INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst., **11**, 3–20 (1998)

MOBILE IP

c. e. perkins*
Sun Microsystems, Mountain View, CA 94303, USA

SUMMARY

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 *Tunnelling*. Route optimization procedures are then outlined and further topics of current interest are described. © 1998 John Wiley & Sons, Ltd.

key words: Mobile IP; mobile networking; IETF; route optimization; IPv6; firewalls; ingress filtering; buffered hand-off

1. INTRODUCTION

Recent years continue to verify the projections of growth in the laptop and mobile computer market. The latest developments in Internet applications and development languages such as Java serve to reinforce this trend, by allowing mobile clients ever fuller access to servers and services located arbitrarily in the Internet, which is itself growing beyond even the optimistic projections of the past.

1.1. *Ubiquitous wireless communications*

Naturally, this tremendous market growth brings along with it a big incentive to improve communications paths between the individual computers and the Internet. One big improvement lies in the changing nature of the communication link itself. Wireless communications devices are becoming available in a wide (and confusing) variety of products. Radio links (especially telephone) and infrared links seem to be among the most popular, but satellite systems are quickly making their entrance into the marketplace. Such devices offer the promise of allowing users to be connected to the global Internet any time from anywhere. This promise still lies far in the future, however, especially for most people unwilling or unable to pay connect charges to cellular telephone companies.

Nevertheless, it is expected that wireless infrastructures will indeed become ubiquitous. In fact, it is quite possible that the combination of various technologies at different frequencies and ranges will provide tens of megabytes per second of wireless data transmission capacity (bandwidth) for every Internet user.

This combination of the huge growth of the information infrastructure within the Internet, the growth of the mobile computing market, and the still-projected growth of the wireless communications market will inexorably lead to a new paradigm for mobile computing. Embedded mobile computer systems seem particularly able to benefit from mobile networking protocols, as will become clear when the protocols are discussed. The user convenience of always having a responsive, low-cost connection, without needing to dial in, seems new, inviting and certain to inspire applications still unimaginable in the current immature state of the Internet.

1.2. *Application transparency and seamless roaming*

New applications will become available that are particularly suited for use on mobile wireless computers. Even so, users will want to use other applications which operate in the same way whether or not the host computer is in motion. For instance, the answers retrieved from a Web search are unlikely to depend upon whether the computer issuing the search is moving. This application transparency is important for the acceptance of mobile computing.

Part of application transparency is the ability for users to move from one wireless area to the next without being required to change the operation of the application or to reconfigure the computer. This will become especially important as each wireless area (cell) decreases in size. Cell size is one factor which determines the maximum number of users that can utilize the wireless infrastructure in a particular region, since users accessing the same channel within a cell will interfere with each other. If multiple users must use the same channel, then some link-layer protocol must be devised to effectively limit the availability of the channel so that each user only has a fraction of the total capacity available. If the cell size can be made smaller, more cells will fit in the region and each user can utilize a greater proportion of the total bandwidth of the cell.

Correspondence to: C. Perkins, Sun Microsystems, Mountain View, CA 94303, USA. Email: cperkins@eng.sun.com

CCC 1074–5351/98/010003–18$17.50

This trend towards smaller cell sizes will emphasize the need for seamless roaming. With huge cell sizes, as in today's cellular telephone architecture, or with satellite communications, roaming (which requires *hand-off*) from one cell to the next can be a relatively infrequent event. With small cell sizes, perhaps the size of an average office area or less, hand-offs will occur more frequently, so users would not tolerate such inconveniences as making new connections on every cell switch. Small cell sizes also have the effect of minimizing battery requirements, which is of great importance to mobile users.

Thus providing application transparency and hiding the effects of mobility and of hand-off from one cell to another will become increasingly important. Most people today are already accustomed to accepting the above-mentioned inconveniences. For one thing, given today's products, there is little choice but accept them. For another, typical cell sizes are large and, moreover, typical wireless access is by way of dialling into a cellular provider's telephone network. Since such dial-in procedures are quite time-consuming, no one expects instant hand-offs. When cell sizes decrease and the computing public recognizes the existence of more convenient alternatives, today's procedures will no longer be acceptable.

1.3. *Portability versus mobility*

Portability, provided for instance by today's telephone access methods for mobile computers, enables a user to establish a link to the Internet upon demand from various points of attachment. True mobility, on the other hand, allows one to continue use across different points of attachment to the Internet and includes portability as a special case. Mobile IP provides mobility and thus portability; other solutions[1] provide only portability. Cellular telephone connections provide mobility to the extent that the telephone connection can be handed off from one base station to the next; from the network (IP) protocol layer the mobile node always appears to be at the same phone number and thus at the same point of attachment to the Internet.

Mobile IP, likewise, makes it appear (to non-mobile-aware nodes) that the mobile node always resides at the same point of attachment to the Internet, called the *home network*. Network-layer entities are responsible for presenting this fiction to the rest of the Internet. The same mechanism works with almost all varieties of communications media. In fact, Mobile IP works just as well with wired media as with wireless media. One could employ Mobile IP to move from one Ethernet to the next; in this case, however, smooth hand-off mechanisms are unlikely to be relevant. *Route optimization* (described in Section 7) offers improvements to basic Mobile IP which can be used for smoother hand-offs from one point of attachment to the next, for nodes that are able to process information about the movement of the mobile nodes.

There are technological barriers, solved by Mobile IP, that prevent the realization of the vision just sketched. Even if the physical wireless links were ubiquitously available to a moving computer, the realities of today's networking infrastructure imply that as long as the computer maintains an established connection, it must keep the same IP address for the connection. For instance, TCP[2] defines a connection as a quadruple containing the IP addresses and the port numbers of the two endpoints; changing any of these causes the connection to be lost. On the other hand, as long as datagrams have the same destination IP address, the datagrams will be routed to the same network. Since mobile nodes move from one network to another, the implication is that an established connection will be broken as soon as the mobile node moves to a new network. This certainly does not meet the criterion of seamless roaming; in fact, one might say that it is seamy roaming.

2. PROTOCOL OVERVIEW

In this section we provide an overview of the Mobile IP protocol.[3] Mobile IP provides a way for datagrams addressed to the mobile node at its *home network* to be delivered to its current point of attachment to the Internet. The current point of attachment is defined by an IP address known to the mobile node, called the *care-of address*. On the mobile node's home network an entity called the *home agent* is responsible for managing the delivery of datagrams to the mobile node at its care-of address when it is no longer present on the home network. Thus the delivery of datagrams to such a mobile node is broken into three phases.

- delivery to the home agent (on the home network);
- delivery to the care-of address;
- delivery to the mobile node.

Figure 1 illustrates the steady state operations which occur during delivery of a datagram to a mobile node. These operations occur by the action of the entities co-operating to support mobility for the mobile node, but the operations themselves do

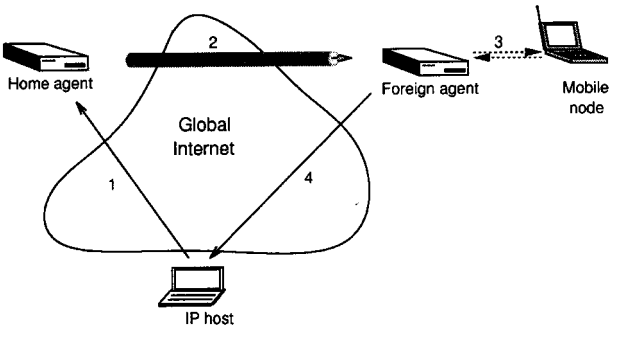

Figure 1. Packet flow in Mobile IP

not constitute mobility protocol operations. Instead, by the action of Mobile IP, the entities are able to support the illustrated operations as needed. The mobile node's home address is always used to maintain its network connections (e.g. TCP connections), and the care-of address is used to provide a route to the current point of attachment of the mobile node.

In the figure a typical and unmodified IP host is shown transmitting a datagram to the mobile node. The datagram traverses the global Internet (step 1) and arrives at the home network, where it is intercepted by the home agent for further processing and delivery. The home agent puts into operation routing mechanisms that cause the datagram to be routed (step 2) to the care-of address of the mobile node; in the figure the care-of address is hosted by another entity known as a *foreign agent*. Note that when a foreign agent is present, the care-of address does not have to be an address of any network interface of the mobile node. The foreign agent (step 3) then delivers the datagram to the mobile node; to do so, it inverts the mechanisms employed by the home agent which caused the datagram to be routed away from the home network to the care-of address, as described in Section 6. When the mobile node wishes to respond to the sending IP host, it can use the foreign agent as a default router; any datagrams sent by the mobile node through the foreign agent can be delivered (step 4) directly to the IP host. Any IP host (mobile, existing or modified as detailed in Section 7 to support mobile nodes) which communicates with a mobile node is called a *correspondent node*.

This simple beginning leads naturally to most of the necessary protocol operations needed for Mobile IP. The protocol must provide ways to do the following operations:

- provide a care-of address to the mobile node;
- inform the home agent about the current care-of address;
- manage the delivery of a datagram to the care-of address even though the datagram is addressed to the mobile node;
- allow the mobile node to determine when movement has occurred to a new point of attachment (so that the home agent can be notified).

As the protocol description proceeds, the need for various additional options will be noted, always coming along with the need for option negotiation between the co-operating entities.

The mobile node acquires a care-of address by processes of advertisement and solicitation which are largely separable from the other protocol operations, although there are numerous dependences. There are two main cases:

- the mobile node gets a care-of address from a foreign agent;
- the mobile node owns or acquires another IP address which it assigns to one of its network interfaces and uses as the care-of address.

The former case will be the subject of Section 3. If the care-of address is assigned to a network interface of the mobile node, it is known as a *collocated* care-of address. Methods by which a mobile node may acquire a collocated care-of address include DHCP[4,5] and PPP,[6–8] but are largely beyond the scope of this article. Rules by which Mobile IP entities handle collocated care-of addresses and care-of addresses obtained from foreign agents are almost identical, but minor differences exist.

