Network Working Group
Internet-Draft
Obsoletes: 5206 (if approved)
Intended status: Standards Track
Expires: July 26, 2016

T. Henderson, Ed.
University of Washington
C. Vogt
J. Arkko
Ericsson Research NomadicLab
January 23, 2016

Host Mobility with the Host Identity Protocol
draft-ietf-hip-rfc5206-bis-10

Abstract

This document defines mobility extensions to the Host Identity Protocol (HIP). Specifically, this document defines a general "LOCATOR_SET" 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_SET parameter can also be used to support end-host multihoming, detailed procedures are out of scope for this document. This document obsoletes RFC 5206.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on July 26, 2016.

Table of Contents

1. Introduction and Scope

The Host Identity Protocol [RFC7401] (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 Encapsulating Security Payload (ESP) Security Associations (SAs)) 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.

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 host mobility, leaving more complicated mobility scenarios, multihoming, and other variations for further study. More specifically:

> This document defines a generalized LOCATOR_SET parameter for use in HIP messages. The LOCATOR_SET parameter allows a HIP host to notify a peer about alternate locators at which it is reachable. The locators may be merely IP addresses, or they may have additional multiplexing and demultiplexing context to aid with the packet handling in the lower layers. For instance, an IP address may need to be paired with an ESP Security Parameter Index (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 the 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_SET parameter is intended to support host multihoming (simultaneous use of a number of addresses), detailed elements of procedure for host multihoming are out of scope.

While HIP can potentially be used with transports other than the ESP transport format [RFC7402], this document largely assumes the use of ESP and leaves other transport formats 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 [I-D.ietf-hip-rfc5204-bis]. Use of the HIP rendezvous server to manage the simultaneous mobility of both hosts is specified herein, but other such scenarios are out of scope for this document. We also do not consider localized mobility management extensions (i.e., mobility management techniques that do not involve directly signaling the correspondent node); this document is concerned with end-to-end mobility. Making underlying IP mobility transparent to the transport layer has implications on the proper response of transport congestion control, path MTU selection, and Quality of Service (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 RFC 2119 [RFC2119].

LOCATOR_SET. The name of a HIP parameter containing zero or more Locator fields.

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 peer host'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

This section is an overview; more detailed specification follows this section.

3.1. Operating Environment

The Host Identity Protocol (HIP) [RFC7401] 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 the ESP transport format [RFC7402] and possibly other protocols in the future.

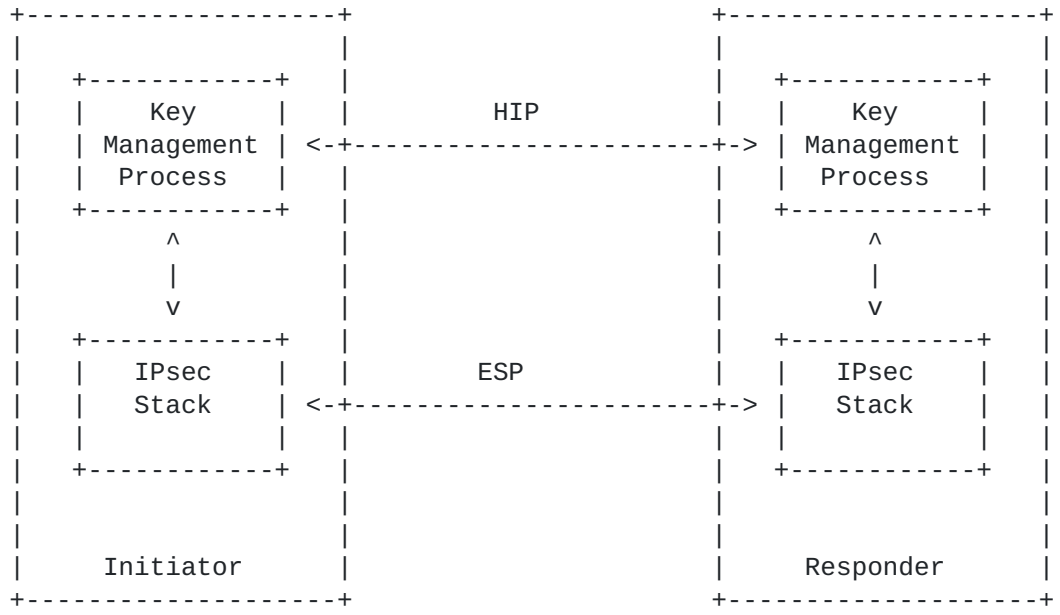

Figure 1: HIP Deployment Model

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 base protocol enable the signaling of new addressing information to the peer in HIP messages. The messages are authenticated 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_SET) parameter, the procedures for sending and processing this parameter to enable basic host mobility, and procedures for a concurrent address verification mechanism.

