Network Working Group
Internet-Draft
Expires: June 28, 2004

P. Nikander
J. Arkko
Ericsson Research Nomadic Lab
December 29, 2003

End-Host Mobility and Multi-Homing with Host Identity Protocol
draft-nikander-hip-mm-01

Status of this Memo

Copyright Notice

Abstract

This document specifies basic end-host mobility and multi-homing mechanisms for the Host Identity Protocol.

Table of Contents

1. Introduction

This document specifies an extension to the Host Identity Protocol [3] (HIP). The extension provides a simple and efficient means for hosts to keep their communications on-going while having multiple IP addresses, either at the same time or one after another. That is, the extension provides basic support for multi-homing, mobility, and simultaneous multi-homing and mobility. Additionally, the extension allows communications to continue even when multi-homing or mobility causes a change of the IP version that is available in the network; that is, if one of the communicating hosts has both IPv4 and IPv6 connectivity, either directly or through a proxy,the other host can alternate between IPv4 and IPv6 without any effects on the upper layer protocols.

This document does not specify any rendezvous or proxy services. Those are subject to other specifications . Hence, this document alone does not necessarily allow two mobile hosts to communicate, unless they have other means for initial rendezvous and solving the simultaneous movement problem.

The Host Identity Protocol [3] (HIP) defines a mechanism that decouples the transport layer (TCP, UDP, etc) from the internetworking layer (IPv4 and IPv6), and introduces a new Host Identity namespace. When a host uses HIP, the transport layer sockets and IPsec Security Associations are not bound to IP addresses but to Host Identifiers. This document specifies how the mapping from Host Identifiers to IP addresses can be extended from a static one-to-one mapping into a dynamic one-to-many mapping, thereby enabling end-host mobility and multi-homing.

In practice, the HIP base exchange [3]creates a pair of IPsec Security Associations (SA) between a pair of HIP enabled hosts. These SAs are not bound to IP addresses, but to the Host Identifiers (public keys) used to create them. However, the hosts must also know

at least one IP address where their peers are reachable. Initially these IP addresses are the ones used during the HIP base exchange.

Since the SAs are not bound to IP addresses, the host are able to receive packets that 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. However, the peers are not able to replys before they can reliably and securely update the set of addresses that they associate with the sending host.

This document specifies a mechanism that allows a HIP host to update the set of addresses that its peers associate with it. The address update is implemented with new HIP parameter types. Due to the danger

of flooding attacks (see [4]), the peers must always check the reachability of the host at a new address before it can use the new addresses. The reachability check is implemented by the challenger creating a new SPI, announcing the new SPI, and waiting for traffic on the new SPI. In addition to initiating the reachability check, announcing the new SPI also acts as an acknowledgement for a preceding address change.

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, end-host and site multi-homing with legacy hosts, and NAT traversal. In these situations there is a need for some helper functionality in the network. This document does not address those needs.

2. Conventions used in this document

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 [1].

3. Terminology

Preferred address An address that a host prefers to receive data.

New preferred address A new preferred address sent by a host to its peers. The reachability of the new preferred address often needs to be verified before it can be taken into use. Consequently, there may simultaneously be an active preferred address, being used, and a new preferred address, whose reachability is being verified.

Interface A logical concept used to group IP addresses together. If a host announces multiple interface, each interface will be associated with a different incoming Security Association.

4. Usage scenarios

In this section we briefly introduce a number of usage scenarios where the HIP mobility and multi-homing facility is useful. To understand these usage scenarios, the reader should be at least minimally familiar with the HIP protocol specification [3]. 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.

4.1 End-host mobility

A mobile IP host must change its IP address according to connectivity. 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.

Although HIP enables ESP and the upper layer to be independent of the internetworking layer, there still needs to be a mapping of the pseudo-IP addresses used in the upper stack (LSI and HIT) to a real IP address. Thus, from the functional point of view, it is sufficient that the peer host learn the new IP address. The upper layer protocols need to know about the address and connectivity change only for QoS and other similar purposes. In most cases, they do not need to know.

4.2 End-host multi-homing

A host may have multiple interfaces, and it is desirable that it can stay reachable through all or any subset of the currently available interfaces. It is expected that the set of available interfaces may change dynamically, and that there may be policies associated with the usage of the different interfaces. For instance, the host may have a fast but low range wireless interface and a slow high range interface. The host may generally prefer to use the fast interface, but it may not be always available.

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.

4.3 Site multi-homing

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 host 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.