In order for the mobile node to notify its home agent about its care-of address, a process of *registration* is specified for use between the mobile node and the home agent and, if present, the foreign agent. After registration the home agent will know the care-of address of the mobile node and therefore deliver datagrams to it. This registration procedure must be made very secure, otherwise any node in the Internet could masquerade as the mobile node and initiate a malicious registration procedure with the intent to disable or usurp communications between the mobile node and the rest of the Internet. For instance, if registration were not secure, a malicious node could supply its own address as the care-of address of the mobile node. Registration amounts to a variant of *remote redirection*, applied from afar to affect the internal state of the home agent. Considering that delivery of datagrams from the home network to the care-of address of the mobile node can potentially be shunted to an unrelated part of the Internet, the characterization makes intuitive sense. Security problems with remote redirects are well understood in today's Internet.[9]

The last piece of Mobile IP involves the delivery of datagrams between the home agent and the care-of address. Mobile IP itself is really only concerned with setting up the delivery path, not carrying out the delivery itself. In order for the home agent to deliver packets to the care-of address, it uses a process known as *encapsulation* or *tunnelling*. The latter name is suggested in Figure 1 by the thick tube (step 2) shielding the datagram from the effects of the global Internet while on its way from the home agent to the care-of address. If the mobile node's home address were visible to routers during the time the datagram travelled between the two tunnel endpoints, it would naturally be routed back to the home network and nothing would be accomplished. Instead, the original datagram is encapsulated by another IP header with the destination of IP address equal to the care-of address and then delivered to the care-of address by the home agent. Thus in the figure, when the foreign agent receives the encapsulated datagram, it merely has to remove the encapsulating (outer) IP header and deliver the resulting inner datagram directly to the mobile node, which is usually presumed to be

on the same link as one of the foreign agent's network interfaces.

Following this overview, then, the next sections describe in detail care-of address advertisement by foreign agents, registration procedures and tunnelling.

3. MOVING ABOUT

As noted above, there are two classifications of care-of addresses. The collocated care-of addresses are acquired by means outside Mobile IP; even static assignment can be used, for instance, a mobile node could use its CDPD (Cellular Digital Packet Data)[10] address as a care-of address whenever it is within range of a CDPD link. In any case the care-of address is considered by the home agent as the tunnel endpoint for any datagram it has to deliver to the mobile node. In this section we consider how Mobile IP enables the acquisition of care-of addresses which are not collocated with the mobile node—in other words, care-of addresses acquired from foreign agents.

The general idea is quite simple. When a mobile node moves within range of a foreign agent, it listens for advertisements which contain a care-of address. If no advertisements are detected, the mobile node may solicit for a care-of address. In many cases the only way the mobile node can detect whether it has moved is by comparing new advertisements with previous ones and determining whether the offered care-of address has changed.

Mobile service advertisements and solicitations are transported via ICMP, with the ICMP payload sometimes containing one or more special Mobile IP extensions. ICMP was chosen because of the perceived similarities between mobility advertisements and the advertisements used in the already existing Router Discovery protocol (RFC 1256).[11] As a general rule, protocol engineers (like programmers) try to reuse existing system components to avoid old errors and benefit from the experience of others.

3.1. *Agent solicitation*

RFC 1256 defines a Router Solicitation ICMP message. It is used by Mobile IP as an Agent Solicitation, without any changes, to find foreign agents or a home agent; in the latter case the mobile node discovers that it has attached to its home network. The message format is illustrated in Figure 2.

In Figure 2 the type is 10, the code is 0 and the checksum is the 16-bit one's complement of the one's complement sum of the ICMP message, starting with the ICMP type.

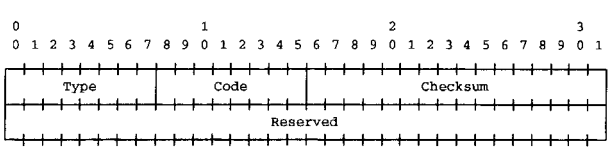

Figure 2. Mobile IP agent solicitation

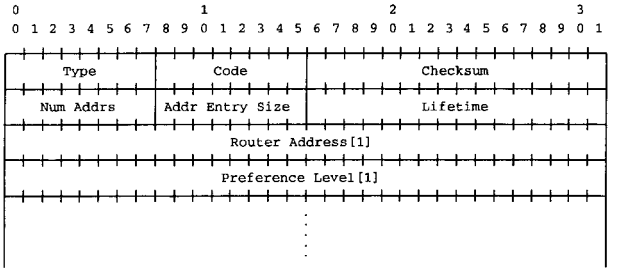

Figure 3. Router advertisement

A mobile node sending a solicitation is required to set the TTL field to 1. The only permissible values for the IP destination are the *all-routers* multicast address, 224.0.0.2, or the *limited broadcast* address, 255.255.255.255, both of which addresses cannot be forwarded by the routers because of the TTL. Any foreign agent or home agent receiving the solicitation will respond with an Agent Advertisement, as detailed in the next subsection.

3.2. *Agent advertisement*

Mobile IP Agent Advertisements are derived from RFC 1256 router advertisements. They are transmitted by mobility agents for use by mobile nodes, containing all the information needed by a mobile node to begin the registration (or deregistration) procedures needed at its current point of attachment. The advertisement message header is the same as for RFC 1256, as illustrated in Figure 3, but contains the Mobility Agent Advertisement, defined below.

In the advertisement the type is 9, the checksum is as defined for the solicitation message, and the code can be either 0 or 16, the latter choice indicating that the mobility agent is not configured to perform as a normal router (and thus that nodes using Router Advertisement will not choose it for normal routing purposes). Agent advertisements can include normal Router Advertisement information, including the addresses for other routers, but this information has to be handled very carefully by mobile nodes (see Section 6.2).

The care-of address in the advertisement is contained in the Mobility Agent extension, illustrated in Figure 4.

The foreign agent can advertise multiple care-of addresses. This feature may become important in the future for describing the relevant hierarchy of

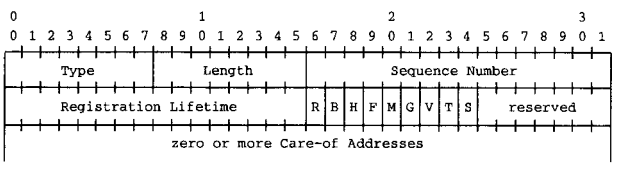

Figure 4. Mobility agent advertisement extension

mobility agents (or security agents) which may serve mobile nodes at a particular location.[12] Other fields are as follows.

Type	16.
Length	$6 + 4N$, where N is the number of care-of addresses advertised.
Sequence Number	The count of agent advertisement messages sent since the agent was initialized.
Registration Lifetime	The longest lifetime (measured in seconds) that this agent is willing to accept in any registration request. A value of 65,535 (all ones) indicates infinity.
R	*Registration required*. Registration using a foreign agent is required (even if using a collocated care-of address).
B	*Busy*. The foreign agent will not accept registrations from new mobile nodes.
H	If set, the agent is a *home agent*.
F	If set, the agent is a *foreign agent*.
M	Received tunnelled datagrams *may* use *minimal encapsulation*.[13]
G	Received tunnelled datagrams *may* use *Generic Routing Encapsulation* (*GRE*).[14]
V	*Van Jacobson header compression*[15] **may** be used over the link with any registered mobile node.
T	*Reverse tunnelling* from foreign agent to home agent is available (Section 9).
S	*Smooth hand-off* is supported (by way of the *Previous Foreign Agent Notification* extension defined with *route optimization* (Section 7.2)).

Note that the 'T' and 'S' flags are not currently defined as part of the base Mobile IP specification.[3]

It is possible for the same mobility agent to serve both as a home agent to mobile nodes with addresses on the home network and as a foreign agent offering care-of addresses to mobile nodes from other networks. A foreign agent that is too busy to serve new mobile nodes sets the 'B' bit, but continues to broadcast advertisements periodically so that current registered mobile nodes (customers) will remain confident that the advertised set of care-of address(es) is still valid.

The information in the advertisement is for use by the mobile node when it registers its current routing information with its home agent. If the mobile node gets an Agent Advertisement from its home agent after having been attached at some other care-of address, the mobile node *must* operate without the services of the home agent, by deregistering its previous care-of address(es) and once again enabling the use of ARP to resolve requests involving its home address.

The 'M' and 'G' flag bits in the advertisement indicate that the foreign agent can handle alternative encapsulation mechanisms. GRE has not yet been shown to interoperate betwen two independent implementations of Mobile IP, but may become important in the future if multiprotocol tunnelling gains use. Minimal encapsulation is more frequently implemented and will probably be used mainly for the cases of wireless mobile routers, wireless mobile nodes with collocated care-of addresses, and wireless nodes implementing route optimization. Saving 8 or 12 bytes in the encapsulation header seems less important for the links between the home agent and the foreign agent if they are all high-speed wired interconnections.

The *registration lifetime* in the Mobility Agent extension of the Agent Advertisement indicates the maximum period of time (typically hours or many minutes) for which the foreign agent is willing to allow use of its care-of addresses for any one registration. If the mobile node wishes to continue use of a care-of address for a longer time, it reregisters the same care-of address. This is to be distinguished from the lifetime field supplied in the ICMP Router Advertisement header (Figure 3), which indicates how long (typically a few seconds) the advertised router information should be considered valid.

4. REGISTRATION

Central to the operation of the home agent is its ability to keep track of the care-of addresses for its mobile nodes. Most of the actual protocol operations associated with Mobile IP have to do with enabling the mobile node to notify its home agent whenever its care-of address changes, by following the registration procedures detailed in this section. *Registration request* messages are sent to the home agent from the care-of address, properly authenticated to avoid malicious disruptions of service to the mobile node. If the home agent approves the registration, as it almost always does for authorized mobile nodes, then it sends an authenticated *registration reply* message to the care-of address. If the care-of address belongs to the mobile node, then the registration process is finished. If the care-of address belongs to a foreign agent, then the foreign agent relays the reply message to the mobile node. In the latter case the foreign agent was also responsible for relaying the request message from the mobile node to the home agent; foreign agents must maintain a certain amount of state to match up replies from home agents with pending registration requests. Other than this, foreign agents play a passive role in the registration process. They typically do not reject registration requests unless the registration conditions indicated in their Agent Advertisment received by the mobile node are not followed.

