HIP Working Group V. Schmitt
Internet-Draft NEC
Expires: May 25, 2007 A. Pathak
IIT Kanpur
M. Komu
HIIT
L. Eggert
M. Stiemerling
NEC
November 21, 2006

HIP Extensions for the Traversal of Network Address Translators
draft-ietf-hip-nat-traversal-00

Status of this Memo

Copyright Notice

Abstract

This document specifies extensions to Host Identity Protocol (HIP) to support traversal of Network Address Translator (NAT) middleboxes. The traversal mechanism tunnels HIP control and data traffic over UDP and enables HIP initiators which MAY be behind NATs to contact HIP responders which MAY be behind another NAT.

Table of Contents

1. 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 e.g. mobility and multihoming in the Internet architecture.

The HIP protocol [I-D.ietf-hip-base] cannot operate across Network Address Translator (NAT) middleboxes, as described in [I-D.irtf-hiprg-nat]. Several different flavors 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. It generally uses the term NAT to refer to both types of middleboxes, unless it needs to distinguish between the two types.

Three basic cases 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 host is assumed to be in the public Internet in both cases. In the third case, both parties are located behind (different) NATs. This document describes extensions for the first case in Section 3.3, for the second case in Section 3.4 and in Section 3.5 for the third case.

The mechanisms described here also cover use of rendezvous server from NATted environments. The use rendezvous server MUST be used when the responder is behind a NAT and the rendezvous MUST be located in a public network. Chaining of NAT enabled rendezvous servers is not possible, altough there may be other kind of rendezvous servers on the path. The limitation of the described rendezvous mechanisms is that it requires NAT boxes supporting both endpoint independent mapping [I-D.srisuresh-behave-p2p-state].

The mechanisms described in this document are based on encapsulating both the control and data traffic in UDP in order to traverse NAT(s). The data traffic is assumed to be ESP. Other types of data traffic are out of scope.

The mobility and multihoming mechanisms of HIP [I-D.ietf-hip-mm], allow HIP hosts to change network location during the lifetime of a HIP association. Consequently, hosts need to start using the proposed NAT traversal mechanisms after a mobility event relocates one or both peers behind a NAT. They may also stop using the proposed mechanisms if they both relocate to the public Internet.

Finally, 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].

2. Detecting NATs

In order to know whether to use the NAT traversal mechanisms, HIP hosts need to detect presence of NAT middleboxes between them. This document does not describe any NAT detection mechanism but rather assumes the NAT is detected using some external mechanism. Hence, no special HIP parameters are required in HIP control messages to detect NATs. The NAT detection MUST occur prior to base exchange, or after node movement, prior to sending UPDATE messages.

For example, STUN [RFC3489] offers a generic mechanism using which a host behind NAT can detect the presence of NAT and type of NAT present. In STUN, the host contacts a STUN server which is located always in public network and the STUN server replies back letting the host know whether the host is behind NAT or in public network. STUN can be used to detect NATs in all but one case where both of the host are behind the same NAT. This is commonly referred as the Hairpin translation [I-D.srisuresh-behave-p2p-state] . The hairpin translation poses an unnecessary overhead in terms of UDP processing of packets and routing of packets through the NAT despite the hosts being located within the same network.

As a solution to the hairpin problem, an implementation MAY choose first to send I1 packets without UDP encapsulation and wait for the response for an implementation specific time. If the initiator does not get a reply from the responder, it then can start retransmitting I1 packets UDP encapsulated. This approach solves the hairpin problem, but incurs extra latency for the HIP connection.

3. HIP Across NATs

HIP based communications between two hosts consists effectively of HIP control traffic and ESP encrypted data traffic. Before ESP data traffic can be sent, the hosts send HIP control messages to negotiate algorithms and exchange keys. After this, the hosts can start sending encrypted ESP data traffic.

The HIP based communications defined in [I-D.ietf-hip-base] works well in public networks. However, this does not work with some legacy NATs which just drop HIP control traffic and ESP data traffic. As a solution for this, we propose UDP encapsulation of control and data traffic using a specific scheme described in this document. The

scheme also allows hosts behind NATs to act as servers.

[RFC3948] describes UDP encapsulation of IPsec ESP transport and tunnel mode. This document only describes the changes required for UDP encapsulation of BEET mode [I-D.nikander-esp-beet-mode].

3.1. Packet Formats

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

3.1.1. Control Traffic

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Source Port            |      Destination Port         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Length              |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
```

```
   ~                     HIP Header and Parameters                  ~
   |                                                                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Format for UDP-encapsulated HIP control traffic.

Figure 1 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 headers checksum is not used because it is redundant and requires the use of inner addresses (extra complexity for UDP-NAT transformations).

3.1.2. Control Channel Keep-Alives

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

3.1.3. Data Traffic

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         Source Port           |       Destination Port        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Length              |           Checksum            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                          ESP Header                           ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Format for UDP-encapsulated IPsec ESP BEET-mode traffic.

Figure 2 shows how IPsec ESP BEET-mode packets are encapsulated within UDP. Again, a minimal UDP packet carries the ESP packet in its payload. 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].

3.1.4. FROM_NAT Parameter

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type              |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                            Address                            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          UDP Port             |          Padding              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Type         [ TBD by IANA (63998 = 2^16 - 2^11 + 2^9 - 2) ]
Length       18
Address      An IPv6 address or an IPv4-in-IPv6 format IPv4 address.
UDP Port     A UDP port number
```

Figure 3: Format for FROM_NAT Parameter

Figure 3 shows FROM_NAT parameter. The use of this parameter is described in later sections.

3.1.5. VIA_RVS_NAT Parameter

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type              |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
```

```
   |                            Address                            |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          UDP Port             |            Padding            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   Type         [ TBD by IANA (64002 = 2^16 - 2^11 + 2^9 + 2) ]
   Length       16
   Address      An IPv6 address or an IPv4-in-IPv6 format IPv4 address
   UDP Port     A UDP port
```

Figure 4: Format for VIA_RVS_NAT Parameter

Figure 4 shows VIA_RVS_NAT parameter. The parameter is used for diagnostic purposes, similarly as VIA_RVS parameter in [I-D.ietf-hip-rvs]. The exact use of this parameter is described in later sections.