Note that a single interface may experience site multi-homing while the host itself may have multiple interfaces.

4.4 Combined mobility and multi-homing

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).

4.5 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 renumber is similar to mobility.

5. Overview of HIP basic mobility and multi-homing functionality

HIP mobility and multi-homing is fundamentally based on the HIP architecture [4], where the transport and internetworking layers are insulated from each other with the new host identity protocol layer. In the HIP architecture, the transport layer sockets are bound to the Host Identifiers (through HIT or LSI in the case of legacy APIs), and the Host Identifiers are translated to the actual IP address.

In the HIP base protocol specification [3], it is defined how two hosts exchange their Host Identifiers and establish a pair of ESP Security Associations (SA). The ESP SAs are then used to carry the actual payload data between the two hosts, by wrapping TCP, UDP, and other upper layer packets into transport mode ESP payloads. The IP header, carrying the ESP header, uses the actual IP addresses in the network.

The base specification does not contain any mechanisms for changing the IP addresses that were used during the base HIP exchange. Hence, in order to remain connected any systems that implement only the space specification and nothing else must retain the ability to receive packets at their primary IP address; that is, those systems cannot change the IP address they are using without causing loss of connectivity.

5.1 Informing the peer about multiple or changed address(es)

This document specifies a new HIP protocol parameter, the REA parameter (see Section 7.1), that allows the hosts to exchange information about their IP address(es), and any changes in their address(es). The logical structure created with REA parameters has three levels: hosts, interfaces, and addresses. This is illustrated in Figure 1.

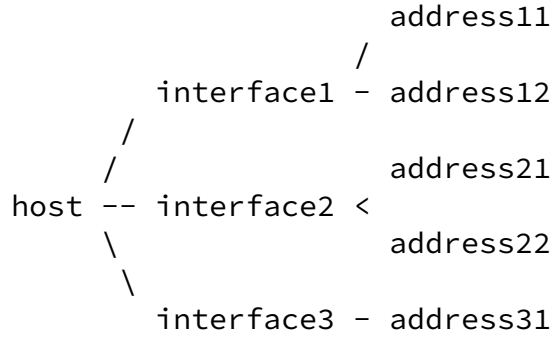


Figure 1

In this document, the interfaces are considered to be logical objects. A host may claim to have any number of interfaces. The purpose of the interfaces 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 interface2, it is likely that both address21 and address22 will simultaneously become obsolete. Note, however, that especially in the case of site multi-homing one of the addresses may become unreachablewhile 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.

All addresses on a single interface share an SA. Each interface has its own SA. In the usual case, the latencies experienced on distinct interfaces may be considerably different. Hence, if multiple interfaces were to share an SA, it would become fairly likely that some of the packets would be discarded due to appearing to have been received outside of the ESP reordering window.

While it would be possible to share an SA between multiple interfaces, there would be no benefit from it. As the interfaces are logical objects, and as the hosts are free to create new interface as demand and to move addresses between interfaces as they will, a conforming host may claim that two physical interfaces are in fact one logical one, thereby allowing the two interfaces to share an SA.

An address may appear on more than one interface. This creates no ambiguity since each interface must have a different SPI, and since the receiver will ignore the IP addresses anyway.

A single REA parameter contains data only about one interface. To signal simultaneously changes on several interfaces, it is necessary to send several REA parameters. The packet structure supports this.

If the REA parameter is send in a NES packet and the sender wants to receive an acknowledgement, it must explicitly indicate so.

Otherwise the recipient of the REA parameter may consider the REA as informational, and act only when it needs to activate a new address.

5.2 Address verification

When a HIP host receives a group of IP addresses from another HIP host in a REA, it does not necessarily know whether the other host is actually reachable at the claimed addresses. In fact, a maliciouspeer host may be intentionally giving a bogus addresses in order to cause a packet flood towards the given address [7]. Thus, before the HIP host can actually use a new address, it must first check that the peer is reachable at the new address. This is

implemented by requesting the peer to create a new outgoing SA, using a new random SPI value, and waiting for data to appear on this new SA.

5.3 Address data structure and status

In a typical implementation, each remote address is represented as a piece of state that contains the following data:

the actual bit pattern representing the IPv4 or IPv6 address,

lifetime (seconds),

status (UNVERIFIED, ACTIVE, DEPRECATED).

The status is used to track the reachability of the address:

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

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

DEPRECATED indicates that the address lifetime has expired

The following state changes are allowed:

UNVERIFIED to ACTIVE The reachability procedure completes succesfully.