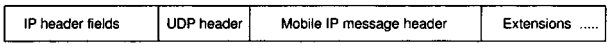

Figure 5. Mobile IP message format

As a matter of terminology, each association between a mobile node and a care-of address is called a *binding*. The home agent, then, is in charge of maintaining a set of bindings for its mobile nodes, and the process of registration is the process of reliably updating bindings at the home agent. Other IP nodes could conceivably maintain bindings for the mobile node; doing so is crucial to the techniques proposed for *route optimization*, as discussed in Section 7. Note, however, that the foreign agent does not necessarily maintain a binding for the mobile node. Maintaining such a binding would allow the foreign agent to encapsulate datagrams for delivery to a remote mobile node. Instead, the foreign agent maintains a *visitor list* entry so that it can know how to handle datagrams that have been decapsulated. When the foreign agent has multiple network interfaces, it does have to know the interface at which the mobile node can receive datagrams.

Mobile IP registration request messages are always sent to UDP port 454. Registration replies are sent back to whichever (arbitrary) source port issued the corresponding request. Registration datagrams are laid out as shown in Figure 5, not including the MAC-layer protocol header. As shown, the registration message header is followed by one or more extensions. The extensions used for authentication (as shown in Figure 8) are currently the only important ones. By protocol, registration requests can be issued no more often than once per second, but the practical limit is much less and is approximately equal to one or two times the *round-trip time* (*RTT*) for packets between the mobile node and the home agent.

4.1. *Registration request*

The Registration Request message is illustrated in Figure 6.

Type	1 (Registration Request).
S	*Simultaneous binding* requested.
B	*Broadcast datagrams* requested.
D	Self-*decapsulation* by mobile node. That is, the mobile node is using a collocated care-of address.
M	*Minimal encapsulation* requested.
G	*GRE* requested.
V	*Van Jacobson compression* requested.
Lifetime	Lifetime of the registration in seconds.
Home Address	The IP address of the mobile node.
Home Agent	The IP address of the mobile node's home agent.
Care-of Address	The IP address for the tunnel endpoint.
Identification	A 64-bit number, constructed by the mobile node, used for matching registration requests with registration replies and for protecting against replay attacks of registration messages.

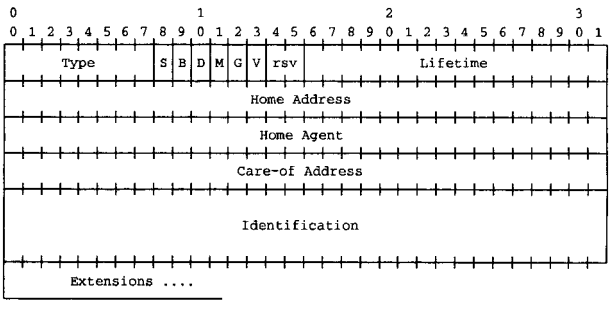

Figure 6. Mobile IP registration request

Most of the fields above have functions which are self-explanatory. The *identification* field will be discussed more fully in connection with the authentication extensions (Section 4.5).

The home agent may, upon request, maintain multiple bindings (multiple care-of addresses) for a mobile node. This could be useful when a mobile node is within range of several wireless transceivers, each of which has a noisy or weak signal. When the home agent has multiple care-of addresses for a mobile node, it replicates each datagram and sends a separate copy down each tunnel for decapsulation at each care-of address. This does not violate the networking semantics of IP, which allows for the possibility that datagrams may be received multiple times at the destination IP address. At the time of writing, however, no implementations of simultaneous bindings are known to the author.

If the mobile node wishes to receive broadcast datagrams from the home network, it has to explicitly notify the home agent of this fact, by setting the 'B' bit. Delivery mechanisms and selection for broadcast datagrams will be discussed in Section 6; both depend upon whether or not the mobile node is using a collocated care-of address, which is indicated by the setting of the 'D' flag.

The choice of the capital letter 'D' to name this flag dates from the time when the collocated care-of address was erroneously known as a *dynamic* care-of address, modelled upon allocation by DHCP. However, the function of a collocated care-of address has very little dependence upon the manner by which a mobile node acquires it, and as previously noted, the acquisition does not have to be dynamic.

4.2. *Deregistration*

If the mobile node returns to its home network, it must notify the home agent to stop delivering

datagrams to its registered care-of address. This is done as a special case of registration, by allowing the mobile node to register its home address as its care-of address. When this happens, the home agent no longer has any role to play. In fact, the home agent takes steps to stay out of the mobile node's way (see Section 6.2). If the mobile node is away from home, has multiple care-of addresses and wishes to deregister one of them, it can do so by registering that care-of address with a zero lifetime. In this way the mobile node always has precise control over its set of care-of addresses. Typically, the mobile node has not requested the use of multiple simultaneous care-of addresses; in this case each new care-of address registration implicitly deregisters the previous care-of address, since the home agent will no longer have a record of it. Notice that rejected registrations have to be returned to the source address of the registration request, even in the case where the source address would otherwise be hidden by encapsulation.

4.3. *Registration reply*

Once the home agent receives a Registration Request, it fashions a Registration Reply (message type 3) for transmission to the mobile node (Figure 7). The Registration Reply also contains status information to indicate whether the registration succeeded or failed. In the unusual case where a foreign agent has to reject a request, it also uses a Registration Reply with the appropriate failure code for this purpose.

Again, the meaning of most of the fields in the reply message is self-explanatory. The lifetime for the binding granted by the home agent may be less than or equal to that requested by the mobile node, but it can never be more than what was requested. The identification field protects against replays (see Section 4.5) and enables the foreign agent to match the reply to a pending request. There are three classes of status codes which are possible:

- success;
- rejection by the foreign agent;
- rejection by the home agent.

Some interesting values for the status code are listed as follows.

Registration successful
0 registration accepted

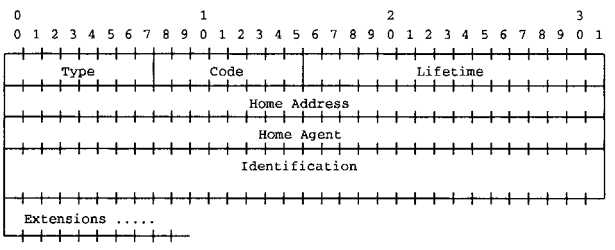

Figure 7. Registration reply packet format

1 registration OK, but simultaneous mobility bindings unsupported

Registration denied by the foreign agent
66 insufficient resources
69 requested lifetime too long
72 requested encapsulation unavailable
73 Van Jacobson compression unavailable
80 home agent unreachable (ICMP error)

Note that errors 69, 72 and 73 should never happen under normal circumstances if the mobile node is paying attention to the properties advertised by the foreign agent. However, it is possible to receive error 66 even when the 'B' bit is not set in the advertisement, owing to the dynamic nature of the foreign agent's workload.

Registration denied by the home agent
131 mobile node failed authentication
133 registration identification mismatch
135 too many simultaneous bindings
136 unknown home agent address

Rejection status 133 usually means that the mobile node and the home agent need to resynchronize the *identification* field used in the Registration Request; some other details about this situation are discussed in Section 4.5. Code 135 is returned only when the home agent does in fact support simultaneous bindings, but the mobile node has tried to register one more care-of address than the home agent is willing to handle. Code 136 is quite useful during the process of *home agent discovery*, described next.

4.4. *Automatic home agent discovery*

Suppose the mobile node is unable to register with its home agent, perhaps receiving code 80 after issuing a registration request to a home agent which had recently been working fine. If repeated attempts do not succeed, the mobile node may decide that its previous home agent address is no longer valid. Alternatively, the mobile node could have somehow wiped out whatever configuration it needed to contact its home agent. In either case the mobile node can use a special procedure to dynamically discover a new home agent address.

When the mobile node does not have a good home agent address, it can send its registration request to the *directed broadcast* address on its home network, which is defined to be its subnet prefix followed by ones in every bit position. For instance, the directed broadcast address for subnet 192.145.210/24 is 192.145.210.255 (where the /24 means that the first 24 bits are the *subnet mask*). This registration request will be rejected by every home agent on the home network. However, each rejection will contain a valid home agent address, so the mobile node can then issue a valid request. The disadvantage of this procedure is that every node on the home network will process the invalid registration request, even those nodes which are not

home agents. On the bright side, the procedure should only occur on very rare occasions.

4.5. *Registration authentication*

Mobile IP registration amounts to sending a *binding update* from the mobile node to the home agent and can have drastic effects on the path taken by datagrams on the way from the home agent to the mobile node. Thus this area of the protocol has to be made secure to avoid attacks by malicious Internet nodes that might wish to disrupt communications with a mobile node or nodes.

Mobile IP requires that entities have to maintain a list of *mobility security associations*, each of which contains sufficient information to perform some mutually agreed upon cryptographic algorithm. This collection of security associations is indexed by a 32-bit number known as an SPI or *security parameters index*. Each authentication extension carries along with it the SPI needed to verify the authentication.

Mobile IP defines several authentication extensions:

- the mobile–home authentication extension;
- the mobile–foreign authentication extension;
- the foreign–home authentication extension.

Each extension has a different type number. For instance, if a foreign agent wishes to authenticate itself to a home agent and has a security association with that home agent, it can append a foreign–home authentication extension (type 33) to the registration request initiated by the mobile node. All three extensions share a common format, illustrated in Figure 8. Since the length is an 8-bit field, all authenticators supplied in these extensions must fit within 251 or fewer octets.

Every Registration Request is required to contain a mobile–home authentication extension. The mobile node and home agent are presumed to set up the needed security association, and identify it by means of an SPI, as part of their configuration process. SPI values 0–255 are reserved. The UDP payload is the data to be authenticated within the registration request; the UDP header and headers preceding the UDP header are excluded. Any alteration to the authenticated data will be detectable, because the authenticator cannot be correctly recomputed by any other entity that does not possess the cryptographic information in the security association selected by the SPI. Importantly, the Identification field of the message is included in the authentication data; this field changes with every registration so that no entity can replay a correct registration sequence at a later time. By implication, the identification should never take the same value in two separate registration requests.