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

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

3.2.1. UDP Encapsulation of IPsec BEET-Mode ESP

In BEET IPsec mode, any present transport-layer checksums in the payload data are consequently 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].

The resulting BEET-mode packet is then 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 as defined in later sections. The UDP checksum MUST be calculated based on an IP header that contains the outer addresses of the SA. 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 5 illustrates the BEET-mode UDP encapsulation procedure for a TCP packet.

```
  ORIGINAL TCP PACKET:
     +-------------------------------------------+
     | inner IPv6 hdr |  ext hdrs  |     |       |
     |   with HITs    | if present | TCP | Data  |
     +-------------------------------------------+

  PACKET AFTER BEET-MODE ESP PROCESSING:
     +-----------------------------------------------------------+
     | inner IPv6 hdr | ESP | dest |     |      |  ESP    | ESP |
     |   with HITs    | hdr | opts.| TCP | Data | Trailer | ICV |
     +-----------------------------------------------------------+
                            |<------- encryption -------->|
                      |<----------- integrity ----------->|

  FINAL PACKET AFTER BEET_MODE IP HEADER PROCESSING:
     +-------------------------------------------------------------+
     | outer IPv4 | UDP | ESP | dest |     |      |  ESP    | ESP |
     |    hdr     | hdr | hdr | opts.| TCP | Data | Trailer | ICV |
     +-------------------------------------------------------------+
                              |<------- encryption -------->|
                        |<----------- integrity ----------->|
```

Figure 5: UDP Encapsulation of an IPsec BEET-mode ESP packet containing a TCP segment.

3.2.2. 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].

The NAT traversal methods described in this section are based on connection reversal and UDP hole punching similar to [I-D.ietf-behave-nat-udp]. However, the methods in this section are adapted for HIP purposes, especially the rendezvous server in mind.

3.3. Initiator Behind NAT

This section discusses mechanisms to reach a HIP responder located in publicly addressable network by a HIP initiator that is located behind a NAT. The case where the responder is using a rendezvous service is also described.

Table 1 lists some short-hand notations used in this section. For simplicity, the ports mangled by NAT are presented as example port numbers (11111 and 22222) instead of symbolic ones. In the examples, we assume that the NAT(s) timeout after I1-R1 exchange for illustration purposes, hence there are different port numbers for I2-R2 exchange.

Notation	Explanation
HIT-I	Initiator's HIT
HIT-R	Responder's HIT
IP-I	Initiator's IP address
IP-R	Responder's IP address
IP-RVS	IP address of the responder's rendezvous server
IP-NAT-I	Public IP of the NAT of the initiator
IP-NAT-R	Public IP of the NAT of the responder
UDP(50500,11111)	UDP packet with source port 50500 and destination port 11111
UDP(11111,22222)	Example port numbers mangled by a NAT
UDP(44444,22222)	Port 44444 is used throughout the examples to denote the NAT mangled source port of I2 as received by the rendezvous server during the registration

Table 1: Notations Used in This Section

3.3.1. NAT Traversal of HIP Control Traffic

This section describes the details of enabling NAT traversal for HIP control traffic for the base exchange [I-D.ietf-hip-base] through UDP encapsulation for the case when initiator of the association is located behind a NAT and responder is located in publicly addressable network. UDP-encapsulated HIP control traffic MUST use the packet formats described in Section 3.1. When sending UDP-encapsulated HIP control traffic, a HIP implementation MUST zero the HIP header checksum before calculating the UDP checksum. The receiver MUST only verify the correctness of the UDP checksum and MUST NOT verify the checksum of the HIP header.

The initiator of a UDP-encapsulated HIP base exchange MUST use the

UDP destination port 50500 for all control packets it sends. It is RECOMMENDED to use 50500 as the source port as well, but an

implementation MAY use a (randomly selected) unoccupied source port. If it uses a random source port, it MUST listen for and accept arriving HIP control/ESP Data packets on this port until the corresponding HIP association is torn down. The random source port is RECOMMENDED to be in the range of the dynamic and private ports (49152-65535). Using a random source port instead of a fixed one makes it possible to have multiple clients behind a NAT middlebox that does only address translation but no port translation. This is referred to as port overloading in [I-D.ietf-behave-nat-udp].

The responder of a UDP-encapsulated HIP base exchange MUST use 50500 as the source port for all UDP-encapsulated control packets it sends. The source address for all the packets that the responder sends MUST be the same as the IP address on which responder receives packets from initiator. The responder MUST NOT respond to any arriving UDP-encapsulated control message with an decapsulated reply. HIP implementations that implement the NAT traversal mechanisms MUST process UDP-encapsulated base exchange messages equivalently to decapsulated messages, i.e., according to [I-D.ietf-hip-base].

The remainder of this section clarifies this process through an example which is illustrated in Figure 6. It shows an initiator with the private IP address I behind a NAT. The NAT has the public IP address as NAT. The responder is located in the public Internet at the IP address R.

```
+---+          +---+                                +---+
|   |----(1)--->|   |---------------(2)-------------->|   |
|   |           | N |                                 |   |
|   |<---(4)----| A |<---------------(3)---------------|   |
| I |           | T |                                 | R |
|   |----(5)--->| - |---------------(6)-------------->|   |
|   |           | I |                                 |   |
|   |<---(8)----|   |<---------------(7)---------------|   |
+---+          +---+                                +---+

1. IP(IP-I, IP-R)       UDP(50500, 50500)  I1(HIT-I, HIT-R)
2. IP(IP-NAT-I, IP-R)   UDP(11111, 50500)  I1(HIT-I, HIT-R)
3. IP(IP-R, IP-NAT-I)   UDP(50500, 11111)  R1(HIT-R, HIT-I)
```

```
   4. IP(IP-R, IP-I)      UDP(50500, 50500)  R1(HIT-R, HIT-I)
   5. IP(IP-I, IP-R)      UDP(50500, 50500)  I2(HIT-I, HIT-R)
   6. IP(IP-NAT-I, IP-R)  UDP(22222, 50500)  I2(HIT-I, HIT-R)
   7. IP(IP-R, IP-NAT-I)  UDP(50500, 22222)  R2(HIT-R, HIT-I)
   8. IP(IP-R, IP-I)      UDP(50500, 50500)  R2(HIT-R, HIT-I)
```

Figure 6: Example of a UDP-encapsulated HIP base exchange (initiator behind a NAT, responder on the public Internet).