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

ACTIVE to DEPRECATED The address 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 address.

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

6. Protocol overview

The readdressing protocol is designed to be piggybacked on a number of existing HIP exchanges. The main packets on which the REA parameters are expected to be carried on are New SPI (NES) packets. However, some implementations may want to experiment with sending REA parameters also on other packets, such as R1 and I2.

The protocol is an asymmetric protcool where one host, called the Mobile Host, informs another host, called the Peer host, about changes of IP addresses on one of its interfaces. The protocol consists of three steps, illustrated in Figure 2.

1. The Mobile Host sends a REA parameter to the Peer host.

2. The Peer Host initiates an address check procedure by sending a new SPI value to a new address. Any packet that contains a new SPI may be used, including NES, I2 and R2. The new SPI value MUST be random, i.e., the Mobile Host MUST NOT be able to guess it. When the Mobile Host receives the new SPI value, it creates a new outgoing SA and starts sending data on it.

3. The Peer Host waits for new data to arrive on the new SA, indicated by the SPI. Once it has succesfully received data on the SA, it marks the new address as reachable.

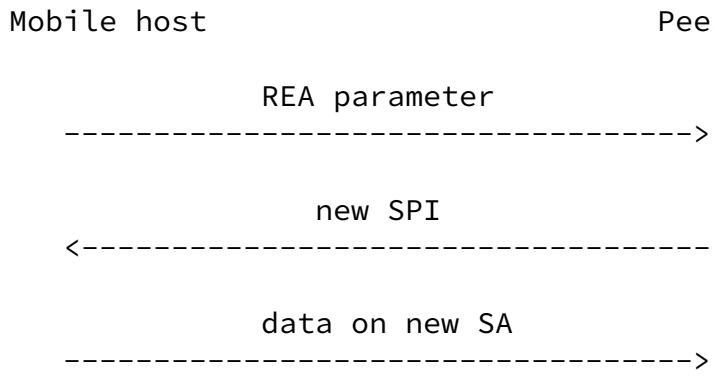


Figure 2

The idea of the address check procedure is that if the Mobile Host is able to succesfully construct the new outgoing SA, using the new SPI value, and send data on that SA, then it must have received the second message in the protocol, and therefore it must be reachable at the said address.

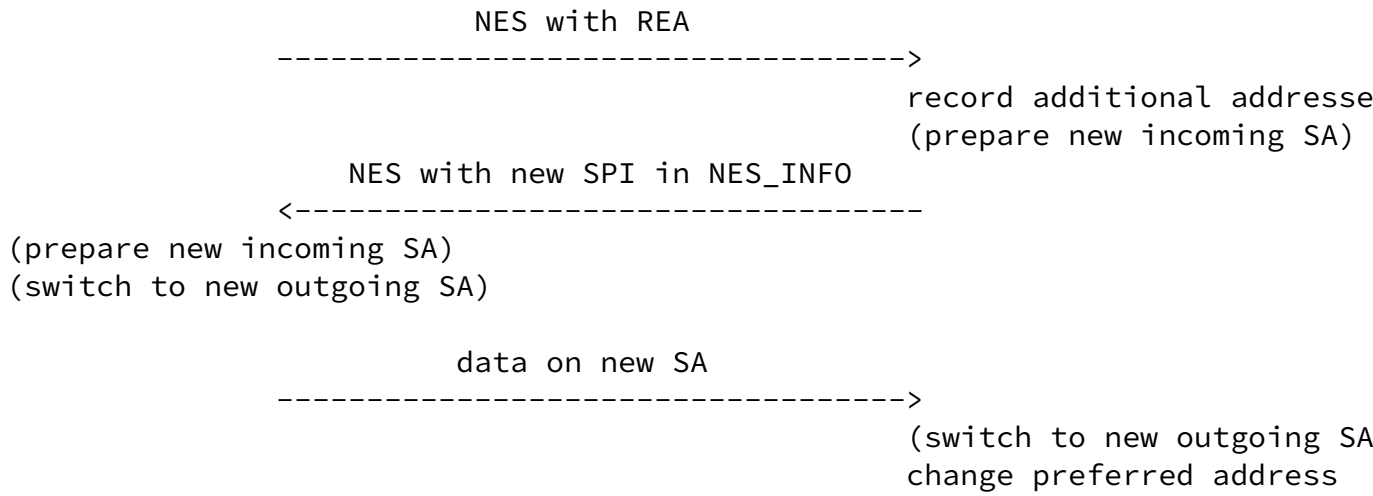
XXX: Residual threat: The Mobile Host able to anticipate the KEY index and guess the SPI value by trying out all? Not really, I

think, since it would require about 2^31 packets on the average...