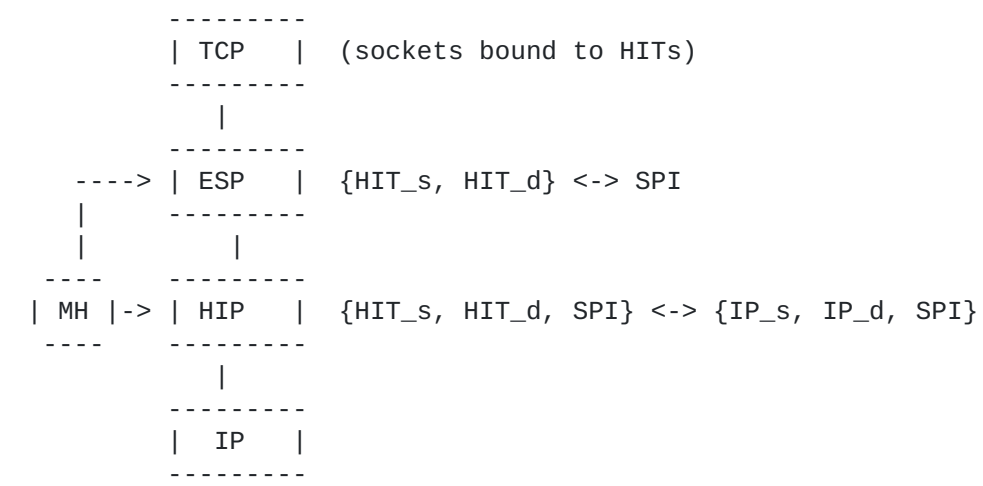

Figure 2: Architecture for HIP Host Mobility (MH)

Figure 2 depicts a layered architectural view of a HIP-enabled stack using the ESP transport format. In HIP, upper-layer protocols (including TCP and ESP in this figure) are bound to Host Identity Tags (HITs) and not IP addresses. The HIP sublayer is responsible for maintaining the binding between HITs and IP addresses. The SPI 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 [RFC7401]. The HIP base exchange establishes the HITs in use between the hosts, the SPIs to use for ESP, and the IP addresses (used in both the HIP signaling packets and ESP data packets). Note that there can only be one such set of bindings in the outbound direction for any given packet, and the only fields used 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 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. However, elements of procedure to recover from simultaneous movement of both hosts are not specified herein. In addition, internal notification of transport layer protocols of the path change (e.g. to reset congestion control variables), and

elements of procedure for traversing middleboxes including network address translators are not covered by this document.

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 violating the ESP anti-replay window. Addresses may also be affiliated with transport ports in certain tunneling scenarios. Locators may simply be traditional network addresses. The format of the locator fields in the LOCATOR_SET parameter is defined in Section 4.

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_SET parameter. This UPDATE packet is acknowledged by the peer. For reliability in the presence of packet loss, the UPDATE packet is retransmitted as defined in the HIP protocol specification [RFC7401]. 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 [RFC7402], 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 [RFC7401] and ESP extension [RFC7402].

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 send packets to these new addresses 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.

3.2. Protocol Overview

In this section, we briefly introduce a number of usage scenarios for HIP host mobility. These scenarios assume that HIP is being used with the ESP transform [RFC7402], 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 [RFC7401]. 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. More specification details are found later in Section 4 and Section 5.

The scenarios below assume that the two hosts have completed a single HIP base exchange with each other. Both of the hosts therefore have one incoming and one outgoing SA. Further, each SA uses the same pair of IP addresses, which are the ones used in the base exchange.

The readdressing protocol is an asymmetric protocol where a mobile 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 majority of the packets on which the LOCATOR_SET parameters are expected to be carried are UPDATE packets.

The scenarios below at times describe addresses as being in either an ACTIVE, UNVERIFIED, or DEPRECATED state. From the perspective of a host, newly-learned addresses of the peer must be verified before put into active service, and addresses removed by the peer are put into a deprecated state. Under limited conditions described below (Section 5.6), an UNVERIFIED address may be used. The addressing states are defined more formally in Section 5.1.

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_SET parameter.

3.2.1. Mobility with a 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, one IP address in use within the HIP session, 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.

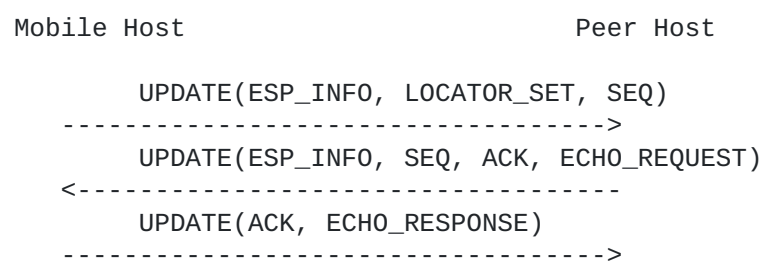

Figure 3: Readdress without Rekeying, but with Address Check

The steps of the packet processing are as follows:

1. The mobile host may be disconnected from the peer host for a brief period of time while it switches from one IP address to another; this case is sometimes referred to in the literature as a "break-before-make" case. The host may also obtain its new IP address before loosing the old one ("make-before-break" case). In either case, upon obtaining a new IP address, the mobile host sends a LOCATOR_SET parameter to the peer host in an UPDATE message. The UPDATE message also contains an ESP_INFO parameter containing the values of the old and new SPIs for a security association. In this case, the OLD SPI and NEW SPI parameters both are set to the value of the preexisting 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_SET 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 [RFC7401].

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 preexisting 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 the address 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.

3.2.2. Mobility with a Single SA Pair (Mobile-Initiated Rekey)

The mobile host may decide to rekey the SAs at the same time that it notifies 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 a 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.