Before beginning the base exchange, the initiator detects that it is behind a NAT. The initiator starts the base exchange by sending a UDP-encapsulated I1 packet to the responder. According to the rules specified above, the source IP address of this I1 packet is IP-I and its source UDP port is 50500. It is addressed to IP-R on port 50500. The NAT in Figure 6 forwards the I1 but substitutes the source address IP-I with its own public address IP-NAT-I and the source UDP port 50500 with 11111.

When the responder in Figure 6 receives the UDP-encapsulated I1 packet on UDP port 50500, it processes it according to [I-D.ietf-hip-base]. The responder replies back with an R1 using the addresses and port information of I1. Thus, the R1 packet is destined to the source IP address and UDP port of the I1, i.e., IP address IP-NAT-I and port 11111. The NAT receives the I1 and substitutes the destination of this packet with the initiator address (IP-I) and port information (50500).

The initiator receives a UDP-encapsulated R1 packet from the responder and processes it according to [I-D.ietf-hip-base]. When it responds with a UDP-encapsulated I2 packet, it uses the same IP source and destination addresses and UDP source and destination ports that it used for sending the corresponding I1 packet, i.e., the packet is addressed from IP-I port 50500 to IP-R port 50500. The NAT again substitutes the source information. To illustrate timeout, the NAT chooses a different source port (22222) for the I2 than for the I1 (11111) in this case.

When a responder receives a UDP-encapsulated I2 packet destined to UDP port 50500, it MUST use the UDP source port contained in this packet for further HIP communications with the initiator. It then processes the I2 packet according to [I-D.ietf-hip-base]. When it responds with an R2 message, it UDP-encapsulates the message, using the UDP source port of the I2 packet as the destination UDP port, and sends it to the source IP address of the I2 packet, i.e., it sends the R2 packet from IP-R port 50500 to IP-NAT-I port 22222. The NAT again replaces the destination information in the R2 with IP-I port 50500

Usually, the I1-R1 and I2-R2 exchanges occur fast enough for the NAT state to persist. This means that the NAT uses the same port for the I1-R1 exchange to translate as the I2-R2 exchange. However, an implementation MUST handle even the case where the NAT state times out between the two exchanges and the I1 and I2 arrive from different UDP source ports and/or IP addresses, as shown in Figure 6.

3.3.2. NAT Traversal of HIP Data Traffic

This section describes the details of enabling NAT traversal of HIP data traffic. As described in Section 3, HIP data traffic is carried in UDP-encapsulated IPsec BEET-mode ESP packets.

3.3.2.1. IPsec BEET-Mode Security Associations

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]. As mentioned in Section 3.3.1, when two peers perform a UDP-encapsulated base exchange, they MUST define a pair of IPsec SAs that result in UDP-encapsulated BEET-mode ESP data traffic.

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

The initiator MUST use UDP destination port 50500 for all UDP-encapsulated ESP packets it sends. It MAY also use port 50500 as source port or it MAY use a random source port. If it uses a random source port, it MUST listen for and accept arriving UDP-encapsulated ESP packets on this port until the corresponding HIP association is torn down.

The responder of a UDP-encapsulated IPsec BEET-mode ESP exchange MUST use 50500 as the source port for all UDP-encapsulated ESP packets it sends. The destination port is the port from which the responder is receiving UDP encapsulated ESP data from the initiator.

Both initiator and responder of a HIP association that uses the NAT traversal mechanism as described in this draft MUST define BEET mode with UDP encapsulation as IPsec mode for SA after a successful base exchange. The inner source address MUST be local HIT used during base exchange and inner destination address MUST be HIT of the respective peer. The other parts of the SA are described in individual sections.

3.3.2.1.1. Security Associations at the Initiator

The initiator of a UDP-encapsulated base exchange defines its outbound SA as shown in Table 2

Field	Value
Outer src address	Same local IP address from which the base exchange packets were transmitted
Outer dst address	Same peer IP address to which base exchange packets were transmitted
UDP src port	Same port number as chosen for I2 packet in base exchange
UDP dst port	Port 50500

Table 2: Outbound SA at initiator

The initiator of a UDP-encapsulated base exchange defines its inbound SA as shown in Table 3

Field	Value
Outer src address	Same peer IP address to which base exchange packets were transmitted
Outer dst address	Same local IP address from which the base exchange packets were transmitted
UDP src port	Port 50500
UDP dst port	Initiator MUST use the UDP source port it uses in the outbound SA here

Table 3: Inbound SA at initiator

3.3.2.1.2. Security Associations at the Responder

The responder of a UDP-encapsulated base exchange defines its outbound SA shown in Table 4.