6.1 Initiating the protocol in NES

The most common case is to carry the readdress protocol on NES packets. In this case, the Mobile Host initiates rekey (usually using the current Diffie-Hellman keys) and includes a REA parameter on the initial NES packet. The Peer host replies to this with a reply NES packet, sent to the new preferred address. As the Mobile Host receives the reply NES, it starts using the new outgoing SA. Finally, as the Peer Host receives traffic on the new incoming SA, it marks the new address as valid and switches over to use the new outgoing SA, associated with the new address.

Mobile host Peer host



The text in (parantheses) indicate functions that are performed anyway, as a part of NES processing, and not new to the REA processing.

Figure 3

Basically, the processing structure is equal to the current NES processing other than that the Peer host records the additional addresses form the REA parameter, sends the NES reply to the new preferred address instead of the current preferred address, and updates the preferred address as soon as it receives data on the new SA.

6.2 Initiating the protocol in R1 or I2

A Responder host MAY include one or more REA 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 REA

parameters, the Initiator SHOULD send the I2 packet to the new preferred address. 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. The I1 destination address and the new preferred address may be identical.

Initiator Responder

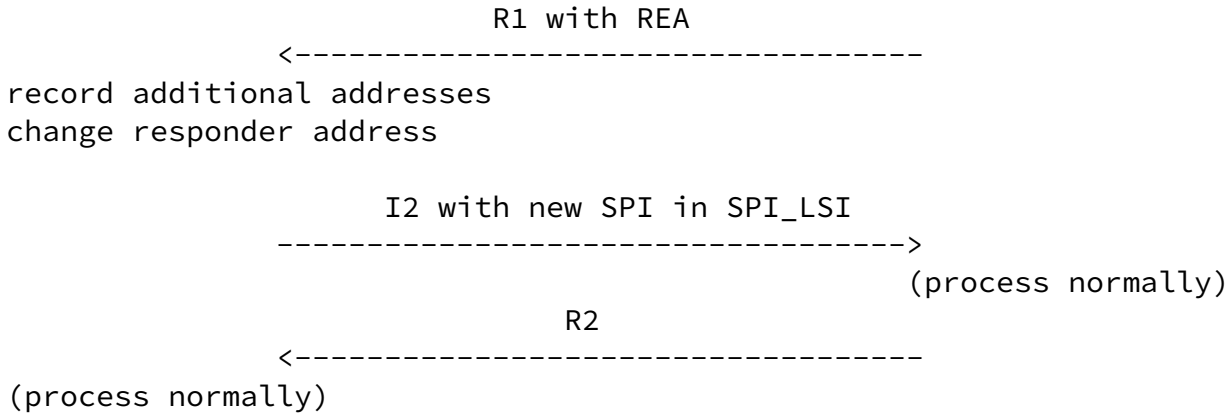


Figure 4

XXX: What about R1 source address? Most probably we want to allow it to be any address to allow an optimized rendezvous server to send an R1 instead of the actual host?

An Initiator MAY include one or more REA parameters in the I2 packet, independent on whether there was REA parameter(s) in the R1 or not. These parameters MUST be protected by the I2 signature. Even if the I2 packet contains REA parameters, the Responder MUST still send the R2 packet to the source address of the I2. The new preferred address SHOULD be identical to the I2 source address.

