INTERNET-DRAFT
<draft-nikander-mobileip-homelessv6-01.txt>
Expires September 2001

P. Nikander
J. Lundberg
Ericsson NomadicLab
C. Candolin
T. Aura
Helsinki Univ. of Tech.

February 2001

Homeless Mobile IPv6

Status of this Memo

Abstract

This document specifies a variant of the Mobility Support in IPv6 protocol [MIP6], mainly intended for mobile hosts that do not need to or want to use any explicit home agent. However, portions of the protocol may be beneficial for other hosts as well, including non-mobile hosts, since it reduces the average header sizes for packets flowing between two mobile hosts or between a mobile and a non-mobile host. It may also be beneficial to multi-homed hosts or sites, since it allows migration of connections from one IP address to another.

The variant can interwork with hosts implementing standard Mobility Support in IPv6 [MIP6], and with hosts implementing the minimal conformance requirements of Section 7.1 of [MIP6].

Four fundamental design principles of the protocol are the following.

(1) Do not require, nor assume, any explicit home addresses or agents.
(2) Try to minimize the average header overhead caused by mobility.
  (2a) Try to assure that the headers are easily compressable.
(3) Try to minimize the changes to Mobile IPv6.

(4) Be backward compatible with Mobile IPv6.

However, it has not been a goal to be fully backward compatible with all applications. That is, while almost all applications should function without any changes, some applications may require application level modifications.

Table of Contents

1. Introduction

This memo specifies Homeless Mobile IPv6, a variant of the Mobility Support in IPv6 [MIP6] (aka Mobile IPv6) protocol. This variant provides mobility and handoff support for hosts that do not have any permanent home address, or that just want to take advantage of the smaller average header size of Homeless Mobile IPv6 vs. standard Mobile IPv6. Homeless Mobile IPv6 is backward compatible and can interwork with hosts that support only standard Mobile IPv6 [MIP6], or just the minimal interoperability requirements of Section 7.1 of [MIP6].

From a strictly technical point of view, Homeless Mobile IPv6 is basically a different way to implement Mobile IPv6, with just minimal protocol changes. In fact, almost all of this specification could stand without any protocol changes, but in that case some of the benefits of this specification would be lost.

On the other hand, this specification has profound implications to the semantics of upper layer protocols, including TCP and UDP, and to a lesser extent, to AH and ESP. Basically, as in standard Mobile IPv6, Homeless Mobile IPv6 allows the applications to maintain transport and higher-layer connections (as well as Security Associations) when a host changes its location, and therefore its IP address. However, the connections are not maintained by providing the upper layer protocols an illusion of a permanent (home) IP address (as in the case of standard Mobile IPv6), but by defining a way to securely maintaining the connections even when the underlying addresses do (visibly) change.

Even though this specification changes the semantics of TCP, UDP, AH, and ESP, it does NOT mandate any (or almost any) changes to the actual TCP or UDP implementations, and the applications (with some exceptions) will work unchanged. The need for changes in the IPSEC protocols, AH and ESP, depend on their implementation. In our estimate, most current IPSEC implementations could be used with this specification without any changes, but they would benefit from changes (see Section 2.4).

Homeless Mobile IPv6 proposes three new Sub-Options. However, only

one of these is crusial (the Security Sub-Option), and we think that it should be added even to the Standard Mobile IPv6. Homeless Mobile IPv6 it does not require any changes to IPv6 routers (other than support for standard Mobile IPv6 at the access routers, as defined in MIPv6 Sec. 7.1 and 7.2 of; see also MIPv6 Sec. 10.9). Homeless Mobile IPv6 relaxes some of the requirements of Mobile IPv6 specification by relying on more state and intelligence on the IP layer of the communicating end hosts.

The basic assumption behind Homeless Mobile IPv6 is that there will be a large number of mobile IPv6 hosts that do not conceptually have any home network or home addresses. Some of these hosts may be client-only hosts, without any home agent kind of functionality, while others may rely on some other means of providing accessibility information, e.g., Dynamic DNS [RFC2136] or SIP [SIP]. Thus, instead of using a Binding Cache to bind a care-of-address to some (arbitrary) home address, Homeless Mobile IPv6 uses a Host Cache (see Sec. 2.1) that keeps up information about IP addresses that belong to a single host. The Host Cache allows the hosts to maintain transport and higher layer connections even if the IP addresses change.

This document should be considered as an appendix to or variant of the Mobility Support in IPv6 [MIP6] specification, and read together with it. In the following, references to [MIP6] are expressed as "MIPv6, Sec N.N"

1.1. Applicability

In principle, the implementation techniques suggested in this specification MAY be applied to any IPv6 protocol stack. This specification is NOT applicable to IPv4. Currently, this specification is purely EXPERIMENTAL in nature.

1.2. New and Renamed Architectural Entities

Homeless Mobile host
: A mobile host that implements the Homeless Mobile IPv6 variant of the Mobile IPv6 protocol as specified in this document

Homeless (supporting) Host (or just Homeless Host)
: A mobile or non-mobile host that implements the Homeless Mobile IPv6 variant of the Mobile IPv6 protocol.

Standard Mobile Host
: A mobile host that does not implement the Homeless Mobile IPv6 variant of the Mobile IPv6 protocol but that is conformant with all of the Mobile IPv6 specification.

Standard Host
: A mobile or non-mobile host that does not implement the Homeless Mobile IPv6 variant of the Mobile IPv6 protocol

but that is conformant with the Mobile IPv6 specification, either all of MIPv6 or just the minimal host requirements as given in MIPv6 Sec 7.1.

1.3. Terminology

Host Cache
: A cache maintained by all Homeless (supporting) Hosts. A Host Cache Entry contains a list of IP addresses that are known (or assumed) to belong to a single host or Host Personality.

Host Cache Entry
: An entry in the Host Cache. A Host Cache Entry contains a number of IP addresses. There is a one-to-one mapping between known Host Cache Entries and Host Personalities (see below).

Local Host Cache Entry
: A Host Cache Entry that contains IP addresses that are local to the host. A host MAY have several Local Host Cache Entries, e.g., to provide several Host Personalities or to differentiate between scoped domains.

Foreign Host Cache Entry
: A Host Cache Entry that contains IP addresses that are not local to the host. For each remote host, or Host Personality, there MUST be only one Foreign Host Cache Entry.

  [Currently, a Foreign Host Cache Entry SHOULD contain only addresses that belong to a single scope and a single scoped domain, as the effects of mixing scoped addresses are unclear. TBD.]

Host Personality
: A logical host identity. A single physical host may have several Host Personalities. In some cases, a cluster of physical hosts may be represented as a single Host Personality. However, this document does not specify any means or methods for how the members of such a cluster may need to co-ordinate their internal states in order to be able to provide such an appearance to the upper layer protocols.

  Host Personalities are NOT real entities, but purely imaginary objects, brought into life by creating Host Cache Entries and destroyed by Host Cache Entry timeouts. Usually (but not necessarily) there is a one-to-one mapping between Host Personalities and reapl physical hosts.

Active Address
: An IP address (in a Host Cache Entry) which can be freely used. Active addresses may be used both for sending new packets and matching received packets.

Tentative Address
: An IP address (in a Host Cache Entry) whose "ownership" status has not been cleared. See Sections 2.4.1. and 3.4.2.

Deprecated Address
: An IP address (in a Host Cache Entry) which has expired.

Activity-Related Lifetime
: IP address lifetime that depends on the last use of the address. An Activity-Related Lifetime specified that the address expires when it has been idle for a certain period of time. In this context, an address is idle if no packets are received with it as a source address.

Absolute Expiration Time
: IP address lifetime that depends on wall clock. An Absolute Expieration Time specifies that the address expires at a specific point of time unless renewed. The address expires even if it is not idle.

Anonymous Authentication Protocol (AAP)
: A lightweight and weak (or at least weaker than IKE) authentication protocol that can be used even in an environment where there is no infrastructure. See Section 2.4.3.

Anonymous Security Association (Anonymous SA)
: An IPsec Security Association created as a result of running the Anonymous Authentication Protocol.

1.4. Protocol Overview

In what follows, we present an overview of functions of the Host Cache in Homeless Hosts, followed by a figure illustrating the data structures that are typically needed to implement Homeless Mobile IPv6. Thereafter, we give an overview of how Mobile IPv6 Binding Updates are used to update the Host Cache, and outline the methods for sending and receiving arbitrary IP packets. Example packet flows are given in Appendix A.

All Homeless (supporting) Hosts maintain a Host Cache. Basically, the Host Cache replaces and enhances the functionality provided by MIPv6 Binding Cache and Binding Update List. Packets transmitted by remote Mobile Hosts, either standard or homeless, may update entries in each Homeless Host's Host Cache.

An entry in the Host Cache contains a list of IP addresses that are known (or assumed) to belong to a single host or Host Personality. An initial Host Cache Entry MAY be based on a DNS reply containing multiple A6 [RFC2874] or AAAA records [RFC1886]. As a host receives Binding Updates from its peer, it SHOULD add, update, and

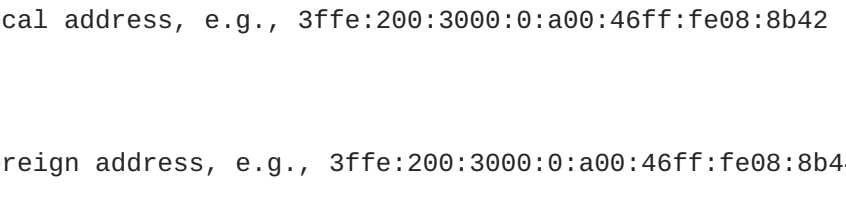
delete IP addresses in the Host Cache as specified in Sections 2.1 and 2.3. Individual IP addresses in the Host Cache expire as defined in Section 2.1.4. To prevent expiration of its IP addresses, stored in a host cache entry at its peers, a mobile host SHOULD periodically send to its peers packets containing Binding Updates.