Field	Value
Outer src address	Same local IP address from which the base exchange packets were transmitted
Outer dst address	Peer IP address of the I2 packet received during the base exchange
UDP src port	Port 50500
UDP dst port	Source UDP port of the I2 packet received from the initiator during base exchange

Table 4: Outbound SA at Responder

Similarly, the responder of a UDP-encapsulated base exchange defines its inbound SA as shown in Table 5

Field	Value
Outer src address	Source IP address of the I2 packet received from the initiator during base exchange
Outer dst	Same local IP address from which the base exchange

address	packets were transmitted
UDP src port	Source UDP port of the I2 packet received from the initiator during base exchange
UDP dst port	Port 50500

Table 5: Inbound SA at responder

3.3.3. Use of the Rendezvous Service when only the Initiator Is Behind NAT

The rendezvous extensions for HIP without NAT traversal have been defined in [rvs]. This section addresses only the scenario where a NATted HIP node uses rendezvous service to contact another HIP node in a publicly addressable network. Figure 7 illustrates the mechanism described in this section.

A rendezvous server MUST listen on UDP port number 50500 for incoming UDP encapsulated I1 packets. However, in this specific case with only initiator behind NAT, the rendezvous server MUST not relay the I1 packets at all because the UDP hole punching does not work. Instead, the rendezvous server replies to the initiator with a NOTIFY message that includes the responder's locator in VIA_RVS parameter.

Upon receiving the NOTIFY with the locators of the responder through the NAT, the initiator MUST send an I1 to the responder. However, it MUST continue retransmissions using the RVS location. This is mandatory because NOTIFY messages are not protected with signatures and can be forged by a rogue host.

When the initiator receives an R1 through the NAT, the responder verifies the integrity of the packet and replies with an I2. The responder should be aware that the I2 may arrive from a different port than the I1. In such a case, the responder should send the R2 to the source port of I2.


```
    +----+           +----+                         +--------+   +----+
    |    |----(1)--->|    |----------------(2)-->|        |   |    |
    |    |           |    |                      | RVS R  |   |    |
    |    |<---(4)----|    |<---------------(3)---|        |   |    |
```

```
     |   |           |   |                     +--------+  |   |
     |   |           | N |                                 |   |
     |   |----(5)--->| A |---------------(6)--------------->|   |
     | I |           | T |                                 | R |
     |   |<---(8)----| - |<--------------(7)---------------|   |
     |   |           | T |                                 |   |
     |   |----(9)--->|   |---------------(10)-------------->|   |
     |   |           |   |                                 |   |
     |   |<---(11)---|   |<--------------(12)--------------|   |
     +---+           +---+                                 +---+

     1.  IP(IP-I, IP-RVS)      UDP(50500, 50500)   I1(HIT-I, HIT-R)
     2.  IP(IP-NAT-I, IP-RVS)  UDP(11111, 50500)   I1(HIT-I, HIT-R)
     3.  IP(IP-RVS, IP-NAT-I)  UDP(50500, 11111)
                                   NOTIFY(HIT-I, HIT-R, VIA_RVS(IP-R))
     4.  IP(IP-RVS, IP-I)    UDP(50500, 50500)
                                   NOTIFY(HIT-I, HIT-R, VIA_RVS(IP-R))
     5.  IP(IP-I, IP-R)        UDP(50500, 50500)   I1(HIT-I, HIT-R)
     6.  IP(IP-NAT-I, IP-R)    UDP(22222, 50500)   I1(HIT-I, HIT-R)
     7.  IP(IP-R, IP-NAT-I)    UDP(50500, 22222)   R1(HIT-R, HIT-I)
     8.  IP(IP-R, IP-I)        UDP(50500, 50500)   R1(HIT-R, HIT-I)
     9.  IP(IP-I, IP-R)        UDP(50500, 50500)   I2(HIT-I, HIT-R)
     10. IP(IP-NAT-I, IP-R)    UDP(33333, 50500)   I2(HIT-I, HIT-R)
     11. IP(IP-R, IP-NAT-I)    UDP(50500, 33333)   R2(HIT-R, HIT-I)
     12. IP(IP-R, IP-I)        UDP(50500, 50500)   R2(HIT-R, HIT-I)
```

Figure 7: Example of a UDP-encapsulated HIP base exchange via RVS (initiator behind a NAT, responder and RVS on the public Internet).

3.4. Responder Behind NAT

This section discusses mechanisms to reach a HIP responder that is located behind a NAT. This section assumes that the initiator is located on publicly addressable network. The initiator contacts the responder through an RVS server.

3.4.1. Rendezvous Client Registration From Behind NAT

The rendezvous client registration [rvs] describes the case when rendezvous client is present in publicly addressable network. This section defines an extension to the rendezvous client registration for the case when the rendezvous client has detected that it is behind a NAT. The process in the NAT case is identical to the case without NAT, except that UDP encapsulation is used. The registration is illustrated in Figure 8.

A node behind a NAT MUST first register to the RVS when it is going to act as a responder for some other nodes. The node (i.e. rendezvous client) performs a base exchange with the RVS over UDP as described in Section 3.3 by sending I1 UDP encapsulated and 50500 as destination port number. RVS sends REG_INFO parameter in R1 to which rendezvous client replies with REG_REQ in I2. Both I1 and R1 are sent using UDP. If RVS grants service to the rendezvous client, it MUST store the source IP address and source port number of the I2 UDP packet that it had received from the rendezvous client during base exchange. The source IP address belongs to the NAT and the source port number is the NAT mangled port. RVS then replies with REG_RESP in R2 over UDP. If the registration process results in a successful REG_RESP, the rendezvous client MUST send NAT keepalives (Section 3.1.2) to keep the mapping in the NAT with the RVS open. The NAT keepalives sent from rendezvous client to the RVS MUST have the same source port as the I2 packet.

