HIP Working Group
Internet-Draft
Intended status: Experimental
Expires: January 7, 2008

M. Komu, Ed.
HIIT
S. Schuetz
M. Stiemerling
NEC
July 6, 2007

HIP Extensions for the Traversal of Network Address Translators
draft-ietf-hip-nat-traversal-02

Status of this Memo

Copyright Notice

Abstract

The Host Identity Protocol (HIP) provides a new namespace that can be used for uniquely identifying hosts in public and also in private address realms. Usually, HIP control and data traffic cannot traverse Network Address Translators (NATs), that hinders general deployment. This document specifies NAT traversal extensions for HIP. As HIP is located between network and transport layer, the

extensions also provide general-purpose NAT traversal support for all high-layer networking applications that run over HIP. The basic design concepts for these extensions have been adopted from the Interactive Connectivity Establishment (ICE) protocol to HIP. Using the specified extensions, two HIP-capable hosts are able to communicate with each other even when they are in different private address realms.

Table of Contents

1. Terminology

In general, this document borrows the terminology from [I-D.ietf-hip-base] and [RFC4423]. Additional terms are defined in the table below." These draft e.g. define "Initiator" and "Responder"

Term	Explanation
Rendezvous server	A host that forwards I1 packets to the Responder
HIP Relay	A host that forwards all HIP control packets between an Initiator and Responder
ESP Relay	A host that forwards ESP traffic between two HIP-enabled hosts
Locator	A routable IPv4 or IPv6 address
Transport locator	Transport layer port and the corresponding IPv4/v6 address
Unreflexive locator	An IPv4 or IPv6 address of a network interface of a host
Relay reflexive transport locator	A translated transport locator of a host as observed by a relay
Peer reflexive transport locator	A translated transport locator of a host as observed by its peer
Leased transport locator	Transport locator of an ESP relay

Table 1: Terminology

2. Introduction

The Host Identity Protocol (HIP) describes a new communication mechanism for Internet hosts [RFC4423]. It introduces a new

namespace and protocol layer between the network and transport layers that decouples the identifier and locator roles to support mobility and multihoming in the Internet architecture. HIP also secures application layer communications using IPsec ESP [I-D.ietf-hip-esp].

The HIP protocol [I-D.ietf-hip-base] cannot operate across legacy NAT middleboxes as described in [I-D.irtf-hiprg-nat]. This document specifies mechanisms that allow HIP to traverse through such NAT middleboxes that are neither HIP-aware nor ESP-aware, without manual configuration of the NAT middleboxes.

HIP introduces a new namespace for hosts that decouples the identity

of a host from its location [RFC4423]. The namespace consists of Host Identifiers which are public keys. The hosts create the corresponding private keys by themselves which makes identity theft more difficult.

The new namespace of HIP has some additional benefits when the extensions defined in this document are used. First, it is possible to address hosts behind a single NAT middlebox in a relatively simple way. The NAT middlebox translates the locators, but the Host Identifiers remain the same and can be used for uniquely identifying a host inside the private address realm. Second, multiple services on different hosts can share the same transport layer port number behind a single legacy NAT. There is no multiplexing issue as long as these hosts have different Host Identifiers and UDP encapsulation is used for traversing the legacy NAT.

Several different types of NATs exist [RFC2663]. This document describes HIP extensions for the traversal of both Network Address Translator (NAT) and Network Address and Port Translator (NAPT) middleboxes. The document generally uses the term NAT to refer to both types of middleboxes, unless it needs to distinguish between the two types.

Three basic scenarios exist for NAT traversal. In the first case, only the Initiator of a HIP base exchange is located behind a NAT. In the second case, only the Responder of a HIP base exchange is located behind a NAT. The respective peer is assumed to be located at a publicly reachable address in both cases. In the third case, both peers are located behind (possible different) NATs. All of the

use cases are addressed in the draft in a unified method that has been adopted from Interactive Connectivity Establishment (ICE) protocol [I-D.ietf-mmusic-ice] and adapted to HIP.

Legacy NAT devices do not operate consistently although the behavior for new NAT devices has been unified in [RFC4787]. The HIP protocol extensions in this document make as little assumptions as possible of the behavior of the NAT devices so that NAT traversal will work even with legacy NAT devices in the most general sense. The purpose of the extensions is to allow two HIP-enabled hosts to communicate with each other even if one or both communicating hosts are in private address realms. With some legacy NAT devices, connecting two hosts behind different address realms is impossible without relaying all traffic through a third party host [I-D.ietf-behave-p2p-state]. As a consequence, the relay host introduces additional hops between the hosts and can become a point of network congestion. In the extensions described in this document, the peers try to avoid the use of a relay for data traffic and only make use of it when necessary.

Hosts that always get a public addresses can use the rendezvous services as described in [I-D.ietf-hip-rvs]. Hosts that can be located in private-address realms may use a transport-layer based relay service as defined in this document. Both rendezvous and relay services forward HIP control packets, but the main difference is that the rendezvous service forwards only the initial I1 packet of the base exchange while all other HIP control packets are sent directly between the communicating hosts. In contrast, the relay service relays all HIP control packets because p2p-unfriendly NAT devices drop the packets otherwise [I-D.ietf-behave-p2p-state]. The peers use the control channel to communicate their current locators to each other to find a direct path for carrying ESP encapsulated data traffic. A direct path between the hosts enables efficient delivery of data traffic without relaying of ESP packets through an intermediary ESP relay. The direct path is searched using connectivity tests.

The basis for the connectivity tests is ICE [I-D.ietf-mmusic-ice]. Two hosts communicate their transport locator (a port and an IP address) to each other in a base exchange. The local locators are paired with peer locators and the pairs are prioritized according to their proximity. The locator pairs are tested sequentially in

priority order using return routability tests [I-D.ietf-hip-mm]. Both sides participate in the connectivity tests. The tests also determine whether transport layer encapsulation is required or not. As a result, the hosts either detect that no transport locator pairs are working, or establish a number of working locator pairs and select a single pair to be used for communication.

The same connectivity tests are also used in situations when a mobile host moves to a different network. The mobile host communicates its new location to the corresponding node through the relay server of its peer and starts the connectivity tests.

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].

3. HIP Across NATs

This section describes NAT traversal between two HIP end-hosts. A successful NAT traversal requires at least the Responder located in a private address realm to register to a relay server. The use of the relay is optional when the Responder is located in a public address realm without rendezvous server.

The base exchange is relayed through the relay server. Next, the

hosts test the reachability between the different locators to construct a direct route. When a direct route is not possible, the hosts resort to ESP relays. When locators of a host change, the hosts test reachability of locators again and select the "optimal" locator. End-hosts can tear down HIP associations using the CLOSE mechanism through the relay.

3.1. Port Number Selection

This document defines only UDP encapsulation for HIP and ESP packets. Further extensions may define bindings for other transport protocols. The RECOMMENDED transport protocol is UDP.

It is RECOMMENDED that an Initiator selects a random port number between the ephemeral port ranged 49152-65535 for initiating a base

exchange even for registration. However, the allocated port MUST be maintained until all of the corresponding Host Associations are closed. Alternatively, a host MAY also use a single fixed port for initiating all outgoing connections.

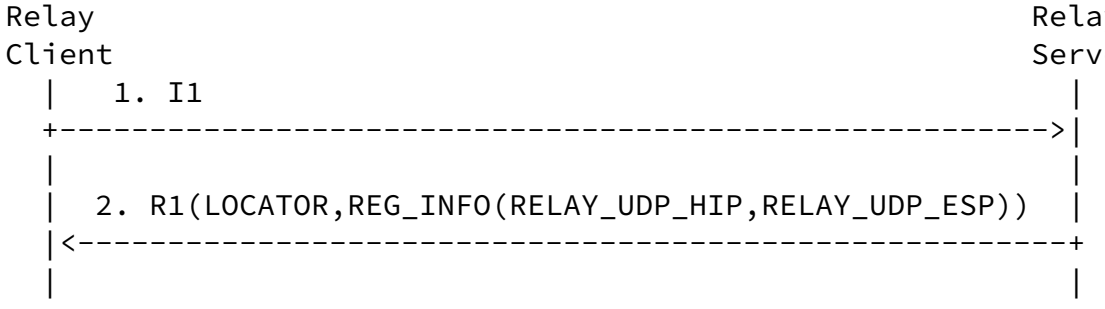
A relay or a Responder without a relay MUST listen at transport port HIPPORT for incoming UDP-encapsulated HIP control packets.

