

**End-Host Mobility and Multihoming with the Host Identity Protocol
draft-ietf-hip-mm-03**

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 28, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document defines mobility and multihoming extensions to the Host Identity Protocol (HIP). Specifically, this document defines a general "LOCATOR" parameter for HIP messages that allows for a HIP host to notify peers about alternate addresses at which it may be reached. This document also defines elements of procedure for mobility of a HIP host-- the process by which a host dynamically changes the primary locator that it uses to receive packets. While the same LOCATOR parameter can also be used to support end-host multihoming, detailed procedures are left for further study.

Table of Contents

1.	Introduction and Scope	4
2.	Terminology and Conventions	6
3.	Protocol Model	7
3.1.	Operating Environment	7
3.1.1.	Locator	9
3.1.2.	Mobility overview	10
3.1.3.	Multihoming overview	10
3.2.	Protocol Overview	10
3.2.1.	Mobility with single SA pair (no rekeying)	11
3.2.2.	Host multihoming	13
3.2.3.	Site multihoming	15
3.2.4.	Dual host multihoming	15
3.2.5.	Combined mobility and multihoming	16
3.2.6.	Using LOCATORs across addressing realms	16
3.2.7.	Network renumbering	16
3.2.8.	Initiating the protocol in R1 or I2	16
3.3.	Other Considerations	17
3.3.1.	Address Verification	17
3.3.2.	Credit-Based Authorization	18
3.3.3.	Preferred locator	19
3.3.4.	Interaction with Security Associations	20
4.	LOCATOR parameter format	23
4.1.	Traffic Type and Preferred Locator	24
4.2.	Locator Type and Locator	25
4.3.	UPDATE packet with included LOCATOR	25
5.	Processing rules	26
5.1.	Locator data structure and status	26
5.2.	Sending LOCATORs	26
5.3.	Handling received LOCATORs	28
5.4.	Verifying address reachability	30
5.5.	Credit-Based Authorization	31
5.5.1.	Handling Payload Packets	31
5.5.2.	Credit Aging	33
5.6.	Changing the preferred locator	34
6.	Security Considerations	36
6.1.	Impersonation attacks	36
6.2.	Denial of Service attacks	37
6.2.1.	Flooding Attacks	37
6.2.2.	Memory/Computational exhaustion DoS attacks	38
6.3.	Mixed deployment environment	38
7.	IANA Considerations	40
8.	Authors	41
9.	Acknowledgments	42
10.	References	43
10.1.	Normative references	43
10.2.	Informative references	43

Henderson (editor)

Expires August 28, 2006

[Page 2]

Appendix A .	Changes from previous versions	44
A.1 .	From nikander-hip-mm-00 to nikander-hip-mm-01	44
A.2 .	From nikander-hip-mm-01 to nikander-hip-mm-02	44
A.3 .	From -02 to draft-ietf-hip-mm-00	44
A.4 .	From draft-ietf-hip-mm-00 to -01	45
A.5 .	From draft-ietf-hip-mm-01 to -02	45
A.6 .	From draft-ietf-hip-mm-02 to -03	45
Author's Address		47
Intellectual Property and Copyright Statements		48

1. Introduction and Scope

The Host Identity Protocol [1] (HIP) supports an architecture that decouples the transport layer (TCP, UDP, etc.) from the internetworking layer (IPv4 and IPv6) by using public/private key pairs, instead of IP addresses, as host identities. When a host uses HIP, the overlying protocol sublayers (e.g., transport layer sockets and ESP Security Associations) are instead bound to representations of these host identities, and the IP addresses are only used for packet forwarding. However, each host must also know at least one IP address at which its peers are reachable. Initially, these IP addresses are the ones used during the HIP base exchange [2].

One consequence of such a decoupling is that new solutions to network-layer mobility and host multihoming are possible. There are potentially many variations of mobility and multihoming possible. The scope of this document encompasses messaging and elements of procedure for basic network-level mobility and simple multihoming, leaving more complicated scenarios and other variations for further study. Specifically,

This document defines a generalized LOCATOR parameter for use in HIP messages. The LOCATOR parameter allows a HIP host to notify a peer about alternate addresses at which it is reachable. The LOCATORS may be merely IP addresses, or they may have additional multiplexing and demultiplexing context to aid the packet handling in the lower layers. For instance, an IP address may need to be paired with an ESP SPI so that packets are sent on the correct SA for a given address.

This document also specifies the messaging and elements of procedure for end-host mobility of a HIP host-- the sequential change in preferred IP address used to reach a host. In particular, message flows to enable successful host mobility, including address verification methods, are defined herein.

However, while the same LOCATOR parameter is intended to support host multihoming (parallel support of a number of addresses), and experimentation is encouraged, detailed elements of procedure for host multihoming are left for further study.

While HIP can potentially be used with transports other than the ESP transport format [5], this document largely assumes the use of ESP and leaves other transport for further study.

There are a number of situations where the simple end-to-end readdressing functionality is not sufficient. These include the initial reachability of a mobile host, location privacy, simultaneous

mobility of both hosts, and some modes of NAT traversal. In these situations there is a need for some helper functionality in the network, such as a HIP Rendezvous server [3]. Such functionality is out of scope of this document. We also do not consider localized mobility management extensions; this document is concerned with end-to-end mobility. Finally, making underlying IP mobility transparent to the transport layer has implications on the proper response of transport congestion control, path MTU selection, and QoS. Transport-layer mobility triggers, and the proper transport response to a HIP mobility or multihoming address change, are outside the scope of this document.

2. Terminology and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119](#) [6].

Locator. A name that controls how the packet is routed through the network and demultiplexed by the end host. It may include a concatenation of traditional network addresses such as an IPv6 address and end-to-end identifiers such as an ESP SPI. It may also include transport port numbers or IPv6 Flow Labels as demultiplexing context, or it may simply be a network address.

Address. A name that denotes a point-of-attachment to the network. The two most common examples are an IPv4 address and an IPv6 address. The set of possible addresses is a subset of the set of possible locators.

Preferred locator. A locator on which a host prefers to receive data. With respect to a given peer, a host always has one active preferred locator, unless there are no active locators. By default, the locators used in the HIP base exchange are the preferred locators.

Credit Based Authorization. A host must verify a mobile or multi-homed peer's reachability at a new locator. Credit-Based Authorization authorizes the peer to receive a certain amount of data at the new locator before the result of such verification is known.

3. Protocol Model

3.1. Operating Environment

The Host Identity Protocol (HIP) [2] is a key establishment and parameter negotiation protocol. Its primary applications are for authenticating host messages based on host identities, and establishing security associations (SAs) for ESP transport format [5] and possibly other protocols in the future.

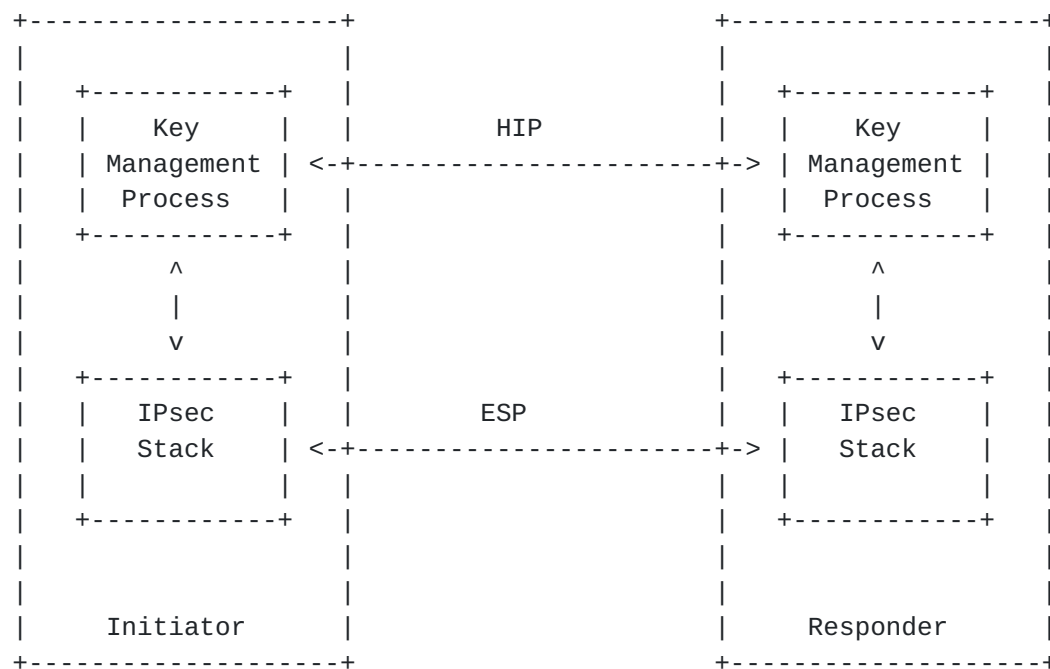


Figure 1: HIP deployment model

The general deployment model for HIP is shown above, assuming operation in an end-to-end fashion. This document specifies extensions to the HIP protocol to enable end-host mobility and multihoming. In summary, these extensions to the HIP protocol can carry new addressing information to the peer and can enable direct authentication of the message via a signature or keyed hash message authentication code (HMAC) based on its host identity. This document specifies the format of this new addressing (LOCATOR) parameter, the procedures for sending and processing this parameter to enable basic host mobility, and procedures for a concurrent address verification mechanism.

