Invited paper

Route Optimization for Mobile IP

Charles E. Perkins [a] and David B. Johnson [b]

[a] *Sun Microsystems, Menlo Park, CA 94025, USA*
[b] *Carnegie Mellon University, Pittsburgh, PA 15213-3891, USA*

Route Optimization has been designed within the IETF to ameliorate the problem of triangle routing, a routing artifact introduced by Mobile IP's requirement to route packets destined for a mobile node by way of its home network. In this article, we describe the current protocol specification for the Route Optimization protocol, concentrating on design decisions and justifications. Once the basic mechanisms are explained, we show how they are applied to enable foreign agents to offer smooth handoffs for mobile nodes, and describe the security operations that enable reliable operation of this handoff between foreign agents with which a mobile node has no pre-existing security relationship.

1. Introduction

Mobile IP has been developed in the Internet Engineering Task Force (IETF) as a solution to provide nearly transparent roaming for IP-addressable mobile nodes [15]. Physical effects such as variable bandwidth make it impossible for roaming to be completely transparent to applications, but such effects are forgivable since they are inevitable. Other problems such as routing anomalies and faulty congestion control are more difficult to understand, more difficult to diagnose, and less likely to be tolerated by typical users. Route Optimization is an attempt to solve the former problem, by reducing or eliminating the routing anomalies introduced by the base Mobile IP specification.

In this paper, we define the problem and give the details of one possible approach towards a solution. As with any routing problem, a robust solution needs to incorporate good security techniques, to avoid the possibility that a malicious network entity might introduce fraudulent routing information and thus disrupt communications. Not solving the problem is better than solving the problem in a way that offers opportunities for corrupting the integrity of the routing tables of computers which need to communicate with mobile nodes.

Section 2 of this paper gives a brief overview of Mobile IP and the *Route Optimization* extensions to it, describing their component parts and design. Section 3 describes the process of *binding cache* maintenance and details each of the messages used by Route Optimization for this purpose. The application of this cache maintenance for providing *smooth handoffs* as a mobile node moves from one foreign agent to the next is presented in section 4, and section 5 then presents the problem of the managing mobility security associations needed by Route Optimization. Accomplishing smooth handoffs requires the use of *registration keys*, and the proposed methods for establishing those keys are detailed in section 6, with the message formats and processing steps detailed in sections 7–9. To further illustrate the power of Route Optimization techniques, in section 10 we show how binding cache maintenance has been applied in the protocol design for mobility support in IPv6, the new version of IP now being designed in the IETF. Lastly, in section 11, we summarize and present conclusions.

2. Overview

The development of powerful laptop computers with medium-speed wireless communications adapters has been the driving motivation for the creation of new protocols. Mobile IP [15] enables a mobile node to move from place to place in the Internet, maintaining active connections and presenting to typical Internet nodes the illusion that it remains present on its *home network*. Communications with the mobile node proceed by use of the mobile node's *home address*, which does not depend on the node's current point of attachment to the Internet. At each point of attachment, the mobile node acquires a *care-of address*, which it must report to its *home agent* on its home network by a process called *registration*.

In all situations considered in the rest of this paper, the mobile node obtains the care-of address by interaction with a *foreign agent* on the *foreign network*. The foreign agent offers the care-of address using ICMP [17] by including it as part of a specially modified *Router Advertisement* [3], which is then known as an *Agent Advertisement*. The Agent Advertisement also includes a maximum time duration, or *lifetime*, for which the mobile node may consider the care-of address valid, and which bounds the lifetime permissible within the mobile node's Registration Request that is subsequently transmitted to the home agent. When a mobile node which is using a care-of address detects that it is no longer receiving the Agent Advertisements from its current foreign agent, or in some other way detects that it no longer is in contact with that foreign agent, it assumes that the care-of address is no longer valid. The mobile node then begins to search for a new care-of address, presumably from another foreign agent.

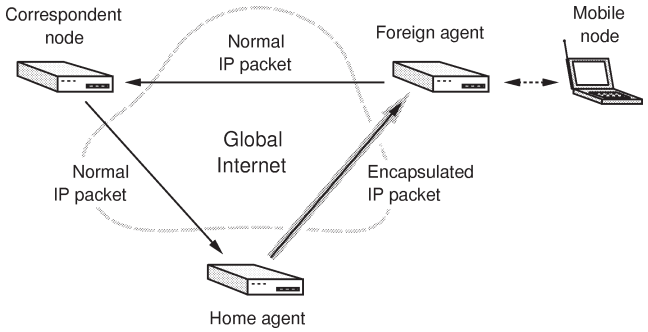

Figure 1. Overview of the base Mobile IP protocol.

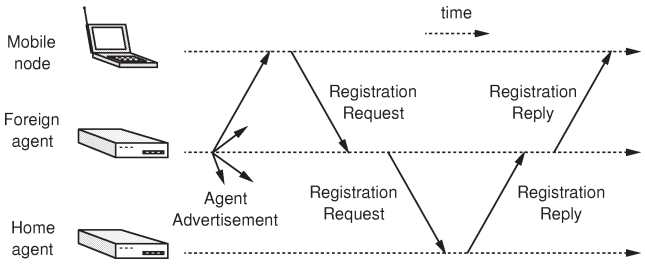

Figure 2. Overview of the Mobile IP registration process.

Figure 1 shows an overview of the base Mobile IP protocol, including the relative placement of Mobile IP agents and networks. A *correspondent node* is any IP host or router that communicates with a mobile node. A correspondent node, itself, may be either mobile or stationary. To send a packet to a mobile node, a correspondent node transmits the packet to the mobile node's home address, which causes the packet to be routed toward the mobile node's home network. There, the packet is intercepted by the mobile node's home agent. The home agent then tunnels the packet to the mobile node's current foreign agent, using the care-of address as the tunnel destination. The foreign agent decapsulates the packet and delivers it locally to the mobile node. If a mobile node sends a packet to a correspondent node, it simply sends it in the same way as if it were at home, but uses its foreign agent as the default router for delivering the packet. The foreign agent, simply acting as a router, then forwards the packet directly to the correspondent node.

The processes of agent advertisement and registration of a care-of address are illustrated in figure 2. Whenever a mobile node registers, its home agent (if it approves the request) associates the mobile node's home address with the care-of address and lifetime together in a routing record known as a *binding*. The home agent thus maintains a *binding cache* containing all the bindings for all those mobile nodes that are using its services while they are away from the home network. The home agent also performs whatever functions are necessary [15] to manage the interception of all packets destined for its registered mobile nodes. Once the home agent has intercepted a packet for a mobile node, it consults its binding cache to *tunnel* (encapsulate) the packet, setting the tunnel destination to be the mobile node's care-of address [13,14]. This tunneling process appears as additional routing overhead on all packets addressed to the mobile node.

This description of packet delivery to and from a mobile node illustrates a routing anomaly caused by the operation of Mobile IP. Packets sent *to* a mobile node are routed indirectly through the mobile node's home agent and are then tunneled to the mobile node, whereas packets sent *from* a mobile node are routed directly to the correspondent node. Round trip communications thus travel along three distinct routing paths, and thus this routing anomaly is known as *triangle routing*. This use of triangle routing can be seen in figure 1. All of these operations are specified by the base Mobile IP standard [15].

For the correspondent node to eliminate triangle routing, it must have some information regarding the current location of the mobile node – in other words, the correspondent node will have to discover the mobile node's care-of address, and to maintain a binding cache giving this care-of address for use in tunneling its own packets to the mobile node's care-of address.

Whenever the care-of address later changes, the correspondent node must update its binding cache. Route Optimization accomplishes this by sending a *Binding Update* to the correspondent node. The production and consumption of these Binding Updates form the heart of the operation of the Route Optimization protocol. And, just as the home agent cannot accept an unauthenticated Registration Request from a mobile node, neither can a correspondent node accept an unauthenticated Binding Update. One of the guiding principles for deciding how to send Binding Updates to correspondent nodes is that the correspondent node must be able to trust the sender. In the current Internet, deployment of authentication technology and key distribution is typically handled interactively, and thus not so amenable to the kind of automatic operation required for Route Optimization. Thus, our proposal has been designed to reduce the number of mobility agents which have to maintain trust with correspondent nodes.

When a mobile node moves to a new care-of address, any existing binding cache entries for the mobile node in different correspondent nodes' binding caches become out-of-date. An out-of-date binding causes a correspondent node to tunnel packets to an old care-of address. Unfortunately, in base Mobile IP, such packets are likely to get dropped because the mobile node's previous foreign agent is not notified of the mobile node's movement to its new care-of address. On the other hand, if the previous foreign agent can maintain a binding cache entry for a mobile node which had previously been visiting, that foreign agent can deliver such misdirected packets to the mobile node's current care-of address. The Binding Update messages used to inform correspondent nodes about a mobile node's care-of address are also used to inform the mobile node's previous foreign agent when the mobile node moves and acquires a new care-of address. Then, the previous foreign agent can act as a temporary forwarder for traffic destined to the mo-

bile node, until all of the relevant correspondent nodes have updated their binding cache entries for the mobile node. This process is called *smooth handoff*, and is described in detail in section 4.