At the protocol level, there are two major differences from MIPv6:

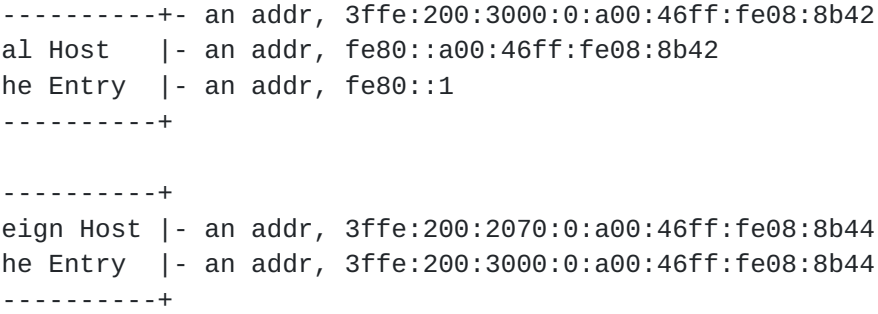
(1) Since the Homeless Host is considered not to have a home, it is always Away from Home (MIPv6 Sec 10.1, Sec 10.3). However, when communicating with another Homeless Host, neither Routing Headers nor Home Address Destination Options are needed.

(2) There are three new Destination Option Sub-Options, Homeless Support Sub-Option, which MAY be sent in a Binding Request or a Binding Update, Alternate Prefixes Sub-Option, which MAY be sent in a Binding Update, and Security Challenge/Response Sub-Option, which MAY be sent in a Binding Acknowledgement or Binding Update.

The main difference to the standard Mobile IPv6 implementation lies in the conceptual data structures needed for the implementation. That is, in a Homeless Host, TCP and UDP protocol control blocks (and possibly AH and ESP Security Associations) are not bound to single IP addresses but to Host Cache Entries. The difference is illustrated in Figure 1, below. It allows two communicating Homeless Hosts to independently select the source and destination IP addresses on packet bases, thereby allowing seamless handover and easy adaptation to local network conditions.



Standard Host Protocol Control Block Bindings



Homeless Host Protocol Control Block Bindings

Figure 1. Differences between Standard and Homeless
Host Conceptual Kernel Data Structures

Initially, a Local Host Cache Entry consists of (a subset of) the known local IP addresses. Correspondingly, a newly created Foreign Host Cache Entry consists either of the single address by which the host is known, or, OPTIONALLY, it MAY consist of several addresses as received by some other means, e.g., in a DNS reply.

A Homeless (supporting) Host MAY, at any time, send Binding Updates to any other Homeless (supporting) Host, thereby attempting to modify the contents of the other host's Host Cache. The Binding Updates MUST be authenticated as specified in MIPv6, Sec 4.4. As a difference to the standard Mobile IPv6, the Binding Updates do not simply replace a single current Care-Of-Address (CoA), but may add, delete, or change entries in the corresponding Host Cache Entry. As in standard Mobile IPv6, the host sending Binding Updates MAY request the receiver to reply with Binding Acknowledgements, see MIPv6 Sec. 4.2. As an additional requirement, before accepting a new CoA as an Active Address, the receiving host MUST check that the peer host is actually reachable at it; see Sec 2.4.1. and 3.4.2.

When sending packets to a Homeless (supporting) Host, the destination address for the packet MAY be freely selected among the Active Addresses in the Foreign Host Cache Entry. The selection MAY be performed separately for each packet. It is RECOMMENDED that the source address of the packet last received from the peer is used as the destination address of packets sent to the peer. Similarily, the source address for each packet MAY be individually selected among those in the Local Host Cache Entry as long as the rules specified in [ADDRSEL] are followed. Especially, the source address MUST be of the same scope as the destination address is. If the scope of the source address is smaller than the global scope, the sending host MUST know that the source and the destination addresses really belong to the same scoped domain. This specification does not offer any means to gain that kind of knowledge.

It is necessary to be more strict when sending ICMP, ND, and other control packets. When sending packets, the Host Cache SHOULD NOT be used for ICMP or ND at all.

When receiving packets from a Homeless (supporting) Host, each packet's destination address MUST be one of the addresses in a Local Host Cache Entry. In other words, when receiving a packet, a Homeless (supporting) Host makes sure that the unicast destination address belongs to at least one of its Local Host Cache Entries. [It is currently unspecified which Local Host Cache Entry the host should select if the destination address matches an address that is present in several Local Host Cache Entries. It looks like this selection should depend on both the interface through which the

packet was received, and possibly also the matching Foreign Host Cache Entry. The real problem here is that we would like to eventually have all Host Cache Entries to carry addresses from several distinct scopes, and that always makes the global addresses to be included in several Local Host Cache Entries. TBD.]

Similarily, in order to be accepted as a packet from a specific host, and thereby being part of a specific connection, the source address of the packet MAY be any of the Active or Deprecated Addresses in the corresponding Foreign Host Cache Entry. That is, the receiving host takes the source address of the received packet, looks for a Foreign Host Cache Entry in its host cache, and if a match is found, uses the Foreign Host Cache Entry to locate the specific protocol control block (pcb) representing the upper layer protocol connection (socket) the packet should be delivered to.

A Tentative Address may be used to positively match a Host Cache Entry if and only if it contains a reachability Response as specified in Sec 3.4.2.

If the source address does not match any Foreign Host Cache Entries in the Host Cache, the receiving host SHOULD NOT create a Foreign Host Cache Entry for the host until it has made sure that the host is actually reachable, and responds to packets sent to it (see Sec 4 to discuss the ramifications of the Host Cache and potential resource exhausting Denial-of-Service attacks). In such a case, a statically allocated Host Cache Entry MAY be used in order to match the packet against partially bound protocol control blocks.

It is RECOMMENDED that the receiving host markes the last used source address of received packets in the corresponding Foreign Host Cache Entry, and uses it as the destination address of the next packet sent to the host.

To maintain full backward compatibility, whenever a Homeless Host communicates with a host that does NOT support Homeless Mobile IPv6, standard Mobile IPv6 facilities must be used. This is specified in more detail in Sections 2.2. and 2.3.

2. Homeless Mobile IPv6 Functions

2.1. Maintaining Host Cache

2.1.1. Creating Host Cache Entries

Local Host Cache Entries are only created through administrative actions. It is RECOMMENDED that each host, by default, creates a single default Local Host Cache Entry. It is also RECOMMENDED that this default Local Host Cache Entry contains all local IP addresses of the host, including the loopback address (::1) and Link Local loopback address (fe80::1). All addresses in the Local Host Cache

Entry are permanent, i.e., they do not expire (unless, of course, the underlying IP addresses expire, e.g., since the prefix expires).

Whenever a Homeless (supporting) Host decides to send a packet to a host that has no Host Cache Entry, it MAY create a new Foreign Host Cache Entry for the host. It is RECOMMENDED that if the upper layer protocol is a user data oriented protocol (e.g. TCP or UDP), and the sending host is initiating the connection, the corresponding Host Cache Entry is created; however, if the sending host is only responding to a connection request, or if the upper layer protocol is control oriented protocol (e.g. ICMP, ND), no Host Cache Entry is created. The case of IPSEC protocols, ESP and AH, are discussed later in Section 2.4.

When a host decides to create a new Foreign Host Cache Entry, the new entry MUST, at minimum, contain one address. If the host has several addresses that are known to belong to a single Host Personality, the newly created Foreign Host Cache Entry MAY contain more than one address. All the addresses initially added to the Foreign Host Cache Entry MUST be temporary and have Activity-Related Lifetime, and SHOULD expire after they have been inactive for DEFAULT-LIFETIME seconds, unless there is some better information available, e.g., DNS caching time.

[For the time being, it is RECOMMENDED that each Foreign Host Cache Entry contains addresses that belong only to a single scope, and within that scope to a single known scoped domain. TBD.]

2.1.2. Sending Binding Updates

When a host creates a new Foreign Host Cache Entry as a result of receiving a packet and replying to it, it is RECOMMENDED that the host includes an Initial Binding Update (see Sec 3.2.1) in the first packet sent to the host. (This requires, naturally, that there is an AH SA between the hosts. See also Section 2.4). On the other hand, if a host creates a new Foreign Host Cache Entry as a result of sending a packet to a previously unknown host, it is RECOMMENDED that an Initial Binding Update is sent only after getting a reply from the host. In this case, if the remote host follows the recommendations of this specification, the reply from the remote host will contain an Initial Binding Update, which, in turn, may be used to trigger sending an Initial Binding Update back.

The Initial Binding Update SHOULD include all those addresses of the sending host that have the same scope and belong to the same scoped domain than the packet destination address, including the address used as the source address of the packet. The Initial Binding Update SHOULD also include a request for a Binding Acknowledgement. The Initial Binding Update MUST be sent in a way that is compatible with standard Mobile IPv6. The exact packet

format is specified in Sec. 3.2.1.