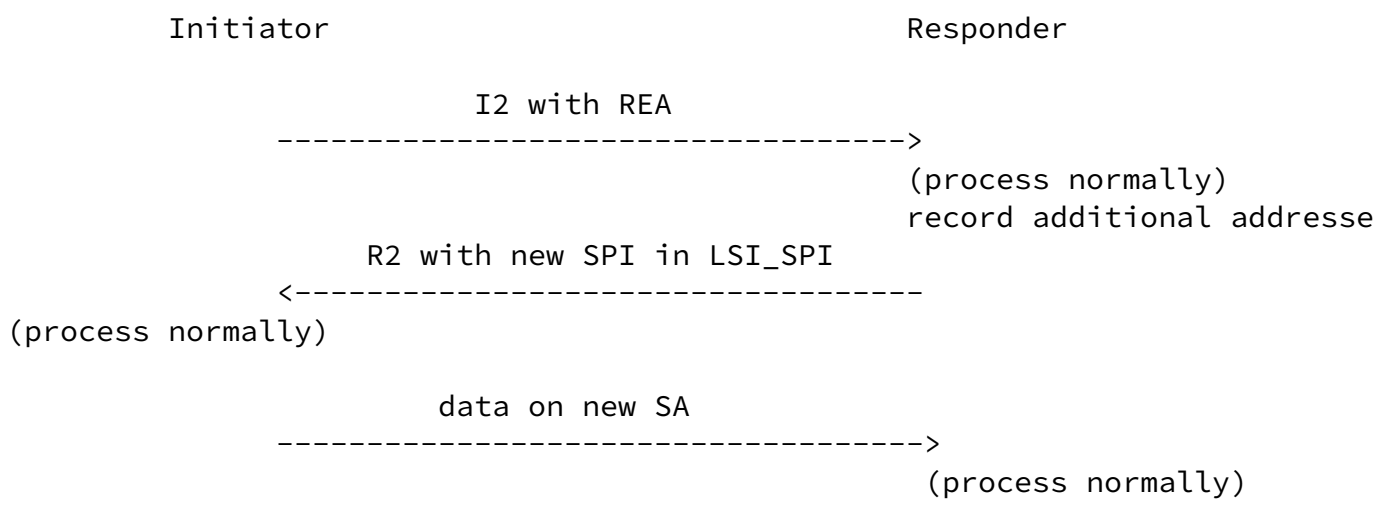


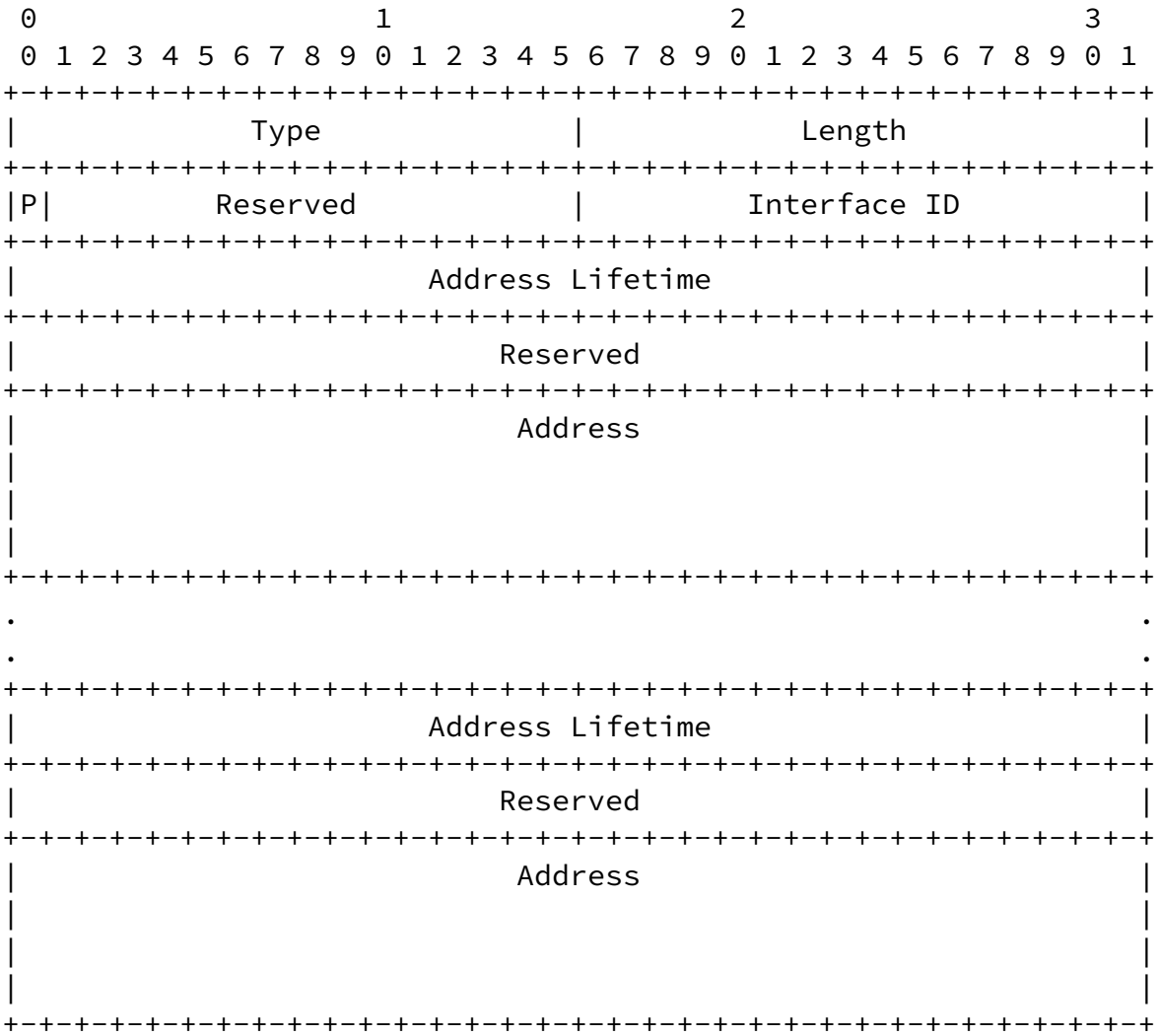
Figure 5

6.3 Explicit address check

When the Peer Host wants to use a new address that has not yet been checked, it must first run check the reachability of the address before sending any large amounts of data to the address. See Section 8.3.

7. Parameter and packet formats

7.1 REA parameter



Type 1 (first non-critical)

Length Length in octets, excluding Type and Length fields.

P Preferred address. The first address in this REA is the new preferred address.

Reserved Zero when sent, ignored when received.

Interface ID Interface ID, as defined by the sending host. The interface IDs may have any values, and need not be consequtively allocated.

Address Lifetime Address lifetime, in seconds.

Reserved Zero when sent, ignored when received.

Address An IPv6 address or an IPv4-in-IPv6 format IPv4 address [2].

The Interface ID field identifies the (logical) interface that this parameter applies to. It is implicitly qualified by the Host Identity of the sending host. The Interface ID groups a set of addresses to an interface. The sending host is free to introduce new interface IDs at will. That is, if a received REA has a new interface ID, it means that all the old addresses, assigned to the other interface IDs, are also supposed to still work, while the new addresses in the newly received REA are supposed to be associated with a new interface. On the other hand, if a received REA has an interface ID that the receiver already knows about, it would replace (all) the address(es) currently associated with the interface with the new one(s).

The Address Lifetime indicates how long the following address is expected to be valid. The lifetime is expressed in seconds. Each address 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 an destination address if an alternate (non-deprecated) is available and has sufficient scope. Since IP addresses are ignored upon reception, deprecation status does not have any affect on the receiver.

The Address field contains an IPv6 address, or an IPv4 address in the IPv4-in-IPv6 format [2]. The latter format denotes a plain IPv4 address that can be used to reach the Mobile Host.