3.2. Relay Registration and NAT Detection

HIP rendezvous servers are used in non-NATted environments and its use is described in [I-D.ietf-hip-rvs]. This section defines the another types middleboxes, called HIP and ESP Relays, which are used in NATted environments.

A HIP relay forwards UDP-encapsulated traffic, and in future extensions, a relay may also forward TCP-encapsulated traffic. A single relay may forward only HIP control packets, ESP traffic or both. A host acting as a Responder in a private address realm SHOULD use a HIP relay for NAT traversal. It is RECOMMENDED that the Responder uses also an ESP relay to guarantee successful NAT traversal with p2p-unfriendly NAT devices.

A relay MUST NOT forward any packets to a host that has not successfully registered to the relay. The registration process follows the generic registration extensions defined in [I-D.ietf-hip-registration]. The registration process is illustrated in Figure 1.



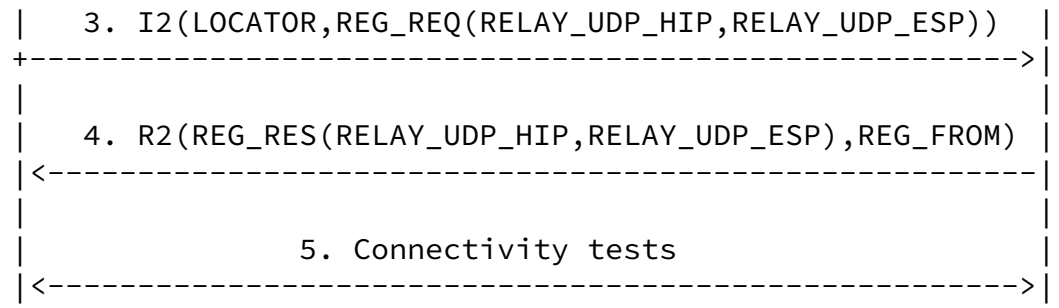


Figure 1: Example registration to a relay

In the above figure, the end-host is referred to as a relay client and the relay middlebox as a relay server. The registration is piggybacked to a base exchange, but it can be done also using HIP UPDATE control packets as described in [I-D.ietf-hip-registration].

In step 1, the relay client starts the registration procedure by sending an I1 packet over the transport layer. The port selection was explained in section Section 3.1.

In step 2, the Responder lists the services that it supports in the R1 packet. The support for HIP-over-UDP relaying is denoted by RELAY_UDP_HIP value and the support for ESP-over-UDP relaying is denoted by a RELAY_UDP_ESP value in the REG_INFO parameter.

In step 3, the Initiator selects the services it registers to and lists them in the REG_REQ parameter. In this example, the Initiator registers both to HIP and ESP relay services.

In step 4, the relay server concludes the registration procedure with an R2 packet and acknowledges the registered services in the REG_RES parameter. The relay may also denote unsuccessful registrations in the REG_FAILED parameter in R2. After the registration, the hosts MUST send periodically NAT keepalive packets to each other as defined later in this document.

In step 5, the client and server handle connectivity tests. The procedure is described in a later section.

When the ESP relay registration was successful, the relay server uses

the source IP address and port of the R2 packet (HIPPORT) to relay

ESP traffic with the client. This address-port pair of the relay is referred to as a "leased transport locator" in this document. As the port number may be shared by multiple clients, the ESP relay MUST multiplex the ESP traffic based on SPIs and not the just the port number.

The R2 packet also includes an REG_FROM parameter that indicates the transport locator of the client as observed by the server. The transport locator may be translated by a number of NAT middleboxes between the client and the server. This locator is referred to as the "relay reflexive transport locator" later in this document.

A single server can provide multiple HIP middlebox services or the services can be distributed among multiple servers. The difference between a HIP rendezvous server [I-D.ietf-hip-rvs] and a HIP relay server depends on the registration. The rendezvous server processing rules apply when the Responder has registered to a middlebox with the RVS registration type. Correspondingly, the middlebox applies the relay extensions defined in this document when the Responder has registered using the relay registration types. When a single server provides both rendezvous and relay services, they are multiplexed depending on the absence or presence of transport layer encapsulation.

The Relay Client MUST include a LOCATOR parameter in I2 which lists all of the locators of the Initiator. The Relay Server MUST include a LOCATOR parameter in R1, but it is RECOMMENDED that the LOCATOR parameter includes only the source transport LOCATOR of R1 as the only locator. The case when the Relay Server includes more locators may require IP header conversion between IPv4 and IPv6, insertion, or removal of, UDP header and fragmentation handling. Multiple locators in R1 is left for further experimentation.

3.3. Base Exchange via Relay

It is RECOMMENDED that the Initiator sends an I1 packet over the transport layer when it is destined to an IPv4 address of the Responder. Respectively, the Responder MUST respond to a such I1 packet with an R1 packet over the transport layer and using the same transport protocol. The rest of the base exchange, I2 and R2, MUST also be sent over the transport layer. However, the transport layer encapsulation can be unnecessary when there are no NATs between the Initiator and Responder. This will be detected in the connectivity tests described in the next section.

When the Initiator has an IPv6 address and it has discovered only an IPv6 address for the peer, it MUST send it directly over IP. In such

a case, the Initiator MUST follow the procedures described in [I-D.ietf-hip-base]. Otherwise, it is RECOMMENDED that the Initiator proceeds as shown in Figure 2.

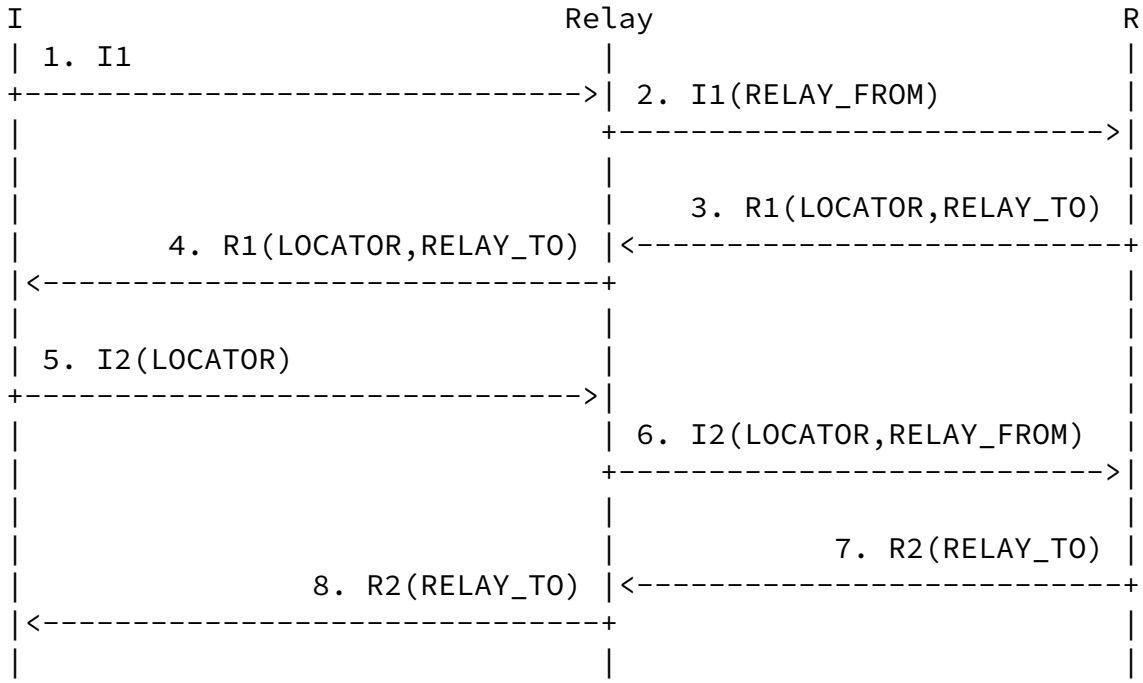


Figure 2: Base Exchange via a relay

In step 1 of the figure, the Initiator discovers the HIT of the Responder and the IPv4 address of the relay of the Responder. The Initiator sends an I1 packet over the transport layer to the HIT of the Responder. The port selection was explained in Section 3.1. The source address is one of the routable addresses of the host is called "unreflexive locators" in this document.