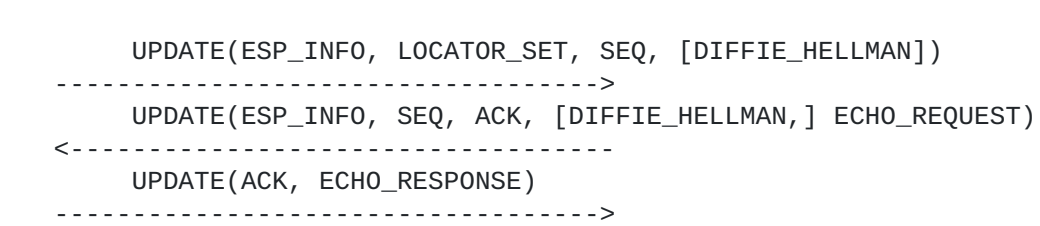

Figure 4: Readdress with Mobile-Initiated Rekey

3.2.3. Mobility messaging through rendezvous server

Section 6.11 of [RFC7401] specifies procedures for sending HIP UPDATE packets. The UPDATE packets are protected by a timer subject to exponential backoff and resent UPDATE_RETRY_MAX times. It may be, however, that the peer is itself in the process of moving when the local host is trying to update the IP address bindings of the HIP association. This is sometimes called the "double-jump" mobility problem; each host's UPDATE packets are simultaneously sent to a stale address of the peer, and the hosts are no longer reachable from one another.

The HIP Rendezvous Extension [I-D.ietf-hip-rfc5204-bis] specifies a rendezvous service that permits the I1 packet from the base exchange to be relayed from a stable or well-known public IP address location to the current IP address of the host. It is possible to support double-jump mobility with this rendezvous service if the following extensions to the specifications of [I-D.ietf-hip-rfc5204-bis] and [RFC7401] are followed.

1. The mobile host sending an UPDATE to the peer, and not receiving an ACK, MAY resend the UPDATE to a rendezvous server (RVS) of the peer, if such a server is known. The host may try the RVS of the peer up to UPDATE_RETRY_MAX times as specified in [RFC7401]. The host may try to use the peer's RVS before it has tried UPDATE_RETRY_MAX times to the last working address (i.e. the RVS may be tried in parallel with retries to the last working address).

2. A rendezvous server supporting the UPDATE forwarding extensions specified herein MUST modify the UPDATE in the same manner as it modifies the I1 packet before forwarding. Specifically, it MUST rewrite the IP header source and destination addresses, recompute the IP header checksum, and include the FROM and RVS_HMAC parameters.

3. A host receiving an UPDATE packet MUST be prepared to process the FROM and RVS_HMAC parameters, and MUST include a VIA_RVS parameter in the UPDATE reply that contains the ACK of the UPDATE SEQ.

4. This scenario requires that hosts using rendezvous servers also take steps to update their current address bindings with their rendezvous server upon a mobility event. [I-D.ietf-hip-rfc5204-bis] does not specify how to update the rendezvous server with a client host's new address. [I-D.ietf-hip-rfc5203-bis] Section 3.2 describes how a host may send a REG_REQUEST in either an I2 packet (if there is no active association) or an UPDATE packet (if such association exists). The procedures described in [I-D.ietf-hip-rfc5203-bis] for sending a REG_REQUEST and REG_RESPONSE to the rendezvous server apply also to this mobility scenario.

3.2.4. Network Renumbering

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

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_SET, 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 [RFC4225]. Therefore, 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 the exchange of a nonce, or the generation of a new SPI and observation of data arriving on the new SPI.

3.3.2. Credit-Based Authorization

Credit-Based Authorization (CBA) allows a host to securely use a new locator even though the peer's reachability at the address embedded in the 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 for the purpose of its attack because bandwidth is an ample resource for many victims.

2. An attacker can often cause unamplified flooding by sending packets to its victim, either by directly addressing the victim in the packets, or by guiding the packets along a specific path by means of an IPv6 Routing header, if Routing headers are not filtered by firewalls.

3. Consequently, the additional effort required to set up a redirection-based flooding attack (without CBA and return routability checks) 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 amplifications. 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, for example, pure direct flooding, where the attacker itself sends bogus packets to the victim.

Figure 5 illustrates Credit-Based Authorization: Host B measures the amount of data recently received from peer A and, when A readdresses, sends packets to A's new, unverified address as long as the sum of the packet 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, multihoming, or any other reason. Not shown in Figure 5 are the results of credit aging (Section 5.6.2), a mechanism used to dampen possible time-shifting attacks.