While using the same messages, smooth handoff introduces a new requirement for establishing trust between a mobile node and each of its foreign agents. In base Mobile IP, and for the purposes of reliably delivering Binding Updates to correspondent nodes, the foreign agents are largely passive agents. No particular trust model is assumed to be in place between the mobile node and the foreign agent. For smooth handoff to work correctly, the previous foreign agents have to be able to authenticate Binding Updates, which means they must share a secret with the node transmitting the binding update. Moreover, to correctly handle packets in flight, the previous foreign agent must get the binding updates as soon as possible. Using the model of a mobile node moving from one wireless access point to the next nearby one, with each access point serving as or for a foreign agent, we choose to enable the new foreign agent to deliver the necessary Binding Update to the previous foreign agent as soon as possible. Establishing a way for the foreign agent to trust the mobile node, while difficult, is possible using any one of various techniques detailed later in the paper. All that the previous foreign agent has to know is that the Binding Update comes from the same mobile node as had been registered with it; this is accomplished by exchanging key information as part of the previous Registration Request and Registration Reply.

As a matter of terminology, the *mobility security associations* between nodes in this paper are usually indicated by a *Security Parameters Index*, or SPI. The SPI indicates which of possibly many security associations between two nodes is to be used when performing the necessary security operations. For instance, an SPI may indicate that authentication is to be performed by comparing the results of an MD5 computation operating on a stream of data, with some secret information included both before and after the data to be authenticated. SPIs are an important field in the registration key messages and other messages relating to a binding with the mobile node's new care-of address. SPIs are selected arbitrarily from the available 32-bit unsigned numbers, except that SPI numbers 0 through 255 are reserved and not allowed to be used in any mobility security association.

3. Binding cache maintenance messages

A correspondent node may create or update a binding cache entry for a mobile node only when it has received and authenticated the mobile node's mobility binding. In addition, a node may use any reasonable strategy for managing the space within its binding cache. When a new entry needs to be added to the binding cache, the node may choose to drop any entry already in the cache, if needed, to make space for the new entry. For example, a *least-recently used* (LRU) strategy for cache entry replacement is likely to work well. This is in contrast to the way that a home agent should manage the registrations for mobile nodes registered with it. A home agent should not drop a registration until the expiration of the lifetime of the binding established during the registration process.

When sending an IP packet, if the sending node has a binding cache entry for the destination node, it should tunnel the packet to the mobile node's care-of address using the encapsulation techniques used by home agents [7,8,13,14].

When a mobile node's home agent intercepts a packet from the home network and tunnels it to the mobile node, as shown in figure 1, the home agent may deduce that the original source of the packet has no binding cache entry for the destination mobile node. The home agent should then send a Binding Update message to the original source node, informing it of the mobile node's current mobility binding. No acknowledgement for such a Binding Update message is needed, since additional future packets from this source node intercepted by the home agent for the mobile node would cause transmission of another Binding Update. For a Binding Update to be authenticated by the original source node, the source node and the home agent must have established a mobility security association.

Similarly, when a foreign agent receives a tunneled packet, if it has a binding cache entry for the destination node, indicating that the mobile node has established a binding elsewhere (not with this foreign agent), the foreign agent may deduce that the tunneling node has an out-of-date binding cache entry for the mobile node. In this case, the foreign agent should send a *Binding Warning* message to the mobile node's home agent, advising it to send a Binding Update message to the node that tunneled this packet. The mobile node's home agent can be determined from the binding cache entry, because the home agent's address is learned from the Binding Update that established the cache entry. The address of the node that tunneled the misdirected packet can be determined from the packet's header, since the address of the node tunneling this packet is the *outer* source address of the encapsulated packet. As in the case of a Binding Update sent by the mobile node's home agent, no acknowledgement of this Binding Warning is needed, since additional future packets for the mobile node tunneled by the same node will cause the transmission of another Binding Warning.

In addition to the Binding Update and Binding Warning messages, Route Optimization makes use of two additional types of messages for binding cache maintenance: Binding Request and Binding Acknowledgement. These four message types are distinguished by a one-octet *type* field, with the message type values chosen from the numbering space defined in the base Mobile IP specification for messages sent to UDP port 434.

The following sections describe in detail each of the Route Optimization messages used for binding cache maintenance, as well as a message extension used by them for authentication. In addition, Route Optimization requires

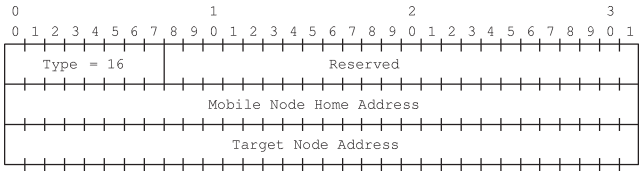

Figure 3. Binding Warning message format.

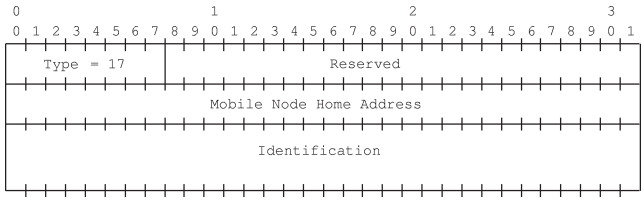

Figure 4. Binding Request message format.

one minor change to the existing Mobile IP Registration Request message: a new flag bit must be added, replacing a previously unused, reserved bit in the message. This modified Registration Request message format is also described below. In the description of each message format, any field marked as *reserved* must be set to all zeroes when transmitted and must be ignored upon reception.

3.1. Binding Warning message

A *Binding Warning* message is used to advise a mobile node's home agent that another node appears to have either no binding cache entry or an out-of-date binding cache entry for some mobile node. When any node receives and decapsulates a tunneled packet for which it is not the current foreign agent for the destination mobile node, if it forwards the packet to a new care-of address based on an entry in its own binding cache, it should send a Binding Warning message to the mobile node's home agent indicated in that binding cache entry. No authentication of the Binding Warning message is necessary, since it does not directly affect the routing of IP packets to the mobile node.

The format of the Binding Warning message is illustrated in figure 3. The mobile node's home IP address is followed by the *target node address*, which is the IP address of the node that tunneled the packet causing the Binding Warning message. A mobile node's home agent will thus receive a Binding Warning message if a node maintaining a binding cache entry for one of the home agent's mobile nodes uses an out-of-date entry. When a home agent receives a Binding Warning message, it should send a Binding Update message including the mobile node's current binding to the target node.

A foreign agent must provide some mechanism to limit the rate at which it sends Binding Warning messages to the same node about any given mobility binding. This rate limiting is especially important because it is expected that, within the short term, many Internet nodes will not support maintenance of a binding cache. In this case, continual transmissions of Binding Warning messages will only waste processing resources at the foreign agent, home agent, and correspondent node, and along the Internet path between these nodes.

3.2. Binding Request message

A *Binding Request* message is used by a node to request a mobile node's current mobility binding from the mobile node's home agent. A node wanting to provide continued service with a particular binding cache entry may attempt to reconfirm that mobility binding before the expiration of the registration lifetime. Such reconfirmation of a binding cache entry may be appropriate when the node has indications (such as an open transport-level connection to the mobile node) that the binding cache entry is still needed. This reconfirmation is performed when the node sends a *Binding Request* message to the mobile node's home agent, requesting a new Binding Update message with the mobile node's current mobility binding. The node maintaining the binding cache entry should also keep track of the home agent's address, to be able to fill in the destination IP address of future Binding Requests.

The format of the Binding Request message is illustrated in figure 4, and contains the home address of the mobile node to which the Binding Request refers. Following that is a 64-bit sequence number, assigned by the node sending the Binding Request message, which is used to assist in matching requests with replies and in protecting against replay attacks.

When the home agent receives a Binding Request message, it consults its list of registered mobile nodes and determines the correct binding information to be sent to the requesting node. Before satisfying the request, the home agent must check whether or not the mobile node has allowed the information to be disseminated. If the mobile node specified the *private* (P) bit in its Registration Request message (as shown in section 3.6), then the home agent returns a Binding Update in which both the care-of address is set equal to the mobile node's home address and the lifetime is set to zero. Such a Binding Update message indicates that the binding cache entry for the specified mobile node should be deleted.

3.3. Binding Update message

The *Binding Update* message is used for notification of a mobile node's current mobility binding. It should be sent by the mobile node's home agent in response to a Binding Request message or a Binding Warning message. It should also be sent by a mobile node, or by the foreign agent with which the mobile node is registering, when notifying the mobile node's previous foreign agent that the mobile node has moved.

