Host Identity Protocol (HIP)
Internet-Draft
Expires: January 17, 2005

L. Eggert
M. Liebsch
NEC

July 19, 2004

Design Aspects of Host Identity Protocol (HIP) Rendezvous Mechanisms
draft-eggert-hip-rendezvous-01

Status of this Memo

Copyright Notice

Abstract

This document discusses design aspects of rendezvous mechanisms for

the Host Identity Protocol (HIP). Rendezvous mechanisms, such as HIP rendezvous servers, improve operation when HIP nodes are multi-homed or mobile. They can also facilitate communication between HIP and non-HIP nodes. Possible rendezvous mechanisms differ in performance, compatibility and impact on the HIP and Internet architectures.

Table of Contents

1. Introduction

The current Internet uses two global namespaces: domain names and IP addresses. The Domain Name System (DNS) provides a two-way lookup service between the two [1]. Domain names are symbolic identifiers for sets of IP addresses.

IP addresses have two uses. First, they are topological locators for network attachment points. Second, they act as names for the attached network interfaces. Saltzer [13] discusses these naming concepts in detail.

Routing and other network-layer mechanisms are based on the locator aspects of IP addresses. Transport-layer protocols and mechanisms typically use IP addresses in their role as names for communication endpoints.

This dual use of IP addresses limits the flexibility of the Internet architecture. The need to avoid readdressing in order to maintain existing transport-layer connections complicates advanced functionality, such as mobility, multi-homing or network composition. Sunshine summarizes the consequences of addressing on advanced network functions [14].

The Host Identity Protocol (HIP) architecture [2] defines a new third namespace. The host identity namespace decouples the name and locator roles currently filled by IP addresses. Instead of mapping domain names directly into IP addresses, HIP maps domain names into host identities, and host identities into IP addresses. Transport-layer mechanisms operate on host identities instead of using IP addresses as endpoint names. Network-layer mechanisms continue to use IP addresses as pure locators.

Without HIP, nodes establish transport-layer connections by first looking up the fully-qualified domain name (FQDN) of a peer in the DNS. A successful DNS lookup returns the peer's IP addresses. A node uses one of the returned IP addresses to initiate transport-layer communication with a peer node.

HIP nodes will also look up the domain name of desired peers in the DNS. When a successful lookup includes a peer's host identities, HIP nodes perform a HIP base exchange before establishing transport-layer connections. The HIP base exchange authenticates the end hosts and

can bootstrap encryption of the subsequent communication with IPsec [15]. The HIP specification [3] discusses the details of the base exchange and the related protocol exchanges.

After the base exchange, HIP nodes use host identities instead of IP

addresses for transport-layer connections with a peer. The HIP layer in the network stack internally translates host identities (HI) into network-layer IP addresses. This additional mapping between host identities and IP addresses (HI->IP) is logically separate from the first mapping between fully-qualified domain names and host identities (FQDN->HI).

For application and transport-layer compatibility, the FQDN->HI mapping must remain in the DNS. However, the HI->IP mapping is internal to the HIP layer and may be performed in a number of ways. Different lookup mechanism may support communication between two mobile or multi-homed HIP nodes better [4].

Transparent communication between HIP and non-HIP nodes places additional restrictions on the lookup mechanisms. For example, non-HIP nodes expect DNS lookups to return IP addresses, requiring the HI->IP mapping (or a representation thereof) to remain in the DNS. Section 4 discusses communication between HIP and non-HIP nodes and describes different alternatives that support it.

2. Communication Between HIP Nodes

In the current Internet, the DNS provides a FQDN->IP mapping. With HIP, it must continue to provide a mapping based on domain names. This allows transport-layer connections to bind to host identities instead of IP addresses transparently.

Instead of mapping domain names directly into IP addresses (FQDN->IP), with HIP the DNS maps them to host identities (FQDN->HI). In a second step, another lookup that is internal to the HIP-layer translates the host identities into IP addresses for network-layer delivery (HI->IP).

Several alternative approaches are possible for maintaining the HI->IP information. The DNS can maintain this mapping along with the FQDN->HI mapping. Alternatively, a database separate from the DNS can manage this information. This section discusses the different approaches and their implications on communication between two HIP nodes. Section 4 will discuss the compatibility aspects of the alternatives described here when HIP and non-HIP nodes communicate.

The HIP architecture and protocol specifications suggest storing host identities along with a node's IP addresses in the DNS [2][3]. The

index for both tables will be domain names. Logically, the DNS will thus contain two separate mappings: FQDN->HI and FQDN->IP.