In step 2, the relay receives the I1 packet at port HIPPORT. If the destination HIT belongs to a registered Responder, the relay processes the packet. Otherwise, the relay MUST drop the packet. The relay MUST append a RELAY_FROM parameter to the I1 packet which preserves the transport source address and port of the Initiator. The relay protects the I1 packet with RELAY_HMAC as described in [I-D.ietf-hip-rvs], except that the parameter type is different. The relay MUST change the transport source to and destination of the packet to match the values the Responder used when registering to the relay, i.e., the reverse of the R2 used in the registration. The relay MUST recalculate the transport checksum and forwards the packet to the Responder.

In step 3, the Responder receives the I1 packet at the transport

layer. The Responder MUST process it according to the rules in [I-D.ietf-hip-base]. In addition, the Responder MUST validate the

RELAY_HMAC according to [I-D.ietf-hip-rvs] and drop the packet if the validation fails. The Responder replies with an R1 packet that MUST contain a LOCATOR parameter that lists the locators of the Responder. The locator list consists of unreflexive, reflexive and leased transport locators of the Responder. The R1 packet also contains a RELAY_TO parameter. The RELAY_TO parameter contains same information as the RELAY_FROM parameter, i.e., Initiator transport locator, but the type of the parameter is different. The RELAY_TO parameter is not integrity protected by the signature of the R1 to allow pre-created R1 packets at the Responder.

In step 4, the relay receives the R1 packet. The relay MUST drop the packet if the source HIT belongs to an unregistered host. The relay MAY verify the signature of the R1 packet and drop it when the signature is invalid. Otherwise, the relay changes the destination transport header to match RELAY_TO information, recalculates transport checksum and forwards the packet.

In step 5, the Initiator receives the R1 packet and processes it accordingly to [I-D.ietf-hip-base]. It replies with an I2 packet that has the same transport locator as R1, but the source and destination ports are swapped. The I2 contains a LOCATOR parameter containing the listing unreflexive, reflexive and leased transport locators of the Initiator

In step 6, the relay receives the I2 packet. The relay appends a RELAY_FROM and a RELAY_HMAC to the I2 packet as in the second step.

In step 7, the Responder receives the I2 packet and processes it according to [I-D.ietf-hip-base]. It replies with an R2 packet and includes a RELAY_TO parameter as in step three. The RELAY_TO parameter is protected by the HMAC.

In step 8, the relay processes the R2 as described in step four. The relay forwards the packet to the Responder.

3.4. Base Exchange without a Relay

A host that has a publicly addressable, fixed IP address MAY exclude

registration to a Relay. As the Relay is not present, the host MUST listen at HIPPORT for transport-encapsulated HIP and ESP packets. An UDP-encapsulated base exchange with such an host does not have the RELAY_TO and RELAY_FROM parameters present. Connectivity tests MUST be handled as defined in the following section before any ESP traffic is allowed.

3.5. Connectivity Tests

The base exchange is completed with an R2 packet. Then, the state of the HIP associations at both peers is ESTABLISHED, but the peers MUST NOT allow any ESP traffic until the connectivity tests described in the next section are performed successfully. All of the locators, except the relay address, are in UNVERIFIED state. In the connectivity tests, the hosts test connectivity between different locator pairs in order to find a working one. The connectivity tests are illustrated in Figure 3. In this example, both hosts are behind NATs.

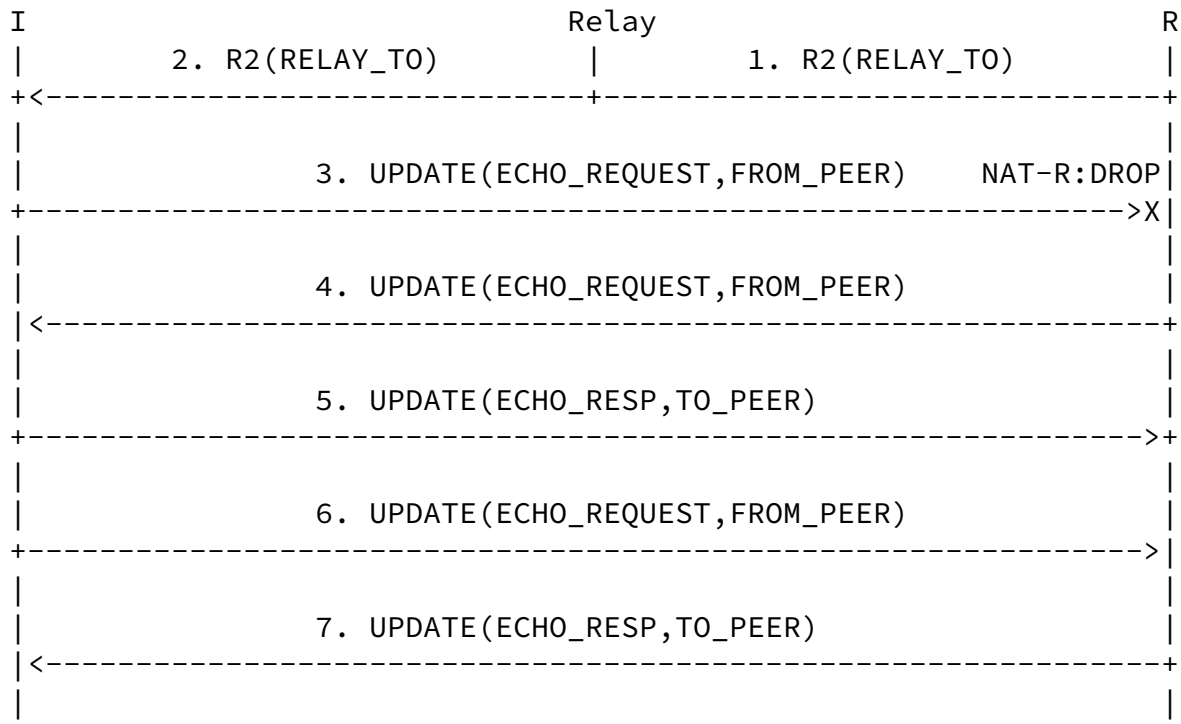


Figure 3: Connectivity tests

The connectivity tests are handled as the mobility extensions defined in [I-D.ietf-hip-mm] and are therefore subject to the same processing rules. The packets include ESP_INFO, SEQ, ACK, HMAC, SIGNATURE parameters that are omitted in this section for simplicity. The differences to the mobility extensions are described in this section.

In steps 1 and 2, the R2 packet is relayed from the Responder through the Relay to the Responder. After this, both hosts start connectivity tests using the return routability tests defined in [I-D.ietf-hip-mm]. The return routability tests are used to probe for connectivity between each locator pair obtained from the local and peer locators obtained during base exchange. The return routability tests are also used as a UDP hole punching mechanism. The tests are carried in certain order which determined by the

priorization algorithm defined in the next section.

As an example, let's consider the case where hosts are testing each others outermost NAT addresses, i.e., relay reflexive transport locators. In step 3, host I sends an UPDATE message containing an ECHO_REQUEST to the R. This will punch a hole the NAT of I, but the NAT of R drops the message because the NAT of R has no state with I.

In step 4, R starts also reachability detection by sending an UPDATE with ECHO_REQUEST. This traverses the NAT of I successfully because Initiator had already punched an hole into its NAT in step 3. The Responder replies using ECHO_RESPONSE in step 5. Upon receiving the ECHO_RESPONSE, the Responder transitions the address pair to VERIFIED state.

In step 6, host I starts a new return routability test either due to a retransmission timer or as a reaction to UPDATE with ECHO_REQUEST received from R. In step 7, host R receives and sends a response to I. Upon receiving the response, host R transitions the locator pair being tested to VERIFIED state.

All locators in UNVERIFIED state MUST be retransmitted RTIME times. The retransmission packets MUST be paced Ta ms apart as defined in [I-D.ietf-mmusic-ice]. The retransmission are ordered in a sequence determined by the priority of the transport locator pairs, as

described in the next section.

The source address of the UPDATE messages containing ECHO_REQUEST parameter is always an unreflexive IPv4 locator of the host. The destination locator is the peer's unreflexive, reflexive or leased transport locator, depending on which address is being tested for reachability. Implementations may add RTT measurement information to the ECHO_REQUEST parameter in addition to a nonce.

The UPDATE messages carrying ECHO_REQUEST include a FROM_PEER parameter. The sender of the UPDATE MUST copy the source address of the UPDATE to the FROM_PEER parameter. When the peer receives the UPDATE, it responds with an UPDATE containing and a ECHO_REQUEST and TO_PEER parameters. The TO_PEER parameter MUST contain the source address of the UPDATE redundantly. The reason from the FROM_PEER and TO_PEER parameters is that it is possible to learn new addresses using them. When there is p2p-unfriendly NAT between the peers, it may cause translate port number of the UPDATE packets to something that has not been communicated through the relay before. Such an addresses are called "peer reflexive transport locators" in this document. The FROM_PEER and TO_PEER parameters can be used for detecting peer reflexive locators. The learned locators are added to the connectivity tests.

