HIP Working Group
Internet-Draft
Intended status: Experimental
Expires: January 16, 2009

M. Komu
HIIT
T. Henderson
The Boeing Company
P. Matthews
(Unaffiliated)
H. Tschofenig
Nokia Siemens Networks
A. Keraenen, Ed.
Ericsson Research Nomadiclab
July 15, 2008

Basic HIP Extensions for Traversal of Network Address Translators
draft-ietf-hip-nat-traversal-04.txt

Status of this Memo

Abstract

The Host Identity Protocol (HIP) provides a new namespace that can be used for uniquely identifying hosts. Existing HIP experimental specifications do not specify protocol operations across Network Address Translators (NATs).

This document specifies basic NAT traversal extensions for HIP. The HIP shim layer is located between the network and transport layer, the extensions can also provide a more general-purpose NAT traversal support for higher-layer networking applications. The extensions are based on the use of the The Interactive Connectivity Establishment (ICE) methodology to discover a working path between two end-hosts.

Table of Contents

1. Introduction

HIP [RFC5201] is defined as a protocol that runs directly over IPv4 or IPv6. This approach is known to have problems traversing NATs. A detailed description of HIP problems with traversing legacy middleboxes is documented in [I-D.irtf-hiprg-nat]. 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 these types of middleboxes.

Currently deployed NAT devices do not operate consistently even though a recommended behavior is described in [RFC4787]. The HIP protocol extensions in this document make as few assumptions as possible about the behavior of the NAT devices so that NAT traversal will work even with legacy NAT devices. The purpose of these 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.

Using the extensions defined in this draft, HIP end-hosts use the HIP control channel to communicate their current locators to each other to find a operational path for the ESP encapsulated data traffic. The hosts test connectivity between different locators and try to discover direct end-to-end path between the end-hosts. However, With some legacy NATs, utilizing the shortest path between two end hosts located behind NATs is not possible without relaying the traffic through a relay, such as a TURN server [RFC5128]. As a consequence, the TURN server increases the roundtrip delay and may become a point of network congestion. With the extensions described in this document, hosts try to avoid the use the TURN server when possible.

This document defines new middlebox extensions to allow NAT traversal for HIP control plane. The HIP Relay extensions define mechanisms to forward HIP control plane. A distinction must be made here between a HIP rendezvous services [RFC5204]) and HIP Relay services. HIP

rendezvous servers solve initial contact and mobility related problems in networks without NATs. HIP Relay servers solve the same problems, in addition to NAT traversal problems. HIP Relay servers can be used both in NATed and non-NATed networks.

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 NATs can drop the packets otherwise [RFC5128].

The basis for the connectivity checks is ICE [I-D.ietf-mmusic-ice].

[I-D.ietf-mmusic-ice] describes ICE as follows:

> "The Interactive Connectivity Establishment (ICE) methodology is a technique for NAT traversal for UDP-based media streams (though ICE can be extended to handle other transport protocols, such as TCP) established by the offer/answer model. ICE is an extension to the offer/answer model, and works by including a multiplicity of IP addresses and ports in SDP offers and answers, which are then tested for connectivity by peer-to-peer connectivity checks. The IP addresses and ports included in the SDP and the connectivity checks are performed using the revised STUN specification [I-D.ietf-behave-rfc3489bis], now renamed to Session Traversal Utilities for NAT."

ICE for SIP is specified in [I-D.ietf-mmusic-ice] and ICE for non-SIP protocols is specified in [I-D.rosenberg-mmusic-ice-nonsip].

Two hosts communicate their locators to each other in the HIP base exchange. They are then paired with the locally operational address of the other endpoint and prioritized according local and recommended policies. These address sets are then tested sequentially based on the procedures specified in ICE. Both sides participate in the connectivity checks. The tests may also discover multiple operational address pairs but determine a single preferred address pair to be used for subsequent communication.

In a nutshell, the extensions in this document defines:

- encapsulatation of HIP packets in UDP
- UDP encapsulation of IPsec ESP packets
- registration extensions for HIP Relay services
- how the "offer" and "answer" are carried in the base exchange
- interaction with ICE connectivity messages
- backwards compatibility issues with rendezvous servers
- a number of optimizations (such when the ICE connectivity tests can be excluded)

2. Terminology

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].

This document borrows terminology from [RFC5201], [RFC5206], [RFC4423], [I-D.ietf-mmusic-ice], and [I-D.ietf-behave-rfc3489bis]. Additionally, the following terms are used:

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

TURN server:

A server that forwards data traffic between two end-hosts

Locator:

As defined in [RFC5206]: "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." It should noticed that "address" is used in this document as a synonym for locator.

HIP Relay:

LOCATOR (written in capital letters) denotes a HIP control message parameter that bundles multiple locators together

ICE Offer:

The Initiator's LOCATOR parameter in HIP I2 control message.

ICE Answer:

The Responder's LOCATOR parameter in HIP R2 control message

Transport address:

Transport layer port and the corresponding IPv4/v6 address

Candidate:

A transport address that has not been verified yet for reachability using ICE

Host candidate:

An IPv4 or IPv6 address of a network interface of a host

Server reflexive transport candidate:

A translated transport address of a host as observed by a HIP Relay or a STUN server

Peer reflexive transport candidate:

A translated transport address of a host as observed by its peer

Relayed transport candidate:

A transport address that exists on a TURN server. Packets that arrive at this address are relayed towards the TURN client.

3. Protocol Description

This section describes the normative behavior of the protocol extension. Examples of packet exchanges are provided for illustration purposes.

3.1. Relay Registration

HIP rendezvous servers operate in non-NATed environments and their use is described in [RFC5204]. This section specifies a new middlebox extension, called the HIP Relay, to perate in NATted

environments. HIP Relay servers forward all HIP control packets between the Initiator and the Responder over UDP.