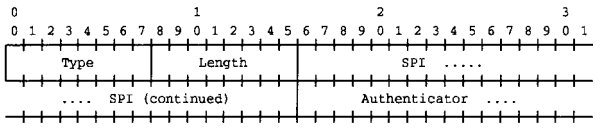

Figure 8. Mobile IP authentication extensions

There are two ways defined to guarantee such uniqueness. First, the mobile node and home agent can use a timestamp (in Network Time Protocol (NTP)[16] format). Every mobile node and home agent is *required* to support the use of timestamps for this purpose, so that interoperability can be guaranteed. As time moves along, the two entities must ensure that no unreasonable timestamp is used in the Identification field. Timestamps from the mobile node in the past, or too far in the future, are rejected by the home agent with error code 133, indicating the need for timestamp resynchronization. This method works fine as long as the method for synchronizing time (usually NTP) at the site is not corrupted. Ensuring this, however, means that NTP is itself run securely, a precaution not universally taken.

An alternative method has been devised that relies on the fact that a 64-bit Identification field is a huge space from which any two values, randomly chosen, are very unlikely to collide. In other words, if the Identification field is chosen each time in a truly random manner, there is unlikely to be any duplication. Unfortunately, true pseudo-random number generators, that do not require substantial processing power, are not so trivial to construct.[17] The protocol is devised so that the home agent supplies random values (called *nonces*) to the mobile node as needed, on the theory that the home agent has more processing power available than a possibly handheld, wireless, battery-powered mobile node.

For purposes of assuring interoperability between Mobile IP entities from different vendors, each mobile node and home agent is *required* to support a particular default algorithm for computing authenticators, and to support the assignment of any SPI value to select that default algorithm. The default algorithm is keyed MD5[18] used in *prefix + suffix* mode to compute a 128-bit message digest of the registration data. Prefix + suffix mode means that the entity producing the authenticator performs the MD5 algorithm on the following information:

$$\texttt{secret} \parallel \textit{data} \parallel \texttt{secret}$$

where `secret` is defined by the mobility security association indexed by the SPI, $\parallel$ is concatenation and *data* is the registration data in the UDP payload. Other authentication algorithms may be used in place of MD5; when other algorithms are used, the only restriction placed is that the authentication be performed on at least the registration message data. At the time of this writing, MD5 is still considered relatively secure, especially in view of the relatively infrequent nature of the registration process.

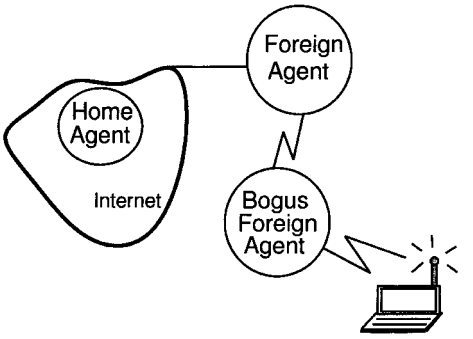

Figure 9. Man-in-the middle registration scenario

5. UNIDENTIFIED FOREIGN AGENTS

As noted above, Mobile IP does not use protocol operations that require verifiable identification of the foreign agent. This has the effect that registration by way of an anonymous foreign agent is vulnerable to a *man-in-the-middle* attack between the foreign agent and the mobile node. In other words, the mobile node can be fooled into thinking it has registered with a *bona fide* foreign agent when, in fact, it has been transacting with an interloper node, as illustrated in Figure 9. Until now, no one in the Mobile IP working group has ever exhibited any real difficulty with the interpolation of a bogus foreign agent into the path taken by datagrams to and from a mobile node. Put simply, if a node *acts* like a foreign agent, it *is* a foreign agent. This property seems to be shared by a bogus foreign agent acting alone or by any combination of a good foreign agent surrounded by multiple bogus foreign agents, as long as they follow protocol.

6. TUNNELLING AND ROUTING

The last major part of the base Mobile IP specification details the methods for delivering datagrams form the home network to the care-of address and thus the presumed location of the mobile node. The method used is encapsulation, by one of three specified encapsulation algorithms. The act of sending an encapsulated datagram to a decapsulating endpoint is called tunnelling the datagram. See Figure 10 for a general view of the encapsulation process. Although tunnelling is useful for a variety of purposes within the Internet today, here we limit consideration to the case of Mobile IP, so that in the figure the encapsulator would be the home agent and the decapsulator would be the owner of the care-of address.

Figure 10. Tunnelling

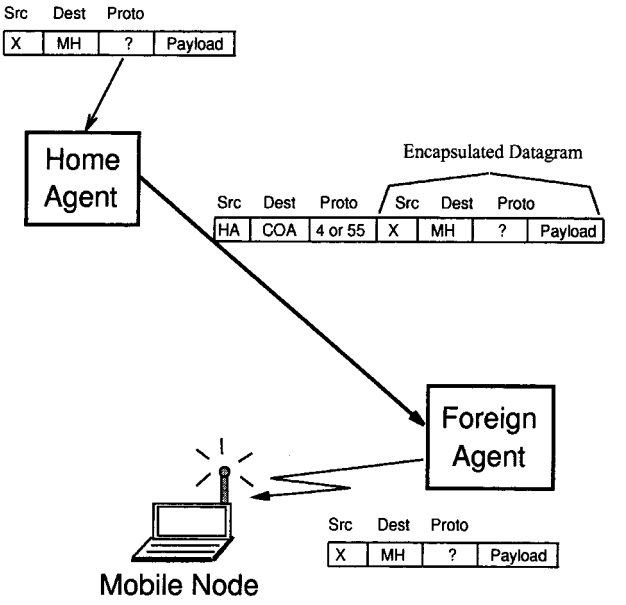

Figure 11. Mobile IP tunnelling operations

The default encapsulation protocol is *IP-within-IP*.[19] IP-within-IP is useful because all nodes are presumed to understand IP, even though nodes in today's Internet are typically not equipped to handle two IP headers in a row. The home agent sets the *protocol* field equal to 4 in the encapsulating header, to indicate that the payload is itself an IP datagram. In other words, the inner encapsulated IP header looks like a *higher-level protocol* header to the outer encapsulating IP header. When minimal encapsulation (described below) is used, the home agent sets the protocol field equal to 55 instead, to indicate the different format of the minimal encapsulation (the higher-level protocol in this case) header. Obviously, in this discussion the term *higher-level protocol* is a misnomer, because IP and minimal encapsulation headers are network-layer headers. The IPv6 discussion in Section 8 shows a better model for header processing.

Consider the encapsulation of a datagram from a correspondent node X for delivery to a mobile node MH, illustrated in Figure 11. The figure shows that the original payload of the correspondent node, with any original higher-level protocol, arrives unchanged at the mobile node. The foreign agent obtains the unchanged datagram by inverting whatever modifications the home agent made for the purposes of tunnelling the datagram.

6.1. *Alternative encapsulation protocols*

The minimal encapsulation header is illustrated in Figure 12. Most of the minimal encapsulation header

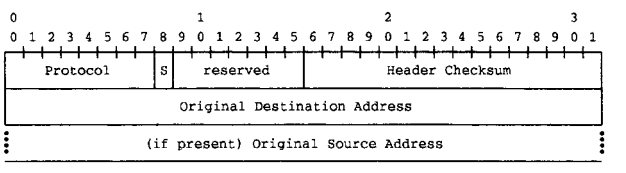

Figure 12. Minimal encapsulation header format

fields are self-explanatory. Whenever the tunnel source is different from the original source of the datagram, the minimal encapsulation header occupies 12 bytes instead of 20 bytes for the IP-within-IP header; in this case the 'S' bit is set. When correspondent hosts use minimal encapsulation in conjunction with route optimization techniques, the minimal header may require only 8 bytes of overhead in the tunnelled datagram.

Although allowed by the Mobile IP specification, Generic Routing Encapsulation (GRE)[14] has not yet been demonstrated in interoperability tests, so should be considered a feature useful for future expansion of the protocol. GRE will not be discussed in detail, but may be very useful for support of Multiprotocol Tunnelling (see Section 6.3).

6.2. *ARP*

One of the central requirements of Mobile IP is that home agents have to deliver datagrams from the home network to a care-of address by the tunnelling operations just described. This implies, however, that datagrams addressed to the mobile node have to somehow arrive at the home agent for further processing whenever the mobile node is not at home.

Mobile IP solves this problem by requiring two things:

- the home agent logically appears to the rest of the Internet to be attached to the home network;
- whenever the home network has any nodes physically attached to it, the home agent must perform proxy ARP on behalf of the mobile node while the mobile node is not physically attached to the home network.

In other words, whenever a node on the home network broadcasts an ARP packet in an attempt to locate the MAC address (layer 2 address) of the mobile node, and the mobile node is away from home, the home agent supplies its own MAC address so that all datagrams destined for the home address of the mobile node will arrive at the home agent instead. This works fine except in the boundary conditions when the mobile node detaches from the home network, or arrives back at the home network after having attached at a care-of address on some other network.

After the mobile node detaches from the home network, the ARP caches of other nodes on the home network may already contain the MAC address of the mobile node, and subsequent communications will fail. In this case the home agent (as soon as a registration is accepted) broadcasts several *gratuitous* (unsolicited) ARP reply packets, which are received by every node on the home network. These *gratuitous ARPs* are supposed to trigger an update to the ARP cache of every node that receives the broadcast. There are instances of implementations where this does not happen correctly, but Mobile IP deftly solves that problem by declaring that the protocol is not applicable to such configurations. Mobile IP can only be operated on networks where every node supports gratuitous ARPs; recent experience shows that there are indeed nodes using popular operating systems that do not comply even with expected network protocol operations such as these. Note that it is possible to avoid the whole issue by configuring mobile nodes with addresses on *virtual* home networks, which do not have to correspond to any physical communications medium; such networks do not have any nodes physically attached to them and thus do not have to support ARP at all.