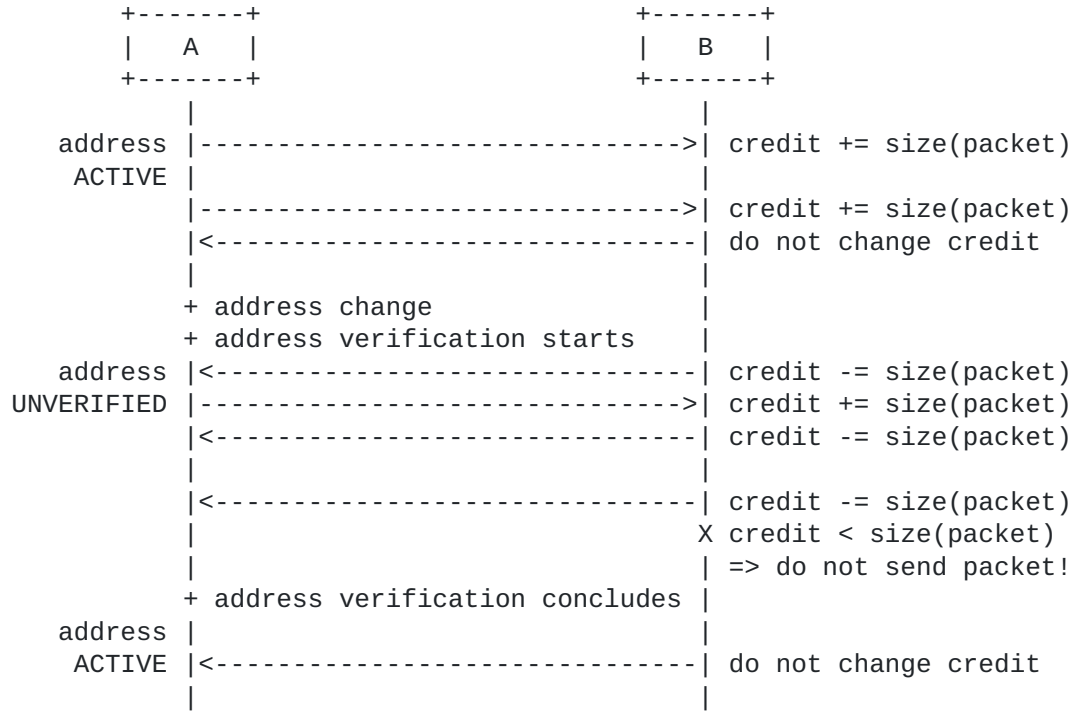

Figure 5: Readdressing Scenario

3.3.3. Preferred Locator

When a host has multiple locators, the peer host must decide 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).

4. LOCATOR_SET Parameter Format

The LOCATOR_SET parameter is a critical parameter as defined by [RFC7401]. It consists of the standard HIP parameter Type and Length fields, plus zero or more Locator sub-parameters. Each Locator sub-parameter contains a Traffic Type, Locator Type, Locator Length, Preferred locator bit, Locator Lifetime, and a Locator encoding. A LOCATOR_SET containing zero Locator fields is permitted but has the effect of deprecating all addresses.

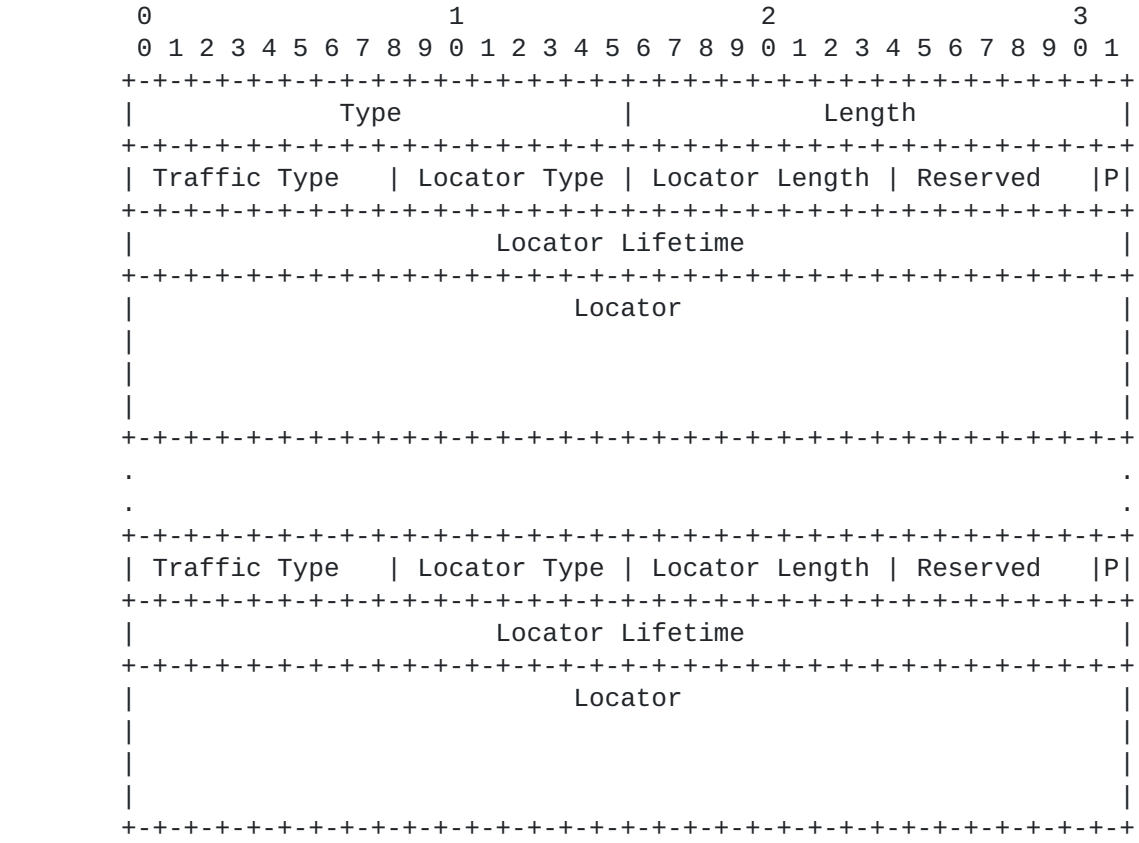

Figure 6: LOCATOR_SET Parameter Format

Type: 193

Length: Length in octets, excluding Type and Length fields, and excluding padding.