Whenever a host learns a new local IP address, e.g., due to physical discovery of a new network, or through neighbor discovery, it SHOULD include a Consequtive Binding Update (see Sec. 3.2.2) in the next packets sent to such hosts in the Host Cache that belong to the same scope and scoped domain as the newly discovered address. [If the host has several Local Host Cache Entries, the situation is apparently not that simple. TBD.] Additionally, the host MAY have a timer so that it will send a Binding Update anyway, after a configurable timeout, in the case there is currently no active traffic between the hosts.

Whenever a host learns that it has lost a local IP address, e.g., due to having lost radio connectivity, it MUST send a deleting Binding Update to its peers, i.e., a Binding Update with zero lifetime, thereby removing the address from their Host Cache Entries. Additionally, it MAY establish forwarding from the lost address, as defined in MIPv6 Sec. 10.9. Naturally, if the lost address was the last address, the host cannot send Binding Updates or establish forwarding. However, even in such a case the host SHOULD follow its Foreign Host Cache Entries, deleting addresses as their expire, and assume that its peers will delete addresses from their Foreign Host Cache Entries as they expire.

2.1.3. Adding Addresses to Host Cache Entries

Whenever a host learns a new local IP address, e.g., due to physical discovery of a new network, or through neighbor discovery, it MUST add it to (at least one of) the Local Host Cache Entry. If the local address has natural lifetime, the lifetime in the Host Cache SHOULD reflect this lifetime.

Whenever a host receives a packet with a Binding Update, and it has a Foreign Host Cache Entry that matches with either the source address of the packet, or the home address in the packet if the Home Address Destination Option is present, the receiving host SHOULD update its Host Cache, as specified in Sec. 2.3.

2.1.4. Removing Addresses from Host Cache Entries

All addresses in a Foreign Host Cache Entry have a defined lifetime. Basically, the host sending a Binding Update SHOULD determine the lifetime for the addresses as defined in MIPv6 Sec 10.8. That is, the lifetime of those kinds of addresses is given as Absolute Expiration Time, i.e., the addresses expire independent of whether they are in active use or not, unless the are explicitly renewed through a Binding Update. As specified in MIPv6 Sec 5.1 and 8.2, a host MAY remove entries from its peers' Host Cache by sending a Binding Update that has a zero lifetime.

Addresses that are entered to a Foreign Host Cache Entry through other means than as a result of processing a received Binding Update SHOULD have a Activity-Related Lifetime of DEFAULT-LIFETIME seconds, unless there is better information about the lifetime through some other means, e.g., DNS. That is, in the default case, the addresses expire when they have not been used for DEFAULT-LIFETIME seconds. If, after entering an address to the Host Cache through some other means than through a Binding Update, and a Binding Update is later received for that address, the lifetime of the address MUST be changed to have an Absolute Expiration Time with the lifetime given in the Binding Update.

Whenever an address expires, it MAY be removed from the Host Cache Entry. However, it is RECOMMENDED that the address is kept (as an deprecated address) for some time, to make it easier to react if the address is still used.

An deprecated address MUST NOT be kept in the Host Cache for longer than MAXIMUM-EXPIRED-LIFETIME seconds.

2.1.5. Removing Host Cache Entries

When the last address in a Host Cache Entry expires, as specified in Sec. 2.1.4., the host MAY remove the Host Cache Entry. However, it is RECOMMENDED that the Host Cache Entry is only removed when the last deprecated address is removed and all open connections referring to the Host Cache Entry are closed.

2.2. Sending Packets

When a Homeless (supporting) Host sends a packet, the amount of information about the destination host may differ. Basically, there are four alternatives:

- the destination host is known to be a Homeless (supporting) Host; there is no difference whether it is a Homeless Mobile Host or a non-mobile Homeless (supporting) Host.
- the destination host is known to be a Standard Mobile Host
- the destination host is known to be a Standard non-mobile Host
- the status of the destination host is not known

(See Appendix D)

In some cases, it is possible that the host wants to send a packet using a Foreign Host Cache Entry where all the addresses are deprecated. Currently, it is RECOMMENDED that in such a case the IP layer behaves as if there was an ICMP Destination Unreachable received.

2.2.1. Sending Packets to a Homeless (supporting) Hosts

When a Homeless (supporting) Host sends a packet to another Homeless (supporting) Host, it may freely select the destination

address from the Active Addresses included in the Foreign Host Cache Entry. Additionally, it may freely select the source address from the addresses in the associated Local Host Cache Entry, as long as the selected source address matches the scope and the scoped domain of the selected destination address as defined in [ADDRSEL].

If the sending host has learned any new local IP addresses since it has last sent a packet to the destination host, it SHOULD include a Binding Update adding the address to the Foreign Host Cache Entry in the destination host (as specified in Section 2.1.2.), or use the technique described in Appendix B. In both cases the packet MUST be protected with AH.

If the sending host wants to use, as the source address of the packet, a new local IP address that has not yet been sent in a Binding Update to the remote host, it can do it in either one of the following ways.

- It MUST include both a Home Address Destination Option containing one of the previously registered addresses, and a Binding Update registering the new source address. The Binding Update SHOULD contain a lifetime that is greater than zero, and it SHOULD have the A-bit (request for Binding Acknowledgement) set. The packet MUST be protected with AH.
- It can use the Binding Update optimization technique described in Appendix B.

If the sending host has lost any local IP addresses since it has last sent a packet to the destination host, it MUST include a Binding Update removing the IP address from the destination host's Host Cache, and the Binding Update SHOULD have the A-bit (request for Binding Acknowledgement) set. The packet MUST be protected with AH. Additionally, it MAY establish forwarding from the lost address, as defined in MIPv6 Sec. 10.9. If the newly lost address is the address which was used as the source address when the previous packet was sent to the correspondent host, the Binding Update SHOULD be sent immediately to remove the lost address.

It is important to note here that no Routing Extensions or Home Address Destination Options are needed in the communication between two Homeless (supporting) Hosts. This means that the average IPv6 header size is just 40 bytes of IP header instead of 40+28+24 bytes of IP header + Routing header + Destination Option header.

2.2.2. Sending Packets to Standard Mobile Hosts

When a Homeless (supporting) Host sends a packet to a Standard Mobile Host, there are basically two cases. If the Standard Mobile Host is at its home network, the case is identical to the Sending Packets to a Standard Host case, and specified in the next Section.

When a Homeless (supporting) Host sends packets to a Standard Mobile Host which is Away from Home (MIPv6 Sec 10.1, Sec 10.3), the sending host MUST include a Routing header as specified in MIPv6 Sec 8.9. Furthermore, the Homeless (supporting) Host must provide an illusion of having a Home Address to the Standard Mobile Host. That is, if the Homeless (supporting) Host decides to use another source IP address than the one the Standard Mobile Host assumes to be the Homeless (supporting) Host's Home Address, the sending host MUST include a Home Address Destination Option to the outgoing packet, and additionally MUST keep sending Binding Updates until a Binding Acknowledgement is received. If a Binding Update is included, the packet MUST be protected with AH.

It is RECOMMENDED that the illusion of being a Standard Mobile Host is provided on a connection-by-connection basis, since a connection-by-connection based solution provides potentially smaller average header size. That is, it is RECOMMENDED that whenever opening a new connection with an already known Standard Mobile Host, the Homeless Mobile Host selects the source address used in the connection independent on the source address used in other connections with the same host. In this way, there is high probability that the new connection will not need Home Address Destination Options even if some of the existing connections do need.

A suggested way to implement the Mobile IPv6 compatible functionality for destination addresses is to mark one of the addresses in the Foreign Host Cache Entry of a Standard Mobile Host as a Home Address, and another address as the current Care-of-Address. The existense of an address marked as a Care-of-Address forces the destination address selection to select the CoA as the destination address. The existense of an address marked as a Home Address signals the packet output routine to include a Routing header containing the Home Address.
A suggested way to implement the Mobile IPv6 compatible functionality for source addresses is to record in the protocol control blocks a pointer to a data structure that contains the following information:
 - the initially selected source address, i.e., emulated Home Address
 - the currently used source address, i.e., emulated Care-of-Address
This data structure may be shared between all protocol control blocks that have the same initially selected source address.

When sending a packet, the sending host compares the selected source address to the initially selected source address. If they are equal, no further processing is needed. If they differ, a Home Address Destination Option needs to be included in the packet. In addition to including the Home Address Destination Option, the sending host compares the selected source address to the emulated Care-of-Address. If they do not match, a Binding Update Destination Option must be included in addition to the Home Address

Destination Option.

2.2.3. Sending Packets to Standard Hosts

When sending packets to a Standard non-mobile Host (or to a Standard Mobile Host currently At Home), the Foreign Host Cache Entry contains only one non-expired address. This address is THE address of the Standard host, or the home address of the Standard Mobile Host.

In this case, the Homeless (supporting) Host MUST emulate a Standard Mobile Host in order to support application portability. That is, whenever the Homeless (supporting) host wants to use a local address different from the initial source address, it MUST include a Home Address Destination Option, and include Binding Update Destination Options until a Binding Acknowledgement is received, as defined above in the previous Section.

2.2.4. Sending Packets to hosts with unknown capabilities

When sending packets to a new host whose capabilities are not known, the sending host MAY send an Initial Binding Update whose purpose is twofold:

- to inform the receiving host that the sending host is a Homeless (supporting) Host, and
- to trigger a similar Binding Update from the receiving host in the case it is a Homeless (supporting) host.