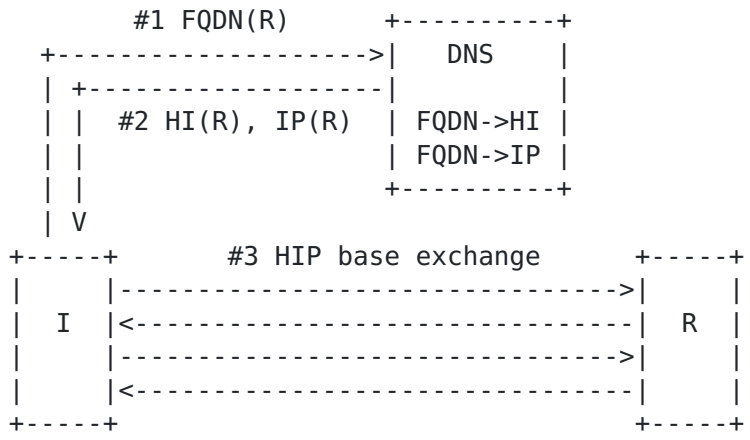

Figure 1: HIP lookup and base exchange

Figure 1 shows the lookup steps and HIP base exchange when a node's host identities are stored alongside its IP addresses. In step #1, the initiator I performs a DNS lookup on R's domain name FQDN(R). The DNS server responds with both R's host identities HI(R) and its IP addresses IP(R) in step #2.

The initiator I uses both pieces of information to perform the HIP base exchange with R in step #3. (The details of the base exchange, specified in [3], are not relevant to this discussion and will thus be omitted.)

Note that the DNS does not currently store the HI->IP mapping directly. Instead, a DNS lookup on a domain name returns both its FQDN->HI and FQDN->IP entries. The HIP stack then implicitly constructs the HI->IP mapping based on the HI and IP information returned by the DNS lookup. In the example in Figure 1, the FQDN(R) lookup in step #1 returns both HI(R) and IP(R) in step #2. HIP implicitly constructs the HI(R)->IP(R) mapping based on the assumption that HI(R) is reachable at IP(R).

One disadvantage of this approach is that a node's domain name is required to obtain both its host identities and its IP addresses. Even if a HIP node already knows the host identity of a HIP peer through other means, it cannot currently obtain the peer's IP addresses through the DNS. The DNS does not maintain an explicit HI->IP table, but instead indexes host identities only by domain names.

A reverse HIP->FQDN DNS mapping could address this limitation. HIP nodes would then look up a HIP peer's domain name through its host identity. They would then use the returned domain name to find the peer's IP addresses in a second lookup. However, the DNS may not be

structurally suited to maintain the reverse HIP->FQDN mapping. As the main Internet-wide database, the DNS is already being overloaded with functionality that might be better handled with new mechanisms [16]. Finally, the additional reverse lookup would increase the latency of the HIP base exchange.

3. Communication Between Mobile or Multi-Homed HIP Nodes

HIP decouples domain names from IP addresses. Because transport protocols bind to host identities, they remain unaware if the set of IP addresses associated with a host identity changes. This change can have various reasons, including, but not limited to, mobility and multi-homing.

Proposed extensions for mobility and multi-homing [4] allow a HIP node to notify its peers about changes in its set of IP addresses. These extensions require an established HIP association between two nodes, i.e., a completed HIP base exchange.

In addition to notifying its current peers about changes in its IP addresses, a HIP node must also update its HI->IP mapping in response to IP address changes. Otherwise, HIP base exchanges from new peers could fail because they try to contact the node at an IP address it is no longer reachable at.

3.1 Mobility and Multi-Homing with DNS Updates

If the DNS indirectly maintains the HI->IP mapping in a FQDN->IP table, nodes can dynamically update their DNS entry in a secure fashion [5][6]. The DNS server maintaining the information will then sign and distribute the updated zone.

Figure 2 shows an example of this scenario. In step #1, R registers its FQDN(R)->IP(R) entry in the DNS. It will dynamically update the DNS entry whenever its IP addresses IP(R) change. Because the DNS always contains R's current IP addresses, node I can perform a HIP base exchange with R at its new IP address (steps #2-4).

One drawback of using dynamic DNS updates in this way is the cost of updating secure zones. Re-signing an entire zone whenever the IP addresses of one entry change places a high cost on the DNS server. Using dynamic DNS to update HI->IP mappings may thus not be

appropriate when changes of IP addresses are frequent.