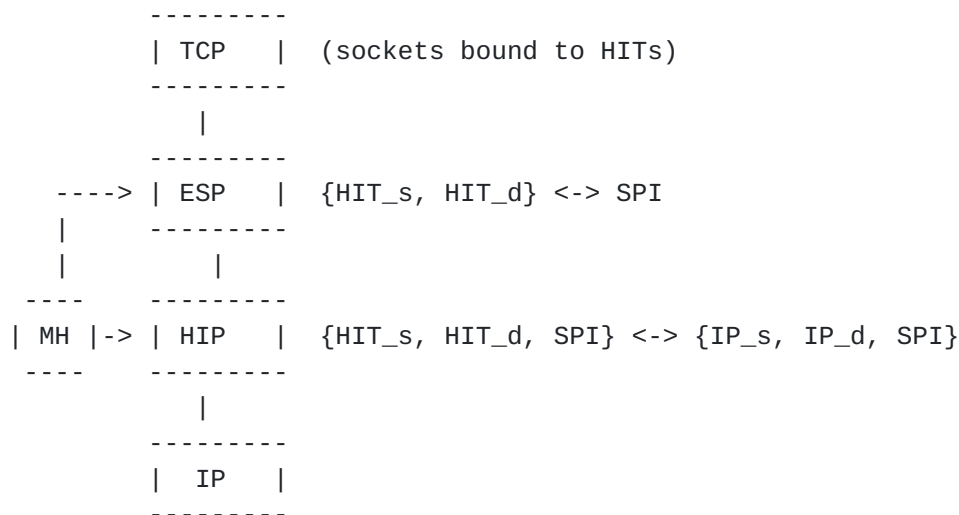


Figure 2: Architecture for HIP mobility and multihoming

Figure 2 depicts a layered architectural view of a HIP-enabled stack using ESP transport format. In HIP, upper-layer protocols (including TCP and ESP in this figure) are bound to HITs and not IP addresses. The HIP sublayer is responsible for maintaining the binding between HITs and IP addresses. The SPI (or other context tag if ESP is not used with HIP), and not necessarily the IP addresses, is used to associate an incoming packet with the right HITs. The block labeled "MH" is introduced below.

Consider first the case in which there is no mobility or multihoming, as specified in the base protocol specification [2]. The HIP base exchange establishes the HITs in use between the hosts, the SPIs to use for ESP, and the IP addresses (used in the HIP signaling packets). Note that there can only be one such binding in the outbound direction for any given packet, and the only selectors for the binding at the HIP layer are the fields exposed by ESP (the SPI and HITs). For the inbound direction, the SPI is all that is required to find the right host context. ESP rekeying events change the mapping between the HIT pair and SPI, but do not change the IP addresses.

Consider next a mobility event, in which a host is still single-homed but moves to another IP address. Two things must occur in this case. First, the peer must be notified of the address change using a HIP UPDATE message. Second, each host must change its local bindings at the HIP sublayer (new IP addresses). It may be that both the SPIs and IP addresses are changed simultaneously in a single UPDATE; the protocol described herein supports this. This document specifies the messaging and elements of procedure for such a mobility event. However, simultaneous movement of both hosts, notification of

transport layer protocols of the path change, and procedures for possibly traversing middleboxes are not covered by this document.

Finally, consider the case when a host is multihomed (has more than one globally routable address) and wants to make these multiple addresses available for use by the upper layer protocols, for fault tolerance. Examples include the use of (possibly multiple) IPv4 and IPv6 addresses on the same interface, or the use of multiple interfaces attached to different service providers. Such host multihoming generally necessitates that a separate ESP SA is maintained for each interface in order to prevent packets that arrive over different paths from falling outside of the ESP replay protection window. Multihoming thus makes possible that the bindings shown on the right side of Figure 2 are one to many (in the outbound direction, one HIT pair to multiple SPIs, and possibly then to multiple IP addresses). However, only one SPI and address can be used for any given packet, so the job of the "MH" block depicted above is to dynamically manipulate these bindings. Beyond locally managing such multiple bindings, the peer-to-peer HIP signaling protocol needs to be flexible enough to define the desired mappings between HITs, SPIs, and addresses, and needs to ensure that UPDATE messages are sent along the right network paths so that any HIP-aware middleboxes can observe the SPIs. This document does not specify the "MH" block, nor does it specify detailed elements of procedure for how to handle various multihoming (perhaps combined with mobility) scenarios. However, this document does describe a basic multihoming case (one host adds one address to its initial address and notifies the peer) and leave more complicated scenarios for experimentation and future documents.

3.1.1. Locator

This document defines a generalization of an address called a "locator". A locator specifies a point-of-attachment to the network but may also include additional end-to-end tunneling or per-host demultiplexing context that affects how packets are handled below the logical HIP sublayer of the stack. This generalization is useful because IP addresses alone may not be sufficient to describe how packets should be handled below HIP. For example, in a host multihoming context, certain IP addresses may need to be associated with certain ESP SPIs, to avoid violation of the ESP anti-replay window [4]. Addresses may also be affiliated with transport ports in certain tunneling scenarios. Or locators may merely be traditional network addresses. In [Section 4](#), a generalized HIP LOCATOR parameter is defined that can contain one or more locators (addresses).

3.1.2. Mobility overview

When a host moves to another address, it notifies its peer of the new address by sending a HIP UPDATE packet containing a LOCATOR parameter. This UPDATE packet is acknowledged by the peer, and is protected by retransmission. The peer can authenticate the contents of the UPDATE packet based on the signature and keyed hash of the packet.

When using ESP Transport Format [5], the host may at the same time decide to rekey its security association and possibly generate a new Diffie-Hellman key; all of these actions are triggered by including additional parameters in the UPDATE packet, as defined in the base protocol specification [2] and ESP extension [5].

When using ESP (and possibly other transport modes in the future), the host is able to receive packets that are protected using a HIP created ESP SA from any address. Thus, a host can change its IP address and continue to send packets to its peers without necessarily rekeying. However, the peers are not able to reply before they can reliably and securely update the set of addresses that they associate with the sending host. Furthermore, mobility may change the path characteristics in such a manner that reordering occurs and packets fall outside the ESP anti-replay window for the SA, thereby requiring rekeying.

3.1.3. Multihoming overview

A related operational configuration is host multihoming, in which a host has multiple locators simultaneously rather than sequentially as in the case of mobility. By using the LOCATOR parameter defined herein, a host can inform its peers of additional (multiple) locators at which it can be reached, and can declare a particular locator as a "preferred" locator. Although this document defines a mechanism for multihoming, it does not define detailed policies and procedures such as which locators to choose when more than one pair is available, the operation of simultaneous mobility and multihoming, and the implications of multihoming on transport protocols and ESP anti-replay windows. Additional definition of HIP-based multihoming is expected to be part of future documents.

3.2. Protocol Overview

In this section we briefly introduce a number of usage scenarios for HIP mobility and multihoming. These scenarios assume that HIP is being used with the ESP transform [5], although other scenarios may be defined in the future. To understand these usage scenarios, the reader should be at least minimally familiar with the HIP protocol

specification [2]. However, for the (relatively) uninitiated reader it is most important to keep in mind that in HIP the actual payload traffic is protected with ESP, and that the ESP SPI acts as an index to the right host-to-host context.

Each of the scenarios below assumes that the HIP base exchange has completed, and the hosts each have a single outbound SA to the peer host. Associated with this outbound SA is a single destination address of the peer host-- the source address used by the peer during the base exchange.

The readdressing protocol is an asymmetric protocol where a mobile or multihomed host informs a peer host about changes of IP addresses on affected SPIs. The readdressing exchange is designed to be piggybacked on existing HIP exchanges. The main packets on which the LOCATOR parameters are expected to be carried are UPDATE packets. However, some implementations may want to experiment with sending LOCATOR parameters also on other packets, such as R1, I2, and NOTIFY.

Hosts that use link-local addresses as source addresses in their HIP handshakes may not be reachable by a mobile peer. Such hosts SHOULD provide a globally routable address either in the initial handshake or via the LOCATOR parameter.

3.2.1. Mobility with single SA pair (no rekeying)

A mobile host must sometimes change an IP address bound to an interface. The change of an IP address might be needed due to a change in the advertised IPv6 prefixes on the link, a reconnected PPP link, a new DHCP lease, or an actual movement to another subnet. In order to maintain its communication context, the host must inform its peers about the new IP address. This first example considers the case in which the mobile host has only one interface, IP address, a single pair of SAs (one inbound, one outbound), and no rekeying occurs on the SAs. We also assume that the new IP addresses are within the same address family (IPv4 or IPv6) as the first address. This is the simplest scenario, depicted in Figure 3.

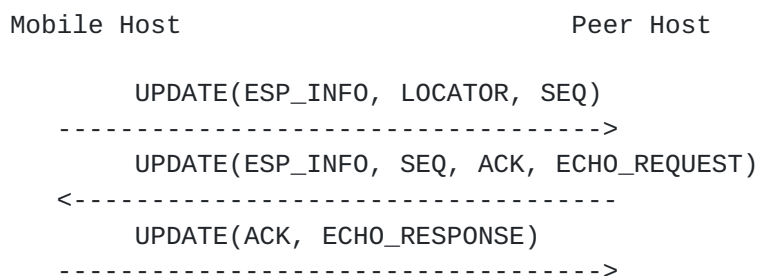


Figure 3: Readdress without rekeying, but with address check

1. The mobile host is disconnected from the peer host for a brief period of time while it switches from one IP address to another. Upon obtaining a new IP address, the mobile host sends a LOCATOR parameter to the peer host in an UPDATE message. The UPDATE message also contains an ESP_INFO parameter with the "Old SPI" and "New SPI" parameters both set to the value of the pre-existing incoming SPI; this ESP_INFO does not trigger a rekeying event but is instead included for possible parameter-inspecting middleboxes on the path. The LOCATOR parameter contains the new IP address (Locator Type of "1", defined below) and a locator lifetime. The mobile host waits for this UPDATE to be acknowledged, and retransmits if necessary, as specified in the base specification [2].
2. The peer host receives the UPDATE, validates it, and updates any local bindings between the HIP association and the mobile host's destination address. The peer host MUST perform an address verification by placing a nonce in the ECHO_REQUEST parameter of the UPDATE message sent back to the mobile host. It also includes an ESP_INFO parameter with the "Old SPI" and "New SPI" parameters both set to the value of the pre-existing incoming SPI, and sends this UPDATE (with piggybacked acknowledgment) to the mobile host at its new address. The peer MAY use the new address immediately, but it MUST limit the amount of data it sends to the address until address verification completes.
3. The mobile host completes the readdress by processing the UPDATE ACK and echoing the nonce in an ECHO_RESPONSE. Once the peer host receives this ECHO_RESPONSE, it considers the new address to be verified and can put it into full use.