End-hosts cannot use the HIP Relay service for forward ESP data plane. Instead, they use TURN servers [I-D.ietf-behave-turn] for relaying the ESP traffic. A HIP end-host SHOULD register to a TURN server before registering to a HIP Relay to guarantee that the host can accept ESP traffic immediately after HIP Relay registration.

A HIP Relay MUST silently drop packets to a HIP Relay Client that has not previously registered with the HIP Relay. The registration process follows the generic registration extensions defined in [RFC5203] and is illustrated in Figure 1.

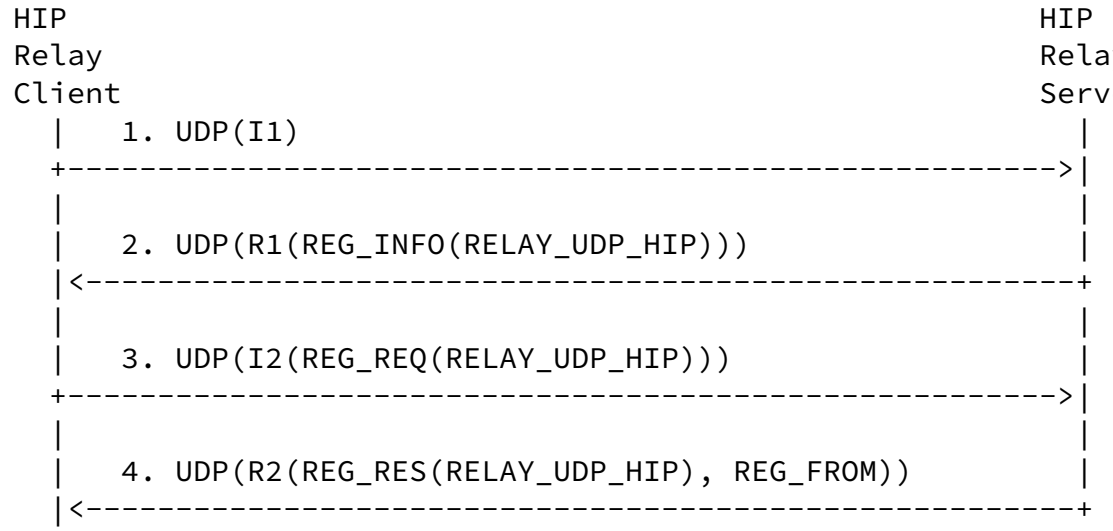


Figure 1: Example Registration to a HIP Relay

In step 1, the Relay Client (Initiator) starts the registration procedure by sending an I1 packet over UDP. It is RECOMMENDED that the Initiator selects a random port number from the ephemeral port range 49152-65535 for initiating a base exchange. However, the allocated port MUST be maintained until all of the corresponding HIP Associations are closed. Alternatively, a host MAY also use a single fixed port for initiating all outgoing connections.

In step 2, the Relay Server (Responder) lists the services that it supports in the R1 packet. The support for HIP-over-UDP relaying is denoted by the RELAY_UDP_HIP value. The R1 SHOULD not contain any NAT transform parameter.

In step 3, the Initiator selects the services it registers for and lists them in the REG_REQ parameter. In this example, the Initiator registers for HIP Relay service.

In step 4, the Responder concludes the registration procedure with an

R2 packet and acknowledges the registered services in the REG_RES parameter. The Responder denotes unsuccessful registrations in the REG_FAILED parameter in R2. The Responder also includes a REG_FROM parameter that contains the transport address of the client as observed by the Relay (Server Reflexive candidate). After the registration, the Initiator sends periodically NAT keepalives.

There are different ways for an Initiator to learn it's publicly visible IP address and port that are referred to as the "server reflexive transport candidate" in this document. It is a local decision on how the end-host learnds the candidate, but either of the following methods is RECOMMENDED:

- The Relay client may use STUN servers to detect the server reflexive locator, as described in [RFC5128].

- Alternatively, the Relay Client can learn it from the REG_FROM parameter when registering to a Relay.

3.2. NAT Transformation Negotiation

This section describes the usage of a new non-critical transform parameter type. The presence of the parameter in HIP base exchange means that the end-host supports ICE connectivity checks. As the parameter is non-critical, it can be ignored by an end-host which means that the host does not support or is not willing to use ICE connectivity checks.

The NAT transform parameter applies to a base exchange between end-hosts, but currently does not apply to with a registration with a HIP Relay server. The NAT transform applies only to a base exchange with transport layer encapsulation and MUST not be included without transport layer encapsulation. The NAT transform negotiation in base exchange is illustrated in Figure 2.