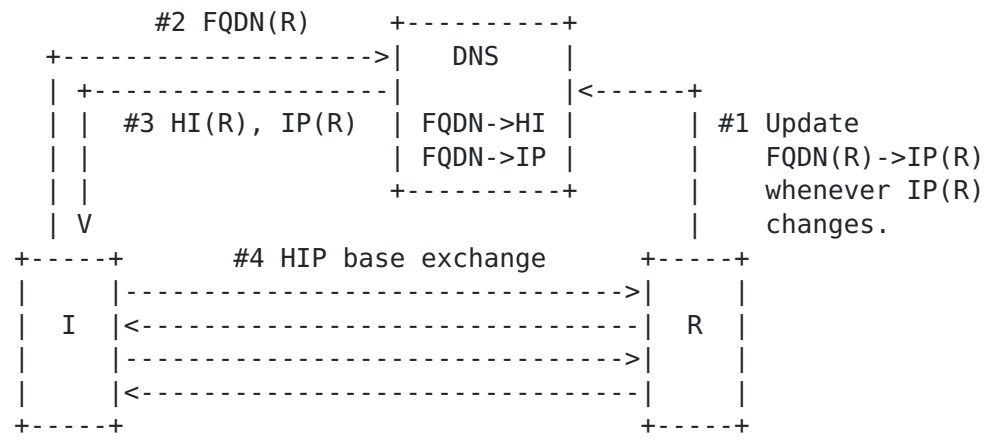

Figure 2: HIP lookup and base exchange with DNS updates

A simple, operational change could help limit the costs of frequent DNS updates. Instead of recomputing a zone after each dynamic update, a DNS server could aggregate the modifications and only perform zone updates periodically. The disadvantage of this approach is that HIP nodes may be unreachable until the DNS server distributes the updated zone.

Another concern with using the DNS to support HIP node mobility is the propagation time of updated DNS entries. DNS servers frequently cache DNS responses to reduce the load on the primary servers. During the time-to-live associated with a DNS response, DNS servers may answer additional requests for the same DNS entry from their local caches instead of contacting the primary servers. Thus, even after a HIP node updates its DNS entry, the DNS can still serve the old entry until the cached responses expire. This can lead to communication problems, because peers may try to contact a HIP node at an IP address it is no longer reachable at.

3.2 Mobility and Multi-Homing with Rendezvous Servers

The HIP architecture tries to greatly reduce the frequency of Dynamic DNS updates by introducing rendezvous servers [2]. Instead of registering its current set of IP addresses in its HI->IP entry in the DNS, a HIP node may instead register the IP addresses of its rendezvous servers. Because the IP addresses of rendezvous servers are assumed to change only infrequently, this approach can significantly reduce the load on DNS servers.

Rendezvous servers maintain a mapping between the host identities of HIP nodes for which they provide service and the node's current IP addresses. HIP nodes must notify their rendezvous servers about any changes in their IP addresses. This approach effectively relocates

the HI->IP information - and the burden of keeping it current - from the DNS to the rendezvous servers. This can reduce update costs under the assumption that rendezvous servers provide more efficient ways of maintaining HI->IP tables.

When a packet destined for one of its HIP nodes arrives at a rendezvous server, it relays the packet to one of the HIP node's current IP addresses. Due to the specifics of the HIP, only the first packet of a HIP base exchange will require such relaying [2]. Subsequent packet of the HIP base exchange and all further data packets will directly flow between the HIP nodes, bypassing the rendezvous server.

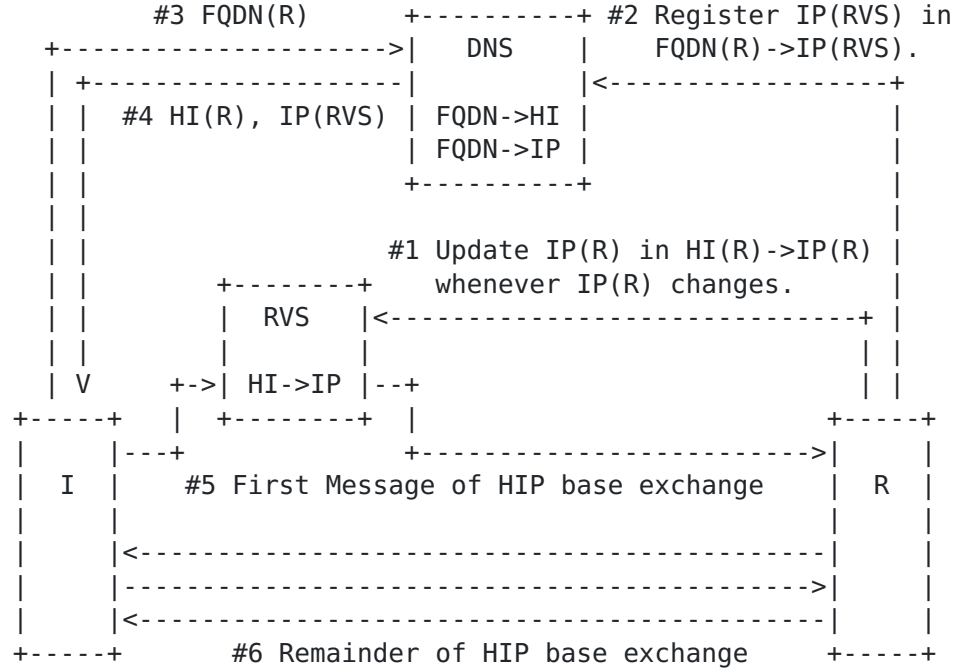

Figure 3: HIP lookup and base exchange with rendezvous server

Figure 3 shows a HIP lookup and base exchange involving a rendezvous server. Here, HIP node R is using rendezvous server RVS. In step #1, it updates RVS with its current IP addresses IP(R). Then, in step #2, R registers the rendezvous server's IP addresses IP(RVS) in its FQDN(R)->IP(RVS) DNS entry.