Traffic Type: Defines whether the locator pertains to HIP signaling, user data, or both.

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 octets 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 octets 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 is, when a "P" bit is set for Traffic Type "2", for example, it means that the locator is preferred for data packets. If there is a conflict (for example, if the "P" bit is set for an address of Type "0" and a different address of Type "2"), the more specific Traffic Type rule applies (in this case, "2"). 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 destination 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 [RFC4291] (128 bits long). This locator type is defined primarily for non-ESP-based usage.

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). This IP address is defined primarily for ESP-based usage.

4.3. UPDATE Packet with Included LOCATOR_SET

A number of combinations of parameters in an UPDATE packet are possible (e.g., see Section 3.2). In this document, procedures are defined only for the case in which one LOCATOR_SET and one ESP_INFO parameter is used in any HIP packet. Furthermore, the LOCATOR_SET SHOULD list all of the locators that are active on the HIP association (including those on SAs not covered by the ESP_INFO parameter). Any UPDATE packet that includes a LOCATOR_SET parameter SHOULD include both an HMAC and a HIP_SIGNATURE parameter. The UPDATE MAY also include a HOST_ID parameter (which may be useful for middleboxes inspecting the HIP messages for the first time). If the UPDATE includes the HOST_ID parameter, the receiving host MUST verify that the HOST_ID corresponds to the HOST_ID that was used to establish the HIP association, and the HIP_SIGNATURE must verify with the public key assodiated with this HOST_ID parameter. The relationship between the announced Locators and any ESP_INFO parameters present in the packet is defined in Section 5.2. This draft does not support any elements of procedure for sending more than one LOCATOR_SET or ESP_INFO parameter in a single UPDATE.

5. Processing Rules

This section describes rules for sending and receiving the LOCATOR_SET parameter, testing address reachability, and using Credit-Based Authorization (CBA) on UNVERIFIED locators.

5.1. Locator Data Structure and Status

In a typical implementation, each locator announced in a LOCATOR_SET parameter is represented by a piece of state that contains the following data:

o the actual bit pattern representing the locator,

- the lifetime (seconds),
- the status (UNVERIFIED, ACTIVE, DEPRECATED),
- the Traffic Type scope of the locator, and
- whether the locator is preferred for any particular scope.

The status is used to track the reachability of the address embedded within the LOCATOR_SET 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 the locator is UNVERIFIED.

ACTIVE to DEPRECATED The locator lifetime expires while the locator 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.

Note that the state of whether or not a locator is preferred is not necessarily the same as the value of the Preferred bit in the Locator sub-parameter received from the peer. Peers may recommend certain locators to be preferred, but the decision on whether to actually use a locator as a preferred locator is a local decision, possibly influenced by local policy.

In addition to state maintained about status and remaining lifetime for each locator learned from the peer, an implementation would typically maintain similar state about its own locators that have been offered to the peer.

Finally, the locators used to establish the HIP association are by default assumed to be the initial preferred locators in ACTIVE state, with an unbounded lifetime.

5.2. Sending LOCATOR_SETs

The decision of when to send LOCATOR_SETs is basically a local policy issue. However, it is RECOMMENDED that a host send a LOCATOR_SET 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 LOCATOR_SETs that force the peer to change the preferred address SHOULD be avoided.

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 the HIP control plane from data plane traffic). Other mobility cases are possible but are left for further study.

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_SET parameter. The ESP_INFO contains the current value of the SPI in both the OLD SPI and NEW SPI fields. The LOCATOR_SET 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. This packet is retransmitted as defined in the HIP protocol specification [RFC7401]. The UPDATE should be sent to the peer's preferred IP address with an IP source address corresponding to the address in the LOCATOR_SET 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_SET 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_SET 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 in which no rekeying is initiated.

5.3. Handling Received LOCATOR_SETs

A host SHOULD be prepared to receive a single LOCATOR_SET parameter in a HIP UPDATE packet. Reception of multiple LOCATOR_SET parameters in a single packet, or in HIP packets other than UPDATE, is outside of the scope of this specification.

This document describes sending both ESP_INFO and LOCATOR_SET parameters in an UPDATE. The ESP_INFO parameter is included when 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_SET parameter contains a complete listing of the locators that the host wishes to make or keep active for the HIP association.

In general, the processing of a LOCATOR_SET depends upon the packet type in which it is included. Here, we describe only the case in which ESP_INFO is present and a single LOCATOR_SET 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.

The processing of ESP_INFO and LOCATOR_SET parameters is intended to be modular and support future generalization to the inclusion of multiple ESP_INFO and/or multiple LOCATOR_SET parameters. A host SHOULD first process the ESP_INFO before the LOCATOR_SET, since the ESP_INFO may contain a new SPI value mapped to an existing SPI, while a Type "1" locator will only contain a reference to the new SPI.