The format of the Binding Update message is illustrated in figure 5, and contains the mobile node's home address and care-of address. Setting the care-of address in the Binding Update equal to the home address of the mobile node

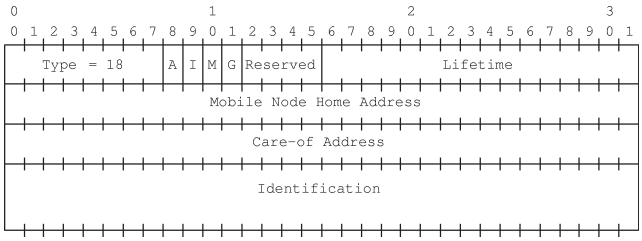

Figure 5. Binding Update message format.

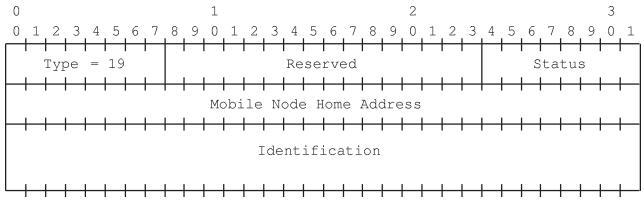

Figure 6. Binding Acknowledgement message format.

means that any existing binding cache entry (and visitor list entry, in the case of a mobile node's previous foreign agent) for the mobile node should be deleted. The Binding Update also contains a *lifetime* for this binding, and any binding cache entry created or modified as a result of receipt of this Binding Update must be deleted once this lifetime expires. A value of all ones for the lifetime indicates infinity; a value of zero means the same thing as setting the care-of address equal to the home address. When sending the Binding Update message, the home agent should set this lifetime to the remaining registration lifetime.

The Binding Update message also contains four defined flag bits. The *acknowledge* (A) bit is set by the node sending the Binding Update message to request a Binding Acknowledgement message be returned. The *identification present* (I) bit is set to indicate that the *identification* field is present in the message. The *minimal encapsulation* (M) and *Generic Routing Encapsulation* (GRE) (G) bits indicate support for receiving packets encapsulated with either of these two optional encapsulation protocols [7,14].

When a node receives a Binding Update message, it must verify the authentication in the message, using the mobility security association it shares with the mobile node's home agent. The authentication data is found in the *Route Optimization Authentication extension* (section 3.5), which must be present in the message. If the authentication succeeds, then a binding cache entry should be updated for use in sending future packets to the mobile node. Otherwise, an authentication exception should be raised.

As with the sending of Binding Warning messages, a home agent must provide some mechanism to limit the rate at which it sends Binding Update messages to to the same node about any given mobility binding. Since in the short term, some nodes may not support maintenance of a binding cache, continual transmissions of Binding Update messages will only waste processing resources at the home agent and correspondent node, and along the Internet path between these nodes.

The 64-bit *identification* field is present unless the Binding Update is sent as part of a *smooth handoff*. When using nonces for replay protection [15], the identification field in the Binding Update message is used slightly differently than when timestamps are used [15], to still allow replay protection even though the Binding Update is not being sent in reply to a request directly from the target node. In this case, the home agent is required to set the high-order 32 bits of the identification field to the value of the nonce that will be used by the home agent in the next Binding Update message sent to this node. The low-order 32 bits of the identification field must be set to the value of the nonce being used for this message.

Thus, on each Binding Update message, the home agent communicates to the target node the value of the nonce that will be used next time, and if no Binding Updates are lost in the network, the home agent and the target node can remain synchronized with respect to the nonces being used. If, however, the target node receives a Binding Update with what it believes to be an incorrect nonce, it may resynchronize with the home agent by using a Binding Request message.

3.4. Binding Acknowledgement message

A *Binding Acknowledgement* message is used to acknowledge receipt of a Binding Update message. It should be sent by a node receiving a Binding Update message if the *acknowledge* (A) bit is set in the Binding Update.

The format of the Binding Acknowledgement message is illustrated in figure 6, and contains a *status* field in addition to the mobile node's home address and the usual identification field for replay protection, copied from the Binding Update message.

If the status is nonzero, the Acknowledgement is negative. For instance, if the Binding Update was not accepted, but the incoming packet has the *acknowledge* flag bit set, then the status code should be set appropriately in the Binding Acknowledgement message. Status values are defined to indicate different conditions for which the Binding Update may have been rejected, including administrative prohibition, lack of sufficient resources, authentication failure, or a mismatch in the expected identification value.

3.5. Route Optimization Authentication extension

The *Route Optimization Authentication* extension is used to authenticate Binding Update and Binding Acknowledgement messages. It has the same format and default algorithm support requirements as the three authentication extensions defined for base Mobile IP [15], but is distinguished by its message type value. The authenticator value is computed from the stream of bytes including the shared secret, the UDP payload, all prior extensions in their entirety (that is, the Route Optimization management message), and the type and length of this extension, but not including the authenticator field itself nor the UDP header.

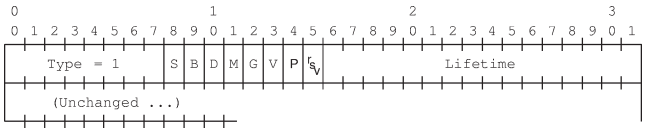

Figure 7. Modified Registration Request message format.

For implementations that can support more than the mandatory base authentication algorithm, other optional authentication algorithms such as the more secure HMAC authenticator [11] could also be used if specified in the mobility security association.

3.6. Modified Registration Request message

The format of the modified *Registration Request* message is illustrated in figure 7. It contains one new bit, the *private* (P) bit, enabling the mobile node to indicate that it would like its home agent to keep its mobility binding *private*. Normally, the home agent sends Binding Update messages to correspondent nodes as needed to allow them to cache the mobile node's binding. If the mobile node sets the *private* bit in the Registration Request message, the home agent is not allowed to send the mobile node's binding in any Binding Update message. Instead, each Binding Update message should give the mobile node's care-of address equal to its home address, and should give a lifetime value of 0.

4. Smooth handoffs

This section provides a description of the proposed operation of smooth handoff from a mobile node's *previous* foreign agent to its *new* foreign agent when the mobile node initiates a new registration.

4.1. Smooth handoff overview

When a mobile node moves and registers with a new foreign agent, the base Mobile IP protocol does not notify the mobile node's previous foreign agent. IP packets intercepted by the home agent after the new registration are tunneled to the mobile node's new care-of address, but packets in flight that had already been intercepted by the home agent and tunneled to the old care-of address when the mobile node moved are likely to be lost and are assumed to be retransmitted by higher-level protocols if needed. The old foreign agent eventually deletes its visitor list entry for the mobile node after the expiration of the registration lifetime.

Route Optimization provides a means for the mobile node's previous foreign agent to be reliably notified of the mobile node's new mobility binding, allowing packets in flight to the mobile node's *previous* care-of address to be forwarded to its *new* care-of address. Any packets tunneled to the mobile node's previous foreign agent, from correspondent nodes with out-of-date binding cache

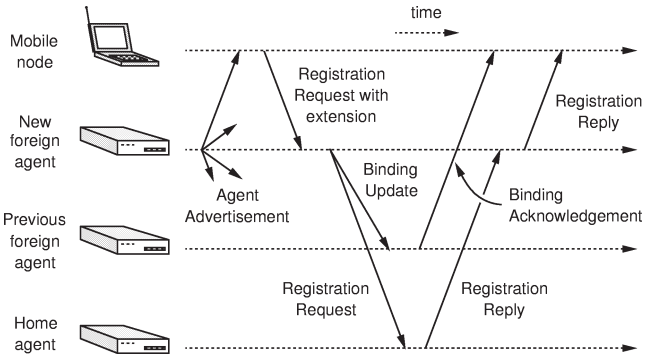

Figure 8. Smooth handoff.

entries for the mobile node, can also be recovered in this way. Finally, this notification allows any resources consumed by the mobile node at the previous foreign agent (such as an allocated radio channel) to be released immediately, rather than waiting for its registration lifetime to expire.

The mobile node requests that its new foreign agent attempt to notify its previous foreign agent on its behalf by including a *Previous Foreign Agent Notification* extension in its Registration Request message sent to the new foreign agent. The new foreign agent then builds a Binding Update message and transmits it to the mobile node's previous foreign agent during registration, requesting an acknowledgement from the previous foreign agent. The notification will typically include the mobile node's new care-of address, allowing the previous foreign agent to create a binding cache entry for the mobile node to serve as a *forwarding pointer* [9] to its new location. This process is illustrated in figure 8. The figure also shows how the binding cache maintenance messages are sent during the Mobile IP registration process. Any tunneled packets for the mobile node that arrive at its previous foreign agent after the forwarding pointer has been created can then be re-tunneled to the mobile node's new care-of address.