In step #3, a second HIP node I issues a DNS lookup on FQDN(R) to obtain R's host identities HI(R) and IP addresses. The lookup returns R's host identities HI(R) in step #4. The DNS reply also

includes the IP addresses of the rendezvous server IP(RVS) (instead of IP(R), because R's current addresses are unknown to the DNS.)

In step #5, node I initiates the HIP base exchange. It addresses the first packet of the HIP base exchange to IP(RVS). Upon receipt, the rendezvous server relays the packet to one of R's current IP addresses IP(R). The remainder of the HIP base exchange then occurs directly between I and R in step #6.

When rendezvous servers maintain the HI->IP information, they may support more efficient update operations compared to dynamic DNS updates (Section 3.1). Unlike the DNS, rendezvous servers do not provide a lookup service. Instead, they use the HI->IP information to actively relay traffic between HIP nodes.

This approach changes the role of the IP addresses stored in a DNS entry. Traditionally, nodes were directly reachable at the IP addresses listed in their DNS entry. HIP rendezvous server change this basic property by replacing the IP addresses of their client nodes in the DNS with their own. The IP addresses in a DNS entry hence no longer directly designate interfaces of an endpoint. Instead, they identify interfaces of a node that can relay packets to the endpoint.

When two HIP nodes communicate, this change has few consequences. HIP decouples higher layers from underlying IP addresses. However, when HIP and non-HIP nodes communicate, this change has a significant impact on the overall architecture. Section 4 will discuss the implications in detail.

4. Communication Between HIP and Non-HIP Nodes

Section 2 and Section 3 have discussed communication between HIP nodes. This section focuses on communication between HIP and non-HIP nodes. Two different scenarios exist. First, a HIP initiator may start communication with a non-HIP recipient. Second, a non-HIP initiator may try to contact a HIP recipient.

Without rendezvous servers, communication between HIP and non-HIP nodes remains identical to the current Internet. Transport-layer protocols bind directly to IP addresses. When IP addresses change, due to mobility or other reasons, transport-layer connections break.

Rendezvous servers may establish some of HIP's benefits even if one of the endpoints does not support it. Rendezvous servers live at static IP addresses. They can maintain ongoing transport-layer

connections by acting as a relays for HIP nodes whose IP addresses may change. The discussion in the remainder of this section assumes that HIP nodes utilize rendezvous servers to maintain the HI->IP information as described in Section 3.

The HIP architecture document [2] discusses the role of rendezvous servers in HIP communication. However, it does not currently describe the details of how rendezvous server relay traffic between HIP and non-HIP nodes. The remainder of this section presents this aspect of rendezvous servers.

4.1 Non-HIP Initiator to HIP Responder

In the first scenario, a non-HIP initiator starts communication with a HIP node. The HIP node is using rendezvous servers. Figure 4 shows this case.

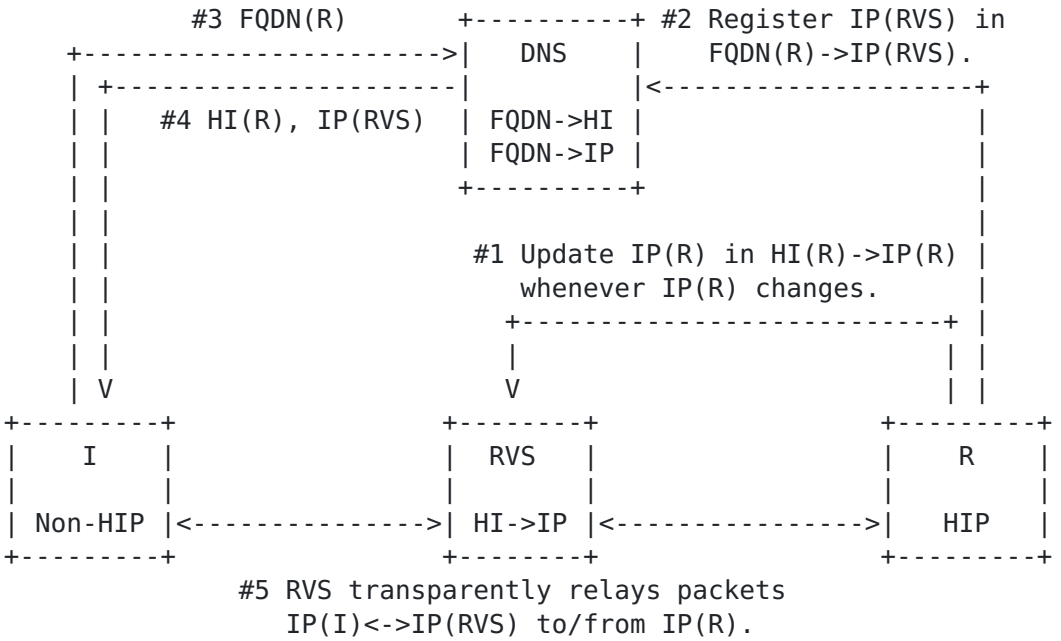

Figure 4: Non-HIP initiator to HIP responder via rendezvous server

Steps #1-4 remain unchanged from the HIP-HIP case shown in Figure 3 and discussed in Section 3.2. HIP node R registers the IP addresses of its rendezvous server RVS in the DNS. It also keeps RVS updated with its current IP addresses IP(R).