When a host receives a validated HIP UPDATE with a LOCATOR_SET and ESP_INFO parameter, it processes the ESP_INFO as follows. The ESP_INFO parameter indicates whether an SA is being rekeyed, created, deprecated, or just identified for the benefit of middleboxes. The host examines the OLD SPI and NEW SPI values in the ESP_INFO parameter:

1. (no rekeying) If the OLD SPI is equal to the NEW SPI and both correspond to an existing SPI, the ESP_INFO is gratuitous (provided for middleboxes) and no rekeying is necessary.

2. (rekeying) 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 by creating a new outbound SA with an SPI corresponding to the NEW SPI, with no addresses bound to this SPI. Note that locators in the LOCATOR_SET parameter will reference this new SPI instead of the old SPI.

3. (new SA) 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 SA and respond with an UPDATE ACK.

4. (deprecating the SA) If the OLD SPI indicates an existing SPI and the NEW SPI is zero, the SA is being deprecated and all locators uniquely bound to the SPI are put into the DEPRECATED state.

If none of the above cases apply, a protocol error has occurred and the processing of the UPDATE is stopped.

Next, the locators in the LOCATOR_SET parameter are processed. For each locator listed in the LOCATOR_SET 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.

The below assumes that all locators are of Type "1" with a Traffic Type of "0"; other cases are for further study.

For each Type "1" address listed in the LOCATOR_SET parameter, the host checks whether 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 corresponding to the SPI that were NOT listed in the LOCATOR_SET parameter as DEPRECATED.

As a result, at the end of processing, the addresses listed in the LOCATOR_SET parameter have either a state of UNVERIFIED or ACTIVE, and any old addresses on the old SA not listed in the LOCATOR_SET parameter have a state of DEPRECATED.

Once the host has processed the locators, if the LOCATOR_SET 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 5.5.

If a host receives a locator with an unsupported Locator Type, and when such a locator is also declared to be the Preferred locator for the peer, the host SHOULD send a NOTIFY error with a Notify Message Type of LOCATOR_TYPE_UNSUPPORTED, with the Notification Data field containing the locator(s) that the receiver failed to process. Otherwise, a host MAY send a NOTIFY error if a (non-preferred) locator with an unsupported Locator Type is received in a LOCATOR_SET parameter.

A host MAY add the source IP address of a received HIP packet as a candidate locator for the peer even if it is not listed in the peer's LOCATOR_SET, but it SHOULD prefer locators explicitly listed in the LOCATOR_SET.

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, when the host is changing its SPI and 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. However, if the host is not changing its SPI, 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.

In some cases, it MAY be sufficient to use the arrival of data on a newly advertised SA as implicit address reachability verification as depicted in Figure 7, instead of waiting for the confirmation via a HIP packet. 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.

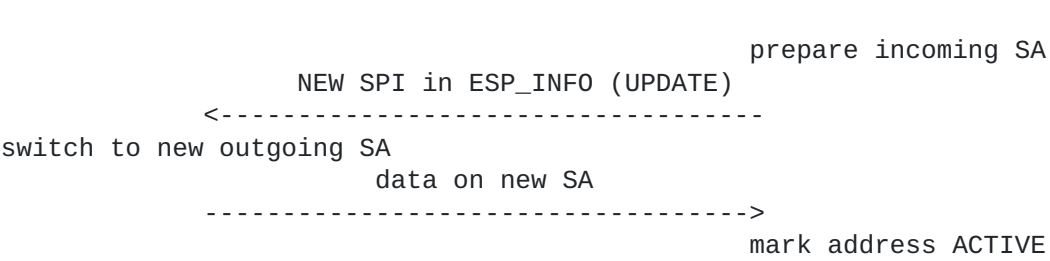

Figure 7: Address Activation Via Use of a New SA

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 5.6. Once address verification succeeds, the status of the new Preferred locator changes to ACTIVE.

5.5. 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_SET 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 that deprecate the Preferred locator arrive, 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, the address verification procedure described above will apply.

5.6. Credit-Based Authorization

To prevent redirection-based flooding attacks, the use of a Credit-Based Authorization (CBA) approach is mandatory when a host sends data to an UNVERIFIED locator. The following algorithm meets the security considerations for prevention of amplification and time-shifting attacks. Other forms of credit aging, and other values for the CreditAgingFactor and CreditAgingInterval parameters in

particular, are for further study, and so are the advanced CBA techniques specified in [CBA-MIPv6].

5.6.1. 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, and when 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 8 depicts the actions taken by the host when a packet is received. Figure 9 shows the decision chain in the event a packet is sent.

