2] R. Atkinson. IP Authentication Header. RFC 1826, August 1995.

[3] Ramon Caceres and Liviu Iftode. Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments. *IEEE Journal on Selected Areas in Communications*, 13(5):850–857, June 1995.

[4] Ramon Caceres and Venkata Padmanabhan. Fast and Scalable Handoffs for Wireless Networks. In *ACM Mobicom 96*, November 1996.

[5] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 1883, December 1995.

[6] Stephen E. Deering, editor. ICMP Router Discovery Messages. RFC 1256, September 1991.

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

[8] Donald E. Eastlake and Charles W. Kaufman. Domain Name System Protocol Security Extensions. draft-ietf-dnssec-secext-09.txt, January 1996. (work in progress).

[9] P. Ferguson. Ingress Filtering in the Internet. draft-ferguson-ingress-filtering-01.txt, November 1996. (work in progress).

[10] V. Gupta and S. Glass. Firewall Traversal for Mobile IP: Goals and Requirements. draft-ietf-mobileip-ft-req-00.txt, January 1997. (work in progress).

[11] Stan Hanks, Tony Li, Dino Farinacci, and Paul Traina. Generic Routing Encapsulation (GRE). RFC 1701, October 1994.

[12] R. Hinden and S. Deering. IP Version 6 Addressing Architecture. RFC 1884, December 1995.

[13] Van Jacobson. Compressing TCP/IP Headers for Low-Speed Serial Links. RFC 1144, February 1990.

[14] D. Johnson and C. Perkins. Mobility Support in IPv6. draft-ietf-mobileip-ipv6-03.txt, November 1996. (work in progress).

[15] David B. Johnson and Charles E. Perkins. Route Optimization in Mobile-IP. draft-ietf-mobileip-optim-05.txt, November 1996. (work in progress).

[16] G. Montenegro. Reverse Tunneling for Mobile IP. draft-ietf-mobileip-tunnel-reverse-00.txt, January 1997. (work in progress).

[17] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP version 6 (IPv6). RFC 1970, August 1996.

[18] C. Perkins. Mobile-IP Local Registration with Hierarchical Foreign Agents. draft-perkins-mobileip-hierfa-00.txt, February 1996. (work in progress).

[19] Charles Perkins. IP Encapsulation within IP. RFC 2003, May 1996.

[20] Charles Perkins. Minimal Encapsulation within IP. RFC 2004, May 1996.

[21] Charles Perkins, Andrew Myles, and David Johnson. IMHP: A Mobile Host Protocol for the Internet. *Computer Networks and ISDN Systems*, 27(3):479–491, December 1994.

[22] Charles E. Perkins and David B. Johnson. Mobility Support in IPv6. In *ACM Mobicom 96*, November 1996.

[23] C. Perkins, Editor. IPv4 Mobility Support. RFC 2002, October 1996.

[24] David C. Plummer. An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Addresses for Transmission on Ethernet Hardware. RFC 826, November 1982.

[25] J. B. Postel, editor. Transmission Control Protocol. RFC 793, September 1981.

[26] J. B. Postel, Editor. Internet Control Message Protocol. RFC 792, September 1981.

[27] J. B. Postel, Editor. Internet Protocol. RFC 791, September 1981.

[28] Joyce K. Reynolds and Jon Postel. Assigned Numbers. RFC 1700, October 1994.

[29] Ronald L. Rivest. The MD5 Message-Digest Algorithm. RFC 1321, April 1992.

[30] Bruce Schneier. *Applied Cryptography*. John Wiley and Sons, 1993.

[31] S. Thomson and T. Narten. IPv6 stateless address autoconfiguration. RFC 1971, August 1996.

[32] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound. Dynamic Updates in the Domain Name System (DNS). draft-ietf-dnsind-dynDNS-11.txt, November 1996. (work in progress).