For this smooth handoff to be secure, during registration with a new foreign agent, the mobile node and the previous foreign agent must have a security association. The security association is used to authenticate the notification sent to the previous foreign agent.

The mobile node is responsible for occasionally retransmitting a Binding Update message to its previous foreign agent until the matching Binding Acknowledgement message is received, or until the mobile node can be sure that the foreign agent has expired its binding. The mobile node is likely to select a small timeout value for the lifetime available to such bindings sent to previous foreign agents.

4.2. Previous Foreign Agent Notification extension

The format of the Previous Foreign Agent Notification extension is illustrated in figure 9. The Previous Foreign Agent Notification extension contains only those values

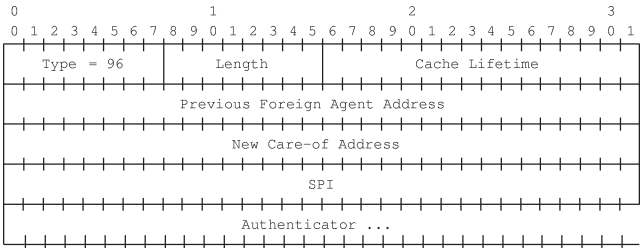

Figure 9. Previous Foreign Agent extension format.

not otherwise already contained in the Registration Request message that are needed for the new foreign agent to construct the Binding Update message. It contains the mobile node's new care-of address (or, sometimes, the mobile node's home address), the previous foreign agent's address, and sufficient information for the new foreign agent to build the expected Binding Update for the previous foreign agent. If the mobile node just wants the previous foreign agent to forget entirely about the mobile node, it uses its home address in the notification extension.

The cache lifetime is copied into the lifetime field of the Binding Update message, and is the number of seconds remaining before the binding cache entry created by the previous foreign agent must be considered expired. A value of zero indicates that the previous foreign agent should not create a binding cache entry for the mobile node once it has deleted the mobile node's registration in its visitor list.

When building the Binding Update, the new foreign agent must create a *Route Optimization Authentication* extension. The *SPI* and authenticator value are copied into the authentication extension by the new foreign agent. This authenticator is calculated by the mobile node only over the predicted Binding Update message body, using a security association shared with its previous foreign agent. The SPI used for the authentication calculation can be one created as part of the establishment of the registration key by one of the methods in section 7.

4.3. Smooth handoffs and Binding Acknowledgements

When a foreign agent receives a *Previous Foreign Agent Notification* message, it creates a *Binding Update* for the previous foreign agent, using the specified SPI and precomputed authenticator sent to it by the mobile node. The Binding Update message is also required to set the *acknowledge* bit, so that the previous foreign agent will know to send a *Binding Acknowledgement* message back to the mobile node.

When the previous foreign agent receives the Binding Update message, it will authenticate the message using the mobility security association indicated by the SPI. If the message authentication is correct, the visitor list entry for this mobile node at the previous foreign agent will be deleted and a Binding Acknowledgement message returned to the sender. In addition, if a new care-of address was included in the Binding Update message, the previous foreign agent will create a binding cache entry for the mobile node. The previous foreign agent can then tunnel packets to the mobile node's new care-of address using that binding cache entry, just as any node maintaining a binding cache.

This Binding Acknowledgement returned by the previous foreign agent is addressed to the mobile node, and thus must be tunneled using the new binding cache entry. The tunneled acknowledgement then should be delivered directly to the new foreign agent, without having to go to the home network. This creates an interesting problem for the new foreign agent when it receives the acknowledgement before the Registration Reply from the home agent. It is suggested that the new foreign agent deliver the acknowledgement to the mobile node anyway, even though the mobile node is technically unregistered. If there is concern that this provides a loophole for unauthorized traffic to the mobile node, the new foreign agent could limit the number of packets delivered to the unregistered mobile node to this single instance. Alternatively, a new extension to the Registration Reply message can be defined to carry along the acknowledgement from the previous foreign agent. This latter approach would have the benefit that fewer packets would be transmitted over bandwidth-constrained wireless media during registration.

When the Binding Acknowledgement message from the previous foreign agent is received by the new foreign agent, it decapsulates it and sends it to the mobile node. In this way, the mobile node can discover that its previous foreign agent has received the Binding Update message. The mobile node must be certain that its previous foreign agent has been notified about its new care-of address, because otherwise the previous foreign agent could become a "black hole" for packets destined for the mobile node based on out-of-date binding cache entries at other nodes. The new foreign agent has no further responsibility for helping to update the binding cache at the previous foreign agent, and does not retransmit the message even if no acknowledgement is received.

If the acknowledgement has not been received after sufficient time, the mobile node is responsible for retransmitting another Binding Update message to its previous foreign agent. Although the previous foreign agent may have already received and processed the Binding Update message (the Binding Acknowledgement message may have been lost in transit to the new foreign agent), the mobile node should continue to retransmit its Binding Update message until the previous foreign agent responds with a Binding Acknowledgement.

It is possible that the binding cache entry created by the previous foreign agent from the information in the Binding Update from the new foreign agent will be deleted from its cache at any time. In this case, the previous foreign agent will be unable to re-tunnel subsequently arriving tunneled packets for the mobile node, and would resort to using a *special tunnel* [16]. Mobile nodes are expected to assign small lifetimes to such bindings so that they will not take

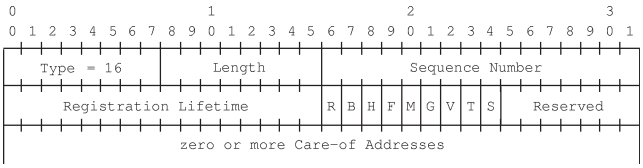

Figure 10. Modified Agent Advertisement message format.

up space in the foreign agent's binding cache for longer than necessary.

4.4. Modified Mobility Agent Advertisement extension

Performing smooth handoffs requires one minor change to the existing Mobile IP *Mobility Agent Advertisement* extension [15]. A new flag bit, the *smooth handoff* (S) bit, replaces a previously unused reserved bit in the extension, to indicate that the foreign agent supports smooth handoffs. By default, every foreign agent that supports smooth handoffs is expected to support at least the establishment of a registration key (see section 6) by using a *Diffie–Hellman* key exchange.

5. Mobility security association management

One of the most difficult aspects of Route Optimization for Mobile IP in the Internet today is that of providing authentication for all messages that affect the routing of packets to a mobile node. In the base Mobile IP protocol, only the home agent is aware of the mobile node's mobility binding and only the home agent tunnels packets to the mobile node. Thus, all routing of packets to the mobile node while away from its home network is controlled by the home agent. Authentication is currently achieved based on a manually established mobility security association between the home agent and the mobile node. Since the home agent and the mobile node are both owned by the same organization (both are assigned IP addresses within the same IP subnet), this manual configuration is manageable, and (for example) can be performed while the mobile node is at home.

However, with Route Optimization, authentication is more difficult to manage, since a Binding Update may in general need to be sent to almost any node in the Internet. Since no authentication or key distribution protocol is generally available in the Internet today, the Route Optimization procedures defined here may make use of the same type of manual key distribution as defined in the base Mobile IP protocol. For use with Route Optimization, a mobility security association held by a correspondent node or a foreign agent must include the same parameters (for instance, the style of replay protection) as required by base Mobile IP [15].

For a correspondent node to be able to create a binding cache entry for a mobile node, the correspondent node and the mobile node's home agent must have established a mobility security association. This mobility security association, though, could conceivably be used in creating and updating binding cache entries at this correspondent node for all mobile nodes served by this home agent. Doing so places the correspondent node in a fairly natural relationship with respect to the mobile nodes served by this home agent. For example, the mobile nodes may represent different people affiliated with the same organization owning the home agent, with which the user of the correspondent node often collaborates. The effort of establishing such a mobility security association with the relevant home agent may be more easily justified (as described below) than the effort of doing so with each individual mobile node. It is similarly possible for a home agent to have a manually established mobility security association with the foreign agents often used by its mobile nodes, or for a particular mobile node to have a manually established mobility security association with the foreign agents serving the foreign networks that it often visits.

In general, if the movement and communication patterns of a mobile node or the group of mobile nodes served by the same home agent are sufficient to justify establishing a mobility security association with the mobile node's home agent, users or network administrators are likely to do so. Rather than storing each mobility security association that it has established with many different correspondent nodes and foreign agents, a home agent may manage its mobility security associations so that each of them can be generated from a single *master key*. With the master key, the home agent could build a key for any given other node, e.g., by computing the node-specific key as

$$\text{MD5}(\text{node-address} \parallel \text{master-key} \parallel \text{node-address})$$

where node-address is the IP address of the particular node for which the home agent is building a key, and master-key is the single master key held by the home agent for all mobility security associations it has established with correspondent nodes. The node-specific key is built by computing an MD5 hash over a string consisting of the master key with the node-address concatenated as a prefix and as a suffix.