This must be made in a way that is compatible with standard Mobile IPv6. The format of the Initial Binding Update is defined in Section 3.2.1.

2.3. Receiving Packets

When a Homeless (supporting) Host receives a packet, the amount of information about the source host may differ. Basically, there are four alternatives:

- the source host is known to be a Homeless (supporting) Host since there is an existing Foreign Host Cache Entry that contains the source address in the packet and the host has indicated its support for Homeless Mobile IPv6; there is no difference whether it is a Homeless Mobile Host or a non-mobile Homeless (supporting) Host.
- the source host is known to be a Standard Mobile Host since there is an existing Foreign Host Cache Entry that contains the source address in the packet, the host has not indicated to support Homeless Mobile IPv6, but it has sent either Home Address Destination Options or Binding Updates.
- the source host is either a Standard Host or a Standard Mobile Host; there is an existing Foreign Host Cache Entry that contains the source address in the packet, the host is known not to support Homeless Mobile IPv6, and the host has not (yet) sent Home Address Destination Options or Binding Updates.
- there is no Foreign Host Cache Entries that would match the

source address of the packet.

2.3.1. Receiving packets from other Homeless Hosts

When a Homeless (supporting) Host receives a packet from another, already known Homeless (supporting) Host, the source address (or the address in the Home Address Destination Option) matches to a Foreign Host Cache Entry that denotes that the sending host is a Homeless (supporting) Host.

If the packet contains a Home Address Destination Option, it MUST also contain a Binding Update, the packet MUST be protected with AH, and the Binding Update SHOULD contain a lifetime that is greater than zero. In such a case, the receiving host SHOULD add the addresses specified in the Binding Update to the Foreign Host Cache Entry. That is, since the packet contains a Home Address Destination Option, the source address of the packet is usually the one added to the Foreign Host Cache Entry. All the addresses MUST be added as Tentative Addresses, and verified as specified in Sec 3.4.2.

Any received packet MAY contain a Binding Update. If the packet contains a Binding Update, it MUST be protected with AH. When processing the Binding Update, the receiving host SHOULD add address(es) to or delete address(es) from the Foreign Host Cache Entry, as specified by the Binding Update. When adding addresses, they MUST be added as Tentative Addresses and verified, as specified in Sec 3.4.2, before their status can be changed to Active.

2.3.2. Receiving packets from Standard Mobile Hosts

Packets that are received from a Standard Mobile Host can be divided into two categories:
- The correspondent node is away from home.
- The correspondent node is at its home network.

When a packet is received from a Standard Mobile Host that is away from home, the packet includes a Home Address Option. The packet SHOULD be matched to a Foreign Host Cache Entry using the address received in the Home Address Option.

Packets received from a Standard Mobile Host that do not include the Home Address Destination Option are handled as if they were received from a Standard Host (Sec 2.3.3).

When a packet containing a Binding Update is received from a Standard Mobile Host, the address received in the Binding Update SHOULD be marked as the current Care-of-Address of the correspondent host as specified in Sec 2.2.2. The Binding Update must be protected with AH.

2.3.3. Receiving packets from Standard Hosts

When receiving packets from a host that is supposed to be a non-mobile Standard Host, the packet is typically a standard IP packet without any Home Address Destination Options or Binding Update Destination Options. In this case, the packet is handled normally, and the associated protocol control block is found through the associated Foreign Host Cache Entry as specified above.

However, since Standard Mobile Hosts do not need to announce their ability to be transferred away from Home, it is possible that a packet contains a Home Address Destination Option or a Binding Update. In such a case, the status of the foreign host SHOULD be changed into a Standard Mobile Host, and the packet SHOULD be handled as specified in Sec 2.3.2.

2.3.4. Receiving packets from hosts that do not have any corresponding Foreign Host Cache Entry

When receiving packets from a host that does not have any corresponding Foreign Host Cache Entry, the receiving host SHOULD NOT create any new Foreign Host Cache Entry upon packet arrival. Instead, the algorithms MAY either use the address directly to determine a possibly existing wildcard protocol control block, or use a statically allocated Foreign Host Cache Entry which is used only for such received packets. (See also Sec. 4.1 where we discuss potential Denial-of-Service attacks.)

2.4. Security

As discussed above in Sections 2.2.1 and 2.3.1., two Homeless (supporting) Hosts MAY use several different IP addresses as the source and destination address in the packets flowing between them (as long as the addresses have the same scope and belong to the same scoped domain). As already mentioned, this behaviour changes the semantics of a number of upper layer protocols, including TCP and UDP on one hand (as discussed above), and possibly also IPSEC AH and ESP on the other hand. Specifically, the method for finding the correct protocol control block for each received packet MUST be changed, and the method for finding the correct Security Association for each received packet MAY be changed, too. The latter issue is discussed in Section 2.4.1, below.

As a related security measure, all Binding Updates and Binding Acknowledgements used in Homeless Mobile IPv6 MUST carry valid AH headers. The purpose of this requirement is to prevent malicious mobile or non-mobile hosts from changing addressing information related to other Homeless Hosts or Standard Mobile Hosts. Unfortunately, as we point out in Section 2.4.1, just relying on AH is not enough. The new approach highlights and worsens a potential security problem already present in the current Mobile IPv6 specification. To protect from address stealing [OWNERSHIP], all

addresses added to a Host Cache Entry must be verified as specified in Sec 2.4.1 and 3.4.2.

As a related measure, using IKE to negotiate IPsec SAs may be too heavy for the purposes of just protecting Binding Updates. The situation is discussed in Section 2.4.2, and an alternative solution is outlined in Section 2.4.3.

2.4.1. Using Security Associations and Accepting New Addresses

As specified in IP Security Architecture [RFC2401] Section 5.2.1, the standard method for determining the correct IPSEC SA for a received packet is to use the packet destination address, IPSEC Protocol type, and the Security Parameter Index (SPI) fields to look up the SA. Because Homeless hosts may use several addresses, it is natural allow any of the local addresses to be used to look up the SA. That is, in the Homeless Mobile IPv6 the SA is not bound to a single address, or to a predefined set of addresses, but to a changing set of addresses. In practice, all the addresses in a Local Host Cache Entry are equally acceptable.

It must be noted that such a usage may be considered as a slight deviation from the IP Security Architecture [RFC2401]. The SAs are still associated with the destination addresses, but the set of destination addresses an SA is valid for may dynamically change. That is, when a host learns a new address, it starts to accept a new SPI/destination address combination as a lookup key for the existing SA. Similarily, at the sending end, when the recipient tells that it has learned a new address, the sender adds the address as one that the SA may be used with. According to RFC2401, such a change can be only accomplished through administrative actions, and possibly requires an IKE negotiation. In the case of Homeless Mobile IPv6, the case may be a side effect of adding a new address into a Host Cache Entry, depending on implementation.

However, we do not consider this behavior as a problem per se. We don't see any security implications at the receiver end. Instead, there is a problem at the sender. If the sender blindly accepts a new address to a Foreign Host Cache Entry, a malicious node can easily divert traffic meant to some other party to itself. (For a specific attack skenario, see Appendix C.)

To prevent this, new addresses sent by a Homeless Mobile Host MUST first be considered as Tentative. Only when the actual reachability of the address has been checked as defined in Sec 3.4.2, it may be changed into an Active Address. [At this writing we do not define a policy when such a checking should be made. It may be made immediately when the new address is learned, when the first packet using the new address is received, or at other point of time.]

To make the behaviour explicit, we specify as follows.

For incoming packets, a host SHOULD accept any IP address in a Local Host Cache Entry together with a valid SPI. That is, instead of having possibly different SPIs for each local IP address, a host does not care which of its local unicast IP addresses an incoming IPSEC protected packet carries as long as the SPI is a valid one, and the packet can be validated.

For outgoing packets, a host SHOULD create a dynamic IPSEC policy entry that maps outgoing IP addresses to SAs based on the Active Addresses in the Host Cache. That is, when selecting which SA to use for protecting an outgoing packet, as defined in [RFC2401] in Section 5.1.1. item 2., the host SHOULD compare the destination address against the Active Addresses in the Foreign Host Cache Entries, and use this information for selecting the right SA. However, the host MUST NOT use any of the Tentative Addresses for this purpose, and it SHOULD NOT use any of the Deprecated Addresses either.

2.4.2. Using Less Secure AH SAs

It is REQUIRED that all Binding Updates and Binding Acknowledgements are protected with AH. This means that a remote Homeless Mobile Host cannot update the corresponding Foreign Host Cache Entry at its peers until there is a valid AH Security Association between the hosts. Consequently, typically one of the first application protocols exchanged between two hosts is IKE, which is used to create the IPSEC AH SA. Unfortunately IKE is quite heavy a protocol, and requires external knowledge which may not be available. However, depending on the situation, it may be sufficient to create the AH SA in some less secure means, e.g., through a so called Anonymous Authentication Protocol specified in Section 2.4.3, below.

Furthermore, in order to simplify the management of the specific AH SA that is used for protecting Binding Updates, it is RECOMMENDED that the specific AH SA is directly associated with the Host Cache Entry, and whenever sending Binding Updates, the associated Foreign Host Cache Entry is directly used to select the SA. This method assures that future Binding Updates sent to the denoted host will automatically get protected with the right SA. It should be noted, however that it MAY be necessary to have some other AH SA to protect traffic for other purposes than authenticating Binding Updates.