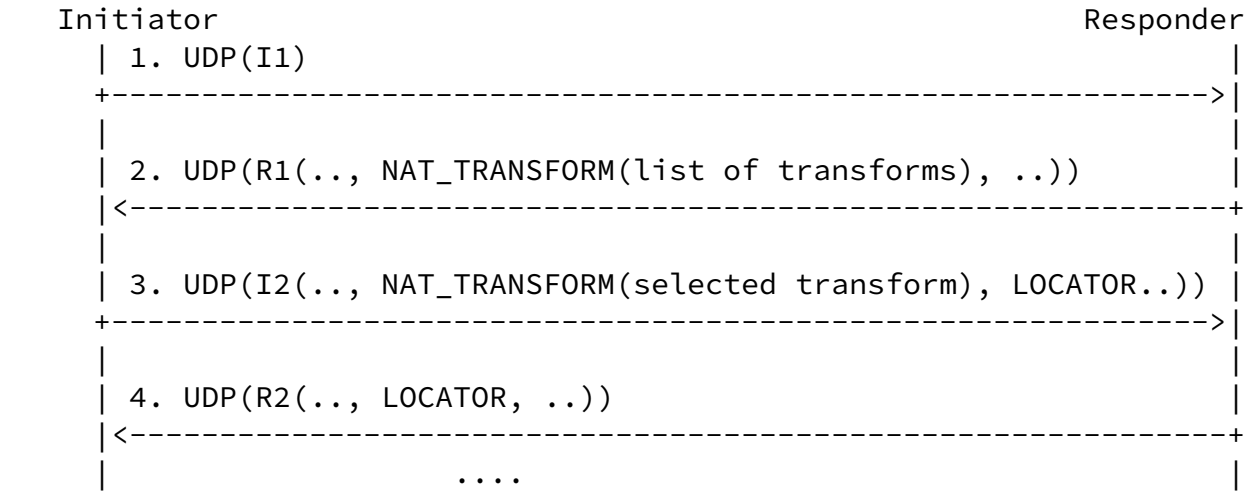


Figure 2: Negotiation of NAT Transforms

In step 1, the Initiator sends an I1 to the Responder. In step 2, the Responder responds with an R1. The R1 contains a list of transforms the Responder supports in NAT_TRANSFORM parameter as shown in Table 1.

Transform Type	Purpose
RESERVED	Reserved for future use
ICE-STUN-UDP	UDP encapsulated control and data traffic with ICE-based connectivity checks using STUN messages

Table 1: Locator Transformations

In step 3, the Initiator sends an I2 that includes a NAT_TRANSFORM parameter. It contains the transform type selected by the Initiator from the list of transforms offered by the Responder. The I2 also includes the locators of the Initiator in a LOCATOR parameter. The locator parameter in I2 is the "ICE offer".

In step 4, the Responder concludes the base exchange with an R2 packet. The Responder includes a LOCATOR parameter in the R2 packet. The locators parameter in R2 is the "ICE answer".

3.3. Base Exchange via HIP Relay

This section describes how Initiator and Responder establish a base exchange through a HIP Relay. The NAT transform negotiation (denoted

as NAT_TFM in the example) was described in previous section and shall not be repeated here. When a Relay receives an R1 or I2 packet

without the NAT transform packet, it drops it and sends a NOTIFY error message to the originator.

It is RECOMMENDED that the Initiator sends an I1 packet encapsulated in UDP when it is destined to an IPv4 address of the Responder. Respectively, the Responder MUST respond to such an 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 use the same transport layer.

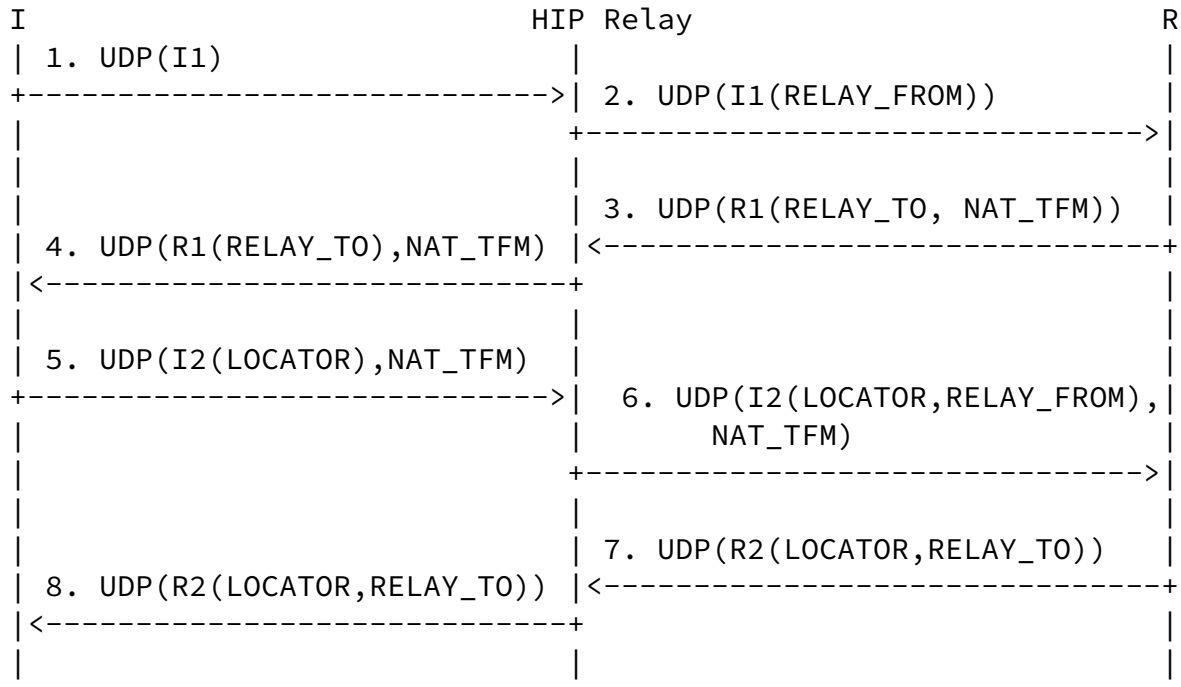


Figure 3: Base Exchange via a HIP Relay

In step 1 of Figure 3, the Initiator sends an I1 packet over the transport layer to the HIT of the Responder. The source address is one of the locators of the host. The locators belonging to the end-hosts are referred as "host candidates" in this document.

In step 2, the HIP 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

silently. The Relay appends a RELAY_FROM parameter to the I1 packet which contains the transport source address and port of the I1 as observed by the Relay. The Relay protects the I1 packet with RELAY_HMAC as described in [RFC5204], except that the parameter type is different. The Relay changes the source and destination ports and IP addresses 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 forward the packet to the Responder.