When non-HIP node I starts communication with R, it performs a DNS lookup on FQDN(R) and receives HI(R) and IP(RVS) in return. Since I does not support HIP, it disregards the host identity HI(R) returned by the DNS lookup. Instead, it sets up transport-layer connections using the IP addresses IP(RVS) obtained from the DNS. The rendezvous

server RVS must then transparently relay the communication to one of R's current IP addresses IP(R) in step #5.

End-to-end communication between I and R is complicated by the fact

that R's DNS entry lists IP addresses IP(RVS). The addresses IP(RVS) belong to the rendezvous server RVS and not R, the endpoint of the communication. I's transport layer will thus bind connections to R to IP addresses IP(I) and IP(RVS). Section 4.3 will discuss the implications.

4.2 HIP Initiator to Non-HIP Responder

This section describes a second scenario, where a HIP node initiates communication with a non-HIP node. Figure 5 shows this case.

As before, the HIP node I keeps its rendezvous server RVS updated about its current IP addresses IP(I) in step #2. It also registers the IP addresses of the rendezvous server IP(RVS) in its DNS entry in step #2, instead of its own.

In step #3, I initiates a transport-layer connection to R by performing a domain name lookup on FQDN(R). The DNS reply in step #4 contains R's IP addresses IP(R) but no host identities, because R is not a HIP node.

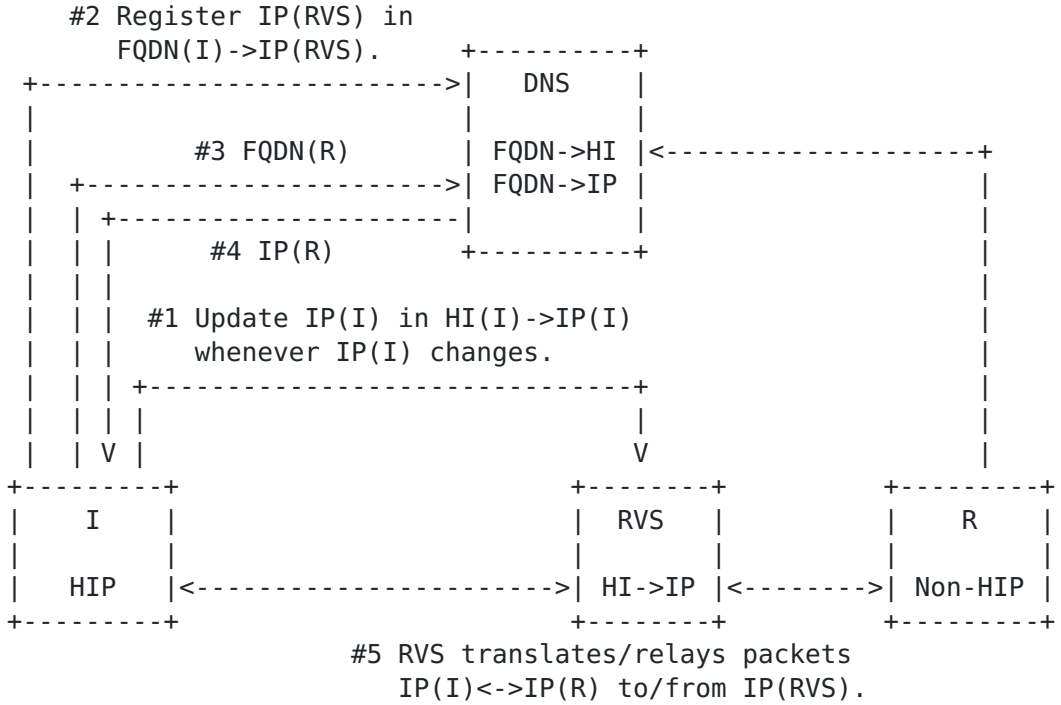

Figure 5: HIP initiator to non-HIP responder via rendezvous server

If I uses IP(R) to establish a direct transport-layer connection with R, the connection will break when R's IP addresses change. Instead, R relays its traffic through rendezvous server RVS in step #5. Since

the IP addresses of RVS are static, the transport-layer connection between I and R remains unaffected from changes to R's IP addresses.

4.3 Discussion

As illustrated by the two scenarios described in Section 4.1 and Section 4.2, rendezvous servers can isolate non-HIP nodes from changes to their HIP peers' IP addresses. Binding transport-layer connections to static IP addresses of rendezvous servers, instead of the more volatile addresses of HIP peers, allows connections between HIP and non-HIP nodes to retain some of the benefits of HIP-HIP connections.