UPDATE packets destined to the unreflexive locators are sent directly over IP. UPDATE packets destined for reflexive peer, relay and leased locators are sent transport layer encapsulated.

Hosts proceed sequentially through the locator pairs in the order described in the next section. A host MUST transition the state of transport locator pairs verified by the return routability tests to the ACTIVE state. Keepalive mechanisms described in later sections MUST be applied to refresh the port state in NAT devices for locators in the ACTIVE state. A host MUST also set up the Security Associations for the inbound ESP traffic for such locators. The selection of a default outbound SA is defined in the next section.

3.6. Selecting an Address Pair

This section describes priority ordering of connectivity tests and locators pair selection based on ICE [I-D.ietf-mmusic-ice]. As part of the priority calculation, each locator has a preference based on

its type. The values for these preferences are shown in Table 2.

Locator Type	Preference
The preferred locator	127
Unreflexive locator	126
Peer reflexive transport locator	120
Relay reflexive transport locator	100
Leased transport locator	0

Table 2: Locator Type Preferences

In addition to the "type" priority, the priority of a locator is also affected by the "local" priority. A (multihoming) host may have multiple locators of same type and SHOULD assign a unique local priority for each locator. Hosts preferring IPv6 communication can assign higher local preferences for IPv6 locators than for unreflexive IPv4 locators. ECHO_REQUEST parameters may include RTT calculation information that an implementation may use to increase the local priority. A host SHOULD calculate locator priority based on the local and type priorities as shown in Figure 4. The locator priority MUST always be included in the type 3 locator fields in LOCATOR parameters as described in section Section 4.4.

```
Locator priority = (2^24) * (type preference) +
                   (2^8) * (local preference)
```

Figure 4: Locator priority

A host SHOULD calculate a priority for each locator pair as shown in Figure 5. I and R denote the priorities of locators of Initiator and Responder. The use of the same formula at both ends gives more guarantees that the peers prefer shortest paths between them. It also converges the selection of the locator pair towards a symmetric pair instead of an asymmetric pair even though it is not completely guaranteed. The reasoning for the formula is described in [I-D.ietf-mmusic-ice].

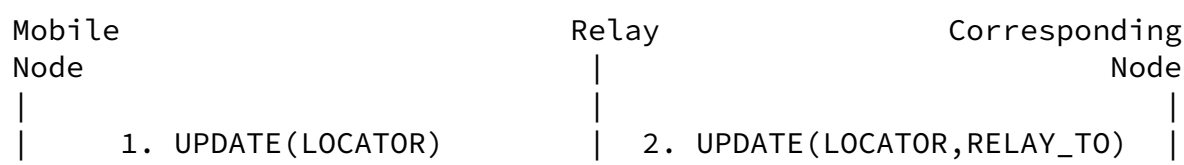
```
Pair priority = 2^32 * MIN(I,R) + 2 * MAX(I,R) + (I > R ? 1 : 0)
```

Figure 5: Pair priority

After reachability tests, both hosts SHOULD assign the transport address pair with the highest pair priority as their default outgoing SA for ESP.

3.7. Mobility

When one of the hosts changes its locators, it has to notify its peers of the address change. This is handled as described in the connectivity tests in Section 3.5 with the exception that the UPDATE with parameter LOCATOR is used as the trigger to start connectivity tests instead of the R2. The UPDATE packet contains a LOCATOR parameter listing unreflexive, reflexive and leased transport locators of the Initiator. This is illustrated in Figure 6.



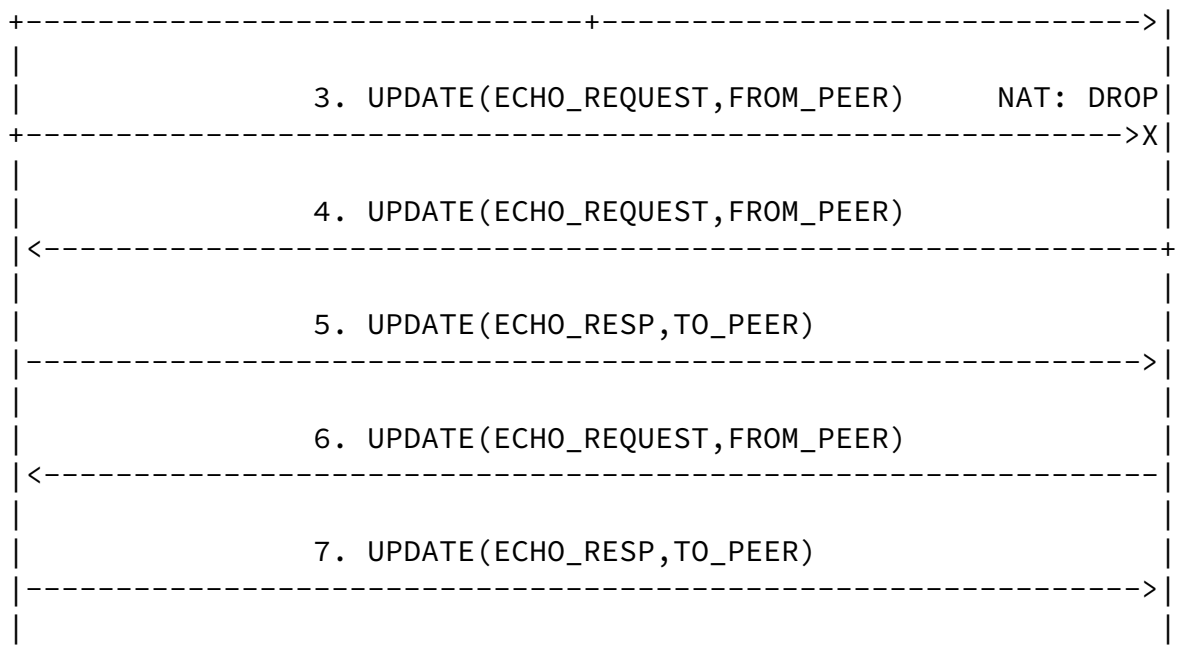


Figure 6: Handover

When a mobile host moves from a private address realm to another, it can obtain the same locator on both networks. To denote that the new locator requires reachability detection, the mobile host MUST use a new SPI for the new locator.

A host can also use the UPDATE mechanism can also be used for switching to a more optimal path after connectivity tests. In the connectivity tests, the host may implement RTT measurements within ECHO_REQUEST and ECHO_RESPONSE messages. In some cases the result of the RTT measurements may indicate that another locator pair is more optimal than the locator pair resulting from the connectivity and priority tests. In such a case, the host MAY send UPDATE with LOCATOR parameter with the optimal locator with the preferred bit on. This gives the highest priority for the most optimal locator and will be used if the connectivity tests succeed.

3.8. NAT Keepalives

A NAT can delete the mapping state after a timeout when there is no traffic refreshing the state. For this reason, both hosts MUST send keep-alives to each other for all locators pairs that are in the ACTIVE state. Keepalives MUST be sent every 20 seconds for UDP. The keepalive is a NOTIFY packet without parameters.

The keep-alives MAY also be used to implement failure detection between end-hosts as in [I-D.oliva-hiprg-reap4hip] (XX FIXME: this needs still more details). The basic idea is to keep track of HIP control and ESP packets received over a transport port. When there is no HIP or ESP traffic (not even keep-alives) arriving during a certain time period, the host switches to an alternative locator pair. The host transitions the default locator pair to the UNVERIFIED state and replaces the currently default SA to correspond to the ACTIVE locator pair with the highest priority. The host may also try to send an UPDATE packet with the LOCATOR parameter after a certain time period if connectivity is still broken.

End-host may also used the keep-alives to detect loss of connectivity with relay server. When this occurs, the end-host can register to a new relay and replace the IP address of the old relay server with a new one in DNS or DHT.

3.9. Closing of HIP Associations

A host closes a HIP association as described in [I-D.ietf-hip-base] except that the CLOSE and CLOSE_ACK packets are sent over transport layer and through the relay as illustrated in Figure 7. Hosts MUST transition the corresponding locator pairs to the DEPRECATED state after a successful CLOSE-CLOSE_ACK exchange. The corresponding inbound and outbound SAs must be deleted on such occasion.