In step 3, the Responder receives the I1 packet. The Responder processes it according to the rules in [RFC5201]. In addition, the Responder validates the RELAY_HMAC according to [RFC5204] and silently drops the packet if the validation fails. The Responder replies with an R1 packet to which it includes a RELAY_TO parameter. The RELAY_TO parameter MUST contain same information as the RELAY_FROM parameter, i.e., the Initiator's transport address and the nonce, 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 drops the packet silently if the source HIT belongs to an unregistered host. The Relay MAY verify the signature of the R1 packet and drop it if the signature is invalid. Otherwise, the Relay rewrites the source address and port, and changes the destination address and port to match RELAY_TO information. Finally, the Relay recalculates transport checksum and forwards the packet.

In step 5, the Initiator receives the R1 packet and processes it according to [RFC5201]. It replies with an I2 packet that uses the destination transport address of R1 as the source address and port. The I2 contains a LOCATOR parameter that lists all the ICE candidates (ICE offer) of the Initiator. The candidates are encoded using the format defined in Section 4.5. The I2 packet MUST also contain the NAT transform parameter with ICE-STUN-UDP or some other transform selected because otherwise the Relay may drop the I2 packet.

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

In step 7, the Responder receives the I2 packet and processes it according to [RFC5201]. It replies with a R2 packet and includes a RELAY_TO parameter as explained in step three. The R2 packet includes a LOCATOR parameter that lists all the ICE candidates (ICE answer) of the Responder. 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. ICE Connectivity Checks

The Responder completes the base exchange with the R2 packet through the Relay. When Initiator successfully receives and processes the R2, both hosts have transitioned to ESTABLISHED state. However, the destination address the Initiator and Responder used for delivering

base exchange packets belonged to the Relay as indicated by the RELAY_FROM and RELAY_TO parameters. Therefore, the address of the Relay MUST not be used for sending ESP traffic unless it was listed as a TURN server in the offer/answer. Instead, the Initiator and Responder MUST start ICE connectivity tests after they have transitioned to ESTABLISHED state after the base exchange when they do not have valid locator pair for ESP traffic and the NAT transform parameter was negotiated successfully.

The ICE connectivity checks are defined in [I-D.ietf-mmusic-ice]. Section Section 4.2 defines the details of the STUN control packets. As a result of the ICE connectivity checks, ICE nominates a single transport address pair to be used if an operational address could be found. The end-hosts MUST use this address pair for the ESP traffic.

3.5. NAT Keepalives

To prevent NAT state from expiring, communicating end-hosts hosts send periodically keepalives to each other. NAT Relays MUST not send any keepalives. An end-host MUST send keepalives every 15 seconds to refresh the UDP port mapping at the NAT(s) when the control or data channel is idle. To implement failure tolerance, an end-host SHOULD have shorter keepalive period.

The keepalives are STUN Binding Indications if the hosts have agreed

on NAT_TRANSFORM during the base exchange, or HIP NOTIFY messages otherwise. A HIP Relay MUST not forward NOTIFY messages.

The communicating hosts MUST send keepalives to each other using the transport locators exchanged in the base exchange when they are in ESTABLISHED state. Also, the Initiator MUST send a NOTIFY message to the Relay to refresh the NAT state alive on the path between the Initiator and Relay when the Initiator has not received any response to its I1 or I2 from the Responder in 15 seconds. The Relay MUST not forward the NOTIFY messages.

3.6. Base Exchange without ICE Connectivity Checks

In certain network environments, the ICE connectivity checks can be omitted to reduce initial connection set up latency because base exchange acts an implicit connectivity test itself. There are three assumptions about such as environments. First, the Responder should have a long-term, fixed locator in the network. Second, the Responder should not have a HIP Relay configured for itself. Third, the Initiator can reach the Responder by simply UDP encapsulating HIP and ESP packets to the host. Detecting and configuring this particular scenario is prone to administrative failure unless carefully planned.

In such a scenario, the Initiator sends an I1 packet over UDP to the Responder. The Responder replies with a R1 packet that does not contain the NAT transform parameter. The Initiator receives the R1 packet and determines from the absence of the NAT transform and RELAY_TO parameters that ICE connectivity checks can be omitted. Finally, the hosts can start to use the locators from the concluding I2 and R2 packets of the base exchange for ESP without ICE connectivity checks.

3.7. Base Exchange without UDP Encapsulation

The Initiator MAY also try to establish a base exchange with the Responder without UDP encapsulation. In such a case, the Initiator sends first an I1 packet without UDP encapsulation to the Responder. After 100 ms, the Initiator MUST then send an UDP-encapsulated I1 packet. For retransmissions, the procedure is repeated.

The I1 packet may arrive directly at the Responder. When the

recipient is the Responder, the proceduces in [RFC5201] are followed for the rest of the base exchange. The Initiator may receive multiple R1 messages from the Responder, but upon receiving a valid R1 without UDP encapsulation, the Initiator MUST ignore further R1 messages with UDP encapsulation encapsulation. The end-hosts do not trigger ICE connectivity checks after the base exchange since the UDP encapsulation was excluded.

The packet may also arrive at a HIP-capable middlebox. When the middlebox is a HIP rendezvous server and the Responder has successfully registered to the rendezvous service, the middlebox follows rendezvous procedures in [RFC5204]. If the middlebox is a HIP Relay server, it drops the I1 packet silently.

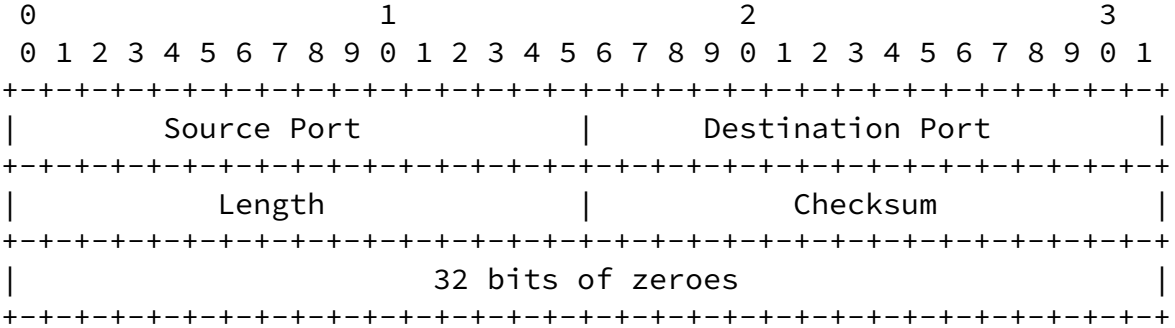
3.8. Sending Control Messages using the Data Plane

The end-hosts MAY send control messages directly to each other using the transport address pair established for data channel without sending the control packets through the Relay. When a host does not get acknowledgements e.g. to an UPDATE or CLOSE message after a timeout based on local policies, the host SHOULD resend the packet through the Relay. This optimization requires further experimentation.