The current HIP architecture document [2] requires HIP nodes using rendezvous servers to register the rendezvous server's IP addresses in the DNS. Consequently, rendezvous servers become explicit connections endpoints. This causes several challenges for end-to-end communication, as discussed in the next sections.

4.3.1 Relaying Overhead

The first issue is relaying overhead. When HIP nodes communicate, rendezvous servers will only need to relay the first packet of a HIP base exchange. The remaining HIP base exchange packets, as well as all subsequent data packets, will flow directly between the HIP nodes.

This is not the case for communication between HIP and non-HIP nodes. A non-HIP node will bind its transport-layer connection to the IP address obtained by looking up the HIP peer's domain name in the DNS. This will be the address of the rendezvous server.

Consequently, all data from the non-HIP to the HIP node will flow through the rendezvous server. This can cause significant relaying overhead. It can also increase the communication delay between the nodes, further affecting performance.

Relaying overhead will be difficult to eliminate. In order to provide some of the benefits of HIP, non-HIP peers communicating with HIP nodes must be able to bind their transport-layer connections to static IP addresses. This constraint implies the presence of a statically addressed relay somewhere in the system.

4.3.2 Return Traffic

A second issue is return traffic from the HIP node to the non-HIP node. Because a non-HIP node binds its transport-layer connection to its peer's IP address, it will not accept return traffic from a

different address than it is sending to. Since all traffic from the non-HIP node is addressed to the rendezvous server, the non-HIP node will expect to receive return traffic from that source address.

Several approaches may address this issue. First, the HIP node may relay all its return traffic through the rendezvous server as well. This causes additional relaying overhead. Second, the HIP node may spoof the IP address of the rendezvous server when sending return traffic. This may cause problems when firewalls along the path perform ingress filtering [7]. Finally, the approach described in Section 5 can also eliminate this issue.

4.3.3 Node Identification

A third issue is identification of the specific HIP node that a rendezvous server must relay arriving packets to. Packets arriving from non-HIP nodes are simple IP packets addressed to the rendezvous server. They do not contain host identities or other information that will allow the rendezvous server to identify the correct HIP node for relaying.

One solution has the rendezvous server use multiple IP addresses. Each of the HIP nodes for which it provides service receives one unique IP address. The HIP node will then register this unique IP address in the DNS. Hence, the rendezvous server can use the destination IP addresses of arriving packets to identify the HIP node to which they must be relayed to. The approach described in Section 5 uses a similar scheme.

A downside of registering unique IP addresses per node is a more complex protocol between rendezvous servers and its HIP nodes. Furthermore, rendezvous servers serving many HIP nodes may require many IP addresses.

4.3.4 Network Address Translation

The HIP architecture document [2] uses the term "forwarding" to describe the operation by which a rendezvous server enables the exchange of packets between communicating nodes. This document uses the term "relaying" instead, to indicate that mechanisms other than IP forwarding may suit the same purpose.

One such approach for relaying packets between HIP and non-HIP nodes is Network Address Translation [8]. When acting as a Network Address Translator, a rendezvous server will rewrite the IP headers of packets exchanged between communicating nodes.

The use of network address translation remains problematic [9][10].

Avoiding its use in the rendezvous server may improve protocol and application compatibility. Section 5 will present a rendezvous mechanism that relays using simple IP forwarding instead, avoiding possible issues due to the use of network address translation.

5. Rendezvous Broker

This section describes rendezvous brokers. Rendezvous brokers provide a modified HIP rendezvous mechanism that addresses some of the issues discussed in Section 4.

Rendezvous brokers are named for their similarity to tunnels brokers [11]. Rendezvous brokers also share commonalities with MobileIP's Home Agents [12] as well as systems for leasing IP subnets [17].

> Note: Rendezvous brokers described in this section may be similar to the "packet forwarding agents" outlined in [18]. While this similarity is under discussion, this document will use the term "rendezvous broker" for clarity. If the two concepts are deemed identical, terminology may change.

Rendezvous brokers are IP routers and manage delegations of globally-routable IP subnets. Rendezvous brokers may be located anywhere in the network. HIP has no concept of home networks (unlike MobileIP [12]) that would tie rendezvous brokers to access networks.

When a HIP node requests rendezvous service, the rendezvous broker delegates a unique, globally-routable IP address (or prefix) to the HIP node. HIP node and rendezvous broker establish a tunnel using the delegated IP address as the HIP node's tunnel endpoint address. The rendezvous broker installs a route towards the delegated IP address via the tunnel. At the end of this process, the HIP node is globally reachable by non-HIP nodes at the delegated IP address obtained from the rendezvous broker.

Figure 6 illustrates this process. In step #1, HIP node R registers its host identity HI(R) with the rendezvous broker RVB. In step #2, R receives an IP address IP(T-R) from RVB. This IP address is globally-routable and delegated to RVB.