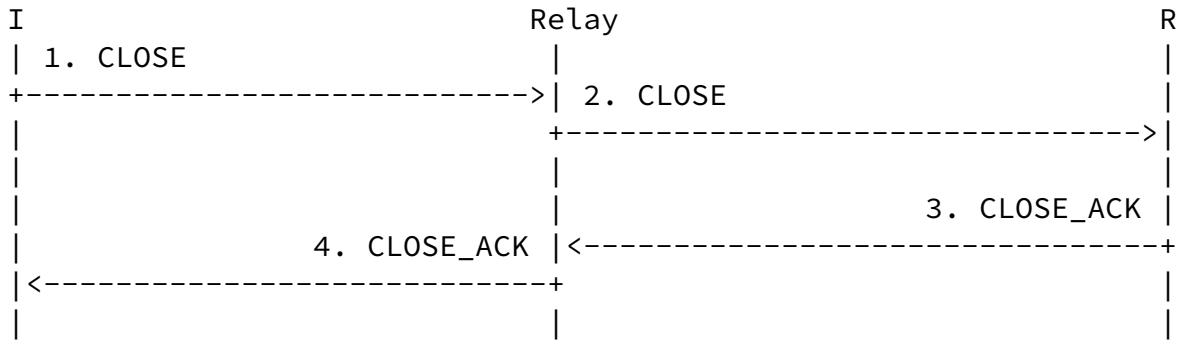


Figure 7: Closing of a HIP association

The hosts may also use the CLOSE mechanism to remove redundant SAs remaining from the connectivity tests. However, the removal can prolong the recovery in the event of connectivity failures.

3.10. Communication with HIP Hosts without NAT Traversal Support

The UDP encapsulation of HIP and ESP control packets has not been defined in any other IETF document and legacy hosts drop all UDP encapsulated HIP and ESP traffic. Processing of unknown locator

types terminates the base exchange or UPDATE. As such, the

extensions defined in this document are not completely backwards compatible and require a minimal support in implementations.

A minimal implementation MUST provide UDP encapsulation of HIP and ESP packets. In such a case, the minimal NAT traversal implementation MUST silently discard the processing of type 3 locators to allow communication with implementations supporting NAT traversal defined in this document. The minimal implementation MUST support UDP keepalives to refresh state of the NAT(s).

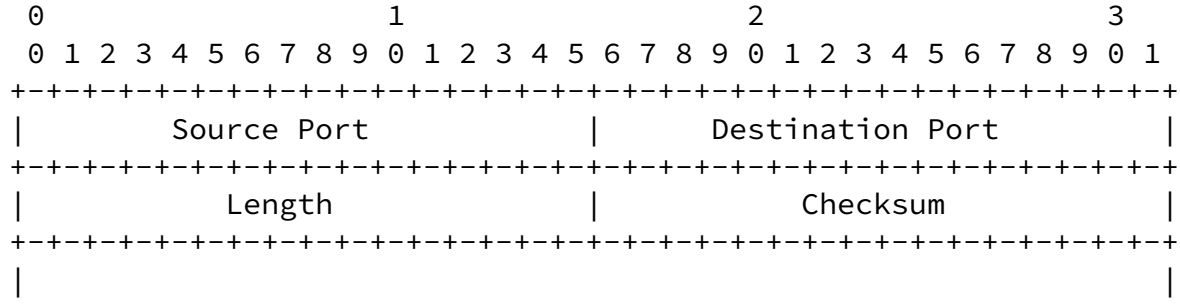
Hosts that conform to [I-D.ietf-hip-mm] respond to UPDATE messages containing an ECHO_REQUEST with an UPDATE message containing an ECHO_RESPONSE. This completes the connectivity tests for the host supporting the extensions defined in this document. As long as the implementation supports UDP encapsulation of HIP control packets, this requires no changes.

The Relay extensions defined in this document do not work with minimalistic implementations. When there is a Relay between the hosts, both the Initiator and Responder MUST support the extensions defined in this document. The presence of RELAY_TO and RELAY_FROM parameters denotes the precence of a relay.

4. Packet Formats

This section defines an UDP-encapsulation packet format for HIP base exchange and control traffic, IPsec ESP BEET-mode traffic and NAT keep-alive packets.

4.1. HIP Control Packets



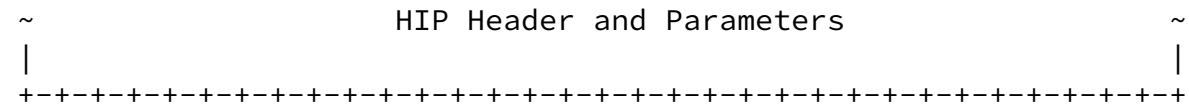


Figure 8: Format for UDP-encapsulated HIP control packets

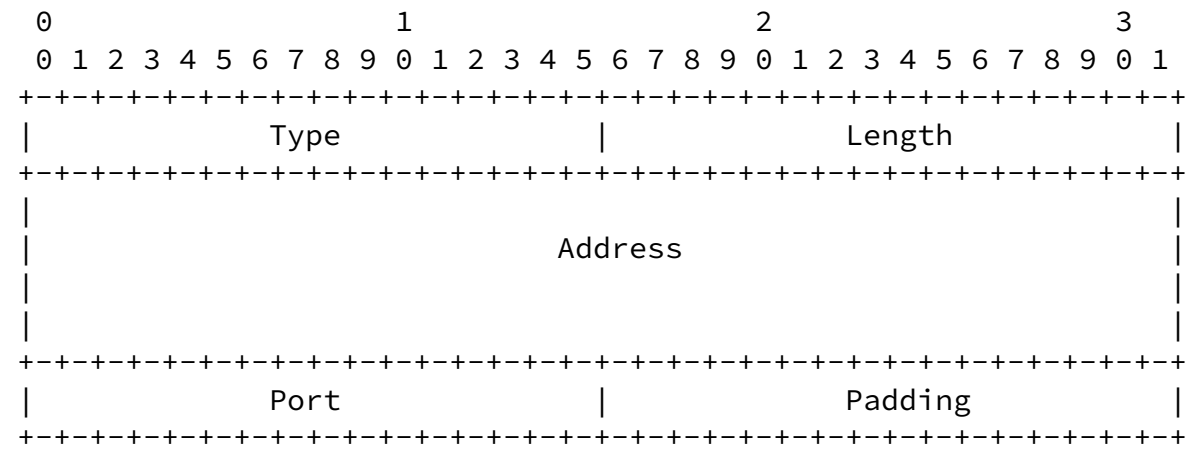
Figure 8 shows how HIP control packets are encapsulated within UDP. A minimal UDP packet carries a complete HIP packet in its payload.

Contents of the UDP source and destination ports are described below. The UDP length and checksum field MUST be computed as described in [RFC0768]. The HIP header and parameter follow the conventions [I-D.ietf-hip-base] with the exception that the HIP header checksum MUST be zero. The HIP header checksum is zero for two reasons. First, the UDP header contains already a checksum. Second, the checksum definition in [I-D.ietf-hip-base] includes the IP addresses in the checksum calculation. The NATs unaware of HIP cannot recompute the HIP checksum after changing IP addresses.

4.2. Control Channel Keep-Alives

The keep-alive for control channel are UDP encapsulated NOTIFY packets [I-D.ietf-hip-base]. The NOTIFY packets MAY contain HIP parameters. The NAT traversal mechanisms encapsulate these NOTIFY packets within the payload of UDP packets.