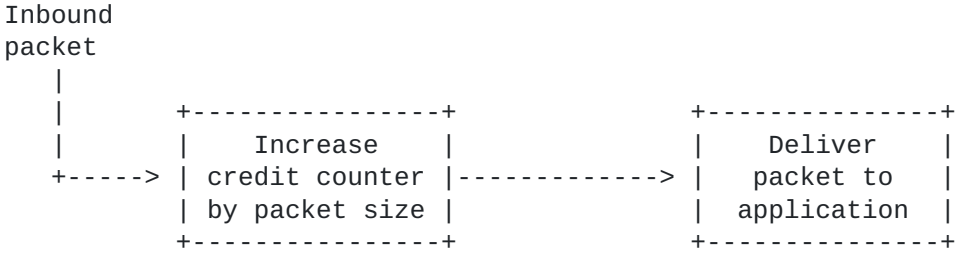

Figure 8: Receiving Packets with Credit-Based Authorization

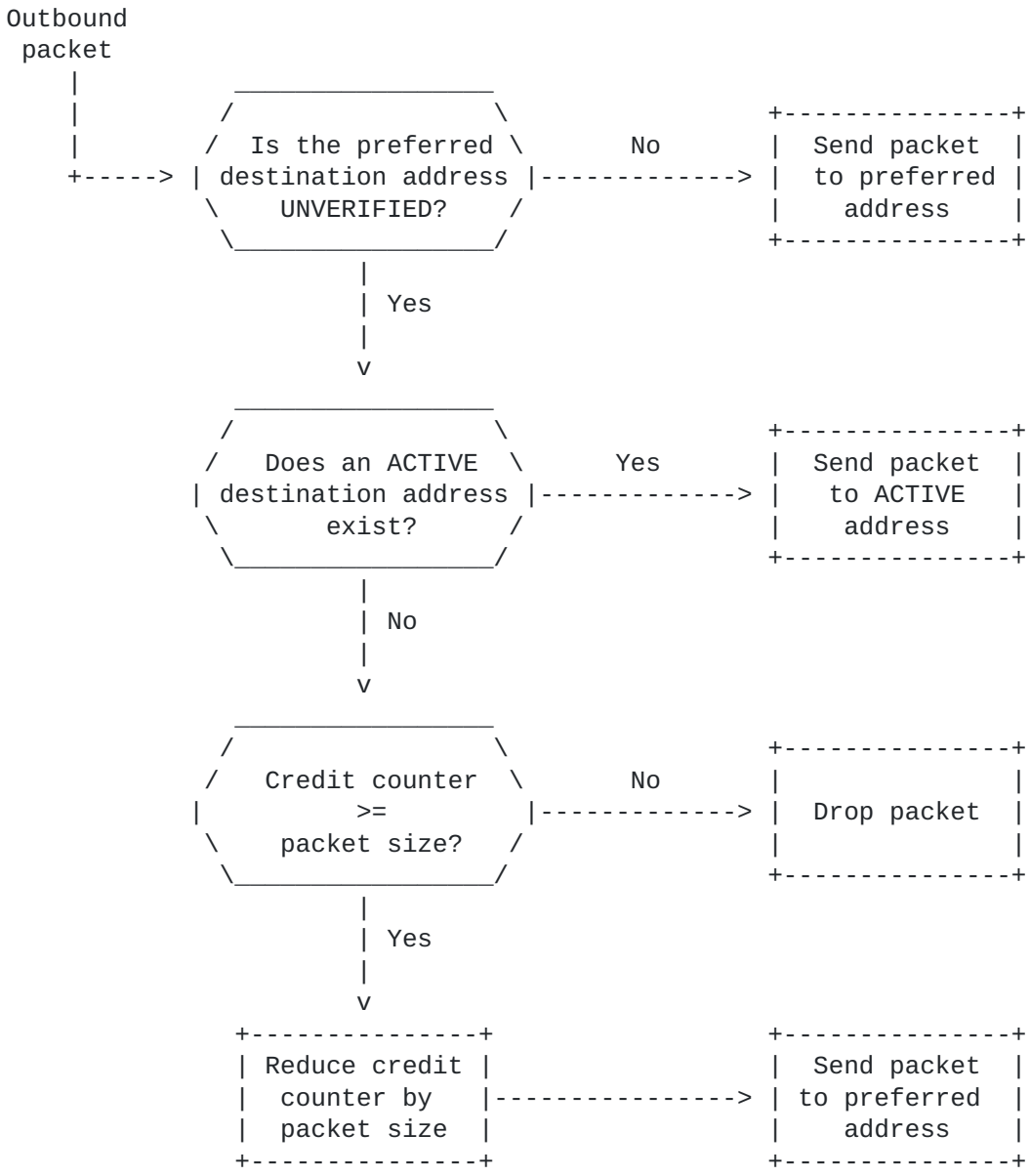

Figure 9: Sending Packets with Credit-Based Authorization

5.6.2. 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 (a fractional value 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:

CreditAgingFactor	7/8
CreditAgingInterval	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.

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 [RFC7401]). 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 3: 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. Security considerations for Credit-Based Authorization are discussed in [SIMPLE-CBA].

6.1. Impersonation Attacks

An attacker wishing to impersonate another host will try to mislead its victim into directly communicating with them, or carry out a man-in-the-middle (MitM) 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 the redirection feature to impersonate a victim's peer from any arbitrary location. The authentication and authorization mechanisms of the HIP base exchange [RFC7401] 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. However, once the opportunistic base exchange has taken place, even a MitM cannot steal the HIP connection anymore 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 lowers a HIP redirection-based flooding attack to the level of a direct flooding attack in which the attacker itself sends the flooding traffic to the victim.

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 [RFC7401]). 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 document [RFC7401].