The rendezvous broker and the HIP node R then establish a tunnel between themselves in step #3. IP(T-R) is the IP address of R's

tunnel endpoint, T-RVB the endpoint address of the rendezvous broker. The tunnel encapsulates packets with IP(RVB) and IP(R). RVB then installs a route that forwards packets addressed to IP(T-R) over the tunnel.

In step #4, R registers the IP address obtained from RVB in its DNS

entry. When the non-HIP initiator I performs a DNS lookup in step #5, it receives IP(T-R) from the DNS in step #6 (along with HI(R), which it ignores). I then initiates a transport-layer connection from IP(I) to IP(T-R). Packets to IP(T-R) will be routed to the RVB, because it is the router for the subnet out of which IP(T-R) was allocated. The RVB will then forward such packets over the tunnel to R due to the route installed in step #3.

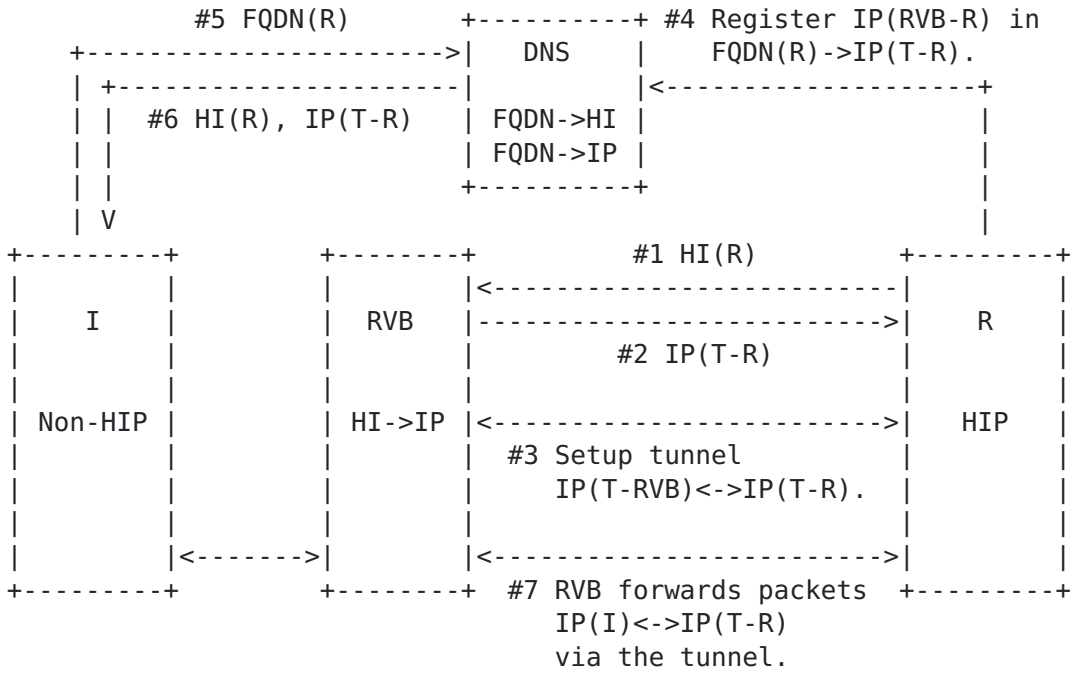

Figure 6: Non-HIP initiator to HIP responder via rendezvous broker

The next sections will compare rendezvous brokers to rendezvous servers and discuss several aspects of rendezvous brokers in more detail.

5.1 Comparison to Rendezvous Servers

Rendezvous brokers address some of the shortcomings of rendezvous servers raised in Section 4.3. One difference is that the IP addresses in a HIP node's DNS entry again identify interfaces of the HIP node itself. With rendezvous servers, the DNS entry instead identifies interfaces of the rendezvous server.

This simplifies the operation of the rendezvous broker. It performs

simple IP forwarding of packets that already carry the addresses of their final source and destination endpoints. Network Address Translation, or other schemes that relay by modifying packet headers, are not required. This may improve application and protocol

compatibility.

Because rendezvous brokers are IP routers, additional mechanisms to identify the correct HIP destination node for arriving packets are not required. The globally-routable destination IP address already acts as a unique indicator of the final destination.

5.2 Mobility

Rendezvous brokers offer mobility support that is equivalent to rendezvous servers. HIP nodes already notify their rendezvous servers when their IP addresses change. Rendezvous brokers also require such notification.

When the IP addresses of a HIP node changes, the rendezvous broker and the HIP node must reconfigure the tunnel between them. This reconfiguration only affects the IP addresses used for tunnel encapsulation. The addresses of the tunnel endpoints remain unchanged. Transport-layer connections bound to a HIP node's tunnel endpoint address thus remain unaffected.

HIP nodes may change rendezvous servers over time and they may use multiple rendezvous servers at the same time. The same is true for rendezvous brokers. Both rendezvous servers and rendezvous brokers may be located anywhere in the network; unlike MobileIP [12], HIP has no notion of home networks. By separating rendezvous mechanisms from topological locations, HIP allows nodes to choose rendezvous servers or Brokers based on local criteria, including network connectivity, location, or mobility.