When the RVS receives an I1 packet from a HIP node to be relayed to the successfully registered rendezvous client behind NAT, RVS MUST relay the I1 over UDP with the destination port as the one stored during registration. The RVS also zeroes the HIP header checksum of the I1. This process is explained in Section 3.4.2.

```
+----+          +----+                                  +----+
|    |----(1)--->|    |---------------(2)--------------->|    |
|    |          | N  |                                  |    |
|    |<---(4)----| A  |<---------------(3)---------------|    |
| I  |          | T  |                                  | R  |
|    |----(5)--->| -  |---------------(6)--------------->|    |
|    |          | I  |                                  |    |
|    |<---(8)----|    |<---------------(7)---------------|    |
+----+          +----+                                  +----+

Initiator = Rendezvous client, Responder = Rendezvous server

1. IP(IP-I, IP-R)     UDP(50500, 50500) I1(HIT-I, HIT-R)
2. IP(IP-NAT-I, IP-R) UDP(33333, 50500) I1(HIT-I, HIT-R)
3. IP(IP-R, IP-NAT-I) UDP(50500, 33333)
      R1(HIT-R, HIT-I, REG_INFO)
4. IP(IP-R, IP-I)     UDP(50500, 50500)
      R1(HIT-R, HIT-I, REG_INFO)
5. IP(IP-I, IP-R)     UDP(50500, 50500)
      I2(HIT-I, HIT-R, REG_REQ)
6. IP(IP-NAT-I, IP-R) UDP(44444, 50500)
      I2(HIT-I, HIT-R, REG_REQ)
7. IP(IP-R, IP-NAT-I) UDP(50500, 44444)
      R2(HIT-R, HIT-I, REG_RES)
8. IP(IP-R, IP-I)     UDP(50500, 50500)
      R2(HIT-R, HIT-I, REG_RES)
```

Figure 8: Rendezvous NAT Client Registration

3.4.2. NAT Traversal of HIP Control Traffic

This section describes the details of enabling NAT traversal for base exchange packets [I-D.ietf-hip-base] through UDP encapsulation, for the case when the HIP initiator is on publicly addressable network and the HIP responder is behind NAT. The process is illustrated in Figure 9.

Before the HIP base exchange starts, the responder of the HIP base exchange MUST have completed a successful rendezvous client registration using the scheme defined in Section 3.4.1.

The initiator of the HIP base exchange sends a plain I1 packet (without UDP encapsulation) to the RVS as described in [rvs]. The RVS relays the inbound I1 packet to the registered rendezvous client. In this case, the incoming I1 is not UDP encapsulated, but the rendezvous client has registered using UDP.

To relay the I1 packet, RVS SHOULD zero the HIP header checksum from

the I1 packet. RVS must add a FROM parameter, as described in [rvs], which contains the IP address of HIP initiator. The FROM parameter is integrity protected by a RVS_HMAC as described in [rvs]. RVS replaces the destination IP address in the IP header of the packet with IP that it had stored during the rendezvous client registration (which is the IP address of the outermost NAT behind which rendezvous client is located). It MUST then encapsulate the I1 packet within UDP. The source port in the UDP header MUST be 50500 and the destination port MUST be the same as the source port number (44444) of the I2 packet which it had stored during the registration process. RVS then recomputes the IP header checksum and sends the packet.

```
                     +-------+
                     |       |
              +----->|  RVS  +-----+           +----+
+---+         |      |       |     |           |    |          +---+
|   |---(1)---+      +-------+     +----(2)--->|    |---(3)--->|   |
|   |                                          | N  |          |   |
|   |<------------------(5)--------------------| A  |<--(4)----|   |
| I |                                          | T  |          | R |
|   |-------------------(6)------------------->| -  |---(7)--->|   |
|   |                                          | R  |          |   |
|   |<------------------(9)--------------------|    |<--(8)----|   |
+---+                                          +----+          +---+
```

1. IP(IP-I, IP-RVS)   I1(HIT-I, HIT-R)
2. IP(IP-RVS, IP-NAT-R) UDP(50500, 44444)
      I1(HIT-I, HIT-R, FROM:IP-I, RVS_HMAC)
3. IP(IP-RVS, IP-R)   UDP(50500, 50500)
      I1(HIT-I, HIT-R, FROM:IP-I, RVS_HMAC)
4. IP(IP-R, IP-I)
      UDP(50500, 50500) R1(HIT-R, HIT-I, VIA_RVS_NAT(RVS-IP, 50500))
5. IP(IP-NAT-R, IP-I)

```
      UDP(44444,   50500) R1(HIT-R, HIT-I, VIA_RVS_NAT(RVS-IP, 50500)
   6. IP(IP-I, IP-NAT-R)   UDP(50500, 44444)   I2(HIT-I, HIT-R)
   7. IP(IP-I, IP-R)       UDP(50500, 50500) I2(HIT-I, HIT-R)
   8. IP(IP-R, IP-I)       UDP(50500, 50500) R2(HIT-R, HIT-I)
   9. IP(IP-NAT-R, IP-I)   UDP(44444, 50500) R2(HIT-R, HIT-I)
```

Figure 9: UDP-encapsulated HIP base exchange (initiator on public Internet, responder behind a NAT).

The relayed I1 packet travels from RVS to the NAT. The NAT changes the destination IP address of the UDP encapsulated I1 packet, and the destination port number in the UDP header. The responder accepts the packet from the RVS and processes it according to [rvs]. The resulting R1 must be encapsulated within UDP. The responder MAY append a VIA_RVS_NAT parameter to the message, which contains the IP

address of the rendezvous and the port the rendezvous used for relaying the R1. The RECOMMENDED source port is 50500 and the destination port number MUST be 50500. The destination address in the IP header MUST be the same as the one specified in the FROM parameter of the relayed I1 packet.

The initiator MUST listen on port 50500 and it receives the UDP encapsulated R1. After verifying the HIP packet, it concludes that the responder is behind a NAT because the packet was UDP encapsulated. The initiator processes the R1 control packet according to [I-D.ietf-hip-base] and replies using I2 that is UDP encapsulated. The addresses and ports are derived from the received R1.

The NAT translates and forwards the UDP encapsulated I2 packet to the responder. The resulting R2 packet is also UDP encapsulated using the address and port information from the received I2 packet.

3.4.3. NAT Traversal of HIP Data Traffic

After a successful base exchange, both of the HIP nodes have all the parameters with them needed to establish UDP BEET mode Security Association. The following section describe inbound and outbound security associations at initiator and responder.