4. Packet Formats

The following subsections define the parameter and packet encodings. All values MUST be in network byte order.

4.1. HIP Control Packets



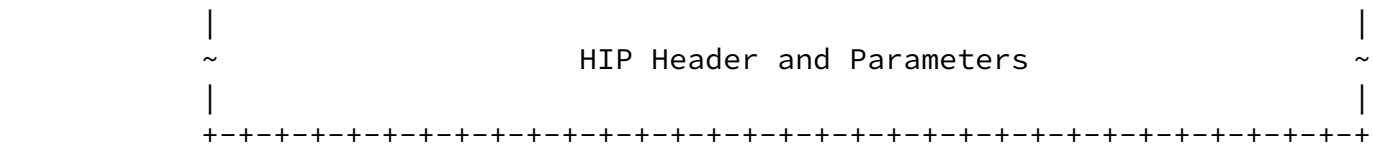


Figure 4: Format for UDP-encapsulated HIP Control Packets

HIP control packets are encapsulated in UDP packets as defined in Section 2.2 of [RFC3948], "rules for encapsulating IKE messages" except for a different port number. Figure 4 illustrates the encapsulation. The UDP header is followed by 32 zero bits that can be used to differentiate HIP control packets from ESP packets. The HIP header and parameters follow the conventions of [RFC5201] 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 [RFC5201] includes the IP addresses in the checksum calculation. The NATs unaware of HIP cannot recompute the HIP checksum after changing IP addresses.

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

4.2. Connectivity Checks

The connectivity checks are performed using STUN Binding Requests. This section describes the details of the parameters in the STUN messages.

The username is formed from the username fragments as defined in section 7.1.1.3 of [I-D.ietf-mmusic-ice]. The requests MUST use STUN short term credentials with HITs of the Initiator and Responder concatenated as a username fragment. The HITs are concatenated according to the Sort(HIT-I, HIT-R) algorithm defined in [RFC5201] section 6.5. The HIT username fragment MUST contain a UTF-8 [RFC3629] encoded sequence and MUST have been processed using SASLPrep [RFC4013] as defined section 15.3 of

[I-D.ietf-behave-rfc3489bis]. The concatenated HIT pair MUST have a fixed size that is accomplished by including the leading zeroes for the HITs.

Drawing of HIP keys is defined in [RFC5201] section 6.5 and drawing of ESP keys in [RFC5202] section 7. Correspondingly, the hosts MUST draw symmetric keys for STUN according to [RFC5201] section 6.5. The hosts draw the STUN keys after HIP keys, or after ESP keys if ESP transform was successfully negotiated in the base exchange. The hosts draw two keys which they MUST use to generate the STUN password. As the STUN password is the same at both ends, the two drawn keys MUST be concatenated with the key for the greater HIT first. Section 15.4 of [I-D.ietf-behave-rfc3489bis] describes how hosts use the password for message integrity of STUN messages.

The connectivity checks MUST contain PRIORITY attribute. They MAY contain USE-CANDIDATE attributes as defined in section 7.1.1.1 of [I-D.ietf-mmusic-ice].

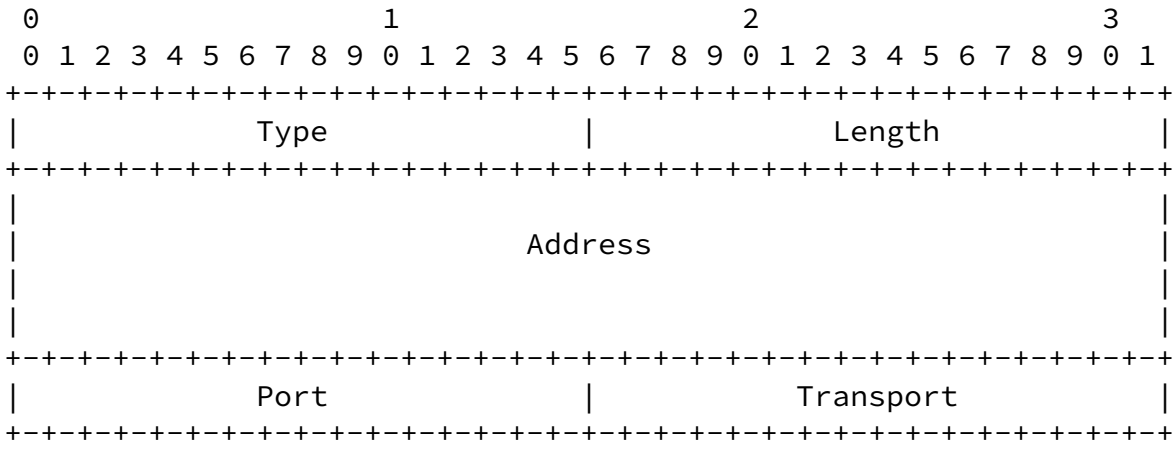
The Initiator is always in the controller role during a base exchange. Hence, the ICE-CONTROLLED and ICE-CONTROLLING attributes are not needed and SHOULD NOT be used. When two hosts are initiating to each other simultaneously, HIP state machine detects it and assigns the host with the larger HIT as the Responder as explained in sections 4.4.2 and and 6.7 in [RFC5201].