7.2 Modified NES_INFO parameter

The NES_INFO parameter is defined in the base specification [3]. The R-bit is defined to indicate wheter a NES is a reply to another NES or not. If a NES is not a reply, the receiver must reply. If a NES is an unexpected reply, the packet is simply dropped.

This specification changes the semantics of the R-bit slightly. (It is expected that this change may be later incorporated to the base specification.) The new semantics of the R-bit are defined as follows.

R	Zero if the sender is requesting an explicit NES_INFO as a reply, one if no reply is needed.

Processing NES packets currently defines the following behaviour.

> If the system is in state E3 and the NES has the R-bit set, the packet is silently dropped.

The new behaviour is defined as follows.

> If the system is in state E3 and the NES_INFO has the R-bit set, the system initiates unidirectional rekeying, as defined in Section 8.3.

7.3 NES packet with included REA

A single REA is included in a NES packet between the NES_INFO and HMAC parameters:

```
IP ( HIP ( REA, NES_INFO, [ DIFFIE_HELLMAN, ] HMAC, HIP_SIGNATURE ) )
```

If there are multiple REA parameters to be sent in a single NES, each of them must be matched with a NES_INFO parameter:

```
IP ( HIP ( REA1, REA2, NES_INFO1, NES_INFO2, [ DH, ] ... ) )
```

7.4 R1 and I2 packets with included REA

The REA parameter is included early in the R1 and I2 packets, since middle boxes may be interested in inspecting them. If a REA is not present, a typical middle box is only interested in the SPI_LSI parameter and the signature.

```
IP ( HIP ( REA,
           BIRTHDAY_COOKIE,
           DIFFIE_HELLMAN,
           HIP_TRANSFORM,
           ESP_TRANSFORM,
           ( HOST_ID | HOST_ID_FQDN ),
           HIP_SIGNATURE_2 ) )


IP ( HIP ( SPI_LSI,
           REA,
           BIRTHDAY_COOKIE,
           DIFFIE_HELLMAN,
           HIP_TRANSFORM,
           ESP_TRANSFORM,
           ENCRYPTED { HOST_ID | HOST_ID_FQDN },
```



```
           HIP_SIGNATURE ) )
```

8. Processing rules

XXX: This section needs more work.