3.4.3.1. Security Associations at the Initiator

The initiator of a base exchange defines its outbound SA as shown in Table 6

Field	Value
Outer src address	Same local IP address from which the base exchange packets were transmitted
Outer dst address	Same peer IP address from which R2 packet was received during base exchange
UDP src port	Port 50500
UDP dst port	Source port of incoming R2 packet during base exchange

Table 6: Outbound SA at initiator

The initiator of a base exchange defines its inbound SA as shown in Table 7

Field	Value
Outer src address	Same peer IP address from which R2 packet was received during base exchange
Outer dst address	Same local IP address from which the base exchange packets were transmitted
UDP src port	Source port of incoming R2 packet during base exchange
UDP dst port	Port 50500

Table 7: Inbound SA at initiator

3.4.3.2. Security Associations at the Responder

The responder of a UDP-encapsulated base exchange defines its outbound SA shown in Table 8.

Field	Value
Outer src address	Same local IP address from which the base exchange packets were transmitted
Outer dst address	Same peer IP as that used during base exchange
UDP src port	Same as source port chosen during base exchange
UDP dst port	Port 50500

Table 8: Outbound SA at Responder

Similarly, the responder of a UDP-encapsulated base exchange defines its inbound SA as shown in Table 9

Field	Value
Outer src address	Source peer IP address as used in base exchange
Outer dst address	Same local IP address from which the base exchange packets were transmitted
UDP src port	Port 50500
UDP dst port	Same as source port chosen during base exchange

Table 9: Inbound SA at responder

3.5. Both Hosts Behind NAT

This section describes the details of enabling NAT traversal for HIP control and ESP data traffic, such as the base exchange [I-D.ietf-hip-base], through UDP encapsulation, for the case when the HIP initiator and the HIP responder are both behind two separate NATs. The described mechanism applies also when the hosts are behind the same NAT but may result in inefficient routing paths, unless the countermeasures described in section Section 2 are followed. The limitation of this approach is that it requires that the NAT boxes support endpoint independent mapping [I-D.srisuresh-behave-p2p-state].

The registration and rendezvous relay are handled similarly as described in Section 3.3.3 and Section 3.4.1. Now that both hosts are behind NATs, both the initiator (Section 3.3) and responder (Section 3.4) mechanisms are combined here.

3.5.1. NAT Traversal of HIP Control Traffic

This section describes traversal mechanism for HIP control traffic in the situation when both the initiator and the responder are behind NATs. Both hosts MUST first detect using external mechanism that they are located behind NAT. The RVS MUST be located on publicly addressable network. Before initiator begins the base exchange, the responder MUST have completed a successful rendezvous client registration with the RVS using the mechanism described in Section 3.4.1.

Initiator of the HIP base exchange starts the base exchange by sending an UDP encapsulated I1 packet to RVS. The UDP packet MUST have destination port number 50500 and initiator is RECOMMENDED to use 50500 as source port number. RVS MUST listen on UDP port 50500. RVS MUST accept the packet as described in Section 3.3.3. As there has been a successful rendezvous client registration between the responder and the RVS as described in Section 3.4.1, the RVS knows the port number which it can use to communicate with the responder through the NAT. RVS MUST add a FROM_NAT parameter to the I1 packet. The FROM_NAT parameter contains the source address of the I1 packet, which is effectively the address of the outermost NAT of the initiator. The RVS copies the source port of the UDP encapsulated I1 packet into the port number field of the FROM_NAT parameter. The FROM_NAT parameter is integrity protected by an RVS_HMAC as described in [rvs]. It MUST replace the destination IP address of the I1 packet by the one it had stored earlier during rendezvous client registration. It MUST replace source IP address of I1 packet with its own address. UDP source port of the relayed I1 packet MUST be 50500 and destination port MUST be the same as one it had stored

during the client rendezvous registration. It MUST recompute the IP header checksum.

In this case, in which the I1 was UDP encapsulated and the rendezvous client is also behind a NAT, the rendezvous server sends two packets.

First, it MUST relay the I1 packet to the responder (rendezvous client) using UDP. Second, it MUST send the locator and port (as observed by the rendezvous) of the responder in a VIA_RVS_NAT parameter in a NOTIFY packet to the inititiator. However, this will actually launch two parallel base exchanges. In the first case, the initiator receives the NOTIFY message, and acts on it as described in section Section 3.3.3, i.e., it sends an I1 directly to the address in the VIA_RVS_NAT parameter and continues to retransmit packet through the RVS. In the second case, the responder will receive the I1 relayed by the rendezvous. The responder acts as described in section Section 3.4.2 by replying with an R1.