2.4.3. Anonymous Authentication

Sometimes there is no other need for IPSEC (or AH) between a specific pair of hosts other than protecting future Binding Updates. That is, two hosts may learn about each other through

some insecure means, e.g., as a result of a mobile host browsing a web site, and continue to talk to each other as either one or both of the hosts move. In such case it may well be that all the information the hosts know about each other is their ability to communicate using specific IP addresses. Thus, there are no external credentials for creating AH SA, e.g., through performing an IKE authentication.

For these kinds of situations, there is clearly a need for a protocol with the following properties.

1) The protocol is lightweight, requiring no heavy cryptographic operations.

2) As a result of running the protocols, the hosts know with _reasonably_ high confidence that they have a common secret that no other host knows. In other words, host A knows that there is some host B, which is currently able to use a specific IP address, and that B knows the same secret value which A knows, and no-one else knows that particular secret (unless, of course A or B themselves have leaked it). That is, we mainly seek for protection against passive attackers, since an active attacker would easily play at B's address.

We denote such an protocol as an "Anonymous Authentication Protocol" (AAP), and any SA created through the protocol as an Anonymous SA. Some more specific requirements for the protocol are given at [Nikander2001]. The exact protocol is beyond the scop of this specification, and currently still to be defined.

If an AAP is used to create an IPSEC AH SA to protect Binding Updates, the resulting Anonymous AH SA MUST NOT be used for any purposes that require strong (identity based) authentication. That is, the Anonymous AH SA does not authenticate anything else but that such connections which were initiated after the authentication was performed will keep connected to the same host even if one or both of the connected hosts move and change their IP addresses.

3. Protocol Details

This section defines the protocol details, including new Destination Option Sub-Options, giving the details of the Binding Update packet formats, and discussing the details of security and the use of AH.

3.1. New Destination Option Sub-Options

Three new Binding Update Destination Option Sub-Options are defined. The Homeless Support Sub-Option SHOULD be included in the first Binding Update sent to a host whose capabilities are unknown, or

that is believed not to know that the sending host does support Homeless Mobile IPv6. That is, it SHOULD be included in the Binding Updates sent to a host until the first Binding Acknowledgement is received from the host.

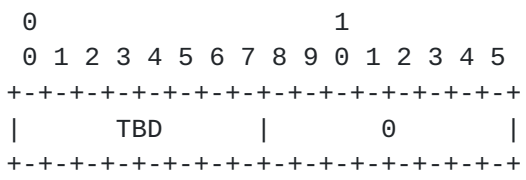
The Alternate Prefixes Sub-Option MAY be included in any Binding Update, but it MUST NOT be expected that the receiving host understands it unless it is known that the receiving host does support Homeless Mobile IPv6.

The Security Challenge/Response Sub-Option is used to check that a remote host is really reachable through a new address it is giving. The purpose of this Sub-Option is to prohibit address stealing attacks.

The general format of suboptions is defined in MIPv6 Sec 5.5.

3.1.1. Homeless Support Sub-Option

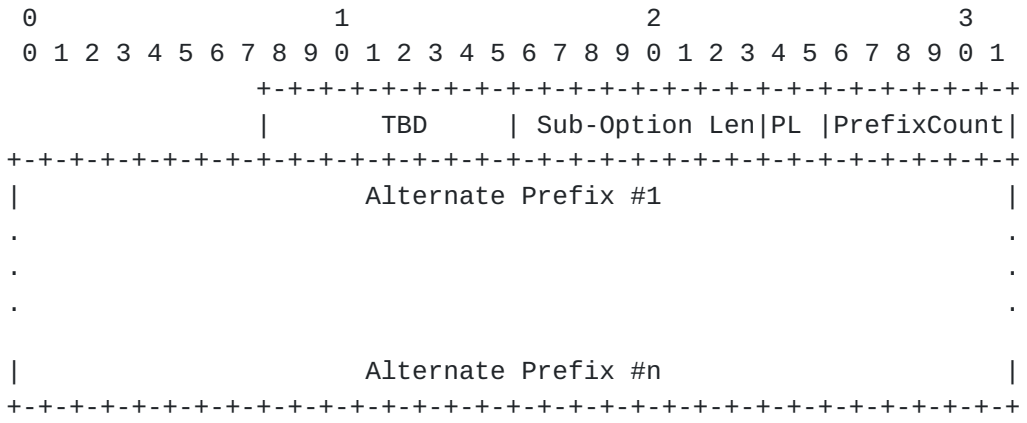
Homeless Support Sub-Option (alignment requirement: none)



The Homeless Support Sub-Option is used in a Binding Update or Binding Request to indicate that the sending host supports Homeless Mobile IPv6.

3.1.2. Alternate Prefixes Sub-Option

Alternate Prefixes Sub-Option (alignment requirement: 4n+1)



The Alternate Prefixes MAY be used in a Binding Update to indicate that there are several alternative addresses that differ only in their prefix. A prefix may have a length of 4, 8, 12, or 16 bytes. Prefix length of 16 bytes can be used to signal alternate addresses

that are completely different, i.e., that do not have any common suffix. More typically, however, would be to use 8 byte prefixes, i.e., having different routing prefixes but identical host IDs.

The suffix shared by all prefixes is defined as follows. If there is an Alternatate Care-Of-Address Sub-Option in the Binding Update Destination Option that preceeds this Alternate Prefixes Sub-Option, the suffix is copied from the Alternate Care-Of-Address Sub-Option. If there are several preceeding Alt CoA Sub-Options, the suffix is copied from the one that is closed to this Alt Prefix Sub-Option. Otherwise, the suffix is copied from the packet source address.

A single Binding Update MAY carry several Alternate Prefix Sub-Options.

Sub-Option Length
As defined in MIPv6, Sec. 5.5. For the Alternate Prefix Sub-Option, the following equation MUST hold.

Sub-Option Length == 1 + ((1 << (2 * (PL + 1))) * Prefix Count)

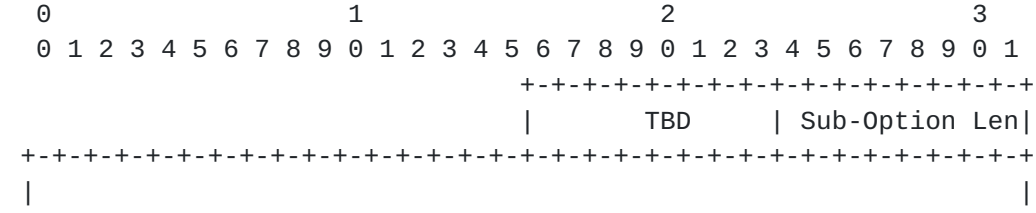
PL
2-bit prefix length.
00 - the precixes are 4 bytes long
01 - the prefixes are 8 bytes long
10 - the prefixes are 12 bytes long
11 - the prefixes are 16 bytes long
i.e., the prefix length is (1 << (2 * (PL + 1)))

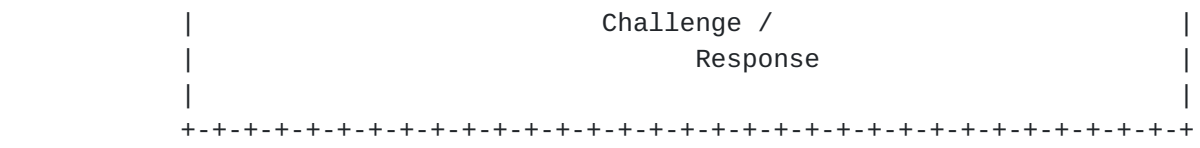
Prefix Count
6-bit unsigned integer, giving the number of prefixes in this Sub-Option. The Maximum number of Alternate Prefixes is limited by the Maximum length of the Sub-Option, i.e., 255 bytes, yielding 63, 31, 21 or 15 prefixes, for the corresponding lengths of 4, 8, 12, and 16 bytes.

Alternate Prefix #k
An alternate prefix, either 4, 8, 12, or 16 bytes long. There MUST be exactly Prefix Count Alternate Prefixes in the Sub-Option.

3.1.3. Security Challenge/Response Sub-Option

Security Challenge/Response Sub-Option (alignment requirement: 4n+2)





As specified in Section 3.4.2, the Security Challenge/Response Sub-Option is used to verify the reachability of hosts so that Tentative Addresses can be securely upgraded to Active Addresses.

Sub-Option Length
: As defined in MIPv6, Sec. 5.5. For the Security Challenge/REsponse Sub-Option, the length is always 16.

Challenge/Response
: 128 bits of a hash code as defined in Section 3.4.2.

3.2. Packet formats

This section details the formats for Binding Updates sent by a Homeless (supporting) Host.

3.2.1. Initial Binding Update

An Initial Binding Update is sent to a host whose capabilities are unknown. Thus, it must be fully compatible with MIPv6, but, at the same time inform the receiving host about the sending host's capabilities.

MIPv6 states the following (MIPv6, Sec 5.1.):

> Any packet that includes a Binding Update option MUST also include a Home Address option. The home address of the mobile node in the binding given in the Binding Update option is indicated by the Home Address field in the Home Address option in the packet.
>
> ....
>
> If the care-of-address for the binding (specified either in an Alternate Care-of-Address Sub-Option in the Binding Update option, if present, or in the Source Address field in the packet's IPv6 header) is equal to the home address of the mobile node, the Binding Update option indicates that any existing binding for the mobile node MUST be deleted.

and (MIPv6, Sec. 5.5)

> ... any of the Mobile IPv6 destination options defined in this document MAY include one or more sub-options.
>
> ....
>
> Sub-Option Type

8-bit identifier of the type of sub-option. In processing a Mobile IPv6 destination option containing a sub-option for which the Sub-Option Type value is not recognized by the receiver, the receiver SHOULD quietly ignore and skip over the sub-option, correctly handling any remaining sub-options in the option.