8.1 Sending REAs

The decision of when to send REAs is basically a local policy issue. However, it is RECOMMENDED that a host sends a REA whenever it recognizes a change of its IP addresses, and assumes that the change is going to last at least for a few seconds. Rapidly sending conflicting REAs SHOULD be avoided.

When a host decided to inform its Peers about changes in its IP addresses, it has to decide how to group the various addresses into interfaces, and whether to include any addresses on multiple interfaces. Since each interface is associated with a different Security Association, the grouping policy may be based on IPsec replay protection considerations. In the typical case, simply basing

the grouping on actual kernel level physical and logical interfaces is often the best policy. Virtual interfaces, such as IPsec tunnel interfaces or Mobile IP home addresses SHOULD NOT be announced.

Once the host has decided on the interfaces and assingment of addresses to the interfaces, it creates a REA parameter for each interface. If there are multiple interfaces and therefore multiple parameters, the parameters MUST be ordered so that the new preferred address is in the first REA parameter.

The REA parameters MAY be sent in R1, I2 and NES packets. If the host does not have any other reason to send R1, I2 or NES, it should generate a new initial NES, SHOULD NOT include any Diffie-Hellman parameter to it, and send the REA parameters in the newly generated NES packet.

If there are multiple REA parameters leading to a packet size that exceeds the MTU, the host SHOULD send multiple packets, each smaller than the MTU. In the case of R1 and I2, the additional packets should be NES packets that are send after the base exchange has been completed.

8.2 Handling received REAs

A host SHOULD be prepared to receive REA parameters in any HIP packets, excluding I1.

When a host receives a REA parameter, it first performs the following operations:

1. The host checks if the Interface ID listed is a new one. If it is a new one, it creates a new logical interface that contains no addresses.

2. For each address listed in the REA parameter, check that the address is a legal unicast or anycast address. That is, the addres 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 address listed in the REA parameter, check if the

address is already listed at the interface. If the address is listed, its lifetime is updated. If the address is status is DEPRECATED, the status is changed to UNVERIFIED. if the address is not listed, the address is added, and its status is set to UNVERIFIED.

4. Mark all addresses at the interface that were NOT listed in the REA parameter as DEPRECATED.

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

Once the host has updated the interface, if the REA parameter contains a new preferred address, the host SHOULD initiate a change of the preferred address. This usually requires that the host first verifies reachability of the address, and only then changes the address. See Section 8.4.

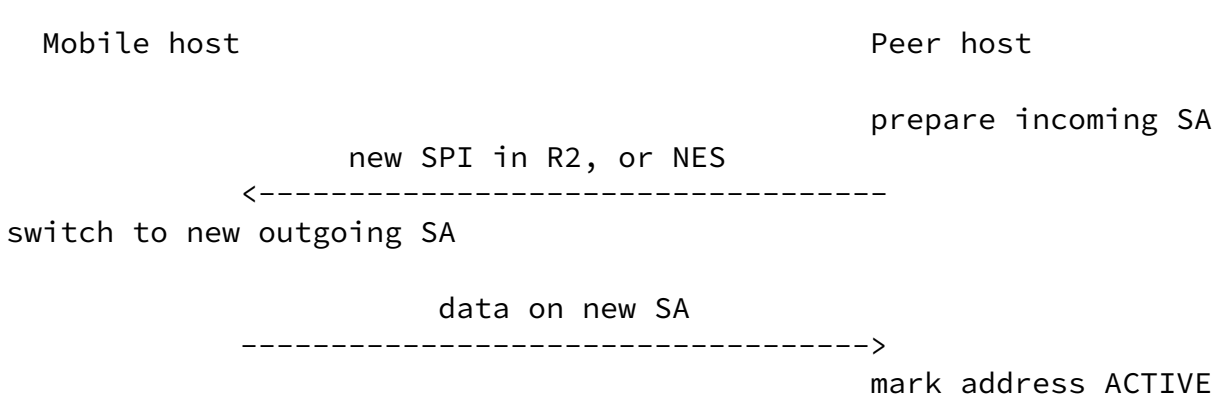
8.3 Verifying address reachability

A host MAY what to verify the reachability of any UNVERIFIED address at any time. However, the exact method of verification depends on whether the host will next send a packet that contains a new SPI value or not. That is, if the host is replying to a R1 with an I2, to an I2 with an R2, or to a initial NES with a reply NES, then the verification is piggybacked on the I2, R2, or reply NES packet. Otherwise the verification is initiated by sending an unidirectional NES packet containing REA and NES_INFO parameters.