When the mobile routing returns to the home network after registering elsewhere, as part of the deregistration process the mobile routing also broadcasts gratuitous ARPs, this time to trigger the insertion of its own MAC address into the other nodes' ARP caches.

This business with gratuitous ARPs is one hint that Mobile IP has tricky dealings with ARP. A mobile node is prohibited from using ARP in almost all cases when it is registered at a care-of address. If the mobile node were to broadcast any ARP request or reply with its home address on a foreign network, it would cause ARP cache entries to be created on the foreign network which could not be updated after the mobile node moved away. Similarly, the mobile node must not respond to any ARP broadcast for its home address when it is not attached to its home network. For this reason, the advertisements of other routers in the modified Router Advertisements (Figure 4) emanating from foreign agents are not really useful. The mobile node has no reliable mechanism specified by Mobile IP to resolve the advertised IP router addresses into usable MAC addresses, and cannot use ARP.

6.3. *Broadcast and multicast*

If a mobile node wishes to receive broadcasts from its home network, it sets the 'B' bit in its Registration Request message. The home agent will then tunnel broadcasts from the home agent to the mobile node's care-of address. However, if a foreign agent at the care-of address detunnelled the datagram and saw that the destination IP address was the limited broadcast address 255.255.255.255, then the foreign agent would drop the datagram. Worse, if the destination address were the subnet directed broadcast address, the foreign agent would route it back to the home network since it would have no reliable way to know that the destination was indeed a directed broadcast address. Thus, when the 'D' bit is not set, Mobile IP specifies that the home agent must doubly encapsulate each broadcast datagram;

- first, with the mobile node's home address as the destination IP address of an encapsulating header;

- finally, with the care-of address as the destination IP address in the second encapsulating header.

If a mobile node has a collocated care-of address, there is no danger of a foreign agent discarding or misdelivering the broadcast. Thus only one encapsulation is needed; when the mobile node decapsulates, it can consume the broadcast directly. Mobile IP allows the mobile node to inform the home agent about this situation by setting the 'D' bit along with the 'B' bit in its registration request. Note that this is the only motivating circumstance where the home agent might care what style of care-of address is being reported by the mobile node during the registration process.

Currently, there is no standard way for a mobile node to be selective about which broadcasts it wishes to receive. The base Mobile IP specification is not helpful in this regard. Selection of broadcasts is regarded as a matter of system configuration between the home agent and the mobile node, without any negotiation defined in the standard. However, there is a draft specification for a registration extension[20] allowing the mobile node to select the broadcasts it wants. The selection can be dynamic, since the applications which need to receive broadcasts on the mobile node will come and go over time as the mobile computer user's needs and activities change.

Multicast is handled similarly to broadcast, and this is an area where the standard may one day be considered insufficient. As is the case with the selection of broadcast packets, mobile nodes may need to specify a selection of multicast datagrams which they need to receive without having to respond to IGMP[21] broadcasts being tunnelled to the mobile node. Another Internet draft[22] similar to the draft for selection of broadcast datagrams proposes a way for a mobile node to negotiate with its home agent for the delivery of particular multicast datagrams.

6.4. *Mobile routers*

Nothing in the Mobile IP protocol specification prohibits a mobile node from actually being a router. This would be the case, for instance, if a router on a ship attached to the Internet at various places during its travels at sea, and served the subnet or subnets of Internet nodes operating on the ship. Note that the Internet nodes on the ship may themselves be mobile. Mobile IP works in all these cases, typically with multiple levels of encapsulation.

Suppose, for example, that a mobile node is the router for certain fixed subnets on a ship and that the routing infrastructure delivers pasckets for all the subnets to the mobile node's on-land home network. The home agent makes it appear as if the mobile node is attached to the home network, and thus tunnels all packets to the mobile node on the ship. Even if there are mobile nodes moving from one ship subnet to another (and thus from one care-of address on the ship to another), this model works just as well. Datagrams from landed sources to one of those mobile nodes will be tunnelled by the landed home agent to the mobile router, decapsulated, and thence forwarded to the ship's home agent for that mobile node on the appropriate ship subnet, and thence tunnelled to the care-of address for the ship's mobile node (which is still on the ship!), and then decapsulated for final delivery to the mobile node. Other cases work similarly.

The only difficulty lies in the manner of establishing routes to the ship's subnets from the mobile node's landed home network. Since the mobile router is often away from the home network, it will be expensive to participate in routing protocols to maintain the appearance that the ship's subnets are attached indirectly to the home network. It is indeed possible for the mobile router to do this, by requesting that its home agend send broadcast packets to it, and by tunnelling back broadcast routing protocol datagrams onto its home network. However, this may become undesirable in some circumstances; if so, then new extensions should be defined so that the mobile router's home agent could advertise the mobile subnets on behalf of the mobile router while it is away from home.

6.5. *Multiprotocol tunnelling*

Viewed in a certain way, Mobile IP's registration procedure is a mechanism for establishing a tunnel between the home agent and a mobile node. In most implementations, discovering which tunnel is needed for a particular mobile node is a matter of searching through a route table indexed by the mobile node's home IP address.

However, the need for tunnel establishment exists for mobile nodes that do not necessarily use IP. Suppose for the moment that a foreign agent existed for a non-IP network-layer protocol and that the foreign agent offered an IP care-of address for such a mobile node. If the operation of the foreign agent is extended to simulate the Mobile IP registration of the mobile node, and if sufficient authorization has been extended to the foreign agent so that the tunnel establishment could be authenticated, then multiprotocol tunnelling will be possible. Certain modifications are needed, naturally.

For one thing, the home agent will have to index the tunnel entry for the mobile node by something other than its IP care-of address. This is a relatively simple matter of implementation and platform-dependent code. More importantly, the information included when encapsulating packets to be delivered to the mobile node by the foreign agent will depend on the specific network-layer protocol which the mobile node is using. Lastly, the home agent will have to use whatever means are provided by the alternative network-layer protocol to attract packets

which are addressed to the mobile node on the home network, unless of course the home network is purely a virtual network.

7. ROUTE OPTIMIZATION

Mobile IP suffers from a problem known as *triangle routing*, which is illustrated in Figure 13.

When a correspondent node needs to send a datagram to a mobile node, it has to go through the home network. When the mobile node sends a datagram to the correspondent node, by normal Internet routing rules, the datagram can be delivered directly to the correspondent node from the care-of address. In the figure a foreign agent is shown, but the routing anomaly does not depend upon the ownership of the care-of address.

This additional network travel for routing of datagrams to the mobile node can be a potential source of a number of troubles:

- increased delays for incoming traffic;
- increased network congestion;
- increased vulnerability to network partitions;
- creation of a routing bottleneck at the home agent.

If the correspondent node were aware of the mobile node's care-of address, then datagrams to the mobile node could be tunnelled directly from the correspondent node to the mobile node without the assistance of the home agent. Route optimization[23] is a process by which correspondent nodes are enabled to perform such tunnelling. Detunnelling such datagrams would not require any change in the already specified behaviour of the foreign agent or mobile node. The big problem is that correspondent nodes in today's Internet are generally unable to accept and store such information about a mobile node's current care-of address, and are similarly unable to perform any sort of tunnelling.

However, working on the assumption that mobility will eventually be important enough so that correspondent nodes will have to support it, we can specify the needed operations. Recall that a *binding* is the association between a mobile node, its care-of address and the lifetime of the association. Following this terminology, route optimization is the process of enabling correspondent nodes to tunnel directly to the care-of address, using bindings which they receive, authenticate and store in a *binding cache*. Note that binding information has to be authenticated by correspondent nodes, just as registrations have to be authenticated by home agents. In fact, there is not much difference in function between the binding update used for route optimization and registration, although the packet formats, and the targets to which they are addressed, are different. Only the registration request, however, causes the recipient (i.e. the home agent) to perform proxy ARP for the sending mobile node.

7.1. *Smooth hand-offs*

One particular case of route optimization deserves special attention, and that is when the entity maintaining a binding cache is the mobile node's previous foreign agent. Consider the illustration in Figure 14. Suppose that when a mobile node moves from one foreign agent to another, the first (i.e. the *previous*) foreign agent is able to receive information from the new (i.e. the *current*) foreign agent about the mobile node's new care-of address. If this operation is sufficiently fast, then datagrams en route to the previous foreign agent can be retunnelled to the current care-of address with possibly no loss of information. Enabling this operation of *smooth hand-off* can be done by using the mechanisms of route optimization to supply a binding update to the pre-

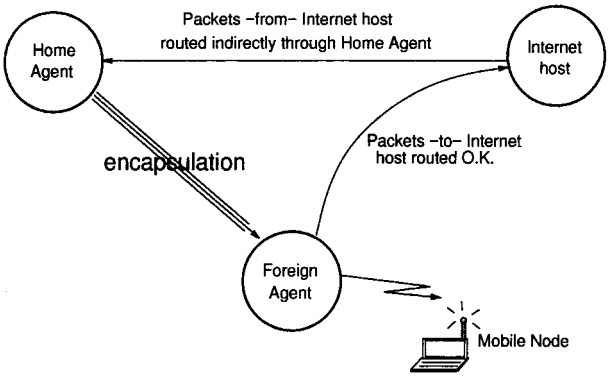

Figure 13. Triangle routing

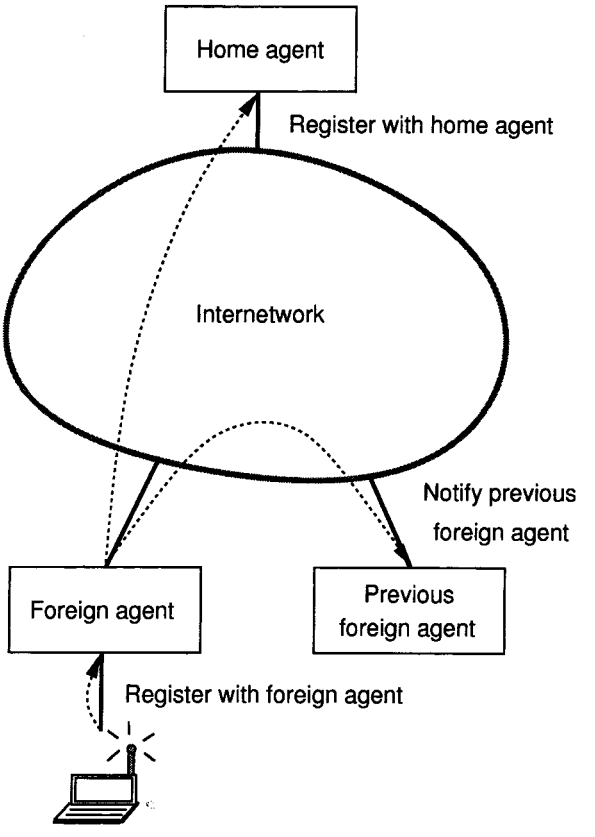

Figure 14. Smooth hand-offs

vious foreign agent in a timely manner. Moreover, smooth hand-offs minimize the time during which packets may be dropped because of registration latency. The previous foreign agent may receive many datagrams destined from the mobile node between the time the mobile node moves to the new foreign agent and the time when the home agent receives the new registration. Dropping these datagrams could seriously impact the performance of TCP on the mobile node,[24] which would translate into an annoying degradation of the interactive response of applications running on the mobile node. Owing to *slow start* mechanisms in TCP,[25] dropped packets cause about twice as much inconvenience as one might intuitively expect otherwise. Since current Web applications all rely on TCP, avoiding dropped packets could substantially improve user perception of interactive performance.