While the peer host is verifying the new address, the new address is marked as UNVERIFIED in the interim, and the old address is DEPRECATED. Once the peer host has received a correct reply to its UPDATE challenge, it marks the new address as ACTIVE and removes the old address.

3.2.1.1. Mobility with single SA pair (mobile-initiated rekey)

The mobile host may decide to rekey the SAs at the same time that it is notifying the peer of the new address. In this case, the above procedure described in Figure 3 is slightly modified. The UPDATE message sent from the mobile host includes an ESP_INFO with the "Old SPI" set to the previous SPI, the "New SPI" set to the desired new SPI value for the incoming SA, and the Keymat Index desired. Optionally, the host may include a DIFFIE_HELLMAN parameter for a new Diffie-Hellman key. The peer completes the request for rekey as is normally done for HIP rekeying, except that the new address is kept

as UNVERIFIED until the UPDATE nonce challenge is received as described above. Figure 4 illustrates this scenario.

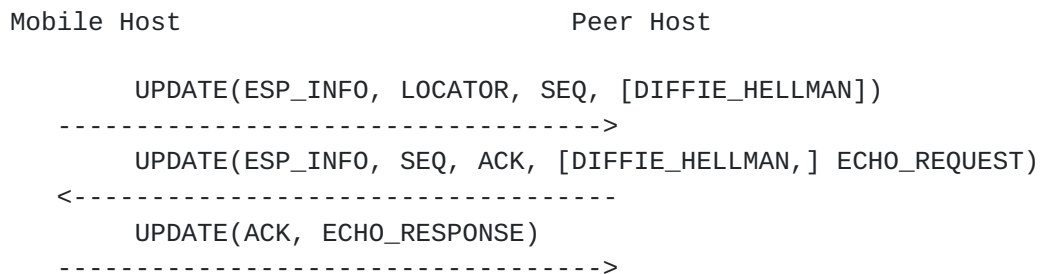


Figure 4: Readdress with mobile-initiated rekey

3.2.1.2. Mobility with single SA pair (peer-initiated rekey)

A second variation of this basic mobility scenario covers the case in which the mobile host does not attempt to rekey the existing SAs, but the peer host decides to do so. This typically results in a four packet exchange, as shown in Figure 5. The initial UPDATE packet from the mobile host is the same as in the scenario for which there is no rekey (Figure 3). The peer may decide to rekey, however, in which case the subsequent three packets follow the normal rekeying procedure described in the ESP specification [5], with the addition of the ECHO_REQUEST and ECHO_RESPONSE nonce for verification of the new address.

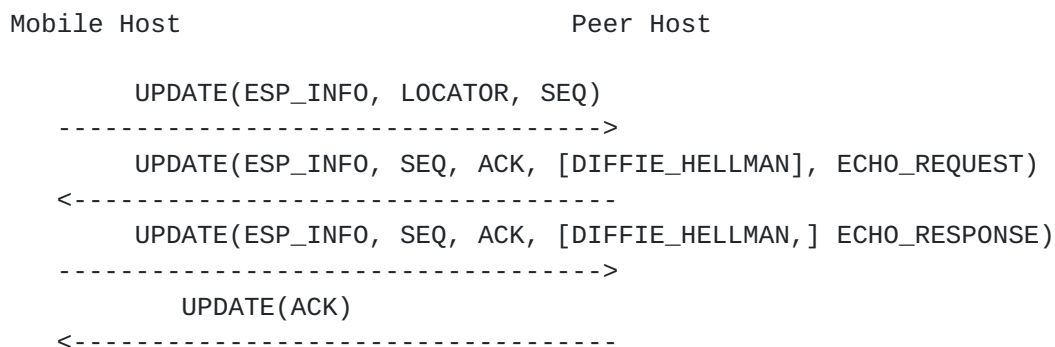


Figure 5: Readdress with peer-initiated rekey

3.2.2. Host multihoming

A (mobile or stationary) host may sometimes have more than one interface or global address. The host may notify the peer host of the additional interface or address by using the LOCATOR parameter. To avoid problems with the ESP anti-replay window, a host SHOULD use a different SA for each interface or address used to receive packets from the peer host.

When more than one locator is provided to the peer host, the host SHOULD indicate which locator is preferred. By default, the addresses used in the base exchange are preferred until indicated otherwise.

Although the protocol may allow for configurations in which there is an asymmetric number of SAs between the hosts (e.g., one host has two interfaces and two inbound SAs, while the peer has one interface and one inbound SA), it is RECOMMENDED that inbound and outbound SAs be created pairwise between hosts. When an ESP_INFO arrives to rekey a particular outbound SA, the corresponding inbound SA should be also rekeyed at that time. Although asymmetric SA configurations might be experimented with, their usage may constrain interoperability at this time. However, it is recommended that implementations attempt to support peers that prefer to use non-paired SAs. It is expected that this section and behavior will be modified in future revisions of this protocol, once the issue and its implications are better understood.

Consider the case between two single-homed hosts, in which one of the host notifies the peer of an additional address. It is RECOMMENDED that the host set up a new SA pair for use on this new address. To do this, the multihomed host sends a LOCATOR with an ESP_INFO, indicating the request for a new SA by setting the "Old SPI" value to zero, and the "New SPI" value to the newly created incoming SPI. A Locator Type of "1" is used to associate the new address with the new SPI. The LOCATOR parameter also contains a second Type 1 locator: that of the original address and SPI. To simplify parameter processing and avoid explicit protocol extensions to remove locators, each LOCATOR parameter must list all locators in use on a connection (a complete listing of inbound locators and SPIs for the host). The multihomed host transitions to state REKEYING, waiting for a ESP_INFO (new outbound SA) from the peer and an ACK of its own UPDATE. As in the mobility case, the peer host must perform an address verification before putting the new address into active use. Figure 6 illustrates the basic packet exchange.

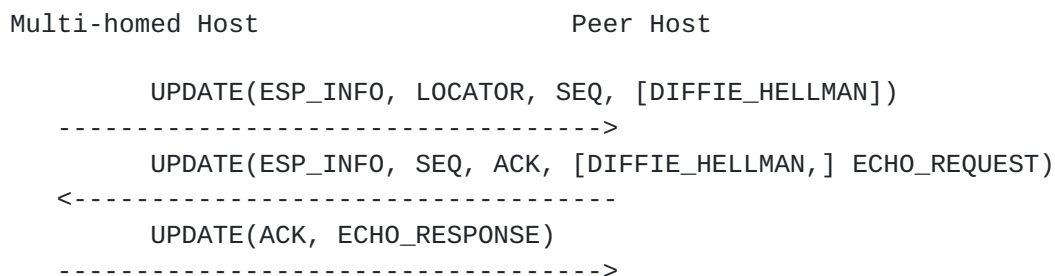


Figure 6: Basic multihoming scenario

Henderson (editor)

Expires August 28, 2006

[Page 14]

When processing inbound LOCATORs that establish new security associations on an interface with multiple addresses, a host uses the destination address of the UPDATE containing LOCATOR as the local address to which the LOCATOR plus ESP_INFO is targeted. Hosts may send UPDATES with the same IP address in the LOCATOR to different peer addresses-- this has the effect of creating multiple inbound SAs implicitly affiliated with different peer source addresses.

3.2.3. Site multihoming

A host may have an interface that has multiple globally reachable IP addresses. Such a situation may be a result of the site having multiple upper Internet Service Providers, or just because the site provides all hosts with both IPv4 and IPv6 addresses. It is desirable that the host can stay reachable with all or any subset of the currently available globally routable addresses, independent on how they are provided.

This case is handled the same as if there were different IP addresses, described above in [Section 3.2.2](#). Note that a single interface may experience site multihoming while the host itself may have multiple interfaces.

Note that a host may be multi-homed and mobile simultaneously, and that a multi-homed host may want to protect the location of some of its interfaces while revealing the real IP address of some others.

This document does not presently specify additional site multihoming extensions to HIP; further alignment with the IETF shim6 working group may be considered in the future.

3.2.4. Dual host multihoming

Consider the case in which both hosts would like to add an additional address after the base exchange completes. In Figure 7, consider that host1 wants to add address addr1b. It would send an UPDATE with LOCATOR to host2 located at addr2a, and a new set of SPIs would be added between hosts 1 and 2 (call them SPI1b and SPI2b). Next, consider host2 deciding to add addr2b to the relationship. host2 now has a choice to which of host1's addresses to initiate an UPDATE. It may choose to initiate an UPDATE to addr1a, addr1b, or both. If it chooses to send to both, then a full mesh (four SA pairs) of SAs would exist between the two hosts. This is the most general case; it often may be the case that hosts primarily establish new SAs only with the peer's preferred locator. The readdressing protocol is flexible enough to accommodate this choice.


```

      -<- SPI1a --                      -- SPI2a ->-
host1 <                               > addr1a <---> addr2a <                               > host2
      ->- SPI2a --                      -- SPI1a -<-

                                addr1b <---> addr2a (second SA pair)
                                addr1a <---> addr2b (third SA pair)
                                addr1b <---> addr2b (fourth SA pair)

```

Figure 7: Dual multihoming case in which each host uses LOCATOR to add a second address

3.2.5. Combined mobility and multihoming

It looks likely that in the future many mobile hosts will be simultaneously mobile and multi-homed, i.e., have multiple mobile interfaces. Furthermore, if the interfaces use different access technologies, it is fairly likely that one of the interfaces may appear stable (retain its current IP address) while some other(s) may experience mobility (undergo IP address change).