4.3. Keepalives

The keepalives for HIP associations agreed that are NAT_TRANSFORM capable are STUN Binding Indications, as defined in [I-D.ietf-behave-rfc3489bis]. The source and destination ports are in the UDP header are the same as used for HIP (50500). However, in contrast to the UDP encapsulated HIP header, there non-ESP-marker between the UDP header and the STUN header is excluded. Keepalives MUST contain the FINGERPRINT STUN attribute but SHOULD NOT contain any other STUN attributes and SHOULD NOT utilize any authentication mechanism. STUN messages are demultiplexed from ESP and HIP control messages using the STUN markers, such as the magic cookie value and the FINGERPRINT attribute.

Keepalives for aHIP associations that are not NAT_TRANSFORM capable are HIP control messages that have NOTIFY as the packet type. The NOTIFY messages do not contain any parameters.

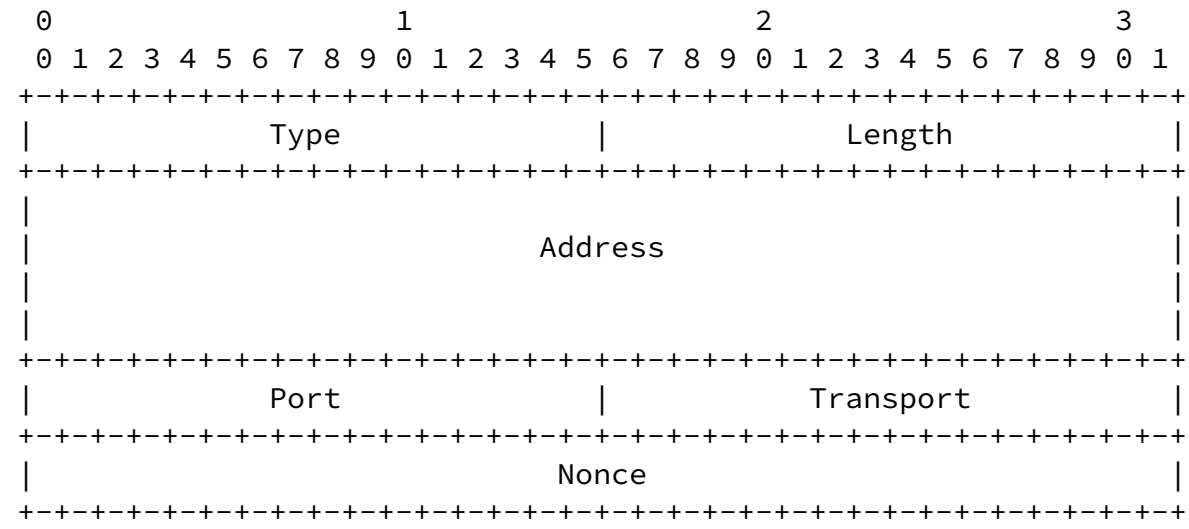
4.4. Relay and Registration Parameters



Type [TBD by IANA:
REG_FROM: (64010 = 2^16 - 2^11 + 2^9 + 10)]

Length 20
Address An IPv6 address or an IPv4 address in "IPv4-compatible IPv6 address" format
Port Transport port number; zero when plain IP is used
Transport Transport protocol type; zero for UDP

Figure 5: Format for REG_FROM parameter



Type [TBD by IANA:
Critical parameters:
RELAY_FROM: (63998 = 2^16 - 2^11 + 2^9 - 2)
RELAY_TO: (64002 = 2^16 - 2^11 + 2^9 + 2)

Length 24
Address An IPv6 address or an IPv4 address in "IPv4-compatible IPv6 address" format
Port Transport port number; zero when plain IP is used
Transport Transport protocol type; zero for UDP
Nonce A nonce assigned by the Relay server.

Figure 6: Format for the RELAY_FROM and RELAY_TO parameters

Format for the REG_FROM parameter is shown in Figure 5, and RELAY_FROM and RELAY_TO in Figure 6. Parameters are identical except for the type and nonce fields.

The nonce field is an experimental field for the RELAY_FROM and RELAY_TO parameters. It allows the Relay to have constant state towards the Initiators without allowing the Responder to send R1 or R2 packets to arbitrary hosts through the Relay.

4.5. LOCATOR Parameter

The generic LOCATOR parameter format is the same as in [RFC5206]. However, presenting ICE candidates requires a new locator type. The generic and NAT traversal specific locator parameters are illustrated in Figure 7.