One idea under current consideration involves buffering packets at one or more foreign agents for more robust delivery to a mobile node. For instance, if packets were buffered while a mobile node moved to a new foreign agent, it could receive the buffered packets and avoid the need for the correspondent node to resend them, similar to the ideas explored in Reference 26. Such buffered hand-off mechanisms fit naturally within the framework of route optimization and the smooth hand-off mechanisms described here.

7.2. *Binding authentication and registration keys*

Route optimization is best understood as doing whatever is needed to securely deliver binding updates. In most cases it is anticipated that it will be easier for the home agent to maintain security relationships with relevant correspondent nodes than for the mobile node to do so. This results from the current difficulty in performing key distribution, which is a natural and essential part of maintaining security associations. In view of that, the binding update is sent from the home agent to the correspondent host. Authenticating the binding update is then done in exactly the same way as authenticating a registration sent from a mobile node to its home agent; namely, by appending an authentication extension to the binding update. The authentication extension for binding updates has the same format as the extension used for registration requests (Figure 8). The same default algorithms and modes are required as in Section 4.5.

Authenticating the binding updates used for smooth hand-offs is a somewhat different story. This is an area which, while tedious, is important for any future robust deployment of the smooth hand-off mechanisms. The reason that authentication is difficult is that generally the mobile node cannot, *a priori*, be presumed to have any existing security relationship with its foreign agents as it moves from one point of attachment to another (as discussed in Section 5). It is more likely that its home agent could provide assistance, acting as a *key distribution centre* (KDC) for the mobile node and its new foreign agent. The key established between the foreign agent and the mobile node (by whatever means) is called a *registration key*.

The route optimization draft protocol[23] makes available various strategies for the mobile node and the foreign agent to establish a registration key. Note that since the home agent always has a security relationship with the mobile node, the home agent can securely deliver any registration key to the mobile node if the home agent is acting as a KDC. The problem is to allow the home agent, acting in that capacity, to securely deliver the registration key (by encrypting it) to the foreign agent. In general terms, and isolated from other details, the procedure is as follows.

1. The foreign agent advertises its willingness to perform smooth hand-offs.
2. If the mobile node has a security relationship with the foreign agent, it can use that security association to secure the future binding update.
3. Otherwise, if the mobile node has a public key, it asks the foreign agent to supply a registration key using the public key to encrypt it.
4. Otherwise, the mobile node includes a *registration key request* extension in its registration request.
5. If the foreign agent has a security association with the home agent, it asks the home agent to supply a key using the security association to encrypt it.
6. Otherwise, if the foreign agent has a public key, it asks the home agent to supply a key using the public key to encrypt it.
7. If all else fails, and the mobile node has included the proper data in its registration key request, the foreign agent performs a Diffie–Hellman[27] key exchange with the mobile node.

It appears that route optimization, while offering more protocol opportunity for bogus foreign agents (see Figure 9) to work their strange designs, is nevertheless no more vulnerable to their malicious intent. In particular, if a bogus foreign agent, by following protocol, supplies an apparently useful registration key to the mobile node, the mobile node may blithely use the registration key as if the bogus foreign agent were honest. Note that any registration key involving an honest foreign agent on the wired side and the home agent as KDC can be trusted by the mobile node, because the home agent authenticates the key selection whether or not it was chosen by the home agent or the honest foreign agent.

All foreign agents advertising support for smooth hand-offs are presumed to support Diffie–Hellman key exchange as a last resort to establish a registration key with the mobile node. The mobile node tells the foreign agent whether it wants this service by including the necessary computational constants

in its registration key request. As far as is possible, route optimization offers ways to get the key by other means, but the expectation is that many mobile nodes will not have any other way. Diffie–Hellman is last resort because it requires expensive exponentiation with very big numbers, which takes a while to compute and could add significantly to battery utilization.

7.3. *Special tunnels*

Suppose that a foreign agent supports smooth hand-offs, as in Figure 14, and receives a tunnelled datagram for a mobile node that is no longer on its visitor's list, and for which the foreign agent has no binding. The foreign agent *may* drop the datagram. Alternatively, to avoid losing such datagrams, the foreign agent can send undeliverable datagrams back to the home network. This will allow the home agent to tunnel the datagrams to the correct foreign agent. Note that the foreign agent is unlikely to know the home agent's IP address unless it is already included in the tunnel header; it will not be there when a correspondent node has tunnelled the datagram. Nevertheless, if the foreign agent keeps the mobile node's home address as the destination IP address, it would probably arrive at the home agent anyway.

If the foreign agent wishes to help avoid the loss of the datagram, it is required to *retunnel* it back to the home network instead of relying on normal Internet routing to deliver the untunnelled datagram to the home network. The datagram will get back to the home network if the destination IP address is the home address of the mobile node. However, if the datagram is not tunnelled back, it could get into an iterated tunnelling loop between the home agent and the incorrect care-of address until the IP header TTL expires. This undesirable situation is prevented by using a tunnel, in this situation called a *special tunnel*.

Upon receipt of a tunnelled diagram destined for the mobile node's address, the home agent is expected to compare the tunnel destination with the (inner) destination of the tunnelled datagram. For special tunnels, both destinations will be equal to the mobile node's home address. The starting point of the special tunnel will be the previous foreign agent's address. If the home agent finds that the mobile node is registered at the source IP address of the special tunnel, then the home agent must not tunnel the datagram back again to the same foreign agent. This last case, which is hopefully rare, could happen if the foreign agent lost track of some visiting mobile nodes. No further protocol is provided to solve this last case. The base Mobile IP protocol provides ways that a mobile node can detect that its foreign agent has rebooted, and thus discover that reregistration may be needed. It is to be hoped that rebooting is the only time when foreign agents forget the list of nodes which are using their services. If indicated by future needs, a new extension could be defined for a registration reply that would be sent to the amnesiac foreign agent.

8. IPv6 MOBILITY

IP version 6 (IPv6) is a new protocol[28–32] being designed as a replacement for IP, which has performed remarkably well during its tenure as the reigning network-level protocol for the Internet. It is anticipated that IP will suffer from exhaustion of its address space sometime very near the beginning of the next century; estimates range from year 2002 to 2006 and later. To be more precise, it is not that each one of the over 4 billion addresses in th 32-bit IP address space will be completely assigned to existing computers; instead, there will be no remaining IP network numbers for allocation to new network installations. Since network numbers form the basis for normal Internet routing, and since IP addresses are almost worthless if they are not routable, this amounts to the same thing as exhausting the address space. Thus IPv6, with its inconceivably larger 128-bit address space, is being prepared to eventually replace the current IP, hereafter called IPv4. This mighty task involves designing replacements for all existing protocols that rely on any details of IPv4, and there are a lot of them.

8.1. *IPv6 overview*

To enable this article to be self-contained, a few important details about IPv6 will be reviewed here. The bits of the address space may be imagined to be divided into roughly equal halves, one part for host identification and one part for routing. The host identification half of the bits allows for complete inclusion of a node's 48-bit IEEE 802 address in the lower half of its IPv6 address, and all IPv6 hosts could be organized into roughly 16 quintillion different routes, where a route is defined by the higher-order address bits of the host's IPv6 address. That ought to be enough routes for a while.

It seems likely, therefore, that IPv6 will solve the existing problems of address space depletion. However, IPv6 includes other features aimed at correcting deficiencies associated with IPv4; even though IPv4 has been able to keep up with growth beyond the imaginations of its original designers, there are nevertheless imperfections in the protocol that could be eliminated. For instance, IPv6 has improved option processing. With IPv6 it is possible to specify that some options do not need the attention of intermediate routers, whereas in IPv4 the use of options is inhibited by the fact that datagrams carrying options are routed quite inefficiently compared with datagrams without options.

A major difference between IPv6 and IPv4 is that any compliant implementation of IPv6 is required to support the processing of some basic security

options for authentication and privacy protection (encryption). This is a matter for great controversy, given the existing laws in various countries of the world which effectively curtail the use of such security techniques. Nevertheless, the community of protocol designers creating IPv6 decided to avoid political arguments (and perhaps even political realities) when making technical decisions about what is needed for the infrastructure of the future Internet. Also, there is no doubt that authentication and privacy are essential for the ability to carry out electronic commerce over the Internet. Thus protocols using IPv6 can be designed under the assumption that authentication and privacy are available. Generally speaking, authentication is based on the possession of a secret key, and the fact that the results of certain cryptographic computations cannot be forged by any machine that does not have access to the secret key. Note the simplification that this would have afforded to the design of registration and route optimization for Mobile IPv4!