Given these provisions, it seems safe to send an initial packet that
- includes a Home Address Destination Option with the Home Address equaling to the source address in the packet,
- includes a Binding Update Destination Option that MUST first carry a Homeless Support Sub-Option, and after that MAY carry one or more Alternate Prefix Sub-Options that enumerate the other IP addresses the host wants to indicate.

According to the MIPv6 specification, a Standard Mobile Host SHOULD treat this as a request to delete a non-existing binding in the Binding Cache, i.e., as a non-op. A Homeless (supporting) Host, instead, would recognize the packet as indicating support for Homeless Mobile IPv6, and will also directly learn the the other addresses that the peer wants to publish.

3.2.2. Consecutive Binding Updates sent to Homeless Hosts

When sending Binding Updates to a host that is known to support Homeless Mobile IPv6, the following relaxations MAY be applied in comparison to the MIPv6 specification:

Any packet that includes a Binding Update NEED NOT to include a Home Address option, if the source address of the packet is already known to the receiver. (cf. MIPv6 Sec 5.1, Sec 8.2)

A packet MAY include several Binding Updates. This may be useful, for example, for including different IP addresses with different lifetimes. If there are several Binding Updates within a single packet, they MUST all have increasing Sequence Numbers. [MIPv6 does not seem to prohibit this, but this kind of usage is not necessarily useful for MIPv6 hosts.]

A packet MAY include several Binding Acknowledgements. This may be used to acknowledge several Binding Updates, received either in one packet or along several packets. [Again, MIPv6 does not seem to prohibit this.] As a shorthand, all Binding Updates within a single packet MAY be positively acknowledged with a single Binding Acknowledgement whose Sequence Number is equal to the highest Sequence Number in the packet that contained the Binding Updates.

Any Binding Update MAY carry more than one Alternate Care-Of-Address and Alternate Prefix Sub-Options.

3.2.3. Consecutive Binding Updates sent to Standard Hosts

When sending Binding Updates to a host that is known not to support Homeless Mobile IPv6, the options must be strictly formatted according to MIPv6 specification.

3.3. Protocol Parameters

The following parameters shall be set by network management. The values listed here are for information only.

DEFAULT-LIFETIME	600 seconds
MAXIMUM-EXPIRED-LIFETIME	1800 seconds

3.4. Security

For the purposes of Homeless Mobile IPv6, there is basically only one security goal: to make sure that connections remain to be connected to the original hosts even if one or both of the hosts move. As was already mentioned in Section 2.4.3, it MAY NOT be necessary to know whom a connection was originally created with. In other words, the goal is make sure that Host Personalities remain integral. Homeless Mobile IPv6 does not have any specific confidentiality goals.

The worst skenario that we have been able to find so far is an address stealing skenario (see Appendix C), where a malicious node claims to be at someone else's address, and move from there to its real address. In that skenario, all the traffic destined to the other host would be sent to the malicious node. Fortunately, a single challenge-response protocol will fix that problem. The details are defined in Section 3.4.2.

Additionally, it MUST be realized that the integrity goals of Homeless Mobile IPv6 are typically much weaker than other typical integrity goals are. Homeless Mobile IPv6 is only interested in making sure that the communicating parties remain the same throughout the communication, and that a malicious communicating party cannot claim to use addresses that it actually cannot use. The realization of the relative weakness of the requirements leads to classification of AH Security Associations, as discussed in Section 3.4.1., below.

3.4.1. Classification of AH Security Associations

Basically, from the point of view of Homeless Mobile IPv6, there are two types of AH Security Associations:

- Generic (or application specific) host-to-host AH Security Associations, which MAY be used for other purposes than authenticating Binding Updates and Binding Acknowledgements, too.

- Anonymous host-to-host AH Security Associations, which SHOULD NOT be used for any other purpose but authenticating Binding Updates and Binding Acknowledegements.

Typically, generic host-to-host AH SAs are created either through manual configuration or through some high level protocol, such as IKE. On the other hand, Anonymous host-to-host AH SAs are typically created through an Anonymous Authentication protocol (see Section 2.4.3).

In order to be secure, a Homeless Mobile IPv6 implementation MUST be able to make a distinction between these two types of associations. In particular, it MUST make sure that Anonymous SAs are used only for securing BUs and BAs, and not for application security. The reachability verification mechanism (described in Section 3.4.2, next), takes care of preventing address stealing.

3.4.2. Verifying reachability

When a Homeless Host receives a Binding Update requesting that a new IP address is added to a Foreign Host Cache Entry, it MUST first add that address as a Tentative Address only. In particular, it MUST NOT assume that the address necessarily belongs to the peer host claiming to own it. If some other host requests to have the same address, the host SHOULD use local security policy to decide what to do. It is RECOMMENDED that the host runs the reachability verification protocol against both of the claimants, and believes the one that checks OK. If both of them check ok, it is RECOMMENDED that the node completely flushes the Host Cache Entries for both claimants.

In order to raise the status of a Tentative Address into an Active Address, a Homeless Host SHOULD run the following reachability verification protocol.

1. Using the SA that protected the received BU, the verifying node creates a challenge and sents it to the address being checked. It is important that the challenge is actually sent to the Tentative Address and not to any other of the addresses in the Foreign Host Cache Entry.

2. When the peer node receives the Challenge, it constructs a corresponding Response and sents it back to the verifying node. The destination address of the Response SHOULD be the source address in the Challenge, but it MAY be any of the other addresses in the Foreign Host Cache Entry that represents the verifying node at the peer host. The source address of the Response may be freely selected as long at it follows the rules specified elsewhere in this specification.

3. When the verifying node receives and positively verifies the

Response, it SHOULD raise the status of the address to an Active Address.

It is important to realize that the protocol only checks that there is a node that is able to receive packets sent to the address being checked and that knows the keying material associated with the SA. Thus, since the peer nodes are assumed not to reveal their keying material, the node is assumed to be the same node that sent the original BU.

The Challenge is created as follows.

```
K_MAT = the authentication keying material associated with the SA
HMAC  = the keyed hashing algorithm used in the SA
ADDR  = the IPv6 address whose reachability is being checked
TIME  = the current time or a nonce in a host specific format

Challenge = HMAC(K_MAT, ADDR | TIME)
```

The Response is created as follows.

```
Reply = HMAC(K_MAT, Challenge | ADDR)
```

The Challenge is sent in the Challenge/Response field of a Security Challenge/Response Sub-Option, included in a Binding Acknowledgement.

The Response is sent in the Challenge/Response field of a Security Challenge/Response Sub-Option, included in a simple Binding Update that MUST NOT contain anything else but a simple binding repeating the address being checked. That is, if the source address contained the address to be checked, the Binding Update would consists of the the actual BU and a Security Challenge/Response Sub-Option. On the other hand, if the source address was different, the BU would contain both Alternate CoA and Security Challenge/Response Sub-Options.

It should be noted that the above defined protocol is actually slight overkill. Since the BA and BU must be protected with the SA anyway, the protocol protects the Challenge and Response twice. Thus, this part of the specification is likely to be revised in a future version.

4. Security and Privacy Considerations

[This section is still not ready, and needs some work.
However, it is much better than it was in the -00.txt version.]

In a way, Homeless Mobile IPv6 loosens up security assumptions about IP addresses. However, to us it seems like this is an intrinsic property of host mobility. We believe that the same security problems, including the address ownership problem, are

present already in the current Mobile IPv6 solution. However, since the presented solution "enhances" mobility by allowing a host to be "present" simultaneous at several topological locations, the security problems look worse.

In this section, we summarize the security and privacy concerns. First, in Section 4.1., we discuss potential Denial-of-Service attacks. Section 4.2. briefly covers the address ownership problem in the context of Binding Updates. Finally, Section 4.3. briefly mentions some privacy problems.

The proposed solutions are presented throughout this document, but especially in sections 2.4 and 3.4, above.

4.1. Potential Denial-of-Service Attacks

The Host Cache introduces an entirely new data structure into the kernel. As state must be created into the Host Cache based on information provided by other hosts, a potential Denial-of-Service vulnerability is created. Ways of reducing the vulnerability of implementations may include creating Host Cache Entries only when absolutely necessary, that is, when the first AH protected Binding Update is received from the correspondent host. Here, the presens of AH protection can be considered as a relative security agains DoS, since creating SAs is expensive compared to just sending packets.

Another potential DoS vulnerability is present in IPsec itself. It is likely that sometime in the future it will be possible to automatically create SAs with any host. It is unrealistic to assume that all of these hosts would be trustworthy. Thus, it will be possible to make other hosts to burne large amounts of CPU by making them to decrypt and check garbage. Fortunately, such attack requires some effort from the side of the attacker, since it needs first to establish the SA. Even further, such attacks are not caused or directly affected by Homeless Mobile IPv6; we just mention this attack here in order to put the other potential DoS attacks into a proper context.