This scheme launches two parallel exchanges, one of which is phased later than the other. Although this kind of operation is not usually very desirable, it is essential to guarantee successful NAT hole punching. The base exchange has been designed to handle simultaneous base exchanges and the race between the two parallel base exchange eventually terminates after initiator is in established state.

```
+---+        +----+         +-------+            +----+        +---+
|   |--(1)-->|    |---(2)-->+       |            |    |        |   |
|   |        |    |         | RVS R |            |    |        |   |
|   |        |    |<--(3a)--+       |---(3b)---->|    |        |   |
|   |        | N  |         +-------+            | N  |        |   |
|   |<-(4a)--| A  |                              | A  |--(4b)->|   |
| I |        | T  |                              | T  |        | R |
|   |--(5a)->| -  |                              | -  |<-(5b)--|   |
|   |        | I  |<-(6b)------------------(6a)->| R  |        |   |
|   |        |    |                              |    |        |   |
....................................................................
+---+        +----+                              +----+        +---+

1.   IP(IP-I, IP-RVS)       UDP(50500, 50500)  I1(HIT-I, HIT-R)
2.   IP(IP-NAT-I, IP-RVS)   UDP(11111, 50500)  I1(HIT-I, HIT-R)

3a.  IP(IP-RVS, IP-NAT-I)   UDP(50500, 11111)
        NOTIFY(HIT-R, HIT-I, VIA_RVS_NAT(IP-NAT-R, 44444)
3b.  IP(IP-RVS, IP-NAT-R)   UDP(50500, 44444)
        I1(HIT-I, HIT-R, FROM_NAT:[IP-NAT-I,11111], RVS_HMAC)

4a.  IP(IP-RVS-R, IP-I)     UDP(50500, 50500)
        NOTIFY(HIT-R, HIT-I, VIA_RVS_NAT(IP-NAT-R, 44444)
4b.  IP(IP-RVS, IP-R)       UDP(50500, 50500)
        I1(HIT-I, HIT-R, FROM_NAT:[NAT-I,11111], RVS_HMAC)

5a.  IP(IP-I, IP-NAT-R)     UDP(50500, 44444) I1(HIT-I, HIT-R)
5b.  IP(IP-R, IP-NAT-I)     UDP(50500, 11111)
        R1(HIT-R, HIT-I, VIA_RVS_NAT(RVS-IP, 50500))

6a.  IP(IP-NAT-I, IP-NAT-R) UDP(11111, 44444) I1(HIT-I, HIT-R)
6b.  IP(IP-NAT-R, IP-NAT-I) UDP(44444, 11111)
        R1(HIT-R, HIT-I, VIA_RVS_NAT(RVS-IP, 50500))
```

Figure 10: UDP-encapsulated HIP base exchange (initiator and responder behind a NAT, RVS on public IP).

3.5.2. NAT Traversal of HIP Data Traffic

After a successful base exchange, both the HIP nodes have all the parameters with them to establish UDP BEET mode Security Association. The following section describes inbound and outbound security associations at initiator and responder.

3.5.2.1. Security Associations at the Initiator

The initiator of a base exchange defines its outbound SA as shown in Table 10

Field	Value
Outer src address	Same local IP address from which the base exchange packets were transmitted
Outer dst address	Same peer IP address from which R2 packet was received during base exchange
UDP src port	Same as the port number chosen to send I2 during base exchange
UDP dst port	Source port of incoming R2 packet during base exchange

Table 10: Outbound SA at initiator

The initiator of a base exchange defines its inbound SA as shown in Table 11

Field	Value
Outer src address	Same peer IP address from which R2 packet was received during base exchange
Outer dst address	Same local IP address from which the base exchange packets were transmitted
UDP src port	Source port of incoming R2 packet during base exchange
UDP dst port	Same as the port number chosen to send I2 during base exchange

Table 11: Inbound SA at initiator

3.5.2.2. Security Associations at the Responder

The responder of a UDP-encapsulated base exchange defines its outbound SA shown in Table 12.

Field	Value
Outer src	Same local IP address from which the base exchange

address	packets were transmitted
Outer dst address	Same peer IP as that used during base exchange
UDP src port	Same as source port chosen send R2 during base exchange

UDP dst port	Same as source port number of I2 packet during base exchange

Table 12: Outbound SA at Responder

Similarly, the responder of a UDP-encapsulated base exchange defines its inbound SA as shown in Table 13

Field	Value
Outer src address	Source peer IP address as used in base exchange
Outer dst address	Same local IP address from which the base exchange packets were transmitted
UDP src port	Same as source Port received from I2 during base exchange
UDP dst port	Same as source port used to send R2 during base exchange

Table 13: Inbound SA at responder

3.6. NAT Keep-Alives

Typically, NATs cache an established binding and time it out if they have not used it to relay traffic for a given period of time. This timeout is different for different NAT implementations. The BEHAVE working group is discussing recommendations for standardized timeout values. To prevent NAT bindings that support the traversal of UDP-encapsulated HIP traffic from timing out during times when there is no control or data traffic, HIP hosts SHOULD send periodic keep-alive messages.

Typically, only outgoing traffic acts refreshes the NAT port state for security reasons. Consequently, both hosts SHOULD send periodic keep-alives for the UDP channel of all their established HIP associations if the channel has been idle for a specific period of time.

For the UDP channel, keep-alives MUST be UDP-encapsulated HIP UPDATE packets as defined in Section 3.1.2. The packets MUST use the same source and destination ports and IP addresses as the corresponding UDP tunnel. The default keep-alive interval for control channels MUST be 20 seconds. The responder of the HIP association should just discard the keep-alives.

3.7. HIP Mobility

After a successful base exchange, either host can change its network location using the mechanisms defined in [I-D.ietf-hip-mm]. This section describes such mobility mechanisms in the presence of NATs. However, double jump scenario, where both hosts move simultaneously, is excluded.

The mobile node can change its location as described in Table 14.

No	From network	To network
1	Behind NAT	Publicly Addressable Network
2	Publicly Addressable Network	Behind NAT
3	Behind NAT-A	Stays behind NAT-A, but different IP
4	Behind NAT-A	Behind NAT-B
5	Publicly Addressable Network	Publicly Addressable Network

Table 14: End host mobility scenarios

The corresponding peer node can be located as follows Table 15

No	Peer Node network
A	Publicly Addressable Network With RVS
B	Publicly Addressable Network Without RVS
C	Behind NAT With RVS
D	Behind NAT Without RVS

Table 15: Peer host Network Scenarios

The NAT traversal mechanisms may not work when the corresponding node is behind a NAT without RVS (case D), except when the mobile node stays behind the same cone NAT (case 3D).

When a host changes its location, it SHOULD detect the presence of NATs along the new paths to its peers using some external mechanism before sending any UPDATE messages. Alternatively, it MAY use some heuristics to conclude that it is behind a NAT rather than incur the latency of running NAT detection first.

The mobile node MUST send the UPDATE packet through the corresponding node's RVS if it has one, in addition to sending it to the corresponding node directly. The mobile node encapsulates the UPDATE packet within UDP only when it is behind a NAT. The corresponding node MUST reply using UDP when the packet was encapsulated within UDP, or without UDP when the UDP header was not present in the UPDATE packet.