Using this scheme, when establishing each mobility security association, the network administrator managing the home agent computes the node-specific key and communicates this key to the network administrator of the other node through some secure channel, perhaps by telephone. The mobility security association is configured at this other node in the same way as any mobility security association. At the home agent, though, no record need be kept that this key has been given out. The home agent need only be configured to know that this scheme is in use for all of its mobility security associations, or perhaps only for a specific set of its mobile nodes. When the home agent needs a mobility security association as part of Route Optimization, it builds the node-specific key based on the master key and the IP address of the other node with which it is attempting to authenticate.

6. Registration key establishment messages

When a mobile node registers with a foreign agent, it typically does not yet share a security association with the foreign agent. However, in order for the foreign agent to process future Binding Updates that it may receive, it must have such a security association. As described earlier, smooth handoffs rely on the Previous Foreign Agent Notification extension (section 4.2), which precedes the transmission of a Binding Update to the previous foreign agent created by the mobile node after it moves. Binding Updates provide the essential mechanism for accomplishing smooth handoffs between a previous foreign agent and a new foreign agent.

Foreign agents are expected to be cheap and widely available as Mobile IP becomes fully deployed. Mobile nodes will likely find it difficult to manage long-term security relationships with so many foreign agents. To securely perform the operations needed for smooth handoffs from one foreign agent to the next, however, any careful foreign agent should require assurance that it is getting authentic handoff information and is not arranging to forward in-flight packets to a forged destination. The registration key establishment messages are used with the Mobile IP Registration Request and Registration Reply messages to establish some trustworthy secret (and SPI) between a mobile node and its foreign agent when none exists beforehand, while allowing the use of fully trustworthy security associations between foreign agents and mobile nodes whenever they do exist.

In Mobile IP, the mobile node often cannot verify the identity of the foreign agent in any absolute terms. It can only act on the presumption that the foreign agent is performing its duties by correctly implementing the protocol. The exact identity of the foreign agent is not crucial to the process of establishing a registration key. Only an agreement to follow the protocol can be expected or enforced. If the mobile node has a way to obtain a certified public key for the foreign agent, then the identity may be established in a firmer fashion, but the needed public key infrastructure is not yet available in the Internet today. Therefore, the methods described here enable a mobile node to create a registration key with an *anonymous* foreign agent (i.e., one whose identity we may be unable to verify) during the registration process. Several methods for establishing a registration key have been proposed, and other methods of establishing keys may become available in the future, using an Internet public key infrastructure or Kerberos. Currently, the following methods have been proposed:

1. If the foreign agent and mobile node share a security association, it can be used to secure the Previous Foreign Agent Notification without need to establish a registration key.

2. If the home agent and foreign agent share a security association, the home agent can choose the new registration key.

3. If the foreign agent has a public key, it can again use the home agent to supply a registration key.

4. If the mobile node includes its public key in its Registration Request, the foreign agent can choose the new registration key.

5. The mobile node and its foreign agent can execute a Diffie–Hellman key exchange protocol [5] as part of the registration protocol.

If a request for key establishment cannot be accommodated by the foreign agent and/or the home agent, then the mobile node's key request must go unfulfilled. This does not mean that the Registration Request itself fails, so it has no effect on the status code returned by the home agent to the mobile node. The mobile node must be able to handle the case in which it has requested a key but the Registration Reply arrives without any key reply extension. This could happen even when the foreign agent has advertised its willingness to offer smooth handoffs, and the mobile node has supplied all the necessary parameters (e.g., for a Diffie–Hellman key exchange).

Once the registration key is established, the method for performing smooth handoff is natural, as described earlier. In the remainder of this section, we give a brief overview of the proposed methods for establishing the registration key enumerated above.

6.1. The home agent as a KDC

The second and third methods listed above make use of the mobility security association shared between the home agent and mobile node, to encode the registration key for delivery to the mobile node. Thus, if the home agent can securely deliver the key to the foreign agent, it can be used as a *Key Distribution Center* (KDC) for the mobile node and its new foreign agent. The mobile node requests this by including a *Registration Key Request* extension in its Registration Request message. When the home agent chooses the registration key, it returns the key in two different extensions to the Registration Reply. One extension has the key encrypted for the foreign agent, and the other extension has the same key encrypted differently for the mobile node.

For the registration key to be established using this method, the home agent must be able to securely transmit an encrypted copy of the registration key to the foreign agent. This is straightforward if the foreign agent already has a mobility security association with the home agent. If mobile nodes from some home network often visit a foreign agent, then the effort of creating such a mobility security association between that foreign agent and the home agent serving their home network may be worthwhile.

If no such mobility security association exists, but the foreign agent has a public key available, it can still ask the home agent to use it to pick a registration key. This may be preferable to asking the mobile node to pick a good registration key, because doing so may depend upon

using resources not available to all mobile nodes; simply selecting cryptographically strong pseudo-random numbers may by itself be a significant computational burden [6]. Moreover, allowing the home agent to pick the key fits well into the existing registration procedures. On the other hand, it is possible that a mobile node could do with less than perfect pseudo-random numbers as long as the registration key were to be used in the restricted fashion envisioned here for smooth handoffs.

6.2. Using the foreign agent as a KDC

When the foreign agent and mobile node share a mobility security association, there is no need to pick a registration key. The mobile node can secure its Binding Update to the foreign agent whenever it needs to, by using the existing security association. This is the most desirable case.

Otherwise, if available, the mobile node can include its public key (such as RSA [18]) in its Registration Request to the foreign agent, using a *Mobile Node Public Key* extension. The foreign agent chooses the new registration key and includes a copy of it in the Registration Request, encrypted with the mobile node's public key, using a *Foreign-Mobile Registration Key Reply* extension.

6.3. Using Diffie–Hellman with the foreign agent

The Diffie–Hellman key-exchange algorithm [5,18] can be used. Diffie–Hellman is a public key cryptosystem that allows two parties to establish a shared secret key, such that the shared secret key cannot be determined by other parties overhearing the messages exchanged during the algorithm. It is already used, for example, in other protocols that require a key exchange, such as in the *Cellular Digital Packet Data* (CDPD) system [2].

Some applications of this technique are known to suffer from a *man-in-the-middle* attack. In other words, a malicious agent *could* pretend to the foreign agent to be the mobile node, and pretend to the mobile node to be the foreign agent, and participate as an unwanted third member in the key exchange. Armed with knowledge of the registration key, the malicious agent *could* at a later time disrupt the smooth handoff, or initiate the handoff prematurely. In Route Optimization, Diffie–Hellman results are authenticated by the home agent to frustrate such man-in-the-middle attacks. Moreover, the mobile node and/or the foreign agent are presumably in direct contact, so that such an attack is detectable if either of the nodes notices the reception of duplicate packets, and corrective action taken.

If Diffie–Hellman were not computationally expensive, it could likely serve the needs of most mobile nodes. But, the algorithm itself uses exponentiations involving numbers with hundreds of digits. That may take a long time for some mobile nodes to compute, time which might come at the expense of interactivity or convenient operation of user application programs. For this reason, Diffie–Hellman may be less desirable than some other methods for establishing registration keys. Since it requires no other configuration, it is nevertheless required in all implementations of foreign agents that advertise support for smooth handoffs.

Briefly, the Diffie–Hellman algorithm involves the use of two large public numbers, a *prime number* and a *generator*. The prime number and the generator must be known by both parties involved in the algorithm, but do not have to be secret; these values may be the same or different for each execution of the algorithm and are not used once the algorithm completes. Each party chooses a private random number, produces a *computed value* based on this random number, the prime and the generator, and sends the computed value in a message to the other party. Each party then computes the (same) shared secret key using its own private random number, the computed value received from the other party, and the prime and generator values. Knowing the computed values and not the chosen random numbers does not enable passive listeners to determine the shared secret key.

To use the Diffie–Hellman algorithm during registration with a foreign agent, the mobile node includes a *Registration Key Request* extension in its Registration Request message, containing its nonzero values for the prime and generator, along with the computed value from its own private random number. The foreign agent then chooses its own private random number and includes a *Diffie–Hellman Registration Key Reply* extension in its Registration Reply message to the mobile node; the extension includes the foreign agent's own computed value based on its chosen random number and the supplied prime and generator values from the mobile node. The mobile node and foreign agent each independently form the (same) shared secret key from their own chosen random number, the computed value supplied by the other party, and the prime and generator values.