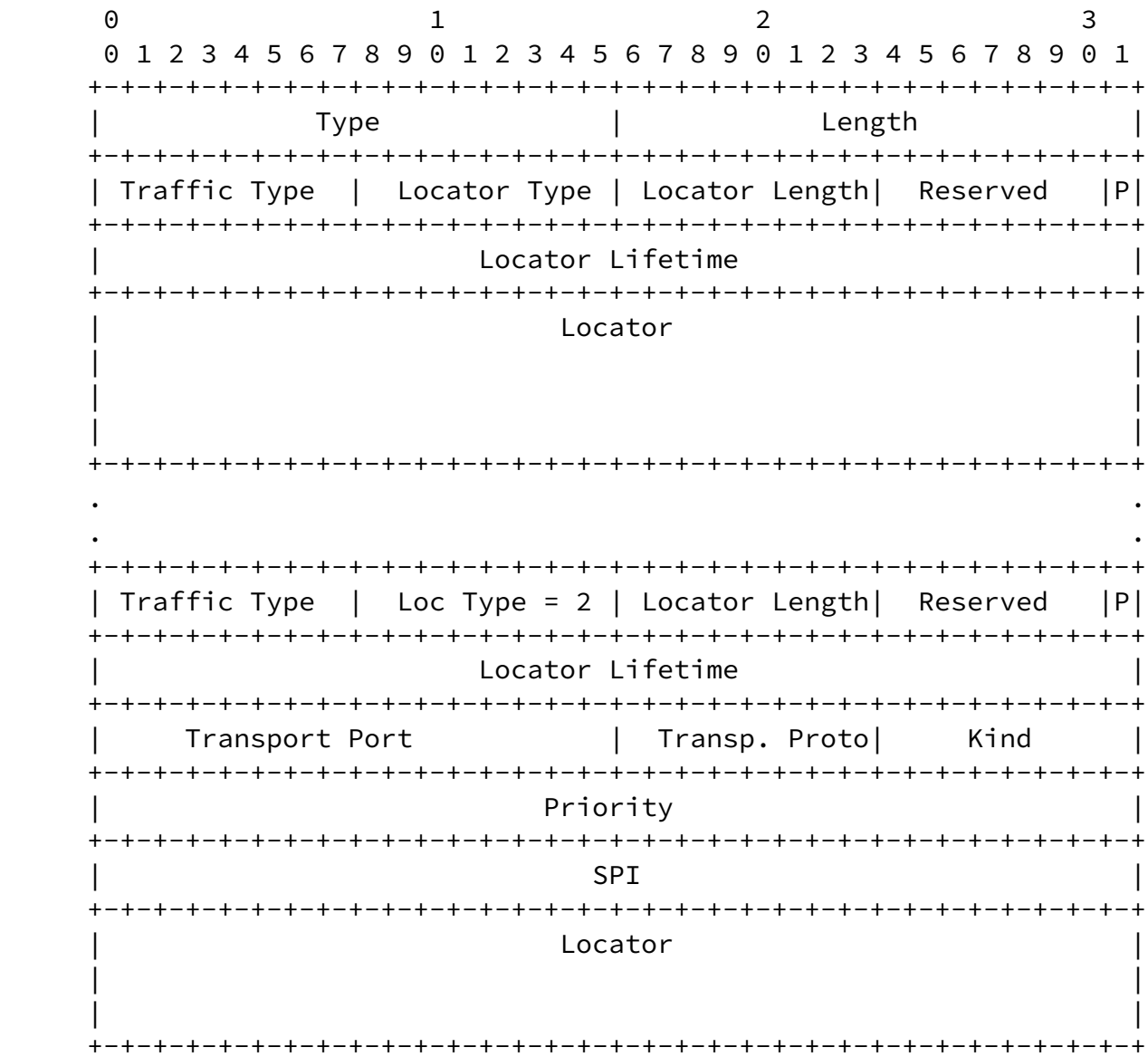


Figure 7: LOCATOR parameter

The individual fields in the LOCATOR parameter are described in Table 2.

Field	Value(s)	Purpose
Type	193	Parameter type
Length	Variable	Length in octets, excluding Type and Length fields and padding
Traffic Type	0-2	Is the locator for HIP signaling (1), for ESP (2), or for both (0)
Locator Type	2	"Transport address" locator type
Locator Length	7	Length of the Locator field in 4-octet units
Reserved	0	Reserved for future extensions
Preferred (P) bit	0	Not used for transport address locators; MUST be ignored by the receiver.
Locator Lifetime	Variable	Locator lifetime in seconds
Transport Port	Variable	Transport layer port number
Transport Protocol	0	0 for UDP
Kind	Variable	0 for host, 1 for server reflexive, 2 for peer reflexive or 3 for relayed address
Priority	Variable	Locator's priority as described in [I-D.ietf-mmusic-ice]
SPI	Variable	SPI value which the host expects to see in

Locator	Variable	incoming ESP packets that use this locator IPv6 address or an "IPv4-compatible IPv6 address" format IPv4 address [RFC3513], obfuscated by XORing it with the owner's HIT

Table 2: Fields of the LOCATOR parameter

4.6. 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 [RFC5204].

4.7. Registration Types

The REG_INFO, REQ_REQ, REG_RESP and REG_FAILED parameters contain values for HIP Relay registration. The value for RELAY_UDP_HIP is 2. The value for RELAY_UDP_ESP is 3.

4.8. ESP Data Packets

[RFC3948] describes UDP encapsulation of the IPsec ESP transport and tunnel mode. On the wire, the HIP ESP packets do not differ from the transport mode ESP and thus the encapsulation of the HIP ESP packets is same as the UDP encapsulation transport mode ESP. However, the (semantic) difference to BEET mode ESP packets used by HIP is that IP header is not used in BEET integrity protection calculation.

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 [RFC5202]. When two peers perform a UDP-encapsulated base exchange, they MUST define a pair of IPsec SAs that produces UDP-encapsulated ESP data traffic.

The management of encryption/authentication protocols and SPIs is defined in [RFC5202]. The UDP encapsulation format and processing of HIP ESP traffic is described in Section 6.1 of [RFC5202].

5. Security Considerations

5.1. Privacy Considerations

The locators are in XORed format in plain text in favor of inspection at HIP-aware middleboxes in the future. The current draft does not specify encrypted versions of LOCATORs even though it could be beneficial for privacy reasons.

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 host addresses. 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 [RFC5128]. With such a legacy NAT, the only option left would be to use a relayed transport address from a TURN server.

As a consequence, a host may support locator-based privacy by leaving out the reflexive candidates. However, the trade-off in using only host candidates can produce suboptimal paths that can congest the TURN server. The use of HIP Relays or TURN Relays can be useful for protection against Denial-of-Service attacks. If a Responder reveals only its HIP Relay addresses and Relayed transport candidates to 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.

5.2. Opportunistic Mode

A HIP Relay server should have one address per Relay Client when a HIP Relay is serving more than one Relay Clients and supports opportunistic mode. Otherwise, it cannot be guaranteed that the Relay can deliver the I1 packet to the intended recipient.