The rendezvous server UPDATE relaying process is similar to I1. The rendezvous server MUST add FROM parameter when it gets a UPDATE packet without UDP encapsulation, or a FROM_NAT parameter when the UPDATE packet it receives is UDP encapsulated and MUST protect the packet with HMACs. Upon replying to the UPDATE, the corresponding node MUST add a VIA_RVS (or VIA_RVS_NAT) parameter to the reply.

When the UDP encapsulation for NAT traversal is used, private IP addresses should be filtered out from the LOCATOR parameter in the HIP control packets. Exposing private addresses may impose privacy related problems.

3.8. HIP Multihoming

Multiple security associations can exists between the same hosts. They may be connected through several paths, some of which may include a NAT and others may not. Implementations that support multihoming MUST support concurrent HIP associations between the same host pair in a way that allows some of them to use UDP encapsulation while others use basic HIP. Implementations MAY distinguish HIP associations based on the SPI instead of a HIT pair for this purpose.

3.9. Firewall Traversal

When the initiator or the responder of a HIP association is behind a firewall, additional issues arise.

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 destination port 50500 from arbitrary source ports and to receive UDP response traffic from that port to the chosen source port.

Most firewall implementations support "UDP connection tracking", i.e., after a host behind a firewall has initiated a 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.

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 port 50500 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 receive UDP traffic on port 50500 from arbitrary source IP addresses and ports.

The NAT traversal mechanisms described in Section 3 require that the firewall - stateful or not - allow inbound UDP traffic to port 50500 and allow outbound UDP traffic to arbitrary UDP ports. If necessary for firewall traversal, ports reserved for IKE MAY be used for initiating new connections, but the implementation MUST be able to listen for UDP packets from port 50500.

4. Security Considerations

Section 5.1 of [RFC3948] describes a security issue for the UDP encapsulation of 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 the 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 the HITs to distinguish between different communication instances.

The rendezvous usage in this draft has been designed to follow the design of the RVS draft [I-D.ietf-hip-rvs] and only I1 relayed. However, as NAT networking presents some additional challenges, it is not possible two follow the RVS design exactly. Particularly, the mechanisms described in Figure 7 and Section 3.5.1 require that the rendezvous server replies back to the initiator with a message which includes the address and port of the responder NAT. Another design choice would have been to relay also the R1 (and I2 in case of both hosts behind NAT) through the rendezvous server to delay the exposure of the responder NAT address and port related information for additional DoS protection. However, this choice was not selected to reduce round trip time. As a consequence, the renzvous client must be accept the risk of lowered privacy protection when it registers to the RVS over UDP as defined in section Figure 8.

5. 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" range, i.e., 50500. Upon publication of this document, IANA is requested to register two UDP ports and the RFC editor is requested to change all occurrences of port 50500 to the port IANA has registered.

6. Acknowledgements

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 and Jukka Ylitalo for their comments on this document.

[I-D.nikander-hip-path] presented some initial ideas for NAT traversal of HIP communication. This document describes significantly different mechanisms that, among other differences, use external NAT discovery and do not require encapsulation servers.

Lars Eggert 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 for InfraHIP research group at Helsinki Institute for Information Technology (HIIT). The InfraHIP project is funded by Tekes, Elisa, Nokia, The Finnish Defence Forces and Ericsson.

7. References

7.1. Normative References

[I-D.ietf-hip-base]
Moskowitz, R., "Host Identity Protocol", draft-ietf-hip-base-05 (work in progress), March 2006.

[I-D.ietf-hip-esp]
Jokela, P., "Using ESP transport format with HIP", draft-ietf-hip-esp-02 (work in progress), March 2006.

[I-D.ietf-hip-mm]
Nikander, P., "End-Host Mobility and Multihoming with the Host Identity Protocol", draft-ietf-hip-mm-03 (work in progress), March 2006.

[I-D.ietf-hip-rvs]
Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", draft-ietf-hip-rvs-04 (work in progress), October 2005.

[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-05 (work in progress), February 2006.

[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.

[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.

[rvs] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension".

7.2. Informative References

[I-D.ietf-behave-nat-udp]
Audet, F. and C. Jennings, "NAT Behavioral Requirements for Unicast UDP", draft-ietf-behave-nat-udp-07 (work in progress), June 2006.

[I-D.irtf-hiprg-nat]
Stiemerling, M., "NAT and Firewall Traversal Issues of Host Identity Protocol (HIP) Communication", draft-irtf-hiprg-nat-02 (work in progress), May 2006.

[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.srisuresh-behave-p2p-state]
Srisuresh, P., "State of Peer-to-Peer(P2P) Communication Across Network Address Translators(NATs)", draft-srisuresh-behave-p2p-state-03 (work in progress), June 2006.

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

[RFC3489] Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489, March 2003.

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

Appendix A. 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

Authors' Addresses

Vivien Schmitt
NEC Network Laboratories
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 90511 0
Fax: +49 6221 90511 55

Email: schmitt@netlab.nec.de
URI: http://www.netlab.nec.de/

Abhinav Pathak
IIT Kanpur
B204, Hall - 1, IIT Kanpur
Kanpur 208016
India

Phone: +91 9336 20 1002
Email: abhinav.pathak@hiit.fi
URI: http://www.iitk.ac.in/

Miika Komu
Helsinki Institute for Information Technology
Tammasaarenkatu 3
Helsinki
Finland

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

Lars Eggert
NEC Network Laboratories
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 90511 43
Fax: +49 6221 90511 55
Email: lars.eggert@netlab.nec.de
URI: http://www.netlab.nec.de/

Martin Stiemerling

NEC Network Laboratories
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 90511 13
Fax: +49 6221 90511 55
Email: stiemerling@netlab.nec.de
URI: http://www.netlab.nec.de/

Full Copyright Statement

Intellectual Property

Acknowledgment

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