IPv6 comes equipped with superior methods for autoconfiguring new nodes as they first attach to the network. Nodes can create their own address by using Stateless Address Autoconfiguration;[33] ARP and Router Advertisement are replaced by Neighbour Discovery.[34] These two protocols are related, and allow leaf nodes to attach to the Internet with globally routable addresses with no further administration. Routers still require the attention of system administration, although there is current work aiming to simplify even that chore.[35]

The last relevant improvement in IPv6 involves the use of routing headers for delivery of packets to their destination. In IPv4 there is a *loose source route* option which allows a source node to require a datagram to visit certain intermediate nodes along the way to its destination. This option was proposed for use with a number of early candidate schemes for IP-layer mobility,[36–39] but routing inefficiencies and security problems prevented adoption of those schemes. The main security difficulty associated with IPv4 source routing is the requirement for *route reversal* by the node which receives the source-routed datagram. Router reversal, which means that the receiver has to source route any responses back through the same intermediate nodes present in the sender's source route, is both a blessing and a curse. Because of the lack of deployed authentication protocols, it opens the door to simple impersonation attacks by any node in the Internet. If a datagram has to be routed back through an intermediate node, that intermediate node can impersonate any other node in the Internet. IPv6 avoids this problem by not requiring the reversal of source-routed packets. Source routing is specified by the inclusion of a *routing header* in packets that need such handling. Routing a packet through a care-of address is sufficient to deliver a packet to a mobile node situated at that care-of address.

8.2. *Mobility support in IPv*6

With the stated assumptions about authentication options, the operations described above are almost enough to enable universal support for IPv6 mobility.[40,41] The missing piece that has to be supplied is the *binding update*. Then a correspondent node (or home agent) uses the care-of address known from the binding update as an intermediate routing point for all packets destined for the mobile node.

Mobility support within IPv6 is natural, especially given the previous developments for IPv4. From Mobile IPv4 are borrowed the concepts of home agent, care-of address and correspondent node. From the route optimization proposal is borrowed the notion of a binding update, which in IPv6 also serves the purpose of tunnel establishment between the home agent and the mobile node.

In IPv6 the mobile node carries out all the needed operations to support its own mobility. Since the mobile node may be presumed to have established the needed security relationships with its correspondent nodes, mobile nodes deliver their own binding updates, in contrast with the situation in IPv4. The home agent hardly enters the picture at all under steady state conditions of established communications between a correspondent node and a mobile node. The mobile node can authentically report the care-of address to its home agent by sending it in a binding update destination option. Note that this option can be sent to the home agent even though the payload portion of the packet containing the binding update is empty. See Figure 15 for the format of the binding update destination option.

To see how this works, suppose that a mobile node departs from its home network and finds a new point of attachment to the IPv6 Internet. Using the above-mentioned methods of Stateless Address Autoconfiguration, the mobile node acquires a *link-local* address and checks that the address is unique (not already in use) on the link. With its new address the mobile node then participates in Router Discovery to acquire a subnet prefix and thus a globally routable IPv6 address. Since the subnet prefix is presumed unique and the link-local address is unique on the link, this procedure is perfect for acquiring a care-of address. Stateful address autoconfiguration[42] may also be used instead, to acquire a globally routable address appropriate for

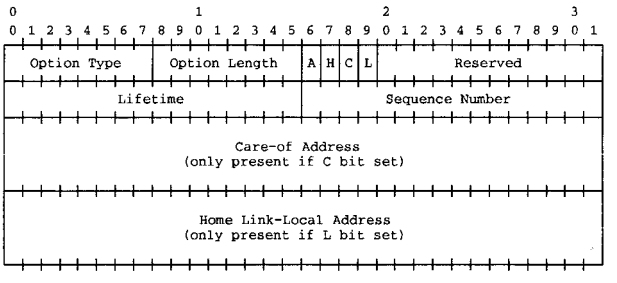

Figure 15. Binding update destination option

the new point of attachment, and such addresses are equally appropriate for use as a care-of address.

Moreover, the care-of address can be reported, in almost exactly the same manner, to every correspondent node that is currently concerned with the whereabouts of the mobile node. When the correspondent node has the mobile node's care-of address, it can include a routing header containing that address in any packet that it needs to send to the mobile node. Two other factors work in favour of this arrangement:

- many correspondent nodes are likely to be within close proximity to the mobile node as it travels;
- the mobile node can update its correspondent nodes as soon as it moves, avoiding any extra latency which would be introduced by the time it takes for the care-of address to reach the home agent.

On the downside, the binding updates are likely to occur over a low-bandwidth channel instead of over the wired infrastructure containing the home agent. Since the binding update is normally only sent as part of data packets which are already being delivered to correspondent nodes, this disadvantage is minimized. The mobile node can check its TCP control blocks to find out those correspondent nodes with which it is engaged in active communication.

Whereas the correspondent nodes use routing headers to deliver packets to the mobile node's care-of address, the home agent cannot do this. Inserting a routing header would disturb any authentication header data that might be present in the packet. Thus, the home agent uses IPv6-within-IPv6[43] encapsulation, just as home agents do in IPv4 mobility. Therefore, when a mobile node receives an encapsulated packet, it knows that the original source of the packet (if not the home agent) needs to receive a binding update. The home agent is involved in sending packets to the mobile node only during exceptional circumstances, and IPv6 mobility offers all the benefits of route optimization.

Just as in IPv4, home agents have to perform proxy services for the mobile node, so that packets sent to the mobile node on the home network will be attracted to the home agent's MAC address. Neighbour Discovery allows for this.[34] The operation formerly performed by *gratuitous ARP* is more rigorously defined and activated by the proxy agent by setting the *override* bit in a Host Advertisement. If the home agent cannot reasonably be expected to know the mobile node's link-local address (and thus its MAC address), this information must be provided by the mobile node as an optional part of one of the binding updates sent to the home agent. One further subtlety arises when there are multiple home agents on the home network. At any particular time, only one of them can serve as the home agent for any particular mobile node (the mobile node's *designated* home agent). If that mobile node needs to send a binding update to a home agent that is not its designated home agent, the mobile node does not want that other home agent to start performing proxy services. Thus, to resolve this ambiguity, binding updates intended for a mobile node's *designated* home agent must set a special flag (the 'H' flag).

Lastly, it should be mentioned that there are no foreign agents specified as part of IPv6 mobility support. The only real loss associated with this would be the ease of performing smooth hand-offs. However, if the mobile node can establish a security association with its default router at each point of attachment, then it can ask previous routers to forward packets sent to its previous care-of addresses, at least until those previous addresses are reused. The request is made by sending a binding update with the 'H' bit set to the previous router of concern. The previous router then acts as a home agent for the care-of address. Methods for acquiring temporary keys with routers at each new point of attachment may be adapted from the registration key acquisition methods specified for IPv4 route optimization. Using this strategy, smooth hand-offs for IPv6 fit naturally without the creation of additional protocol.

In the rest of the paper, IP will again be used to mean IPv4.

9. FIREWALLS, INGRESS FILTERING AND REVERSE TUNNELLING

Mobile IP allows the mobile node to issue datagrams using its home address as the source IP address in the IP header. This often means that the source address is not topologically correct from the perspective of the foreign agent's first hop router, or the mobile node's first hop router when there is no foreign agent. However, recent administrative practice for enterprise networks follows the recommendations of an Internet Draft advising the use of *ingress filtering*.[44] This means that enterprise routers may be configured to drop packets unless the source address appears to be topologically correct. The rationale for doing so is to enable the administrator to avoid giving the impression of allowing employees to issue possibly troublesome packets. If every enterprise does ingress filtering, so the theory goes, then no enterprise can harbour anonymous wrongdoers. Any wrongdoing will be traceable by virtue of the correctness of the source IP address.

Whether or not this is a valid strategy, it is a strategy that Mobile IP must learn to deal with until the strategy is replaced by better one. In the meantime, one way for mobile nodes to get around the problem is to incur (yet) another routing penalty, and *reverse tunnel* packets back to their home network, from which further delivery can take place and from which the use of the home address as the source IP address will be topologically correct.[45,46]

In IPv6 a better solution is available for ingress

filtering. A *Home Address* destination option is proposed,[40] by which a mobile node can inform a correspondent node that the sender's true address is found in the proposed option, instead of in the source IP address of the packet as usual. This would allow a mobile node to appear to a foreign administrative domain to have a topologically correct source address (e.g. its care-of address), but to keep its connections open with correspondent nodes based on its home address which would be provided in the new option.

Another difficulty is posed by the placement of firewalls[47] between the home enterprise and the rest of the Internet. This means that the mobile node might find it difficult to get registration packets back to the home agent once leaving the enterprise. Moreover, the placement of firewalls at other locations relative to the mobile node may also be problematic, but no *new* problems are introduced in the latter case by Mobile IP. Firewalls are, basically, a big protocol headache, in more ways (not considered here) than are relevant to Mobile IP.

Administrators should configure their firewalls to admit packets if they are destined to arrive at a trusted home agent. Home agents are likely to be decapsulating agents, and tunnelling can be used as a dangerous form of redirection. Thus the home agents *must* exert some policy control over the further delivery of decapsulated packets, rather than just blindly reinserting them onto the home network. Another strategy for allowing registration to proceed is for the mobile node to explicitly maintain or establish tunnels to enable the entry of its registration messages into the home enterprise.[48–50]

The proposal for *hierarchical foreign agents* is being modified to provide for hierarchies of security domains, each of which is protected by a firewall. As long as two agents attempting to establish chained tunnels trust each other, and the other agents in each domain trust one of the agents negotiating the tunnel, a tunnel can be established and parametrized as required. Surrogate tunnels can be managed without any protocol changes to Mobile IP.

10. CURRENT STATUS

Mobile IP is progressing nicely through the IETF standardization process, except for concerns related to ingress filtering and firewall traversal. There have been two successful interoperability Testathons, both generously hosted by FTP Software. As a result, there are about a dozen known independent implementations, plenty enough to show the viability of the protocol. A number of options in the current Proposed Standard[3] have not been tested at the interoperability sessions, notably *simultaneous registrations*, the MIB specification,[51] GRE[14,52] and the ability to forward broadcast and multicast packets from the home agent to the mobile node's care-of address.