Establishing a registration key using Diffie–Hellman is computationally expensive, but the use of Diffie–Hellman described here is designed to allow the Diffie–Hellman computations to be overlapped with other activities. The mobile node may choose (or be manually configured with) the prime and generator values at any time, or may use the same two values for a number of registrations. The mobile node may also choose its private random number and calculate its computed value at any time. For example, after completing one registration, the mobile node may choose the private random number for its next registration and begin the computation of its new computed value based on this random number, such that it has completed this computation before it is needed in its next registration. Even more simply, the mobile node may use the same private random number and computed value for any number of registrations. The foreign agent may choose its private random number and begin computation of its computed value based on this number as soon as it receives the mobile node's Registration Request message.

This method could be extended to support other similar key exchange algorithms, either by adding a new request and reply extension for each, or by adding a field in the extensions to indicate which algorithm is to be used.

7. Messages requesting a registration key

This section describes the extensions that may be used by mobile nodes or foreign agents to request the establishment of a registration key. See section 9 for appropriate algorithms which allow each node to tailor the use of these extensions to most closely fit its configured requirements.

7.1. Foreign Agent Key Request extension

If the foreign agent receives a Registration Key Request from a mobile node and it has a security association with the home agent, it may append the *Foreign Agent Key Request* extension to the Registration Request after the *Mobile-Home Authentication* extension. The home agent will use the SPI specified in the key request extension to encode the registration key in the subsequent Registration Reply message. The format of the Foreign Agent Key Request extension is illustrated in figure 11.

7.2. Mobile Node Public Key extension

If the mobile node has a public key, it can ask its prospective foreign agent to choose a registration key, and to use the mobile node's public key to encode the chosen registration key. No eavesdropper will be able to decode the registration key, even if it is broadcast to all entities with access to the network medium used by the mobile node. If using the public key, the foreign agent should still include the selected key in the Registration Request before it goes to the home agent. Then, the home agent can authenticate the selected encoded registration key as part of the Registration Reply message. The format of the *Mobile Node Public Key* extension is illustrated in figure 12.

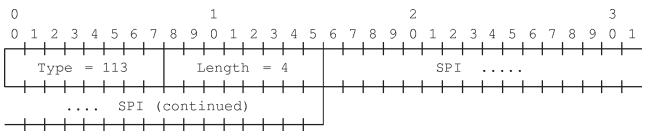

Figure 11. Foreign Agent Key Request extension format.

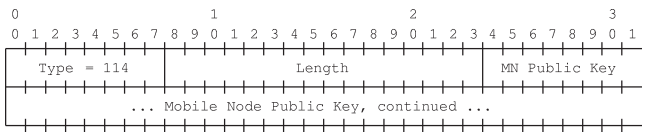

Figure 12. Mobile Node Public Key extension format.

7.3. Foreign Agent Public Key extension

The format of the *Foreign Agent Public Key* extension is illustrated in figure 13. Using this extension, if the foreign agent has a public key, it can ask the home agent to choose a registration key, and to use the foreign agent's public key to encode the chosen registration key. Then, the home agent can authenticate the selected encoded registration key as part of the Registration Reply message. The SPI field in the extension is provided for the home agent to transcribe into the eventual *Foreign Agent Public Key Reply* extension to the Registration Reply message.

7.4. Registration Key Request extension

The *Registration Key Request* extension, illustrated in figure 14, may be included in a Registration Request message sent to a foreign agent. If the length of the parameters in the key request extension are all zero, then the mobile node is asking the foreign agent to supply a key by any means it has available except for Diffie–Hellman.

If the lengths are nonzero, then the mobile node is enabling the foreign agent to also perform the Diffie–Hellman key exchange algorithm (as described in section 6.3) if the other possible key establishment methods are not available. The foreign agent should then select a good pseudo-random registration key, and include a *Diffie–Hellman Registration Key Reply* extension, in the Registration Request message sent to the home agent to complete the key exchange. The home agent will also include the same extension in the Registration Reply sent to the mobile node, which will then be authenticated as part of the reply message.

The *Prime* and *Generator* fields give the two public values for this execution of the Diffie–Hellman algorithm. The *Computed Value* is the public computed value from the mobile node for this Diffie–Hellman exchange. The values *Prime*, *Generator*, and *Computed Value* must all be the same length, which must be a multiple of 8 bits. Correspondingly, the *Length* field is a number which is 3 times the length (in bytes) of each of *Prime*, *Generator*, and *Computed Value*.

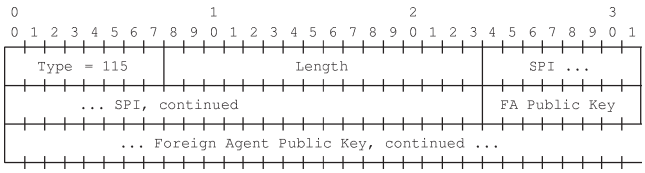

Figure 13. Foreign Agent Public Key extension format.

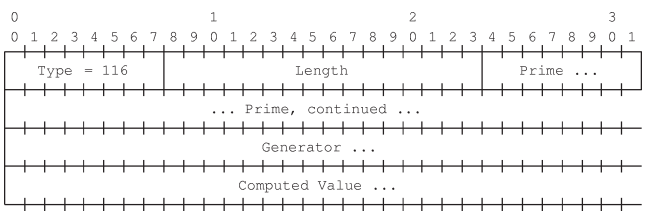

Figure 14. Registration Key Request extension format.

8. Extensions to supply a registration key

This section describes the extensions that may be used to supply a registration key to a requesting entity, either a foreign agent or a mobile node. These extensions are the counterparts to the corresponding extensions used to request registration keys that were described in the last section.

8.1. Home-Mobile Key Reply extension

The *Home-Mobile Key Reply* extension may be used in Registration Reply messages to send a registration key from the mobile node's home agent to the mobile node. When used, the home agent is required to also include a key reply extension in the Registration Reply message, which gives a copy of the same key to the mobile node's new foreign agent. The Home-Mobile Key Reply extension, illustrated in figure 15, is authenticated along with the rest of the Registration Reply message, and thus no additional authenticator is included in the extension. The SPI used to encode the registration key may be different from the SPI used to authenticate the Registration Reply message.

The *Mobile Node Encrypted Key* is the registration key, chosen by the home agent, encrypted under the mobility security association between the home agent and the mobile node. The same key must be sent, encrypted for the foreign agent in a Foreign Agent Key Reply extension in this Registration Reply message.

8.2. Foreign Agent Key Reply extension

The *Foreign Agent Key Reply* extension, illustrated in figure 16, may be used in Registration Reply messages to send a registration key from the mobile node's home agent to the mobile node's new foreign agent. The registration key is encrypted under the mobility security association between the home agent and the foreign agent. The key that is sent in this extension must also be sent by the home agent to the mobile node, encoded for the mobile node in a *Home-Mobile Key Reply* extension in the same Registration Reply message.

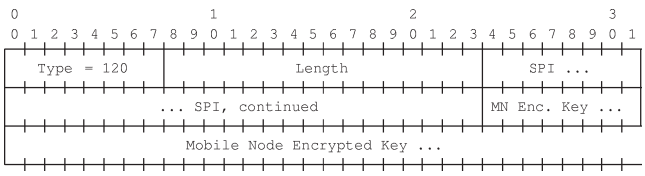

Figure 15. Home-Mobile Key Reply extension format.

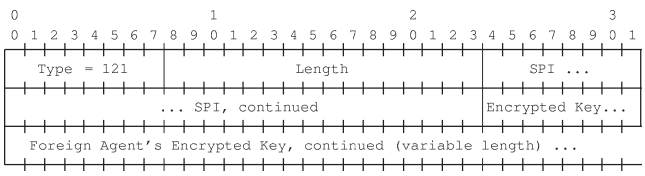

Figure 16. Foreign Agent Key Reply extension format.

Authentication of the key is performed by use of data within a Home-Foreign Authentication extension to the Registration Reply message, which is required when the Foreign Agent Key Reply extension is used. Replay protection is accomplished using the Identification field in the Registration Request message, which is also used by the foreign agent to identify the pending registration data.

8.3. Mobile Node Public Key Reply extension

The Mobile Node Public Key Reply extension is illustrated in figure 17. When the mobile node sends a *Mobile Node Public Key Request* to its prospective foreign agent, the foreign agent can immediately select a registration key. The foreign agent encodes this registration key into the *Mobile Node Public Key Reply* extension to the Registration Request, along with an SPI for future reference to the key. The home agent subsequently transcribes the extension without change into the Registration Reply message.

8.4. Foreign Agent Public Key Reply extension

In response to a *Foreign Agent Public Key Request* extension, the home agent will select a registration key and encode it into two separate key reply extensions of the Registration Reply message. The Foreign Agent Public Key Reply message is illustrated in figure 18. The Foreign Agent Public Key Reply extension contains the registration key encrypted with the public key of the foreign agent. The *Foreign Agent's Encrypted Key* is a pseudo-random number, chosen by the home agent, and encrypted using the foreign agent's public key. The SPI, provided by the foreign agent for transcribing into this extension, is ultimately targeted for use by the mobile node.