4.2. Address ownership and Binding Updates

A Binding Update allows a remote host to effectively change the routing table of the local host. Such a mechanism is an effective impersonation, man-in-the-middle, and denial-of-service attack tool. For this reason, MIPv6 requires that all BUs must be protected with AH.

In addition to being simply protected with AH, there must be some mechanism that makes sure that the sender of the BU is indeed authorized to change the routing information for the address given in the BU. In the case of standard MIPv6, routing information is

only changed for the home address, and therefore it is sufficient to be authorized to change routing information for it. However, in the case of Homeless Mobile IPv6, all the addresses in a Host Cache Entry are equal. Therefore, it is equally important to check that the remote host is authorized for all of the addresses.

Currently, there are no existing PKI or other infrastructure that would provide the needed information, and it is doubtful whether there will ever be. Therefore, we have selected a simplistic approach. The recipient of the BU checks that the sender is indeed reachable through the address to be added to the Host Cache Entry. If the sender is reachable, i.e., if it can read packets sent to that address, it is assumed to be authorized to modify routing information for that address.

The reasoning goes as follows. If a host is able to act as if it is at address A, then it is either actually at address A, or it is able to eavesdrop all traffic that is sent from the current address to address A. If there is no other host at address A, we can't do much since we cannot effectively make a distinction between a host genuinely at that location and a malicious host eavesdropping all traffic sent to that location. On the other hand, if there is another host at address A, there two more possibilies. First, it is possible that the eavesdropper is also able to block traffic. In that case, again, there isn't much we can do. However, if the eavesdropper cannot block traffic, then also the "real" host at A receives packets sent to A. In that case, the "real" host at A can send error messages indicating that it is getting unsolicited challenges. However, no such mechanism has been specified (yet), and we are still trying to figure out the value of such messages. [TBD]

One further condition must still be considered. Let us consider a situation where initially the host at address A is absent, and then comes back at some later point of time. This case is more tricky, since by the time the "real" host comes back, there is already an authenticated binding directing all traffic sent to it to somewhere else. Fortunately, even this kind of attack, like the ones above, require that the attacker can eavesdrop the traffic at least when the attack is initiated. This limits the number of potential attackers considerably. The other aspects of this skenario are being studied. [TBD]

Thus, by the argument above, a host that is able to reply to packets sent to a specific address is also able to otherwise divert traffic sent to the address, at least in the case that we receive no notifications about unsolicited challenges. Thus, it seems to be safe to use the simple reachability check in order to accept routing information.

4.3. Privacy

[Ability to change addresses at will. Ability to send Binding Updates inside ESP. TBD.]

[Local Host Cache Entries corresponding to several distinct Host Personalities to enhance privacy. TBD.]

5. General Comments and Open Issues

5.1. Application Backwards Compatibility

According to our current understanding, most applications SHOULD work without any changes on the top of Homeless Mobile IPv6. In general, applications that just open a connected socket, and use it in an "address-agnostic" way do work just fine. However, there are a small collection of TCP applications, FTP being the prime example, that use IP addresses at the application level, and a slightly larger collection of UDP applications that do the same. Of these, those applications that store the address for a long time and use it later, will definitely break. But, in our opinion, those applications should be rewritten in any case. On the other hand, if the IP address is used at the application level just for a short time, the application has a good chance of working anyway.

5.2. Cache Entry Creation Policy

We need to clarify the policy for creating Foreign Host Cache Entries when acting as a server (answering party). Basically, the method must be such that there are no easy IP address-spoofing Denial-of-Services attacks.

- The upper layer PDU might contain information, such as application-level authentication data, that proves the foreign host to be honest. In that case, a cache entry could be created without any worries.

- In the case of protocols like TCP, the cache entry is still created too early (= when sending SYN|ACK).

- For stateless upper-level protocols, the cache entry should exist only between receiving a packet and sending a reply. After that, the entry is wasting cache space.

We do not know how to solve the problem. However, the following ideas have come to our mind.

- Classify cache entries into "trusted" and "untrusted". For untrusted entries, use the DEFAULT-LIFETIME instead of the one specified in the last Binding Update. When the cache is filling up, erase random untrusted entries.

- Provide and API that lets the upper-layer (or application)

protocols say that a certain cache entry is trusted.
The TCP protocol could mark the cache entry as trusted after
receiving the ACK (3rd packet).

- Provide an API that lets upper-layer (or application)
protocols erase cache entries or prevent their creation.
The stateless upper-layer protocol could erase the cache
entry after sending a reply. (It might be
better not to create any cache entries and to pass the
source IP addresses to the upper layer protocol.)

- Create Host Cache Entries only after the first AH protected
Binding Update is received from the correspondent host.

The problem is that only the upper-layer protocol can determine whether a cache entry should be trusted to not. (For example, only the TCP layer knows that the ACK proves that the foreign host has received the SYN|ACK. This cannot be reasoned by the network layer without making some strong assumptions about all upper-layer protocols.) But if we let the upper-layer protocols manage the cache, as we suggest above, it will become confusing when cache entries are shared by several upper-layer protocols and even by several users.

5.3. Address Confusion

Consider the following scenario:

1. Stateless Mobile Host M1 has care-of-address ADDR1.
2. M1 opens a connection to a Standard Mobile Host C.
3. M1 obtains a new address ADDR2, loses its old address ADDR1, and sends the corresponding Binding Updates to C. From now on, M1 creates the illusion to C that ADDR1 is his home address.
4. Stateless Mobile Host M2 gets the old care-of-address ADDR1.
5. M2 opens a connection to a Standard Mobile Host C. C becomes confused because M2 appears to have the same home address as M1.

This problem will not occur if the care-of-addresses are not reused by different mobile hosts. For example, composing the care-of-address of a unique hardware address solves it.

6. Intellectual Property Right Notice

For Ericsson policy on IPR issues, see the Ericsson General IPR statement for IETF, http://www.ietf.org/ietf/IPR/ERICSSON-General

References

[MIP6] David B. Johnson, Charles Perkins, ``Mobility Support in IPv6'', work in progress, Internet-Draft draft-ietf-mobileip-ipv6-13.txt, Internet Engineering Task Force, November 2000.

[RFC2136]
P. Vixie (Ed.), S. Thomson, Y. Rekhter, J. Bound ``Dynamic Updates in the Domain Name System,'' RFC 2136, ISC & Bellcore & Cisco & DEC, April 1997.

[SIP] Handley, Schulzrinne, Schooler, Rosenberg, ``SIP: Session Initiation Protocol'', work in progress, Internet Draft draft-ietf-sip-rfc2543bis-01.txt, Internet Engineering Task Force, August 2000.

[RFC2874]
M. Crawford and C. Huitema, ``DNS Extensions to Support IPv6 Address Aggregation and Renumbering'', RFC 2874, July 2000.

[RFC1886]
S. Thomson and C. Huitema, ``DNS Extensions to support IP version 6'', RFC 1886, December 1995.

[ADDRSEL]
R. Draves, ``Default Address Selection for IPv6'', work in progress, Internet Draft draft-ietf-ipngwg-default-addr-select-02.txt, November 2000.

[OWNERSHIP]
P. Nikander, ``An Address Ownership Problem in IPv6'', work in progress, Internet Draft draft-nikander-ipng-address-ownership-00.txt, Internet Engineering Task Force, February 2001.

[RFC2401]
S. Kent and R. Atkinson, ``Security Architecture for the Internet Protocol'', RFC2401, November 1998.

[Nikander2001]
P. Nikander, ``Denial-of-Service, Address Ownership, and Early Authentication in the IPv6 World'', unpublished manuscript submitted for publication, available from the author upon request.

Authors' Addresses

Pekka Nikander
Ericsson Research NomadicLab, Espoo, Finland
phone: +358-40-721 4424
email: Pekka.Nikander@nomadiclab.com