Along with the base protocol, the encapsulation specifications[13,19] and the MIB specification are currently Proposed Standards. Mobile IP cannot advance to Draft Standard until all such features have either undergone interoperability testing or else have been deleted from the protocol specification. The MIB specification is not a feasible candidate for deletion. Besides these problems, however, it is the opinion of the author that a better resolution of the security problems is needed.

As indicated above, future users are likely to demand a high level of application transparency, and Mobile IP has been designed to meet that demand. On the other hand, today's users are largely satisfied with a level of service which can be met by other means, for instance those which have been incrementally provided based on other protocols such as PPP. This makes sense, given that wide-area mobility is currently based on dial-up protocols.[53] Moreover, local-area mobility (by way of, for instance, wireless LANs) has not met market projections. As a result, the deployment of Mobile IP to this date has been slow, although bright spots exist, particularly at academic institutions. This should be accelerated by the existence of various freeware implementations, some of which are available as follows:

- http://www.monarch.cs.cmu.edu/(CMU);
- http://www.cs.pdx.edu/research/SMN (Portland State);
- http://mip.ee.nus.sg (University of Singapore);
- http://www.mcl.cs.columbia.edu/source.html (Columbia Mobile IP);
- ftp://ftp.it.kth.se/pub/klemets/klemets.tar.gz ('MINT');
- http://anchor.cs.binghamton.edu/mobileip/ (Linux MH & 'agent');
- http://mosquitonet.stanford.edu/software/mip.html (MosquitoNet).

11. 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.

REFERENCES

1. W. A. Simpson (ed.), 'The Point-to-Point Protocol (PPP)', *RFC 1661*, 1994.
2. J. B. Postel (ed.), 'Transmission Control Protocol', *RFC 793*, 1981.

3. C. Perkins (ed.), 'IPv4 mobility support', *RFC 2002*, 1996.
4. R. Droms, 'Dynamic Host Configuration Protocol', *RFC 2131*, 1997.
5. S. Alexander and R. Droms, 'DHCP options and BOOTP vendor extensions', *RFC 2132*, 1997.
6. W. Simpson, 'The Point-to-Point Protocol (PPP) for the transmission of multi-protocol datagrams over point-to-point links', *RFC 1331*, 1992.
7. G. McGregor, 'The PPP Internet Protocol Control Protocol (IPCP)', *RFC 1332*, 1992.
8. J. Solomon and S. Glass, 'Mobile-IPv4 configuration option for PPP IPCP', *draft-ietf-pppext-ipcp-mip-03.txt*, 1998 (work in progress).
9. V. L. Voydock and S. T. Kent, 'Security mechanisms in high-level networks', *ACM Comput. Surv.*, **15**, 135–171 (1983).
10. *Cellular Digital Packet Data Specification*, CDPD Consortium, Chicago, IL, 1993.
11. S. E. Deering (ed.), 'ICMP Router Discovery messages', *RFC 1256*, 1991.
12. C. Perkins, 'Mobile-IP local registration with hierarchical foreign agents', *draft-perkins-mobileip-hierfa-00.txt*, 1996 (work in progress).
13. C. Perkins, 'Minimal encapsulation within IP', *RFC 2004*, 1996.
14. S. Hanks, T. Li, D. Farinacci and P. Traina, '*Generic Routing Encapsulation (GRE)', RFC 1701*, 1994.
15. V. Jacobson, 'Compressing TCP/IP headers for low-speed serial links', *RFC 1144*, 1990.
16. D. Mills, 'Simple Network Time Protocol (SNTP) version 4 for IPv4, IPv6 and OSI', *RFC 2030*, 1996.
17. D. E. Eastlake, S. D. Crocker and J. I. Schiller, 'Randomness recommendations for security', *RFC 1750*, 1994.
18. R. L. Rivest, 'The MD5 message-digest algorithm', *RFC 1321*, 1992.
19. C. Perkins, 'IP encapsulation within IP', *RFC 2003*, 1996.
20. B. Patel and C. Perkins, 'Preference for broadcast datagram support with Mobile IP', *draft-perkins-mobileip-bcastpref-00.txt*, 1996 (work in progress).
21. S. Deering, 'Host extensions for IP multicasting', *RFC 1112*, 1989.
22. P. Bhattacharya, B. Patel and C. Perkins, 'Preference for multicast datagram support with Mobile IP', *draft-partha-mobileip-mcastpref-07.txt*, 1996 (work in progress).
23. C. E. Perkins and D. B. Johnson, 'Route optimization in Mobile-IP', *draft-ietf-mobileip-optim-07.txt*, 1997 (work in progress).
24. R. Caceres and L. Iftode, 'Improving the performance of reliable transport protocols in mobile computing environments', *IEEE J. Select. Areas Commun.*, **SAC-13**, 850–857 (1995).
25. V. Jacobson, 'Congestion avoidance and control', *Proc. ACM SIGCOMM '88 Workshop*, August 1988, pp. 314–329.
26. R. Caceres and V. Padmanabhan, 'Fast and scalable handoffs for wireless networks', *Proc. ACM Mobicom 96*, 56–66 November 1996.
27. W. Diffie and M. Hellman, 'New directions in cryptography', *IEEE Trans. Inf. Theory*, 644–654 (1976).
28. S. Deering and R. Hinden, 'Internet Protocol, version 6 (IPv6) specification', *RFC 1883*, 1995.
29. R. Hinden and S. Deering, 'IP version 6 addressing architecture', *RFC 1884*, 1995.
30. A. Conta and S. Deering, 'Internet Control Message Protocol (ICMPv6) for the Internet Protocol version 6 (IPv6)', *RFC 1885*, 1995.
31. C. Huitema, *IPv6—The New Internet Protocol*, Prentice-Hall PTR, Upper Saddle River, NJ, 1996.
32. S. O. Bradner and A. Mankin, *IPng Internet Protocol Next Generation*, 2nd edn, IPng Series, Addison-Wesley, Reading, MA, 1996.
33. S. Thomson and T. Narten, 'IPv6 stateless address autoconfiguration', *RFC 1971*, 1996.
34. T. Narten, E. Nordmark and W. Simpson, 'Neighbor discovery for IP version 6 (IPv6)', *RFC 1970*, 1996.
35. M. Crawford and R. Hinden, 'Router renumbering for IPv6', *draft-ietf-ipngwg-router-renum-01.txt*, 1997 (work in progress).
36. Y. Rekhter and C. Perkins, 'Loose source routing for mobile hosts', *Internet Draft*, 1992 (work in progress).
37. C. Perkins, A. Myles and D. Johnson, 'IMHP: a Mobile Host Protocol for the Internet', *Comput. Netw. ISDN Syst.*, **27**, 479–491 (1994).
38. D. B. Johnson, 'Scalable and robust internetwork routing for mobile hosts', *Proc. 14th Int. Conf. on Distributed Computing Systems*, June 1994, pp. 2–11.
39. C. Perkins and A. Myles, 'Mobile IP', *Proc. Int. Telecommunications Symp.*, August 1994, pp. 415–419.
40. D. Johnson and C. Perkins, 'Mobility support in IPv6', *draft-ietf-mobileip-ipv6-04.txt*, 1997 (work in progress).
41. C. E. Perkins and D. B. Johnson, 'Mobility support in IPv6', *Proc. ACM Mobicom 96*, 27–37 November 1996.
42. J. Bound and C. Perkins, 'Dynamic Host Configuration Protocol for IPv6', *draft-ietf-dhc-dhcpv6-13.txt*, 1998 (work in progress).
43. A. Conta and S. Deering, 'Generic packet tunneling in IPv6', *draft-ietf-ipngwg-ipv6-tunnel-08.txt*, 1998 (work in progress).
44. P. Ferguson and D. Senie, *Network Ingress Filtering: Defeating Denial of Service attacks which employ IP Source Address Spoofing*, RFC 2267, January 1998.
45. G. Montenegro, 'Reverse tunneling for Mobile IP', *draft-ietf-mobileip-tunnel-reverse-02.txt*, 1997 (work in progress).
46. S. Cheshire and M. Baker, 'Internet mobility 4×4', *ACM SIGCOMM Comput. Commun. Rev.*, **26**, 318–329 (1996).
47. W. R. Cheswick and S. Bellovin, *Firewalls and Internet Security*, Addison-Wesley, Reading, MA, 1994.
48. P. Calhoun and C. Perkins, 'Tunnel Establishment Protocol (TE)', *draft-calhoun-tep-00.txt*, 1997 (work in progress).
49. G. Montenegro, 'Tunnel Set-up Protocol (TSP)', *draft-montenegro-tsp-00.txt*, 1997 (work in progress).
50. V. Gupta and S. Glass, 'Firewall traversal for Mobile IP: goals and requirements', *draft-ietf-mobileip-ft-req-00.txt*, 1997 (work in progress).
51. D. Cong, M. Hamlen and C. Perkins, 'The definitions of managed objects for IP mobility support using SMIv2', *RFC 2006*, 1996.
52. S. Hanks, T. Li, D. Farinacci and P. Traina, 'Generic routing encapsulation over IPv4 networks', *RFC 1702*, 1994.
53. C. Rigney, A. Rubens, W. Simpson and S. Willens, 'Remode Authentication Dial in User Service (RADIUS)', *RFC 2138*, 1997.

Author's biography:

Charles E. Perkins is a Senior Staff Engineer at Sun Microsystems, developing Service Location Protocol and investigating dynamic configuration protocols for mobile networking. He is serving as document editor for the Mobile IP working group of the Internet Engineering Task Force (IETF) and is author or co-author of standards-track documents in the svrloc, dhc (Dynamic Host Configuration) and IPng working groups, as well as serving on the Internet Architecture Board (IAB). Charles has recently authored a book on Mobile IP and has published a number of papers in the areas of mobile networking, *ad hoc* networking, route optimization for mobile networking, resource discovery and automatic configuration for mobile computers. He is associate editor for *Mobile Communications and Computing Review*, the official publication of ACM SIGMOBILE, and area editor for the journals *Wireless Networks* and *Transactions on Networking*. Please refer to the author's home page for additional information on related projects (http://www.svrloc.org/~charliep).