The use of LOCATOR plus ESP_INFO should be flexible enough to handle most such scenarios, although more complicated scenarios have not been studied so far.

3.2.6. Using LOCATORS across addressing realms

It is possible for HIP associations to migrate to a state in which both parties are only using locators in different addressing realms. For example, the two hosts may initiate the HIP association when both are using IPv6 locators, then one host may loose its IPv6 connectivity and obtain an IPv4 address. In such a case, some type of mechanism for interworking between the different realms must be employed; such techniques are outside the scope of the present text. If no mechanism exists, then the UPDATE message carrying the new LOCATOR will likely not reach the destination anyway, and the HIP state may time out.

3.2.7. Network renumbering

It is expected that IPv6 networks will be renumbered much more often than most IPv4 networks are. From an end-host point of view, network renumbering is similar to mobility.

3.2.8. Initiating the protocol in R1 or I2

A Responder host MAY include one or more LOCATOR parameters in the R1 packet that it sends to the Initiator. These parameters MUST be protected by the R1 signature. If the R1 packet contains LOCATOR

parameters with a new preferred locator, the Initiator SHOULD directly set the new preferred locator to status ACTIVE without performing address verification first, and MUST send the I2 packet to the new preferred locator. The I1 destination address and the new preferred locator may be identical. All new non-preferred locators must still undergo address verification.

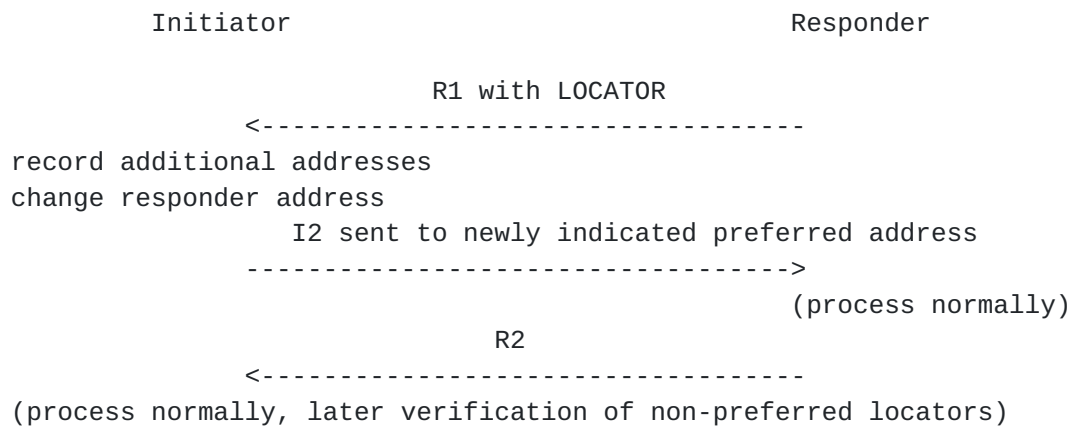


Figure 8: LOCATOR inclusion in R1

An Initiator MAY include one or more LOCATOR parameters in the I2 packet, independent of whether there was a LOCATOR parameter in the R1 or not. These parameters MUST be protected by the I2 signature. Even if the I2 packet contains LOCATOR parameters, the Responder MUST still send the R2 packet to the source address of the I2. The new preferred locator SHOULD be identical to the I2 source address. If the I2 packet contains LOCATOR parameters, all new locators must undergo address verification as usual.

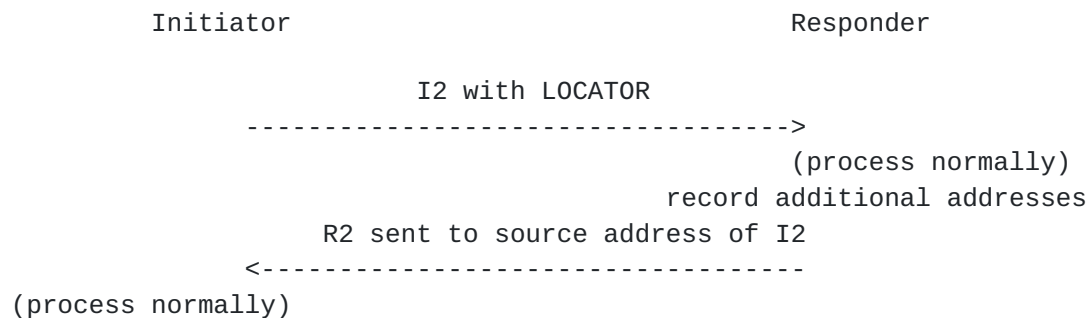


Figure 9: LOCATOR inclusion in I2

3.3. Other Considerations

3.3.1. Address Verification

When a HIP host receives a set of locators from another HIP host in a

LOCATOR, it does not necessarily know whether the other host is actually reachable at the claimed addresses. In fact, a malicious peer host may be intentionally giving bogus addresses in order to cause a packet flood towards the target addresses [8]. Likewise, viral software may have compromised the peer host, programming it to redirect packets to the target addresses. Thus, the HIP host must first check that the peer is reachable at the new address.

An additional potential benefit of performing address verification is to allow middleboxes in the network along the new path to obtain the peer host's inbound SPI.

Address verification is implemented by the challenger sending some piece of unguessable information to the new address, and waiting for some acknowledgment from the responder that indicates reception of the information at the new address. This may include exchange of a nonce, or generation of a new SPI and observing data arriving on the new SPI.

3.3.2. Credit-Based Authorization

Credit-Based Authorization allows a host to securely use a new locator even though the peer's reachability at the address embedded in this locator has not yet been verified. This is accomplished based on the following three hypotheses:

1. A flooding attacker typically seeks to somehow multiply the packets it generates itself for the purpose of its attack because bandwidth is an ample resource for many attractive victims.
2. An attacker can always cause unamplified flooding by sending packets to its victim directly.
3. Consequently, the additional effort required to set up a redirection-based flooding attack would pay off for the attacker only if amplification could be obtained this way.

On this basis, rather than eliminating malicious packet redirection in the first place, Credit-Based Authorization prevents any amplification that can be reached through it. This is accomplished by limiting the data a host can send to an unverified address of a peer by the data recently received from that peer. Redirection-based flooding attacks thus become less attractive than, e.g., pure direct flooding, where the attacker itself sends bogus packets to the victim.

Figure 10 illustrates Credit-Based Authorization: Host B measures the bytes recently received from peer A and, when A readdresses, sends

packets to A's new, unverified address as long as the sum of their sizes does not exceed the measured, received data volume. When insufficient credit is left, B stops sending further packets to A until A's address becomes ACTIVE. The address changes may be due to mobility, due to multihoming, or due to any other reason.

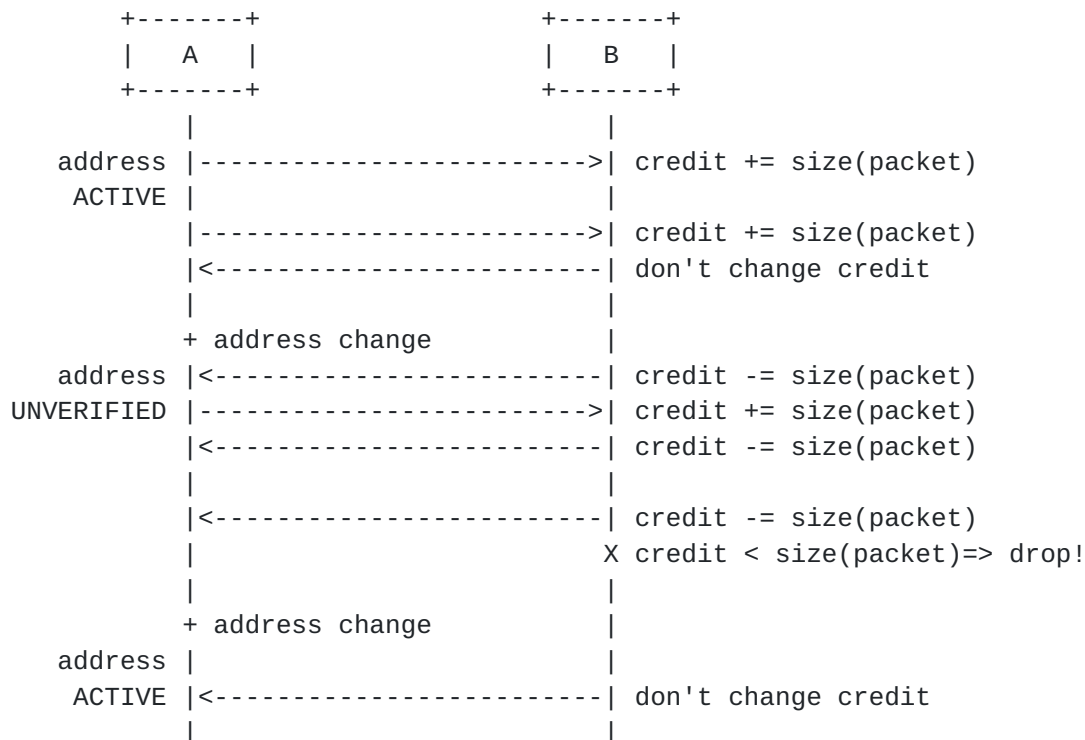


Figure 10: Readdressing Scenario

3.3.3. Preferred locator

When a host has multiple locators, the peer host must decide upon which to use for outbound packets. It may be that a host would prefer to receive data on a particular inbound interface. HIP allows a particular locator to be designated as a preferred locator, and communicated to the peer (see [Section 4](#)).