4.3. RELAY_FROM, RELAY_TO and RELAY_VIA Parameters



Type [TBD by IANA:

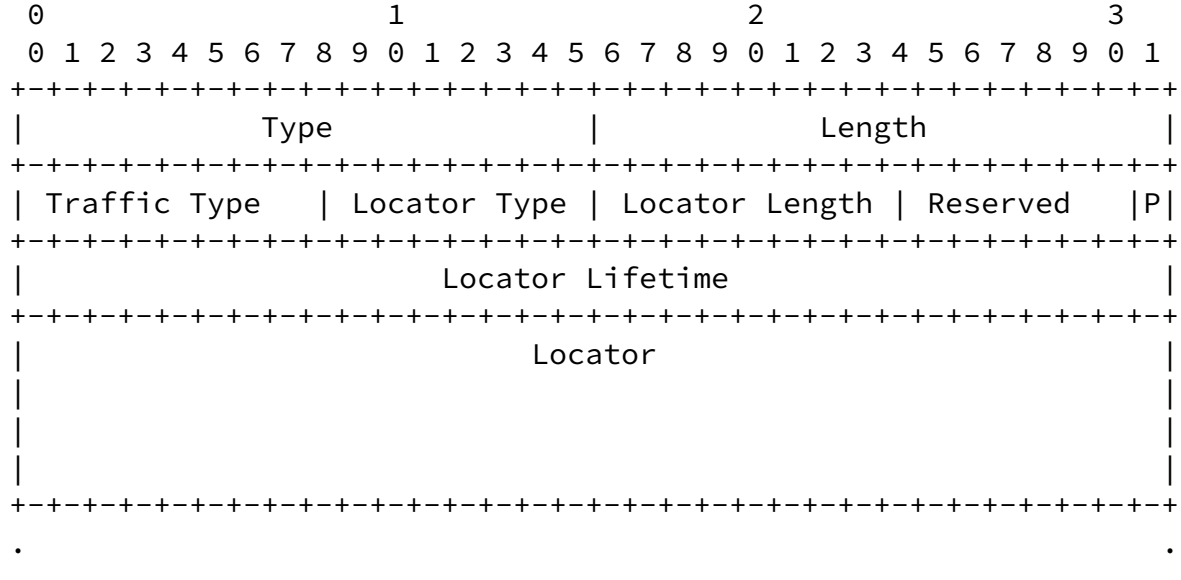
```
                RELAY_FROM: (63998 = 2^16 - 2^11 + 2^9 - 2)
                RELAY_TO:   (64002 = 2^16 - 2^11 + 2^9 + 2)
                RELAY_VIA:  (64006 = 2^16 - 2^11 + 2^9 + 6) ]
   <!-- AG: those are not described?
                TO_PEER:    (64010 = 2^16 - 2^11 + 2^9 + 10)
                REG_FROM:   (64010 = 2^16 - 2^11 + 2^9 + 12) ]
   -->
   Length      18
   Address     An IPv6 address or an IPv4 address in IPv4-in-IPv6
               format.
   Port        Transport port number
```

Figure 9: Format for the RELAY_FROM, RELAY_TO and RELAY_VIA parameters

Figure 9 shows the format of RELAY_FROM, RELAY_TO and RELAY_VIA parameters.

4.4. LOCATOR Parameter

The generic LOCATOR parameter format is the same as in [I-D.ietf-hip-mm]. However, presenting transport locators requires a new locator type. The generic and NAT specific locator parameters are illustrated in Figure 10.



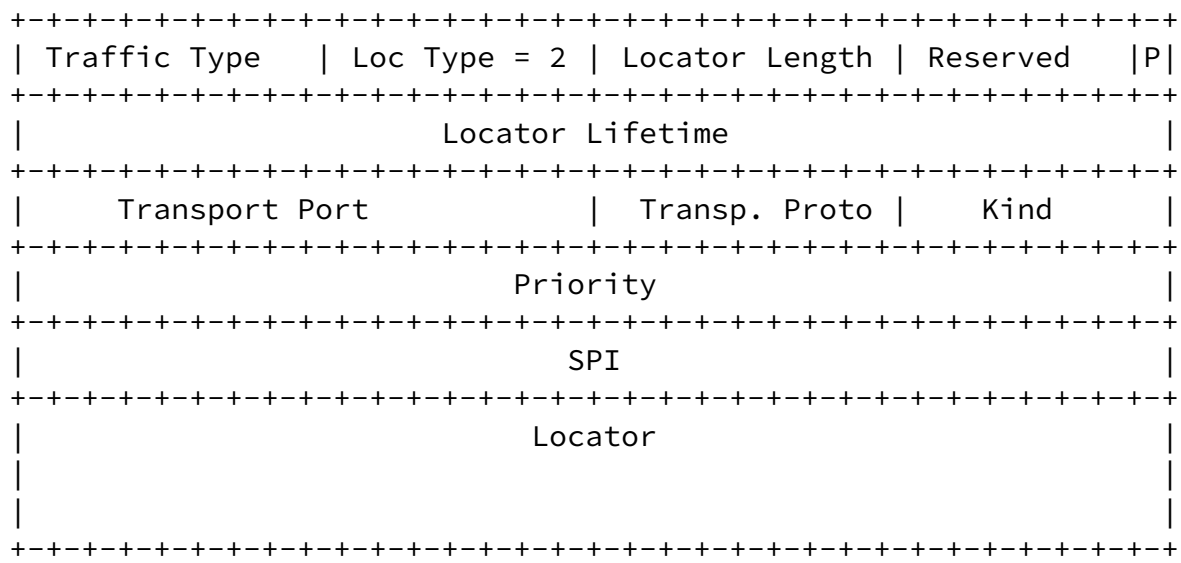


Figure 10: Locator parameter

The individual fields in the LOCATOR parameter are described in Table 3.

Field	Value(s)	Purpose
Type	193	Parameter type
Length	Variable	Length in octets, excluding Type and Length fields, and excluding padding.
Traffic Type	0-2	2 for unreflexive and leased, 1 for relay reflexive
Locator Type	3	Transport locator
Locator Length	19	Length of the Locator field in 4-octet units
Reserved	0	Reserved for future extensions
Preferred (P) bit	0	Usually zero for type 3 locators
Locator Lifetime	Variable	Locator lifetime in seconds
Transport Port	Variable	Zero for unreflexive and greater than zero otherwise

Transport Protocol	0	Zero for UDP
Kind	Variable	0 for unreflexive, 1 for relay reflexive, 2 for leased
Priority	Variable	Locator preference, see Section 3.6
SPI	Variable	0 for relay reflexive, otherwise greater than zero
Locator	Variable	An IPv6 address or an IPv4-in-IPv6 format IPv4 address[RFC2373]

Table 3: Fields of the locator parameter

4.5. RELAY_HMAC

The RELAY_HMAC parameter value has the TLV type 65520 (2^16 - 2^5 + 2^4). It has the same semantics as RVS_HMAC [I-D.ietf-hip-rvs].

4.6. Registration Types

The REG_INFO, REQ_REQ, REG_RESP and REG_FAILED parameters contains values for relay registration. The value for RELAY_UDP_HIP is 2. The value for RELAY_UDP_ESP is 3.

4.7. ESP Data Packets

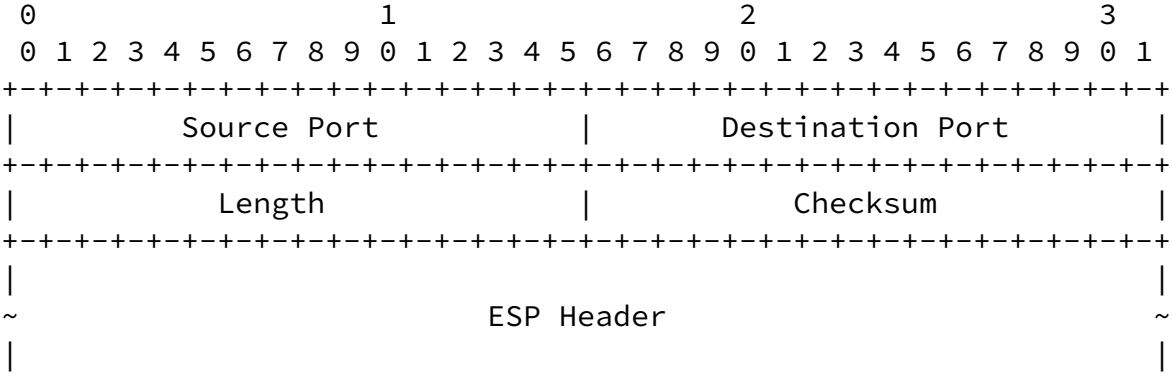


Figure 11: Format for UDP-encapsulated IPsec ESP BEET-mode traffic

Figure 11 shows how IPsec ESP BEET-mode packets are encapsulated within UDP. Again, a minimal UDP packet carries the ESP packet in its payload. The contents of the UDP source and destination ports are described in later sections. The UDP length and checksum field MUST be computed as described in [RFC0768].

4.8. UDP Encapsulation/Decapsulation of IPsec BEET-Mode ESP

[RFC3948] describes UDP encapsulation of the IPsec ESP transport and tunnel mode. This section describes the UDP encapsulation of the BEET mode.