A central server that has to deal with a large number of mobile clients may consider increasing the SA lifetimes to try to slow down the rate of rekeying UPDATEs or increasing the cookie difficulty to slow down the rate of attack-oriented connections.

6.3. Mixed Deployment Environment

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

1. A HIP host redirects its connection onto a non-HIP host. The non-HIP host will drop the reachability packet, so this is not a threat unless the HIP host is a MitM that could somehow respond successfully to the reachability check.

2. A non-HIP host attempts to redirect their connection onto a HIP host. This falls into IPv4 and IPv6 security concerns, which are outside the scope of this document.

3. A non-HIP host attempts to steal a HIP host's session (assume that Secure Neighbor Discovery is not active for the following). The non-HIP host contacts the service that a HIP host has a connection with and then attempts to change its IP address to steal the HIP host's connection. What will happen in this case is implementation dependent but such a request should fail by being ignored or dropped. Even if the attack were successful, the HIP host could reclaim its connection via HIP.

4. A HIP host attempts to steal a non-HIP host's session. A HIP host could spoof the non-HIP host's IP address during the base exchange or set the non-HIP host's IP address as its preferred address via an UPDATE. Other possibilities exist, but a simple solution is to prevent the use of HIP address check information to influence non-HIP sessions.

7. IANA Considerations

The following changes to the "Host Identity Protocol (HIP) Parameters" registries are requested.

The existing Parameter Type of 'LOCATOR' (value 193) should be renamed to 'LOCATOR_SET' and the reference should be updated from RFC5206 to this specification.

The existing Notify Message Type of 'LOCATOR_TYPE_UNSUPPORTED' (value 46) should have its reference updated from RFC5206 to this specification.

8. Authors and Acknowledgments

Pekka Nikander and Jari Arkko originated this document, and Christian Vogt and Thomas Henderson (editor) later joined as co-authors. Greg Perkins contributed the initial draft of the security section. Petri Jokela was a co-author of the initial individual submission.

The authors thank Jeff Ahrenholz, Baris Boyvat, Rene Hummen, Miika Komu, Mika Kousa, Jan Melen, and Samu Varjonen for improvements to the document.

9. References

9.1. Normative references

[I-D.ietf-hip-rfc5203-bis]
Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Registration Extension", draft-ietf-hip-rfc5203-bis-09 (work in progress), June 2015.

[I-D.ietf-hip-rfc5204-bis]
Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", draft-ietf-hip-rfc5204-bis-07 (work in progress), December 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <http://www.rfc-editor.org/info/rfc4291>.

[RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <http://www.rfc-editor.org/info/rfc7401>.

[RFC7402] Jokela, P., Moskowitz, R., and J. Melen, "Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP)", RFC 7402, DOI 10.17487/RFC7402, April 2015, <http://www.rfc-editor.org/info/rfc7402>.

9.2. Informative references

[CBA-MIPv6] Vogt, C. and J. Arkko, "Credit-Based Authorization for Mobile IPv6 Early Binding Updates", February 2005.

[RFC4225] Nikander, P., Arkko, J., Aura, T., Montenegro, G., and E. Nordmark, "Mobile IP Version 6 Route Optimization Security Design Background", RFC 4225, DOI 10.17487/RFC4225, December 2005, <http://www.rfc-editor.org/info/rfc4225>.

[SIMPLE-CBA] Vogt, C. and J. Arkko, "Credit-Based Authorization for Concurrent Reachability Verification", February 2006.

Appendix A. Document Revision History

To be removed upon publication

Revision	Comments
draft-00	Initial version from RFC5206 xml (unchanged).
draft-01	Remove multihoming-specific text; no other changes.
draft-02	Update references to point to -bis drafts; no other changes.
draft-03	issue 4: add make before break use case issue 6: peer locator exposure policies issue 10: rename LOCATOR to LOCATOR_SET issue 14: use of UPDATE packet's IP address
draft-04	Document refresh; no other changes.
draft-05	Document refresh; no other changes.
draft-06	Document refresh; no other changes.
draft-07	Document refresh; IANA considerations updated.
draft-08	Remove sending LOCATOR_SET in R1, I2, and NOTIFY (multihoming) State that only one LOCATOR_SET parameter may be sent in an UPDATE packet (according to this draft) (multihoming) Remove text about cross-family handovers (multihoming)
draft-09	Add specification text regarding double-jump mobility procedures.
draft-10	issue 21: clarified that HI MAY be included in UPDATE for benefit of middleboxes changed one informative reference from RFC 4423-bis to RFC 7401

		removed discussion about possible multiple LOCATOR_SET and ESP_INFO parameters in an UPDATE (per previous mailing list discussion)
		removed discussion about handling LOCATOR_SET parameters in packets other than UPDATE (per previous mailing list discussion)

Authors' Addresses

Thomas R. Henderson (editor)
University of Washington
Campus Box 352500
Seattle, WA
USA

EMail: tomhend@u.washington.edu

Christian Vogt
Ericsson Research NomadicLab
Hirsalantie 11
JORVAS FIN-02420
FINLAND

EMail: christian.vogt@ericsson.com

Jari Arkko
Ericsson Research NomadicLab
JORVAS FIN-02420
FINLAND

Phone: +358 40 5079256
EMail: jari.arkko@ericsson.com