In general, when multiple locators are used for a session, there is the question of using multiple locators for failover only or for load-balancing. Due to the implications of load-balancing on the transport layer that still need to be worked out, this draft assumes that multiple locators are used primarily for failover. An implementation may use ICMP interactions, reachability checks, or other means to detect the failure of a locator.

3.3.4. Interaction with Security Associations

This document specifies a new HIP protocol parameter, the LOCATOR parameter (see [Section 4](#)), that allows the hosts to exchange information about their locator(s), and any changes in their locator(s). The logical structure created with LOCATOR parameters has three levels: hosts, Security Associations (SAs) indexed by Security Parameter Indices (SPIs), and addresses.

The relation between these entities for an association negotiated as defined in the base specification [2] and ESP transform [5] is illustrated in Figure 11.

```

      -<- SPI1a --                      -- SPI2a ->-
host1 <                      > addr1a <---> addr2a <                      > host2
      ->- SPI2a --                      -- SPI1a -<-

```

Figure 11: Relation between hosts, SPIs, and addresses (base specification)

In Figure 11, host1 and host2 negotiate two unidirectional SAs, and each host selects the SPI value for its inbound SA. The addresses addr1a and addr2a are the source addresses that each host uses in the base HIP exchange. These are the "preferred" (and only) addresses conveyed to the peer for each SA; even though packets sent to any of the hosts' interfaces can arrive on an inbound SPI, when a host sends packets to the peer on an outbound SPI, it knows of a single destination address associated with that outbound SPI (for host1, it sends a packet on SPI2a to addr2a to reach host2), unless other mechanisms exist to learn of new addresses.

In general, the bindings that exist in an implementation corresponding to this draft can be depicted as shown in Figure 12. In this figure, a host can have multiple inbound SPIs (and, not shown, multiple outbound SPIs) between itself and another host. Furthermore, each SPI may have multiple addresses associated with it. These addresses bound to an SPI are not used as SA selectors. Rather, the addresses are those addresses that are provided to the peer host, as hints for which addresses to use to reach the host on that SPI. The LOCATOR parameter allows for IP addresses and SPIs to be combined to form generalized locators. The LOCATOR parameter is used to change the set of addresses that a peer associates with a particular SPI.



Figure 12: Relation between hosts, SPIs, and addresses (general case)

A host may establish any number of security associations (or SPIs) with a peer. The main purpose of having multiple SPIs is to group the addresses into collections that are likely to experience fate sharing. For example, if the host needs to change its addresses on SPI2, it is likely that both address21 and address22 will simultaneously become obsolete. In a typical case, such SPIs may correspond with physical interfaces; see below. Note, however, that especially in the case of site multihoming, one of the addresses may become unreachable while the other one still works. In the typical case, however, this does not require the host to inform its peers about the situation, since even the non-working address still logically exists.

A basic property of HIP SAs is that the inbound IP address is not used as a selector for the SA. Therefore, in Figure 12, it may seem unnecessary for address31, for example, to be associated only with SPI3-- in practice, a packet may arrive to SPI1 via destination address address31 as well. However, the use of different source and destination addresses typically leads to different paths, with different latencies in the network, and if packets were to arrive via an arbitrary destination IP address (or path) for a given SPI, the reordering due to different latencies may cause some packets to fall outside of the ESP anti-replay window. For this reason, HIP provides a mechanism to affiliate destination addresses with inbound SPIs, if there is a concern that anti-replay windows might be violated otherwise. In this sense, we can say that a given inbound SPI has an "affinity" for certain inbound IP addresses, and this affinity is communicated to the peer host. Each physical interface SHOULD have a separate SA, unless the ESP anti-replay window is loose.

Moreover, even if the destination addresses used for a particular SPI are held constant, the use of different source interfaces may also cause packets to fall outside of the ESP anti-replay window, since the path traversed is often affected by the source address or

interface used. A host has no way to influence the source interface on which a peer uses to send its packets on a given SPI. Hosts SHOULD consistently use the same source interface and address when sending to a particular destination IP address and SPI. For this reason, a host may find it useful to change its SPI or at least reset its ESP anti-replay window when the peer host readdresses.

An address may appear on more than one SPI. This creates no ambiguity since the receiver will ignore the IP addresses as SA selectors anyway. However, this document does not specify such cases.

If the LOCATOR parameter is sent in an UPDATE packet, then the receiver will respond with an UPDATE acknowledgment. If the LOCATOR parameter is sent in a NOTIFY, I2, or R2 packet, then the recipient may consider the LOCATOR as informational, and act only when it needs to activate a new address. The use of LOCATOR in a NOTIFY message may not be compatible with middleboxes.

Locator Type: Defines the semantics of the Locator field.

Locator Length: Defines the length of the Locator field, in units of 4-byte words (Locators up to a maximum of 4*255 bytes are supported).

Reserved: Zero when sent, ignored when received.

P: Preferred locator. Set to one if the locator is preferred for that Traffic Type; otherwise set to zero.

Locator Lifetime: Locator lifetime, in seconds.

Locator: The locator whose semantics and encoding are indicated by the Locator Type field. All Locator sub-fields are integral multiples of four bytes in length.

The Locator Lifetime indicates how long the following locator is expected to be valid. The lifetime is expressed in seconds. Each locator MUST have a non-zero lifetime. The address is expected to become deprecated when the specified number of seconds has passed since the reception of the message. A deprecated address SHOULD NOT be used as a destination address if an alternate (non-deprecated) is available and has sufficient scope.

4.1. Traffic Type and Preferred Locator

The following Traffic Type values are defined:

0: Both signaling (HIP control packets) and user data.

1: Signaling packets only.

2: Data packets only.

The "P" bit, when set, has scope over the corresponding Traffic Type that precedes it. That is, if a "P" bit is set for Traffic Type "2", for example, that means that the locator is preferred for data packets. If there is a conflict (for example, if P bit is set for an address of Type "0" and a different address of Type "2"), the more specific Traffic Type rule applies. By default, the IP addresses used in the base exchange are preferred locators for both signaling and user data, unless a new preferred locator supersedes them. If no locators are indicated as preferred for a given Traffic Type, the implementation may use an arbitrary locator from the set of active locators.

4.2. Locator Type and Locator

The following Locator Type values are defined, along with the associated semantics of the Locator field:

- 0: An IPv6 address or an IPv4-in-IPv6 format IPv4 address [[7](#)] (128 bits long).
- 1: The concatenation of an ESP SPI (first 32 bits) followed by an IPv6 address or an IPv4-in-IPv6 format IPv4 address (an additional 128 bits).

4.3. UPDATE packet with included LOCATOR

A number of combinations of parameters in an UPDATE packet are possible (e.g., see [Section 3.2](#)). Only one LOCATOR parameter is used in any HIP packet, and this LOCATOR SHOULD list all of the locators that the host wishes to make available for the HIP association. Any UPDATE packet that includes a LOCATOR parameter SHOULD include both an HMAC and a HIP_SIGNATURE parameter.

5. Processing rules

5.1. Locator data structure and status

In a typical implementation, each outgoing locator is represented by a piece of state that contains the following data:

- o the actual bit pattern representing the locator,
- o lifetime (seconds),
- o status (UNVERIFIED, ACTIVE, DEPRECATED).

The status is used to track the reachability of the address embedded within the LOCATOR parameter:

UNVERIFIED indicates that the reachability of the address has not been verified yet,

ACTIVE indicates that the reachability of the address has been verified and the address has not been deprecated,

DEPRECATED indicates that the locator lifetime has expired

The following state changes are allowed:

UNVERIFIED to ACTIVE The reachability procedure completes successfully.

UNVERIFIED to DEPRECATED The locator lifetime expires while it is UNVERIFIED.

ACTIVE to DEPRECATED The locator lifetime expires while it is ACTIVE.

ACTIVE to UNVERIFIED There has been no traffic on the address for some time, and the local policy mandates that the address reachability must be verified again before starting to use it again.

DEPRECATED to UNVERIFIED The host receives a new lifetime for the locator.

A DEPRECATED address MUST NOT be changed to ACTIVE without first verifying its reachability.

5.2. Sending LOCATORS

The decision of when to send LOCATORS is basically a local policy

issue. However, it is RECOMMENDED that a host sends a LOCATOR whenever it recognizes a change of its IP addresses in use on an active HIP association, and assumes that the change is going to last at least for a few seconds. Rapidly sending conflicting LOCATORS SHOULD be avoided.

When a host decides to inform its peers about changes in its IP addresses, it has to decide how to group the various addresses with SPIs. The grouping should consider also whether middlebox interaction requires sending (the same) LOCATOR in separate UPDATES on different paths. Since each SPI is associated with a different Security Association, the grouping policy may also be based on ESP anti-replay protection considerations. In the typical case, simply basing the grouping on actual kernel level physical and logical interfaces may be the best policy. Grouping policy is outside of the scope of this document.

Note that the purpose of announcing IP addresses in a LOCATOR is to provide connectivity between the communicating hosts. In most cases, tunnels or virtual interfaces such as IPsec tunnel interfaces or Mobile IP home addresses provide sub-optimal connectivity. Furthermore, it should be possible to replace most tunnels with HIP based "non-tunneling", therefore making most virtual interfaces fairly unnecessary in the future. Therefore, virtual interfaces SHOULD NOT be announced in general. On the other hand, there are clearly situations where tunnels are used for diagnostic and/or testing purposes. In such and other similar cases announcing the IP addresses of virtual interfaces may be appropriate.