8.5. Diffie–Hellman Key Reply extension

The *Diffie–Hellman Registration Key Reply* extension, illustrated in figure 19, should be included in a Registration Request message sent by a foreign agent to the home agent, when the following conditions are met:

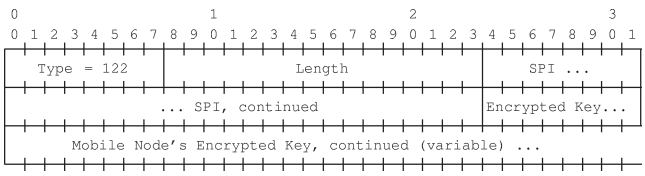

Figure 17. Mobile Node Public Key Reply extension format.

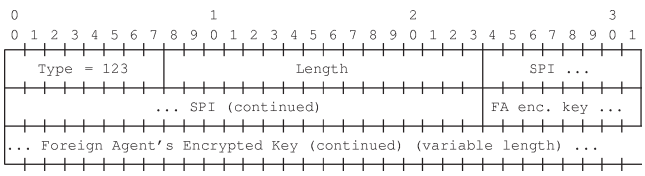

Figure 18. Foreign Agent Public Key Reply extension format.

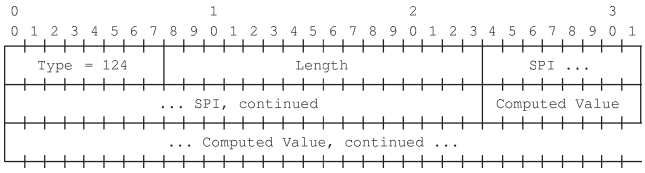

Figure 19. Diffie–Hellman Key Reply extension format.

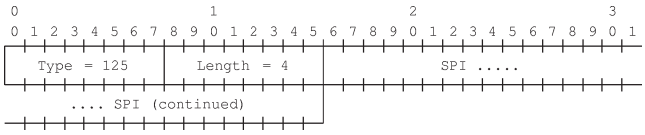

Figure 20. SPI extension format.

- the mobile node has included a *Registration Key Request* extension with nonzero prime and generator in its Registration Request message to the foreign agent, and
- the foreign agent has no public key or security association with the home agent or mobile node.

The *Computed Value* field gives the computed value from the foreign agent for this execution of the Diffie–Hellman algorithm. The values of the prime and generator are taken from the Registration Key Request extension from the mobile node's Registration Request message. The foreign agent supplies a new SPI along with the new registration key, so that the new key will be useful in the same way as registration keys created by any other method.

8.6. SPI extension

The SPI extension is included in Registration Reply messages when needed to specify the SPI to be associated with the registration key, which may be different from the SPI used to encrypt the key. The format of the SPI extension is shown in figure 20.

9. Registration Key Request processing

This section describes the processing steps taken by a mobile node, foreign agent, and home agent in establishing a registration key for the mobile node during registration. These steps use the message extensions described in section 8.

9.1. Mobile Node Key Requests

If the mobile node receives an Agent Advertisement from a foreign agent with the *smooth handoff* bit set, it may initiate a smooth handoff with its previous foreign agent, as well as asking its new foreign agent to aid in supplying a registration key for the new registration. The following algorithm may be used by the mobile node during registration to allow flexibility in the selection of the new registration key. Any particular mobile node may be configured to use one, none, or any subset of the key establishment procedures made available as part of the Route Optimization protocol.

1. If the *smooth handoff* bit is not set in the Agent Advertisement, do not include a Registration Key Request in the Mobile IP registration message, and skip all of the following steps.
2. If the mobile node has a registration key with the previous foreign agent, append the Previous Foreign Agent Notification extension.
3. If a security association exists with the new foreign agent, use it to secure any future Previous Foreign Agent Notification extensions.
4. Otherwise, if a public key is owned by the mobile node, append the Mobile Node Public Key Request extension.
5. Otherwise, if a Diffie–Hellman key exchange is desired, include a value for the prime, generator, and mobile node computed value in a Mobile Node Key Request extension.
6. Otherwise, append the Registration Key Request extension with empty (zero-length) prime, generator, and computed value.

In this way, the mobile node can get a registration key whenever one can be produced by any of the several available mechanisms.

9.2. Home agent processing for Registration Key Requests

When the home agent receives a Registration Request message, an extension requesting a registration key may be present in the message, asking the home agent to provide a registration key to the mobile node and its foreign agent, as described in section 6. In that event, the home agent employs a good algorithm for producing random keys [6] and encrypts the result separately for use by the foreign agent and by the mobile node. The chosen key is encrypted under the mobility security association shared between the home agent and the mobile node, and the encrypted key is placed in a Home-Mobile Registration Key Reply extension (section 8.1) in the Registration Reply message. The same key is also encrypted under the mobility security association shared between the home agent and the foreign agent, and the encrypted key is placed in a Home-Foreign Registration Key Reply extension (section 8.2) in the Registration Reply message.

If the home agent cannot satisfy a request to select a registration key, it may still satisfy the registration attempt. In this case, the home agent returns a Registration Reply message indicating success, but does not include any key reply extension.

9.3. Home agent supplying registration keys

When the home agent receives a Registration Request message with registration key extensions, it usually performs one of two operations:

- the home agent selects and encodes a registration key for both the mobile node and the foreign agent, or
- it transcribes the registration key already selected by the foreign agent into the appropriate extension to the Registration Reply message.

Either operation ensures that the mobile node and home agent are dealing with the same foreign agent.

When building the Registration Reply, the home agent should follow an algorithm such as the one suggested below, to be useful for the range of registration key establishment scenarios that are possible given the current route optimization protocol.

1. If the Registration Request contains extensions indicating that the foreign agent has already selected a registration key for use by the mobile node, copy those extensions into the Registration Reply.
2. Otherwise, if the Registration Request contains a Foreign Agent Key Request, then there is a security association with the foreign agent; append the Mobile Node Key Reply extension and the Foreign Agent Key Reply extension to the Registration Reply.
3. Otherwise, if a public key is available for the foreign agent (perhaps contained in the Foreign Agent Public Key extension), append the Mobile Node Key Reply extension and the Foreign Agent Public Key Reply extension to the Registration Reply.
4. Otherwise, do nothing. A registration key cannot be provided, but the registration can still be accepted according to the base Mobile IP specification.

In all cases, the home agent authenticates the Registration Key Reply extensions using the Home-Mobile Authentication extension.

9.4. Foreign agent processing for key requests

The foreign agent, when it receives a request from a mobile node for a registration key, is faced with a variety of possible actions. The action selected by the foreign agent depends on the resources it has available. The foreign agent typically attempts to reduce as much as possible the computational burden placed on the mobile node, but relies on the security association with the greatest cryptographic strength to encode the registration key. Furthermore, if the foreign agent performs the key selection, it still supplies the encoded key in an extension to the Registration Request message, so that the process of registration will also have the effect of authenticating its choice of registration key to the mobile node. This strategy reduces the opportunity for attackers to mount *man-in-the-middle* attacks.

The following steps illustrate an algorithm that may be used when a foreign agent gets a Registration Request with one of the key request extensions included.

1. If the Registration Request contains a Previous Foreign Agent Notification extension, perform the indicated function by building and transmitting the appropriate Binding Update message to the previous foreign agent.
2. If there is a security association with the home agent, append a Foreign Agent Key Request extension to the Registration Request.
3. Otherwise, if the foreign agent has a public key, append the Foreign Agent Public Key Request to the the Registration Request.
4. Otherwise, if the mobile node's public key is available, use it to pick a good registration key and append the Foreign Agent Public Key Reply extension.
5. Otherwise, if the mobile node has indicated that a Diffie–Hellman key exchange is desired, perform the computation and use the result when appending the Diffie–Hellman Key Reply extension to the Registration Request.
6. Otherwise, forward the Registration Request to the home agent without any further changes.

10. Route Optimization in IPv6

The development of Route Optimization techniques for IPv4 has played an important role in the development of mobility support for IP Version 6 (IPv6) [4,10]. Most IPv4 correspondent nodes will not support the processing of Binding Updates for years. IPv6 correspondent nodes, on the other hand, should be able to support the analogous features in IPv6 much sooner. There are three reasons for the difference:

- There is not a sizable deployment of IPv6 nodes to impede the introduction of appropriate protocol features.
- Efficient support of mobility was mandated by the IPng Directorate and the IESG (Internet Engineering Steering Group) [1].
- The base IPv6 protocol supports mobility more naturally than IPv4, so that there is less work needed overall to create products that implement the standard.