Any of the I2, R2, NES-reply or unidirectional-NES packets cause either the creation or change of the outgoing SA in the Mobile Host. Furthermore, as part of the process to send R2, NES-reply or unidirectional-NES, the Peer Host MUST prepare a new incoming SA, using the SPI value included in R2 or NES, so that it is prepared to receive traffic on the new SA.

Note that in the case of receiving a REA on an R1 and replying with an I2, receiving the corresponding R2 is sufficient for marking the Responder's primary address active, and there is no need to wait for data to appear on the SA. On the other hand, marking the address

active as a part of receiving data on the SA is an idempotent operation, and does not cause any harm.



8.4 Changing the preferred address

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

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

1. If the new preferred address is not listed on any interface, the procedure fails.

2. If the new preferred address has DEPRECATED status and there is at least one non-depraceted address, the host selects one of the non-deprecated addresses as a new preferred address and continues.

3. If the new preferred address has ACTIVE status, the preferred address is changed and the procedure succeeds.

4. If the new preferred address has UNVERIFIED status, the host starts to verify its reachability. Once the verification has succeeded, the preferred address change is completed, unless a new change has been initiated in the mean while.

9. Policy considerations

XXX: This section needs to be written.

The host may change the status of unused ACTIVE addresses into UNVERIFIED after a locally configured period if inactivity.

10. Security Considerations

XXX: This section requires lots of more work.

(Initial text by Jari Arkko): If not controlled in some manner, messaging related to address changes would create the following types of vulnerabilities:

Revealing the contents of the (cleartext) communications

Hijacking communications and man-in-the-middle attacks

Denial of service for the involved nodes, by disabling their ability to receive the desired communications

Denial of service for third parties, by redirecting a large amount of traffic to them

Revealing the location of the nodes to other parties

In HIP, communications are bound to the public keys of the end-points and not to IP addresses. The REA message is signed with the sender's public key, and hence it becomes impossible to hijack the communications of another host through the use of the REA message. Similarly, since only the host itself can sign messages to move its traffic flows to a new IP address, denial of service attacks through REA can not cause the traffic flows to be sent to an IP address that the host did not wish to use. Finally, In HIP all communications are encrypted with ESP, so a hijack attempt would also be unable to reveal the contents of the communications.

Malicious nodes that use HIP can, however, try to cause a denial of service attack by establishing a high-volume traffic flow, such as a video stream, and then redirecting it to a victim. However, the return routability check provides some assurance that the given address is willing to accept the new traffic. Only attackers who are on the path between the peer and the new address could respond to the test.

11. IANA Considerations

12. Acknowledgments

Normative references

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

[3] Moskowitz, R., Nikander, P. and P. Jokela, "Host Identity Protocol", draft-moskowitz-hip-07 (work in progress), June 2003.

[4] Moskowitz, R., "Host Identity Protocol Architecture", draft-moskowitz-hip-arch-03 (work in progress), May 2003.

Informative references

[5] Bellovin, S., "EIDs, IPsec, and HostNAT", IETF 41th, March 1998.

[6] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", draft-iab-sec-cons-03 (work in progress), February 2003.

[7] Nikander, P., "Mobile IP version 6 Route Optimization Security Design Background", draft-nikander-mobileip-v6-ro-sec-01 (work in progress), July 2003.

Authors' Addresses

Pekka Nikander
Ericsson Research Nomadic Lab

JORVAS FIN-02420
FINLAND

Phone: +358 9 299 1
EMail: pekka.nikander@nomadiclab.com

Jari Arkko
Ericsson Research Nomadic Lab

JORVAS FIN-02420
FINLAND

Phone: +358 9 299 1
EMail: jari.arkko@nomadiclab.com

Appendix A. Changes from previous versions

A.1 From -00 to -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).

Appendix B. Implementation experiences

Intellectual Property Statement

Full Copyright Statement

Acknowledgement

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