Once the host has decided on the groups and assignment of addresses to the SPIs, it creates a LOCATOR parameter that serves as a complete representation of the addresses and affiliated SPIs intended for active use. We now describe a few cases introduced in [Section 3.2](#). We assume that the Traffic Type for each locator is set to "0" (other values for Traffic Type may be specified in documents that separate HIP control plane from data plane traffic). Other mobility and multihoming cases are possible but are left for further experimentation.

1. Host mobility with no multihoming and no rekeying. The mobile host creates a single UPDATE containing a single ESP_INFO with a single LOCATOR parameter. The ESP_INFO contains the current value of the SPI in both the "Old SPI" and "New SPI" fields. The LOCATOR contains a single Locator with a "Locator Type" of "1"; the SPI must match that of the ESP_INFO. The Preferred bit SHOULD be set and the "Locator Lifetime" is set according to local policy. The UPDATE also contains a SEQ parameter as usual and is protected by retransmission. The UPDATE should be sent to

the peer's preferred IP address with an IP source address corresponding to the address in the LOCATOR parameter.

2. Host mobility with no multihoming but with rekeying. The mobile host creates a single UPDATE containing a single ESP_INFO with a single LOCATOR parameter (with a single address). The ESP_INFO contains the current value of the SPI in the "Old SPI" and the new value of the SPI in the "New SPI", and a "Keymat Index" as selected by local policy. Optionally, the host may choose to initiate a Diffie Hellman rekey by including a DIFFIE_HELLMAN parameter. The LOCATOR contains a single Locator with "Locator Type" of "1"; the SPI must match that of the "New SPI" in the ESP_INFO. Otherwise, the steps are identical to the case when no rekeying is initiated.
3. Host multihoming (addition of an address). We only describe the simple case of adding an additional address to a single-homed, non-mobile host. The host SHOULD set up a new SA pair between this new address and the preferred address of the peer host. To do this, the multihomed host creates a new inbound SA and creates a new ESP_INFO parameter with an "Old SPI" parameter of "0", a "New SPI" parameter corresponding to the new SPI, and a "Keymat Index" as selected by local policy. The host adds to the UPDATE message a LOCATOR with two Type "1" Locators: the original address and SPI active on the association, and the new address and new SPI being added (with the SPI matching the "New SPI" contained in the ESP_INFO). The Preferred bit SHOULD be set depending on the policy to tell the peer host which of the two locators is preferred. The UPDATE also contains a SEQ parameter and optionally a DIFFIE_HELLMAN parameter, and follows rekeying procedures with respect to this new address. The UPDATE message SHOULD be sent to the peer's preferred address with a source address corresponding to the new locator.

The sending of multiple LOCATORS, locators with Locator Type "0", and multiple ESP_INFO parameters is for further study.

5.3. Handling received LOCATORS

A host SHOULD be prepared to receive a LOCATOR parameter in any HIP packet, excluding I1.

This document describes sending both ESP_INFO and LOCATOR parameters in an UPDATE. The ESP_INFO parameter is included if there is a need to rekey or key a new SPI, and is otherwise included for the possible benefit of HIP-aware middleboxes. The LOCATOR parameter contains a complete map of the locators that the host wishes to make or keep active for the HIP association.

In general, the processing of a LOCATOR depends upon the packet type in which it is included and upon whether ESP_INFO parameter is included. Here, we describe only the case in which ESP_INFO is present and a single LOCATOR and ESP_INFO are sent in an UPDATE message; other cases are for further study. The steps below cover each of the cases described in [Section 5.2](#).

When a host receives a LOCATOR parameter in a validated HIP packet, it first performs the following operations:

1. The host checks if the New SPI listed in the ESP_INFO is a new one. If it is a new one, it creates a new inbound SA with that SPI that contains no addresses. If it is an existing one, it prepares to change the address set on the existing SPI.
2. For each locator listed in the LOCATOR parameter, check that the address therein is a legal unicast or anycast address. That is, the address MUST NOT be a broadcast or multicast address. Note that some implementations MAY accept addresses that indicate the local host, since it may be allowed that the host runs HIP with itself.
3. For each Type 1 address listed in the LOCATOR parameter, check if the address is already bound to the SPI indicated. If the address is already bound, its lifetime is updated. If the status of the address is DEPRECATED, the status is changed to UNVERIFIED. If the address is not already bound, the address is added, and its status is set to UNVERIFIED. Mark all addresses on the SPI that were NOT listed in the LOCATOR parameter as DEPRECATED. As a result, the SPI now contains any addresses listed in the LOCATOR parameter either as UNVERIFIED or ACTIVE, and any old addresses not listed in the LOCATOR parameter as DEPRECATED.
4. If the LOCATOR is paired with an ESP_INFO parameter, the ESP_INFO parameter is processed as follows:
 1. If the Old SPI indicates an existing SPI and the New SPI is a different non-zero value, the existing SA is being rekeyed and the host follows HIP ESP rekeying procedures. Note that the Locators in the LOCATOR parameter will use this New SPI instead of the Old SPI.
 2. If the Old SPI value is zero and the New SPI is a new non-zero value, then a new SA is being requested by the peer. This case is also treated like a rekeying event; the receiving host must create a new inbound SA and respond with an UPDATE ACK.

3. If the Old SPI indicates an existing SPI and the New SPI is zero, the SPI is being deprecated and all locators uniquely bound to the SPI are put into DEPRECATED state.
4. If the Old SPI equals the New SPI and both correspond to an existing SPI, the ESP_INFO is gratuitous (provided for middleboxes) and no rekeying is necessary.
5. Mark all locators on each SPI that were NOT listed in the LOCATOR parameter as DEPRECATED.

As a result, each SPI now contains any addresses listed in the LOCATOR parameter either as UNVERIFIED or ACTIVE, and any old addresses not listed in the LOCATOR parameter as DEPRECATED.

Once the host has updated the SPI, if the LOCATOR parameter contains a new preferred locator, the host SHOULD initiate a change of the preferred locator. This requires that the host first verifies reachability of the associated address, and only then changes the preferred locator. See [Section 5.6](#).

5.4. Verifying address reachability

A host MUST verify the reachability of an UNVERIFIED address. The status of a newly learned address MUST initially be set to UNVERIFIED unless the new address is advertised in a R1 packet as a new preferred locator. A host MAY also want to verify the reachability of an ACTIVE address again after some time, in which case it would set the status of the address to UNVERIFIED and reinitiate address verification

A host typically starts the address-verification procedure by sending a nonce to the new address. For example, if the host is changing its SPI and is sending an ESP_INFO to the peer, the new SPI value SHOULD be random and the value MAY be copied into an ECHO_REQUEST sent in the rekeying UPDATE. If the host is not rekeying, it MAY still use the ECHO_REQUEST parameter in an UPDATE message sent to the new address. A host MAY also use other message exchanges as confirmation of the address reachability.

Note that in the case of receiving a LOCATOR on an R1 and replying with an I2, receiving the corresponding R2 is sufficient proof of reachability for the Responder's preferred address. Since further address verification of such address can impede the HIP base exchange, a host MUST NOT verify reachability of a new preferred locator that was received on a R1.

In some cases, it may be sufficient to use the arrival of data on a

newly advertised SA as implicit address reachability verification, instead of waiting for the confirmation via a HIP packet (e.g., Figure 14). In this case, a host advertising a new SPI as part of its address reachability check **SHOULD** be prepared to receive traffic on the new SA. Marking the address **ACTIVE** as a part of receiving data on the SA is an idempotent operation, and does not cause any harm.

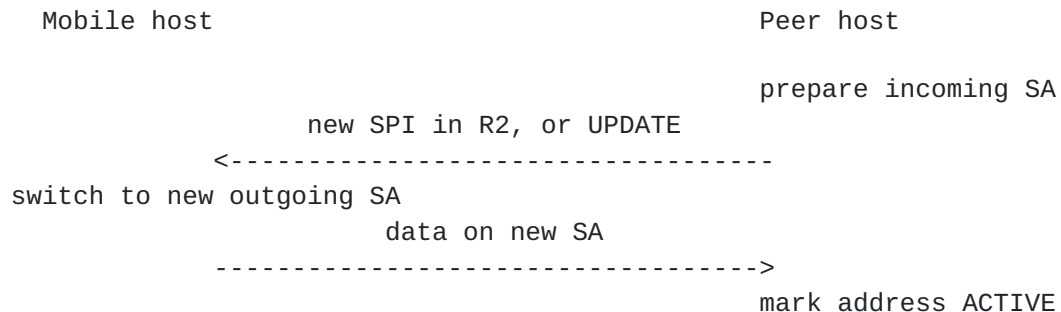


Figure 14: Address activation via use of new SA

When address verification is in progress for a new preferred locator, the host **SHOULD** select a different locator listed as **ACTIVE**, if one such locator is available, to continue communications until address verification completes. Alternatively, the host **MAY** use the new preferred locator while in **UNVERIFIED** status to the extent Credit-Based Authorization permits. Credit-Based Authorization is explained in [Section 5.5](#). Once address verification succeeds, the status of the new preferred locator changes to **ACTIVE**.

5.5. Credit-Based Authorization

5.5.1. Handling Payload Packets

A host maintains a "credit counter" for each of its peers. Whenever a packet arrives from a peer, the host **SHOULD** increase that peer's credit counter by the size of the received packet. When the host has a packet to be sent to the peer, if the peer's preferred locator is listed as **UNVERIFIED** and no alternative locator with status **ACTIVE** is available, the host checks whether it can send the packet to the **UNVERIFIED** locator: The packet **SHOULD** be sent if the value of the credit counter is higher than the size of the outbound packet. If the credit counter is too low, the packet **MUST** be discarded or buffered until address verification succeeds. When a packet is sent to a peer at an **UNVERIFIED** locator, the peer's credit counter **MUST** be reduced by the size of the packet. The peer's credit counter is not affected by packets that the host sends to an **ACTIVE** locator of that peer.