5.3. Base Exchange Replay Protection for HIP Relay Server

On certain scenarios, it is possible that an attacker, or two

attackers, can replay an earlier base exchange through a Relay server by masquerading as the original Initiator and Responder. The attack does not require the attacker(s) to compromise the private key(s) of the attacked host(s). However, Responder has to be disconnected from the Relay in order to masquarade successfully as the Responder.

The Relay can protect itself against Replay attacks by involving in the base exchange by introducing nonces that the end-hosts (Initiator and Responder) have to sign. The Relay MAY add ECHO_REQUEST_M parameters to the R1 and I2 messages as described in [I-D.heer-hip-middle-auth] and drops the I2 or R2 messages if the corresponding ECHO_RESPONSE_M parameters are not present.

5.4. Demuxing Different HIP Associations

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 HIP ESP transport format because the Responder use HITs to distinguish between different Initiators.

6. 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 Parameter Types by assigning new HIP Parameter Type values for the new HIP Parameters: RELAY_FROM, RELAY_TO and REG_FROM (defined in Section 4.4) and

RELAY_HMAC (defined in Section 4.6). NAT_TRANSFORM is also a new parameter.

7. Contributors

This draft is a product of a design team which also included Marcelo Bagnulo and Jan Melen who both have made major contributions to this document.

8. Acknowledgments

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 Andrei Gurtov, Simon Schuetz, Martin Stiemerling, Lars Eggert, Vivien Schmitt, Abhinav Pathak for their contributions and 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 and Jani Hautakorpi for their comments on this document.

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 and Birdstep.

9. References

9.1. Normative References

[I-D.ietf-behave-rfc3489bis]
Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for (NAT) (STUN)", draft-ietf-behave-rfc3489bis-16 (work in progress), July 2008.

[I-D.ietf-behave-turn]
Rosenberg, J., Mahy, R., and P. Matthews, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)",

draft-ietf-behave-turn-08 (work in progress), June 2008.

[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-19 (work in progress), October 2007.

[I-D.rosenberg-mmusic-ice-nonsip]
Rosenberg, J., "NICE: Non Session Initiation Protocol (SIP) usage of Interactive Connectivity Establishment (ICE)", draft-rosenberg-mmusic-ice-nonsip-00 (work in progress), February 2008.

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

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

[RFC3513] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC4013] Zeilenga, K., "SASLprep: Stringprep Profile for User Names and Passwords", RFC 4013, February 2005.

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

[RFC5201] Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, "Host Identity Protocol", RFC 5201, April 2008.

[RFC5202] Jokela, P., Moskowitz, R., and P. Nikander, "Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP)", RFC 5202, April 2008.

[RFC5203] Laganier, J., Koponen, T., and L. Eggert, "Host Identity Protocol (HIP) Registration Extension", RFC 5203, April 2008.

[RFC5204] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", RFC 5204, April 2008.

[RFC5206] Nikander, P., Henderson, T., Vogt, C., and J. Arkko, "End-

Host Mobility and Multihoming with the Host Identity Protocol", RFC 5206, April 2008.

9.2. Informative References

[I-D.heer-hip-middle-auth]
Heer, T., Wehrle, K., and M. Komu, "End-Host Authentication for HIP Middleboxes", draft-heer-hip-middle-auth-01 (work in progress), July 2008.

[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.

[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.

[RFC5128] Srisuresh, P., Ford, B., and D. Kegel, "State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs)", RFC 5128, March 2008.

Appendix A. IPv4-IPv6 Interoperability

Currently Relay Client and Server do not have to run any ICE connectivity tests as described in Section 3.6. However, it could be useful for IPv4-IPv6 interoperability when the Relay Server actually includes both the NAT transform parameter and multiple locators in R2. The interoperability benefit is that the Relay could support IPv4-based Initiators and IPv6-based Responders by converting the network headers and recalculating UDP checksums.

Such an approach is underspecified in this document currently. It is not yet recommended because it may consume resources at the Relay and requires also similar conversion support at the TURN relay for data

packets.

Appendix B. Base Exchange through a Rendezvous Server

This section describes handling for a scenario where Initiator looks

up the information of the Responder from DNS and discovers a RVS record [RFC5204]. In such a case, the Initiator uses its own HIP Relay to forward HIP traffic to the Rendezvous server. The Initiator will send the I1 message using the its HIP Relay server which will then forward it to the RVS server of the responder. The responder will send the R1 packet directly to the Initiator's HIP Relay server and the following I2 and R2 packets are also sent directly using the Relay.

In case the Initiator is not able to distinguish which records are RVS address records and which are Responders address records, then the Initiator SHOULD first try to contact the Responder directly and if none of the addresses is reachable it MAY try out them using its own HIP Relay as described in the above.

Appendix C. Document Revision History

To be removed upon publication

Revision	Comments
draft-ietf-nat-traversal-00	Initial version.
draft-ietf-nat-traversal-01	Draft based on RVS.
draft-ietf-nat-traversal-02	Draft based on Relay proxies and ICE concepts.
draft-ietf-nat-traversal-03	Draft based on STUN/ICE formats.
draft-ietf-nat-traversal-04	Issues 25-27,29-36

Authors' Addresses

Miika Komu

Helsinki Institute for Information Technology
Metsanneidonkuja 4
Espoo
Finland

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

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

Email: thomas.r.henderson@boeing.com

Philip Matthews
(Unaffiliated)

Email: philip_matthews@magma.ca

Hannes Tschofenig
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland

Phone: +358 (50) 4871445
Email: Hannes.Tschofenig@gmx.net
URI: http://www.tschofenig.com

Ari Keraenen (editor)
Ericsson Research Nomadiclab

Hirsalantie 11
02420 Jorvas
Finland

Phone: +358 9 2991
Email: ari.keranen@ericsson.com

Full Copyright Statement

Intellectual Property

pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

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

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