One of the main features of IPv6 that is most useful for mobility support is the requirement that all IPv6 nodes support authentication. Since the nodes have to perform authentication, APIs have been developed to enable the functions at the network protocol layer, and the APIs can be used to do key management. With keys distributed to the correspondent nodes, the mobile node can expect its correspondents to accept authenticated Binding Updates. When the correspondents are able to keep track of the mobile node's care-of addresses, they can (just as in IPv4 Route Optimization) cause packets to go directly to the mobile node without ever passing through the home network or home agent.

There are further advantages inherent in IPv6 for mobility support, many of which are not relevant to Route Optimization. One important feature, however, is the new organization for IPv6 options. Options can now be specified to be end-to-end, instead of hop-by-hop as with IPv4. For this reason, in IPv6, Binding Updates are delivered in end-to-end options (called *destination options*). Moreover, since normal packets can carry destination options, Binding Updates can be delivered to correspondent nodes without adding additional control packets to the network, thus reducing network load. Moreover, since intermediate nodes (routers) do not process the destination option, there is little routing penalty for the transmission of binding updates to correspondent nodes.

Given the expected universal deployment of authentication algorithms in IPv6 nodes, it is more reasonable to expect that a mobile node can maintain a security association with all its correspondents. For this reason, Binding Updates are delivered directly from the mobile node to the correspondent node in IPv6, in contrast to the method used in IPv4 whereby home agents are responsible for this chore. The mobile node can tell directly when a correspondent node needs a Binding Update. Whenever the mobile node moves to a new point of attachment to the Internet, any of its correspondents with active connections will need to get the new care-of address. The mobile node then inserts a Binding Update destination option into the next packet that it needs to transmit to any particular correspondent. Furthermore, the mobile node can tell if an incoming packet was originated by a correspondent node with an out-of-date or nonexistent binding for the mobile node, since such packets have to be tunneled to the mobile node from the home agent instead of directly from the correspondent node.

Lastly, the IPv6 analog for smooth handoffs from one point of attachment to the next deserves mention. Since each mobile node can acquire a care-of address by using Neighbor Discovery [12] and IPv6 address autoconfiguration [19], there is no need for any foreign agent to supply this information to the mobile node. With 128-bit addresses, there is not the same incentive for conserving the IPv6 address space, either. So, when a mobile node moves to a new network, there is no foreign agent left behind to help with forwarding data packets still in flight to the old network.

To solve this problem, the mobile node can deliver a Binding Update for its previous care-of address to the default router on its previous network. This default router need not even know anything about the mobile node's home address. In fact, the previous router acts as a home agent for the mobile node's previous care-of address.

11. Conclusions

In this paper, we have presented the current proposed protocol definition for *Route Optimization*, by which is meant the elimination of *triangle routing* whenever the correspondent node is able to perform the necessary protocol operations. The Route Optimization protocol definition is largely concerned with supplying a *Binding Update* to any correspondent node that needs one (and can process it correctly). The Binding Update message is also used in conjunction with the Previous Foreign Agent Notification extension to allow for smooth handoffs between foreign agents.

Furthermore, we have presented some methods for establishing registration keys for use by mobile nodes and foreign agents supporting smooth handoffs. We have given detailed processing requirements for mobile nodes, foreign agents, and home agents. Finally, the essential features of Route Optimization, as realized in IPv6, have been identified. We have discussed the differences between IPv4 and IPv6, with the hope that in so doing, the design space for Route Optimization will be more fully understood, and that IPv6's ability to support mobility will be more fully appreciated.

Acknowledgements

The work of David Johnson in this paper was supported in part by the National Science Foundation (NSF) under CAREER Award NCR-9502725, by the Air Force Materiel Command (AFMC) under DARPA contract numbers F19628-93-C-0193 and F19628-96-C-0061, and by the AT&T Foundation under a Special Purpose Grant in Science and Engineering. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of NSF, AFMC, DARPA, the AT&T Foundation, Sun Microsystems, Carnegie Mellon University, or the U.S. Government.

References

[1] S. Bradner and A. Mankin, The recommendation for the IP Next Generation Protocol, RFC 1752 (January 1995).
[2] CDPD consortium, *Cellular Digital Packet Data Specification* (Chicago, IL, 1993).
[3] S.E. Deering, ed., ICMP router discovery messages, RFC 1256 (September 1991).
[4] S. Deering and R. Hinden, Internet Protocol, version 6 (IPv6) specification, RFC 1883 (December 1995).
[5] W. Diffie and M. Hellman, New directions in cryptography, IEEE Transactions on Information Theory 22 (November 1976) 644–654.
[6] D.E. Eastlake, S.D. Crocker and J.I. Schiller, Randomness recommendations for security, RFC 1750 (December 1994).
[7] S. Hanks, T. Li, D. Farinacci and P. Traina, Generic Routing Encapsulation (GRE), RFC 1701 (October 1994).
[8] S. Hanks, T. Li, D. Farinacci and P. Traina, Generic routing encapsulation over IPv4 networks, RFC 1702 (October 1994).
[9] D.B. Johnson, Scalable and robust internetwork routing for mobile hosts, in: *Proceedings of the 14th International Conference on Distributed Computing Systems* (June 1994) pp. 2–11.
[10] D. Johnson and C. Perkins, Mobility support in IPv6, draft-ietf-mobileip-ipv6-06.txt (August 1998). (Work in progress.)
[11] H. Krawczyk, M. Bellare and R. Cannetti, HMAC: keyed-hashing for message authentication, RFC 2104 (February 1997).

[12] T. Narten, E. Nordmark and W. Simpson, Neighbor discovery for IP version 6 (IPv6), RFC 1970 (August 1996).
[13] C. Perkins, IP encapsulation within IP, RFC 2003 (May 1996).
[14] C. Perkins, Minimal encapsulation within IP, RFC 2004 (May 1996).
[15] C. Perkins, ed., IP mobility support, RFC 2002 (October 1996).
[16] C.E. Perkins and D.B. Johnson, Special tunnels for Mobile IP, draft-ietf-mobileip-spectun-00.txt (November 1997). (Work in progress.)
[17] J.B. Postel, ed., Internet control message protocol, RFC 792 (September 1981).
[18] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C* (Wiley, New York, 2nd ed., 1996).
[19] S. Thomson and T. Narten, IPv6 stateless address autoconfiguration, RFC 1971 (August 1996).

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 the editor for ACM/IEEE Transactions on Networking and for ACM/URSI/Baltzer Wireless Networks in the area of wireless 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 mobileip, svrloc, dhc (Dynamic Host Configuration) and IPng working groups. Charles is also associate editor for Mobile Communications and Computing Review, the official publication of ACM SIGMOBILE. He is serving on the Internet Architecture Board (IAB). Charles has 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 lead guest editor of an upcoming issue of the ACM/Baltzer journal Mobile Networks and Applications. Charles has served on various committees for the National Research Council, and is currently the chairperson of the Nomadicity Working Team of the Cross-Industry Working Team (XIWT). His previous projects included developing multiprocessor operating systems using Mach, and adapting Unix to fit on PDAs (personal digital assistants), multiprocessor systems and user interface prototyping systems. Charles holds a B.A. in mathematics and a M.E.E. degree from Rice University, and a M.A. in mathematics from Columbia University. He is a member of ISOC, ACM, IEEE, and the IETF.
E-mail: cperkins@eng.sun.com

David B. Johnson is an Assistant Professor in the School of Computer Science at Carnegie Mellon University. He also holds a courtesy faculty appointment in the Electrical and Computer Engineering Department at Carnegie Mellon, and is a member of CMU's Information Networking Institute. His research interests include network protocols, distributed systems, and operating systems. Prior to joining the faculty at CMU in 1992, he was on the faculty at Rice University for three years as a Research Scientist and Lecturer in the Computer Science Department. He received a B.A. in computer science and mathematical sciences in 1982, an M.S. in computer science in 1985, and a Ph.D. in computer science in 1990, all from Rice University. Professor Johnson is currently leading the Monarch Project at CMU, developing adaptive networking protocols and protocol interfaces to allow truly seamless wireless and mobile host networking. Related to this research, he has been active in the Internet Engineering Task Force (IETF) for many years and is one of the principal designers of the IETF Mobile IP protocol for IPv4 and IPv6. In 1997, he was Program Chain for the Third Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '97) and is currently Technical Vice Chair for Mobile Systems for the 19th International Conference on Distributed Computing Systems (ICDCS '99). He is Executive Committee member and the Treasurer for the ACM Special Interest Group on Mobile Computing and Communications (SIGMOBILE), an Area Editor for the ACM/Baltzer journal Mobile Networks and Applications (MONET) and the ACM SIGMOBILE magazine Mobile Computing and Communications Review (MC2R), and a Guest Editor for an upcoming issue of IEEE Journal on Selected Areas in Communications (J-SAC) on mobile ad hoc networking. He is a member of the IEEE Computer Society, IEEE Communications Society, ACM, USENIX, Sigma Xi, and the Internet Society.
E-mail: dbj@cs.cmu.edu