Figure 15 depicts the actions taken by the host when a packet is received. Figure 16 shows the decision chain in the event a packet is sent.

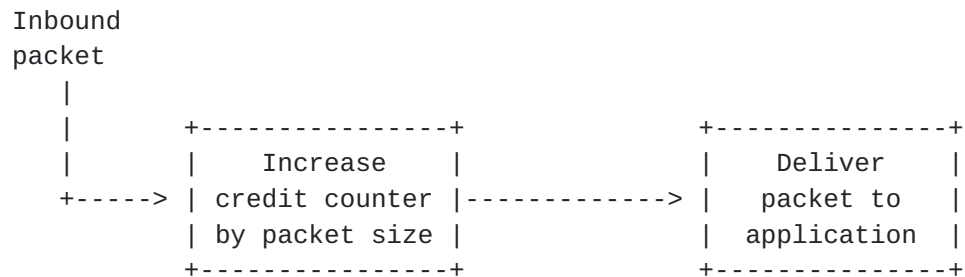


Figure 15: Receiving Packets with Credit-Based Authorization

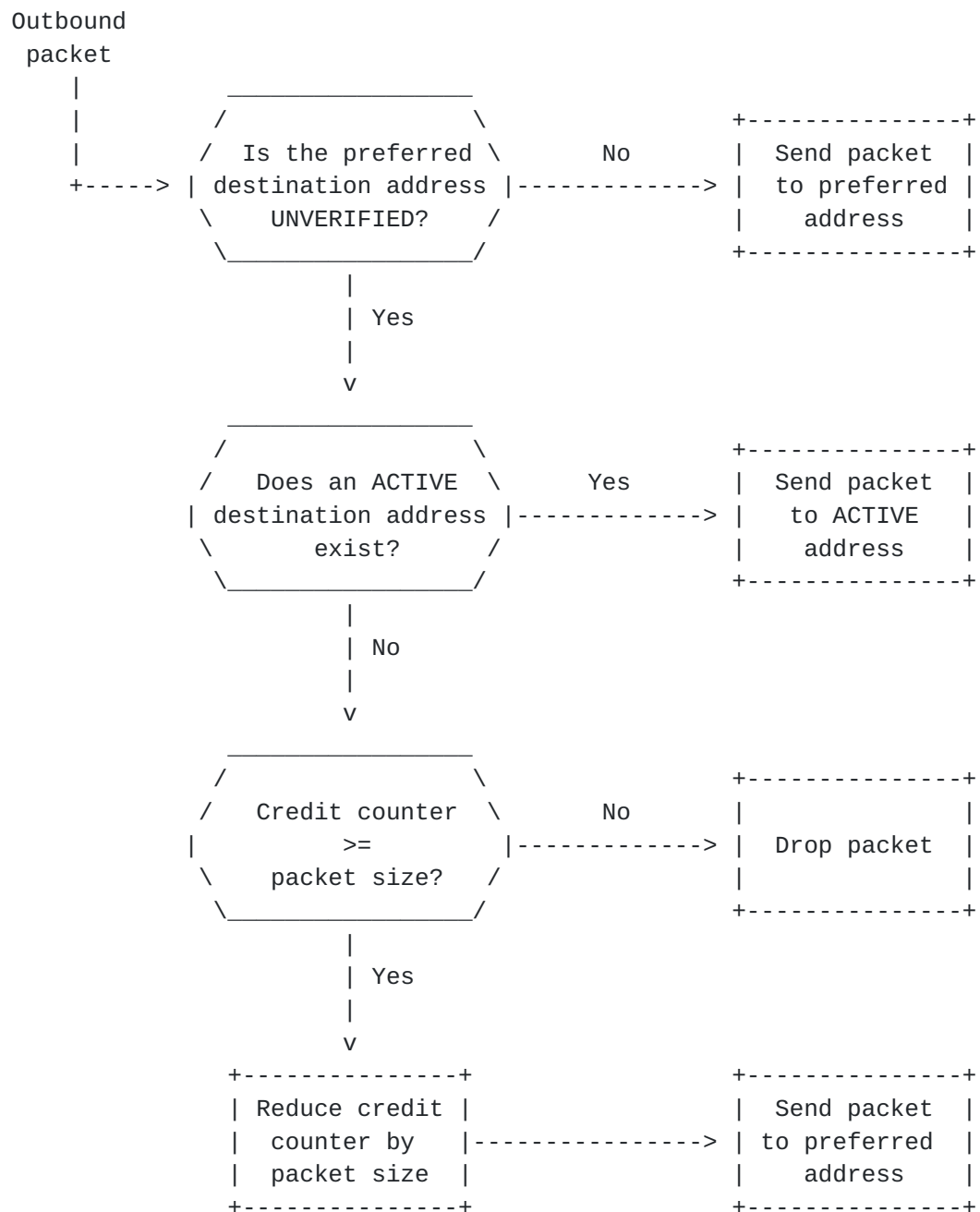


Figure 16: Sending Packets with Credit-Based Authorization

5.5.2. Credit Aging

A host ensures that the credit counters it maintains for its peers gradually decrease over time. Such "credit aging" prevents a malicious peer from building up credit at a very slow speed and using this, all at once, for a severe burst of redirected packets.

Credit aging may be implemented by multiplying credit counters with a factor, `CreditAgingFactor`, less than one in fixed time intervals of `CreditAgingInterval` length. Choosing appropriate values for `CreditAgingFactor` and `CreditAgingInterval` is important to ensure that a host can send packets to an address in state `UNVERIFIED` even when the peer sends at a lower rate than the host itself. When `CreditAgingFactor` or `CreditAgingInterval` are too small, the peer's credit counter might be too low to continue sending packets until address verification concludes.

The parameter values proposed in this document are as follows:

<code>CreditAgingFactor</code>	7/8
<code>CreditAgingInterval</code>	5 seconds

These parameter values work well when the host transfers a file to the peer via a TCP connection and the end-to-end round-trip time does not exceed 500 milliseconds. Alternative credit-aging algorithms may use other parameter values or different parameters, which may even be dynamically established.

5.6. Changing the preferred locator

A host MAY want to change the preferred outgoing locator for different reasons, e.g., because traffic information or ICMP error messages indicate that the currently used preferred address may have become unreachable. Another reason may be due to receiving a `LOCATOR` parameter that has the P-bit set.

To change the preferred locator, the host initiates the following procedure:

1. If the new preferred locator has `ACTIVE` status, the preferred locator is changed and the procedure succeeds.
2. If the new preferred locator has `UNVERIFIED` status, the host starts to verify its reachability. The host SHOULD use a different locator listed as `ACTIVE` until address verification completes if one such locator is available. Alternatively, the host MAY use the new preferred locator, even though in `UNVERIFIED` status, to the extent Credit-Based Authorization permits. Once address verification succeeds, the status of the new preferred locator changes to `ACTIVE` and its use is no longer governed by Credit-Based Authorization.
3. If the peer host has not indicated a preference for any address, then the host picks one of the peer's `ACTIVE` addresses randomly

or according to policy. This case may arise if, for example, ICMP error messages arrive that deprecate the preferred locator, but the peer has not yet indicated a new preferred locator.

4. If the new preferred locator has DEPRECATED status and there is at least one non-deprecated address, the host selects one of the non-deprecated addresses as a new preferred locator and continues. If the selected address is UNVERIFIED, this includes address verification as described above.

6. Security Considerations

The HIP mobility mechanism provides a secure means of updating a host's IP address via HIP UPDATE packets. Upon receipt, a HIP host cryptographically verifies the sender of an UPDATE, so forging or replaying a HIP UPDATE packet is very difficult (see [2]). Therefore, security issues reside in other attack domains. The two we consider are malicious redirection of legitimate connections as well as redirection-based flooding attacks using this protocol. This can be broken down into the following:

Impersonation attacks

- direct conversation with the misled victim
- man-in-the-middle attack

DoS attacks

- flooding attacks (== bandwidth-exhaustion attacks)
 - * tool 1: direct flooding
 - * tool 2: flooding by zombies
 - * tool 2: redirection-based flooding
- memory-exhaustion attacks
- computational exhaustion attacks

We consider these in more detail in the following sections.

In [Section 6.1](#) and [Section 6.2](#), we assume that all users are using HIP. In [Section 6.3](#) we consider the security ramifications when we have both HIP and non-HIP users.

6.1. Impersonation attacks

An attacker wishing to impersonate will try to mislead its victim into directly communicating with them, or carry out a man in the middle attack between the victim and the victim's desired communication peer. Without mobility support, both attack types are possible only if the attacker resides on the routing path between its victim and the victim's desired communication peer, or if the attacker tricks its victim into initiating the connection over an incorrect routing path (e.g., by acting as a router or using spoofed DNS entries).

The HIP extensions defined in this specification change the situation in that they introduce an ability to redirect a connection (like IPv6), both before and after establishment. If no precautionary measures are taken, an attacker could misuse this feature to impersonate a victim's peer from any arbitrary location. The authentication and authorization mechanisms of the HIP base exchange [2] and the signatures in the UPDATE message prevent this attack. Furthermore, ownership of a HIP association is securely linked to a HIP HI/HIT. If an attacker somehow uses a bug in the implementation or weakness in some protocol to redirect a HIP connection, the original owner can always reclaim their connection (they can always prove ownership of the private key associated with their public HI).