4.8.1. UDP Encapsulation of IPsec BEET-Mode ESP

During the HIP base exchange, the two peers exchange parameters that enable them to define a pair of IPsec ESP security associations (SAs), as described in [I-D.ietf-hip-esp]. When two peers perform a UDP-encapsulated base exchange, they MUST define a pair of IPsec SAs that produces UDP-encapsulated BEET-mode ESP data traffic.

The management of encryption/authentication protocols and security parameter indices (SPIs) is defined in [I-D.ietf-hip-esp]. Additional SA parameters, such as IP addresses and UDP ports, MUST be defined according to this section. Two SAs MUST be defined on each host for one HIP association; one for outgoing data and another one for incoming data.

The BEET mode provides limited tunnel mode semantics without the regular tunnel mode overhead [I-D.nikander-esp-beet-mode]. In the BEET mode, transport-layer checksums in the payload data are based on the HITs. The packet MUST then undergo BEET-mode ESP cryptographic processing as defined in Section 5.3 of [I-D.nikander-esp-beet-mode].

Next, the resulting BEET-mode packet is UDP encapsulated. For this purpose, a UDP header MUST be inserted between the IP and ESP header. The source and destination ports are filled in. The UDP checksum MUST be calculated based on the outer addresses (locators) of the IPsec security association. The other fields of the UDP header are

computed as described in [RFC0768].

The resulting UDP packet MUST then undergo BEET IP header processing as defined in Section 5.4 of [I-D.nikander-esp-beet-mode].

Figure 12 illustrates the BEET-mode UDP encapsulation procedure for a TCP packet.

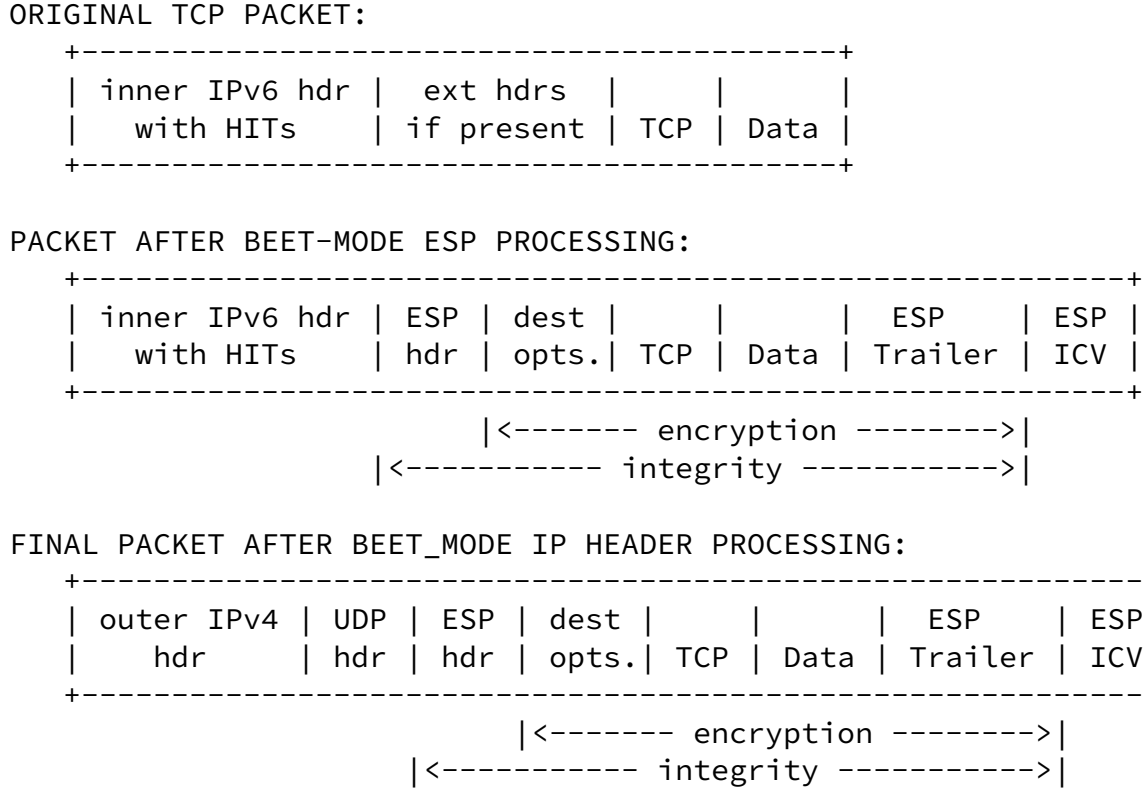


Figure 12: UDP encapsulation of an IPsec BEET-mode ESP packet containing a TCP segment

4.8.2. UDP Decapsulation of IPsec BEET-Mode ESP

An incoming UDP-encapsulated IPsec BEET-mode ESP packet is decapsulated as follows. First, if the UDP checksum is invalid, then the packet MUST be dropped. Then, the packet MUST be verified as defined in [I-D.nikander-esp-beet-mode]. If verified, the ESP data contained in the payload of the UDP packet MUST be decrypted as described in [I-D.nikander-esp-beet-mode].

5. Firewall Traversal

This section describes firewall traversal issues separately from NAT issues. When the Initiator or the Responder of a HIP association is behind a firewall, additional issues arise.

The NAT traversal mechanisms described in Section 3 require that the firewall - stateful or not - allows UDP traffic. At the minimum, successful firewall control packet traversal requires that the host behind the firewall is allowed to communicate packets with a HIP relay (or a Responder without Relay) that is listening on UDP port HIPPORT. Successful ESP data packet traversal requires the same for the ESP relay. For unrelayed traffic, the destination port HIPPORT should be open at the firewall to all hosts behind the firewall.

Most firewall implementations support "UDP connection tracking", i.e., after a host behind a firewall has initiated UDP communication to the public Internet, the firewall relays UDP response traffic in the return direction. If no such return traffic arrives for a specific period of time, the firewall stops relaying the given IP address and port pair. The mechanisms described in Section 3 already enable traversal of such firewalls, if the keep-alive interval used is less than the refresh interval of the firewall.

When the Initiator is behind a firewall, the NAT traversal mechanisms described in Section 3 depend on the ability to initiate communication via UDP to the destination port HIPPORT from arbitrary source ports and to receive UDP response traffic from that port to the chosen source port. If the Initiator is behind a firewall that does not support "UDP connection tracking", the NAT traversal mechanisms described in Section 3 can still be supported, if the firewall allows permanently inbound UDP traffic from the port HIPPORT and destined to arbitrary source IP addresses and UDP ports.

When the Responder is behind a firewall, the NAT traversal mechanisms described in Section 3 depend on the ability to send and receive UDP traffic originating from HIPPORT of the HIP and ESP relays. When unrelayed traffic is preferred, arbitrary source IP addresses and ports are required.

6. Security Considerations

6.1. A Difference to RFC3948

Section 5.1 of [RFC3948] describes a security issue for the UDP encapsulation in the standard IP tunnel mode when two hosts behind different NATs have the same private IP address and initiate

communication to the same Responder in the public Internet. The Responder cannot distinguish between two hosts, because security associations are based on the same inner IP addresses.

This issue does not exist with the UDP encapsulation of IPsec BEET mode as described in Section 3, because the Responder use HITs to distinguish between different communication instances.

6.2. Privacy Considerations

The LOCATORs are sent in plain text. Alternatively, they could be encrypted. This option was not chosen to allow packet inspection by middleboxes. Plain text locators may be useful for HIP-aware middleboxes in the future.

It is possible that an Initiator or Responder may not want to reveal all of its locators to its peer. For example, a host may not want to reveal the internal topology of the private address realm and it discards unreflexive locators. Such behavior creates non-optimal paths when the hosts are located behind the same NAT. Especially, this could be a problem with a legacy NAT that does not support routing from the private address realm back to itself through the outer address of the NAT. This scenario is referred to as the hairpin problem [I-D.ietf-behave-p2p-state]. With such a legacy NAT, the only option left would be to use a leased transport locator from a relay. As a consequence, a host may support locator-based privacy by leaving out the reflexive locators. Using only unreflexive locators can produce suboptimal paths possibly causing congestion.

The use of relays can be useful for protection against Denial-of-Service attacks. If a Responder reveals only its HIP and ESP relay addresses to malign Initiators, the Initiators can only attack the relays that does not prevent the Responder from initiating new outgoing connections if a path around the relay exists.

6.3. Opportunistic Mode

The use of opportunistic HIP is NOT RECOMMENDED and its use is not defined in this document. In opportunistic HIP, the Initiator sends the I1 message with null destination HIT. Private address realms do not have unique addresses by definition. Therefore, opportunistic mode is subject to failure even when there are no attackers present.