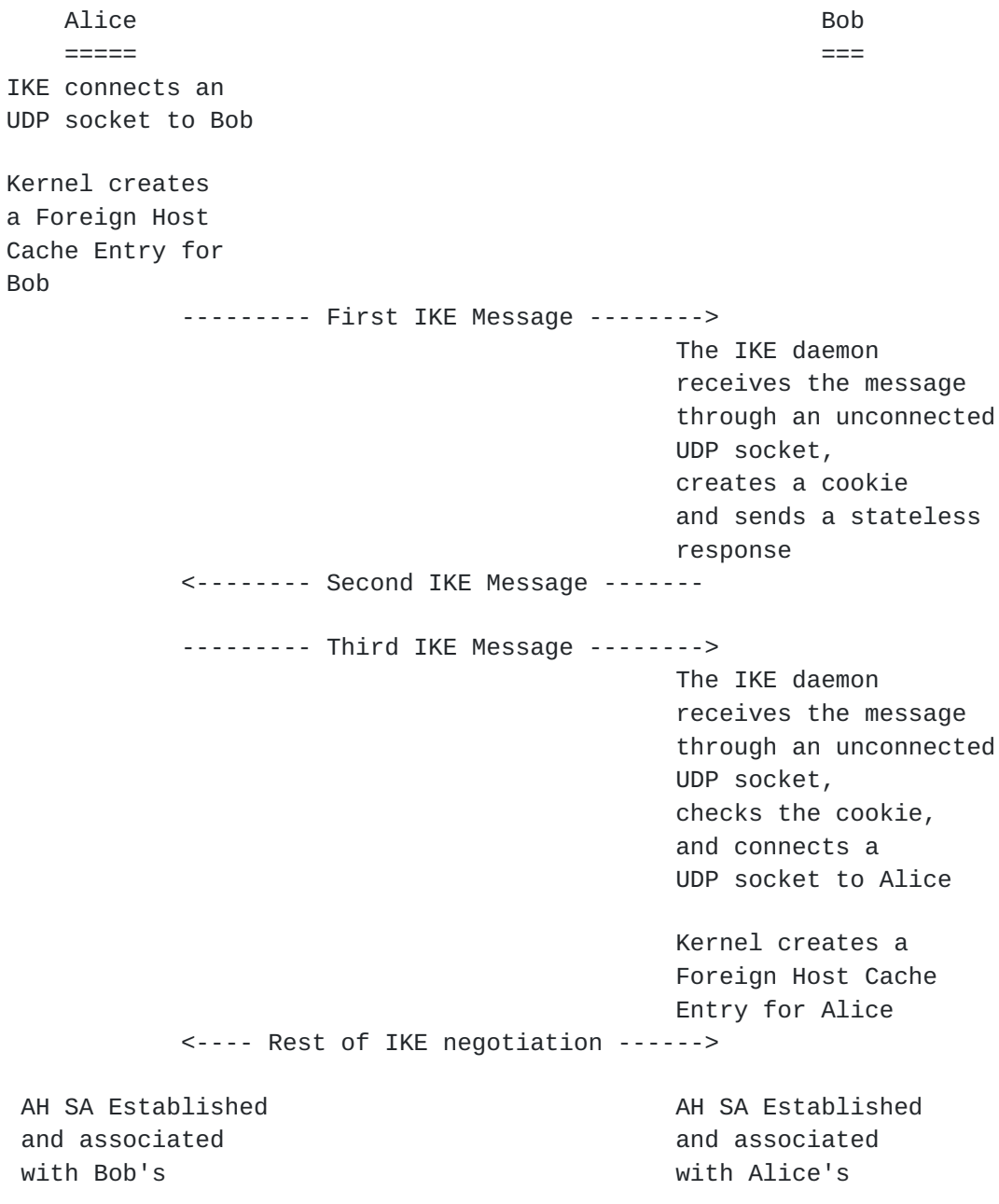
Janne Lundberg
Ericsson Research NomadicLab, Espoo, Finland
phone: +358-9-2991
email: Janne.Lundberg@nomadiclab.com

Catharina Candolin
Telecommunications Software and Multimedia Lab
Helsinki University of Technology, Finland
email: Catharina.Candolin@hut.fi

Tuomas Aura
Microsoft Research, Cambridge, UK
email: tuomaura@microsoft.com

Appendix A. Example Packet Flows

A.1. Alice contacts Bob, initiates IKE negotiation creating an AH SA, and Alice and Bob exchange Binding Updates.



```
   Foreign Host                                    Foreign Host
   Cache Entry                                     Cache Entry

              ----- Initial Binding Update------->

              <---- Initial Binding Update -------
```

A.2. **Alice contacts Bob, performs Anonymous Authentication,** and Alice and Bob exchange Binding Updates

```
      Alice                                                  Bob
      =====                                                  ===
   IKE connects an
   UDP socket to Bob

   Kernel creates
   a Foreign Host
   Cache Entry for
   Bob
              ----- First Anon Auth Message ------>
                                              Bob creates a response
                                              that contains all his
                                              state, including the
                                              AH SA, and does not
                                              create any state yet
             <----- Second Anon Auth Message -----
                    + AH
                    + Initial Binding Update
   Alice creates
   an AH SA and
   associates it with
   the Foreign Host
   Cache Entry
             ------ Third Anon Auth Message ----->
                    + AH
                    + Initial Binding Update    Bob checks that the
                                                the state can be
                                                correctly recovered,
                                                creates a Foreign
                                                Host Cache Entry and
                                                an AH SA, and associates
                                                the AH SA with the
                                                Foreign Host Cache
                                                Entry
```

Appendix B. **Binding Update Optimization**

This appendix presents an alternative way of performing Binding Updates between two Homeless Hosts. This alternative further reduces overhead caused by Binding Updates.

A Homeless Host can inform a correspondent host of a new address that can be used to reach it by sending a packet with the new address as the source address in a packet. No Binding Update Option is needed. In the receiving host, the new address is added to the Host Cache Entry which is identified by the SPI in the AH header in the packet.

The packet MUST be protected with AH, and a Binding Acknowledgement MUST be sent as a result of receiving an address through this optimization. The lifetime of the address is set to DEFAULT-LIFETIME. This technique MUST NOT be used as the Initial Binding Update, and the destination node MUST be known to be a Homeless Host.

Appendix C. An attack against "address ownership"

A usual misconception in security is that cryptography means security. That is, most people think that "if I have an IPSEC security association with Bob, then Bob must be honest and trustworthy". Unfortunately, this assumtion is not necessarily true. Now, as long as IPsec is used for VPN only, the problem doesn't really surface. Once we start to use IPsec for multiple purposes, or simply for signalling with several hosts, the problem becomes more visible. In this example, we show how this problem surfaces with standard Mobile IPv6.

C.1. An attack skenario

Let us consider a server serving a number of future mobile terminals. The actual nature of the server is not important. The terminals and the server communicate with Mobile IPv6. The server is meant to be open, i.e., to serve any hosts using Mobile or non-mobile IPv6.

Following the cryptographic tradition, let us call the server Alice. To simplify the situation, let us have only two clients: Bob, who is honest, and Mallory, who is malicious and whose aim is to "steal" or at least disturb communication with Alice and Bob. It is important to notice that Mallory has selected Bob as his target, and he attempts to perform his attack in such a way that neither Alice or Bob are aware of the attack; the result of a succesfull attack is that Mallory controls, at least to a degree, all communication between Alice and Bob.

Now, if we assume that the MIP6 implementation that Alice uses fully conforms to the standards but is simple minded, the following attack would work.

1. Mallory contacts Alice and creates a pair of AH SAs with her.

2. Using the SAs created in Step 1, Mallory sends a MIP6 Binding Update to Alice, claiming that his Home Address is that of Bob's and that Bob (the Home Address) is currently visiting at

Mallory's (care-of-address).

3. Since the Binding Update is protected with AH (see MIPv6 Sec 4.4 and 8.2), Alice accepts the Binding Update. (Note that this is a mistake at Alice's part. However, giving Alice the required knowledge to make the right decision is hard. For more discussion, see [address-ownership-problem].

4. As part of the Binding Update processings (sec 8.3), Alice creates a new Binding Cache Entry telling that all future communications to Bob's Home Address should be sent to the address Mallory gave (the CoA).

Now, let us assume that Bob now wants to create a TCP connection with Alice. Thus, he sends a TCP SYN packet, using his home address, to Alice. Alice receives this packet normally, and passes it to TCP. Alice's TCP creates a new TCB and sends a SYN ACK packet, using Bob's home address as the destination address.

5. Now, as a part of output processing (MIP6 sec 8.9), Alice checks its Binding Cache, find a Binding matching the destination address, and adds a Routing Header to route the packet through Mallory to Bob.

6. Mallory receives the SYN ACK packet from Alice.

7. Mallory has now a number of options to further fool Bob.

   a) Mallory may choose to claim Bob that Alice is a mobile node herself, and currently at the location where Mallory is. To do this, he creates a pair of IPsec SAs with Bob (or uses existing ones), and sends a Binding Update claiming that Alice is currently at his address. (This assumes, of course, that Bob makes the same mistake Alice made above at step 3.)

   b) Mallory may choose to claim Bob that Alice is currently (better) reachable through him. To do this, he replaces the routing header that Alice inserted with his own, and uses an existing AH SA (or creates a new) between himself and Bob to protect the routing header to get replies back, as specified in RFC2460 section 8.4. The real difference between this and the a) alternative is that Bob does not insert a Home Address destination option or a Binding Update, but relies on Bob reversing the Routing Header since it is protected with AH.

8. Independent on whether Mallory decides use Binding Updates or Routing Headers, he further has two options on how to handle the data stream in the future

   a) Mallory may decide to act as a man-in-the-middle, passing

data between Alice and Bob, and modifying it at need. Furthermore, if the IPsec implementation Alice and Bob are using is simple minded enough, he _may_ be able to fool Alice that she is securely (i.e. with IPSec) talking with Bob, and fool Bob that he is securely talking with Alice.

b) Mallory may decide to play Alice to Bob, and completely terminate Bob's session with Alice.

This ends the attack description; the only purpose of it is to be illustrate the problem.

C.2. Attack analysis

Even an superficial analysis on the attack description reveals the basic problem: Alice is using an "untrustworthy" SA to accept Binding Updates concerning Bob, and Bob is using equally "untrustworthy" SA to verify Binding Updates or Routing Headers. If we look at a little bit closer to steps 3. and 7. above, we can describe the problem as an authorization problem.

First, in step 3., the SA that Mallory has created with Alice should NOT be authorized to create a Binding Cache Entry for Bob's home address. Similarily, in step 7a., the SA that Mallory has created with Bob should NOT be authorized to create a Binding Cache Entry for Alice. Still, in step 7b., the SA that Mallory has created with Bob should NOT be authorized to accept the Routing Header as a reversible Routing Header (RFC2460 sec 8.4.).

Appendix D. State machine for backward compatibility

This appendix defines a state machine implementation for keeping track of the status of the peer hosts. (See Sections 2.2. and 2.3.)

TBD.