MitM attacks are always possible if the attacker is present during the initial HIP base exchange and if the hosts do not authenticate each other's identities, but once the base exchange has taken place even a MitM cannot steal an opportunistic HIP connection because it is very difficult for an attacker to create an UPDATE packet (or any HIP packet) that will be accepted as a legitimate update. UPDATE packets use HMAC and are signed. Even when an attacker can snoop packets to obtain the SPI and HIT/HI, they still cannot forge an UPDATE packet without knowledge of the secret keys.

6.2. Denial of Service attacks

6.2.1. Flooding Attacks

The purpose of a denial-of-service attack is to exhaust some resource of the victim such that the victim ceases to operate correctly. A denial-of-service attack can aim at the victim's network attachment (flooding attack), its memory, or its processing capacity. In a flooding attack the attacker causes an excessive number of bogus or unwanted packets to be sent to the victim, which fills their available bandwidth. Note that the victim does not necessarily need to be a node; it can also be an entire network. The attack basically functions the same way in either case.

An effective DoS strategy is distributed denial of service (DDoS). Here, the attacker conventionally distributes some viral software to as many nodes as possible. Under the control of the attacker, the infected nodes, or "zombies", jointly send packets to the victim. With such an 'army', an attacker can take down even very high bandwidth networks/victims.

With the ability to redirect connections, an attacker could realize a DDoS attack without having to distribute viral code. Here, the attacker initiates a large download from a server, and subsequently redirects this download to its victim. The attacker can repeat this

with multiple servers. This threat is mitigated through reachability checks and credit-based authorization. Both strategies do not eliminate flooding attacks per se, but they preclude: (i) their use from a location off the path towards the flooded victim; and (ii) any amplification in the number and size of the redirected packets. As a result, the combination of a reachability check and credit-based authorization makes a HIP redirection-based flooding attack as effective and applicable as a normal, direct flooding attack in which the attacker itself sends the flooding traffic to the victim.

This analysis leads to the following two points. First, when a reachability packet is received, this nonce packet **MUST** be ignored if the HIT is not one that is currently active. Second, if the attacker is a MitM and can capture this nonce packet then it can respond to it, in which case it is possible for an attacker to redirect the connection. Note, this attack will always be possible when a reachability packet is not sent.

6.2.2. Memory/Computational exhaustion DoS attacks

We now consider whether or not the proposed extensions to HIP add any new DoS attacks (consideration of DoS attacks using the base HIP exchange and updates is discussed in [2]). A simple attack is to send many UPDATE packets containing many IP addresses that are not flagged as preferred. The attacker continues to send such packets until the number of IP addresses associated with the attacker's HI crashes the system. Therefore, there **SHOULD** be a limit to the number of IP addresses that can be associated with any HI. Other forms of memory/computationally exhausting attacks via the HIP UPDATE packet are handled in the base HIP draft [2].

6.3. Mixed deployment environment

We now assume an environment with both HIP and non-HIP aware hosts. Four cases exist.

1. A HIP user redirects their connection onto a non-HIP user. The non-HIP user will drop the reachability packet so this is not a threat unless the HIP user is a MitM and can respond to the reachability packet.
2. A non-HIP user attempts to redirect their connection onto a HIP user. This falls into IPv4 and IPv6 security concerns, which are outside the scope of this document.
3. A non-HIP user attempts to steal a HIP user's session (assume that Secure Neighbor Discovery is not active for the following). The non-HIP user contacts the service that a HIP user has a

connection with and then attempts to use a IPv6 change of address request to steal the HIP user's connection. What will happen in this case is implementation dependent but such a request should be ignored/dropped. Even if the attack is successful, the HIP user can reclaim its connection via HIP.

4. A HIP user attempts to steal a non-HIP user's session. This could be problematic since HIP sits 'on top of' layer 3. A HIP user could spoof the non-HIP user's IP address during the base exchange or set the non-HIP user's IP address as their preferred address via an UPDATE. Other possibilities exist but a simple solution is to add a check which does not allow any HIP session to be moved to or created upon an already existing IP address.

7. IANA Considerations

This document defines a LOCATOR parameter for the Host Identity Protocol [\[2\]](#). This parameter is defined in [Section 4](#) with a Type of 193.

8. Authors

Pekka Nikander originated this Internet Draft. Tom Henderson, Jari Arkko, Greg Perkins, and Christian Vogt have each contributed sections to this draft.

9. Acknowledgments

The authors thank Mika Kousa, Jeff Ahrenholz, and Jan Melen for many improvements to the draft.

10. References

10.1. Normative references

- [1] Moskowitz, R. and P. Nikander, "Host Identity Protocol Architecture", [draft-ietf-hip-arch-03](#) (work in progress), August 2005.
- [2] Moskowitz, R., "Host Identity Protocol", [draft-ietf-hip-base-04](#) (work in progress), October 2005.
- [3] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", [draft-ietf-hip-rvs-04](#) (work in progress), October 2005.
- [4] Kent, S., "IP Encapsulating Security Payload (ESP)", [RFC 4303](#), December 2005.
- [5] Jokela, P., "Using ESP transport format with HIP", [draft-ietf-hip-esp-01](#) (work in progress), October 2005.
- [6] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [7] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", [RFC 2373](#), July 1998.

10.2. Informative references

- [8] Nikander, P., Arkko, J., Aura, T., Montenegro, G., and E. Nordmark, "Mobile IP Version 6 Route Optimization Security Design Background", [RFC 4225](#), December 2005.

Appendix A. Changes from previous versions

A.1. From `nikander-hip-mm-00` to `nikander-hip-mm-01`

The actual protocol has been largely revised, based on the new symmetric New SPI (NES) design adopted in the base protocol draft version -08. There are no more separate REA, AC or ACR packets, but their functionality has been folded into the NES packet. At the same time, it has become possible to send REA parameters in R1 and I2.

The Forwarding Agent functionality was removed, since it looks like that it will be moved to the proposed HIP Research Group. Hence, there will be two other documents related to that, a simple Rendezvous server document (WG item) and a Forwarding Agent document (RG item).

A.2. From `nikander-hip-mm-01` to `nikander-hip-mm-02`

Alignment with base-00 draft (use of UPDATE and NOTIFY packets).

The "logical interface" concept was dropped, and the SA/SPI was identified as the protocol component to which a HIP association binds addresses to.

The RR was (again) made recommended, not mandatory, able to be administratively overridden.

A.3. From -02 to [draft-ietf-hip-mm-00](#)

REA parameter type value is now "3" (was TBD before).

Recommend that in multihoming situations, that inbound/outbound SAs are paired to avoid ambiguity when rekeying them.

Clarified that multihoming scenario for now was intended for failover instead of load-balancing, due to transport layer issues.

Clarified that if HIP negotiates base exchange using link local addresses, that a host SHOULD provide its peer with a globally reachable address.

Clarified whether REAs sent for existing SPIs update the full set of addresses associated with that SPI, or only perform an incremental (additive) update. REAs for an existing SPI should list all current addresses for that SPI, and any addresses previously in use on the SPI but not in the new REA parameter should be DEPRECATED.

Clarified that address verification pertains to *outgoing* addresses.

When discussing inclusion of REA in I2, the draft stated "The Responder MUST make sure that the puzzle solution is valid BOTH for the initial IP destination address used for I1 and for the new preferred address." However, this statement conflicted with [Appendix D](#) of the base specification, so it has been removed for now.

A.4. From [draft-ietf-hip-mm-00](#) to -01

Introduction section reorganized. Some of the scope of the document relating to multihoming was reduced.

Removed empty appendix "Implementation experiences"

Renamed REA parameter to LOCATOR and aligned to the discussion on redefining this parameter that occurred on the RG mailing list.

Aligned with decoupling of ESP from base spec.

A.5. From [draft-ietf-hip-mm-01](#) to -02

Aligned with [draft-ietf-hip-base-03](#) and [draft-ietf-hip-esp-00](#)

Address verification is a MUST (C. Vogt, list post on 06/12/05)

If UPDATE exceeds MTU because of too many locators, do not split into multiple UPDATES, but instead rely on IP fragmentation (C. Vogt, list post on 06/12/05)

New value for LOCATOR parameter type (193), per 05/31/05 discussion on the WG list

Various additions related to Credit-Based Authorization due to C. Vogt

Security section contributed by Greg Perkins, with subsequent editing from C. Vogt and P. Nikander

Reorganization according to [RFC 4101](#) guidance on writing protocol models

Open issue: LOCATOR parameter semantics (implicit/explicit removal)

A.6. From [draft-ietf-hip-mm-02](#) to -03

Aligned with [draft-ietf-hip-base-05](#) and [draft-ietf-hip-esp-02](#)

Further clarification that the scope of this draft is primarily limited to the case in which ESP is used

New layered architectural overview in [Section 3](#)

Limited the scope of multihoming description to just a single host adding a single new address; other cases left for further study

Require that ESP_INFO be included on all UPDATE packets relating to mobility and multihoming (for middleboxes)

New convention for use of "Old SPI" and "New SPI" values to signal new SPIs (Old SPI == 0, New SPI != 0) and gratuitous ESP_INFOS with no rekeying (Old SPI == New SPI != 0).

Only specify the use of Locator Type of 1 when using ESP, for simplicity of receiver processing.

Removed multiple addresses in LOCATOR example of [section 3.2.2](#), because it is not clear that the example is correct (requires further study)

Corrected mention of sending ECHO_REQUEST nonce in R2 (should be sent in separate UPDATE because R2 is not an acknowledged packet)

Removed first four paragraphs of [Section 5](#), which were redundant with previous introductory material.

Rewrote Sections [5.2](#) and [5.3](#) on sending and receiving LOCATOR, to more explicitly cover the scenario scope of this document.

Removed unwritten "Policy Considerations" section

Author's Address

Tom Henderson
The Boeing Company
P.O. Box 3707
Seattle, WA
USA

Email: thomas.r.henderson@boeing.com

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