In a normal HIP base exchange, a well-behaving Responder drops the I1 packet when the destination HIT does not belong to it. An attacker could respond to the I1, but the base exchange would eventually fail as the attacker would fail to prove its ownership of the destination HIT of the I1.

7. IANA Considerations

This section is to be interpreted according to [RFC2434].

This draft currently uses a UDP port in the "Dynamic and/or Private Port" and HIPPORT. Upon publication of this document, IANA is requested to register a UDP port and the RFC editor is requested to change all occurrences of port HIPPORT to the port IANA has registered. The HIPPORT number 50500 should be used for initial experimentation.

This document updates the IANA Registry for HIP Parameters Types by assigning new HIP Parameter Types values for the new HIP Parameters defined in Section 4: o RELAY_FROM (defined in Section 4.3) o RELAY_TO (defined in Section 4.3) o RELAY_VIA (defined in Section 4.3) o RELAY_HMAC (defined in Section 4.5)

8. Acknowlegements

The authors would like to thank Lars Eggert, Vivien Schmitt, Abhinav Pathak and Andrei Gurtov for their contributions to previous versions of this draft. Thanks for Philip Matthews on introducing ICE concepts to the authors and for proposing the initial design. Thanks for Jonathan Rosenberg and the rest of the MMUSIC WG folks for the excellent work on ICE. In addition, the authors would like to thank Tobias Heer, Teemu Koponen, Juhana Mattila, Jeffrey M. Ahrenholz, Thomas Henderson, Kristian Slavov, Janne Lindqvist, Pekka Nikander, Lauri Silvennoinen, Jukka Ylitalo, Juha Heinanen, Joakim Koskela, Samu Varjonen, Dan Wing, Hannes Tschofenig and Jani Hautakorpi for their comments on this document.

[I-D.nikander-hip-path] presented some initial ideas for NAT traversal of HIP communication. The idea was based on NAT detection using extra parameters in the base exchange. This document takes a

different approach based on ICE.

Simon Schuetz and Martin Stiemerling are partly funded by Ambient Networks, a research project supported by the European Commission under its Sixth Framework Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Ambient Networks project or the European Commission.

Miika Komu is working in the Networking Research group at Helsinki Institute for Information Technology (HIIT). The InfraHIP project was funded by Tekes, Telia-Sonera, Elisa, Nokia, the Finnish Defence Forces and Ericsson. Miika Komu wrote draft-ietf-hip-nat-02 version from scratch based on ICE-related comments from Philip Matthews.

9. References

9.1. Normative References

[I-D.ietf-hip-base]
Moskowitz, R., "Host Identity Protocol", draft-ietf-hip-base-08 (work in progress), June 2007.

[I-D.ietf-hip-esp]
Jokela, P., "Using ESP transport format with HIP", draft-ietf-hip-esp-06 (work in progress), June 2007.

[I-D.ietf-hip-mm]
Henderson, T., "End-Host Mobility and Multihoming with the Host Identity Protocol", draft-ietf-hip-mm-05 (work in progress), March 2007.

[I-D.ietf-hip-registration]
Laganier, J., "Host Identity Protocol (HIP) Registration Extension", draft-ietf-hip-registration-02 (work in progress), June 2006.

[I-D.ietf-hip-rvs]
Laganier, J. and L. Eggert, "Host Identity Protocol (HIP)

Rendezvous Extension", draft-ietf-hip-rvs-05 (work in progress), June 2006.

[I-D.ietf-mmusic-ice]
Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", draft-ietf-mmusic-ice-16 (work in progress), June 2007.

[I-D.nikander-esp-beet-mode]
Melen, J. and P. Nikander, "A Bound End-to-End Tunnel (BEET) mode for ESP", draft-nikander-esp-beet-mode-07 (work in progress), February 2007.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2373] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC4423] Moskowitz, R. and P. Nikander, "Host Identity Protocol (HIP) Architecture", RFC 4423, May 2006.

9.2. Informative References

[I-D.ietf-behave-p2p-state]
Srisuresh, P., "State of Peer-to-Peer(P2P) Communication Across Network Address Translators(NATs)", draft-ietf-behave-p2p-state-03 (work in progress), July 2007.

[I-D.irtf-hiprg-nat]
Stiemerling, M., "NAT and Firewall Traversal Issues of Host Identity Protocol (HIP) Communication",

draft-irtf-hiprg-nat-04 (work in progress), March 2007.

[I-D.nikander-hip-path]
Nikander, P., "Preferred Alternatives for Tunnelling HIP (PATH)", draft-nikander-hip-path-01 (work in progress), March 2006.

[I-D.oliva-hiprg-reap4hip]
Oliva, A. and M. Bagnulo, "Fault tolerance configurations for HIP multihoming", draft-oliva-hiprg-reap4hip-00 (work in progress), July 2007.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[RFC3948] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, January 2005.

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

Appendix A. Differences to ICE

The protocol extensions defined in this draft are based on ICE. The extensions are a rough translation of ICE concepts to HIP protocol. The translation preserved certain concepts as they are, but there are subtle differences. This section tries to explain how ICE concepts were mapped to HIP protocol and what are the differences.

The terminology for this draft is a hybrid of ICE and HIP terminology. "Agent" was translated to "host" in favour of HIP terminology. Transport address was changed to transport locator. Similarly, address pair is denoted as locator pair. This document does not really talk about "candidate addresses", but just "locators" which may or may not be verified using the return routability tests,

in favour of mobility terminology in [I-D.ietf-hip-mm]. Host candidate of ICE became unreflexive locator, server reflexive candidate was mapped to relay reflexive transport locator, peer reflexive candidate was mapped to peer reflexive locator and relayed candidate became leased transport locator.

The component, base and foundation terms are not used in the document as there is only a single "media stream" for all (ESP) traffic between two hosts.

There is no "lite" version ICE in this document, just full, as the full version is the preferred one also for ICE. One specific scenario defined in this document has some resemblance to the lite ICE. When a Responder is a publicly accessible server with fixed address, it may exclude the use of the relay. In that case, it does not have to handle the RELAY parameters but still has to respond to the connectivity checks.

A connectivity check is not a STUN Binding Request. Instead, it is return routability check as defined in [I-D.ietf-hip-mm]. "Triggered check" occurs when a host receives a UPDATE with ECHO_REQUEST and it responds using a ECHO_RESPONSE and sends its own ECHO_REQUEST. A "check list" is effectively a LOCATOR parameter as defined in [I-D.ietf-hip-mm]. The term "ordinary check" is not really used in this document as it HIP packets are retransmitted periodically when the LOCATORs are in UNVERIFIED state. "Valid list" corresponds to locator pairs that have been verified successfully by the return routability tests.

The peers trigger the connectivity checks after the base exchange or after a base exchange. The conclusion of the connectivity checks, i.e., selection of the final address pair, differs the most as a result of fitting the ICE nomination algorithm to HIP mobility mechanisms. There is no "controlling agent" and the end-hosts make a

local decision on which locator pair to choose. This could lead to asymmetric address pairs, but the priority algorithm guarantees that the address pairs converge. Also, there is are no aggressive and regular nomination modes as a consequence of the lack of controlling agent.

ICE uses TLS, usernames and passwords as security mechanisms. HIP

has built-in security mechanisms that preferred over the ones that are used in ICE.

Appendix B. Document Revision History

To be removed upon publication

Revision	Comments
schmitt-00	Initial version.
ietf-00	Officially adopted as WG item. Solved issues 1-9,11,12
ietf-01	Solved remaining issues except that relaying ESP and mobility were still incomplete.
ietf-02	Miika rewrote almost from scratch based on ICE. Editorial corrections from Simon and Andrei.

Authors' Addresses

Miika Komu (editor)
Helsinki Institute for Information Technology
Metsanneidonkuja 4
Espoo
Finland

Phone: +358503841531
Fax: +35896949768
Email: miika@iki.fi
URI: http://www.hiit.fi/

Simon Schuetz
NEC Network Laboratories
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 4342 165
Fax: +49 6221 4342 155
Email: simon.schuetz@netlab.nec.de
URI: http://www.netlab.nec.de/

Martin Stiemerling
NEC Network Laboratories
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 4342 113
Fax: +49 6221 4342 155
Email: stiemerling@netlab.nec.de
URI: http://www.netlab.nec.de/

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF
Administrative Support Activity (IASA).