5.3 Tunneling

This document does not further define the specifics of the tunneling mechanism used between a rendezvous broker and its HIP nodes. Possible tunneling mechanisms include [19][20][21][22][23]. Different tunneling mechanisms incur different overheads. Some may also offer better traversal of Network Address Translators or firewalls.

Similarly, the tunnel setup protocol between rendezvous brokers and HIP nodes is currently unspecified. Candidate tunnel management approaches include [24][25][26].

Rendezvous brokers forward all traffic from non-HIP nodes to HIP nodes over tunnels. For the return traffic from HIP nodes to non-HIP nodes two options exist. First, return traffic could also flow over tunnel. Second, return traffic could flow through the base network over one of the HIP node's interfaces. The second alternative may

offer increased performance due to the avoidance of triangle routing. However, firewalls that perform ingress filtering could prevent communication [7].

Another aspect of using tunnels to connect rendezvous brokers and their HIP nodes is reduced Maximum Transmission Units. Implementation issues in the network stacks of end systems and routers can lead to communication problems in such scenarios [27].

6. Location Privacy with HIP

Section 3.2 discussed HIP rendezvous servers and Section 5 discussed HIP rendezvous brokers. One common characteristic of these approaches is end-to-end addressing, i.e., initiator and responder of HIP associations eventually learn the other's current IP address. Because IP addresses have topological relevance, they may allow to deduce additional information about the peer, such as their ISP or even geographical location. In some cases, this is undesirable.

The current HIP architecture does not support location privacy. It exposes peer IP addresses, which in their function as locators can be used to deduce additional information about peers. If location privacy is a non-functional requirement for HIP, the current architecture must be augmented.

Rendezvous brokers, as described in Section 5, maintain bindings between peers' local and globally visible IP addresses. Rendezvous brokers may thus already support some degree of location privacy, because they only make a host's global IP addresses visible to its peers. This, however, requires all traffic to flow through the broker.

One key limitation of the current HIP architecture with regard to location privacy is that it implicitly requires the end hosts to resolves their peers' host identities into the corresponding IP addresses. This does not change even with rendezvous brokers; they still reveal the peers' global IP addresses to the end hosts.

The following sections discuss extensions to the current HIP architecture that establish location privacy, i.e., do not reveal the IP addresses associated with a node's network interfaces to its peers.

6.1 Location Privacy Between HIP Nodes

6.1.1 Distributing HIP Functionality

With HIP, transport-layer services bind to host identities instead of

IP addresses. Resolution of the destination's host identity to its associated IP addresses implicitly occurs at the host identity layer and may involve additional communication with remote entities, such as the DNS. This is because the network layer requires packets to be addressed with the appropriate IP addresses to allow traffic forwarding towards the final destination. Figure 7 illustrates this view of the HIP protocol stack.

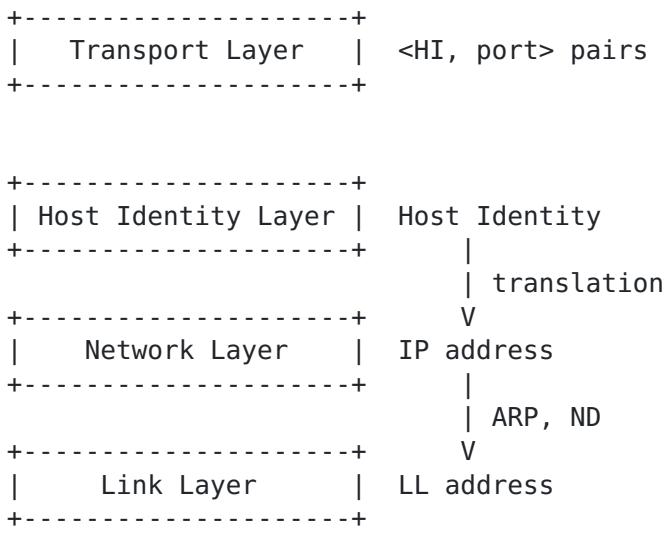

Figure 7: HIP architecture reference model

One approach to support a limited degree of location privacy using HIP is to have both the initiator and the responder of a HIP association use rendezvous brokers throughout a communication. This approach, however, still reveals the remote host's globally visible IP addresses, because the tunneled IP packet between a host and its RVB contains the peer's global IP address. It does hide local movement and the associated changes of a host's local IP address though.

A more complete approach to establish location privacy is to split up the HIP architecture and relocate some pieces of the HIP functionality into new network entities. Figure 8 shows one example of such an approach of splitting and relocating HIP functionality and the following sections discuss it in more detail.

When HIP functionality is relocated, destination networks become similar to IP-based mobile communication networks. They comprise of various network components (agents, brokers, servers, gateways) and

access routers that provide mobile hosts with wired or wireless access to an infrastructure. In such environments, HIP should not expose the IP addresses of a host to its peers. Consequently, it must hide or obfuscate them at least on the last hop between a host

and its access router.

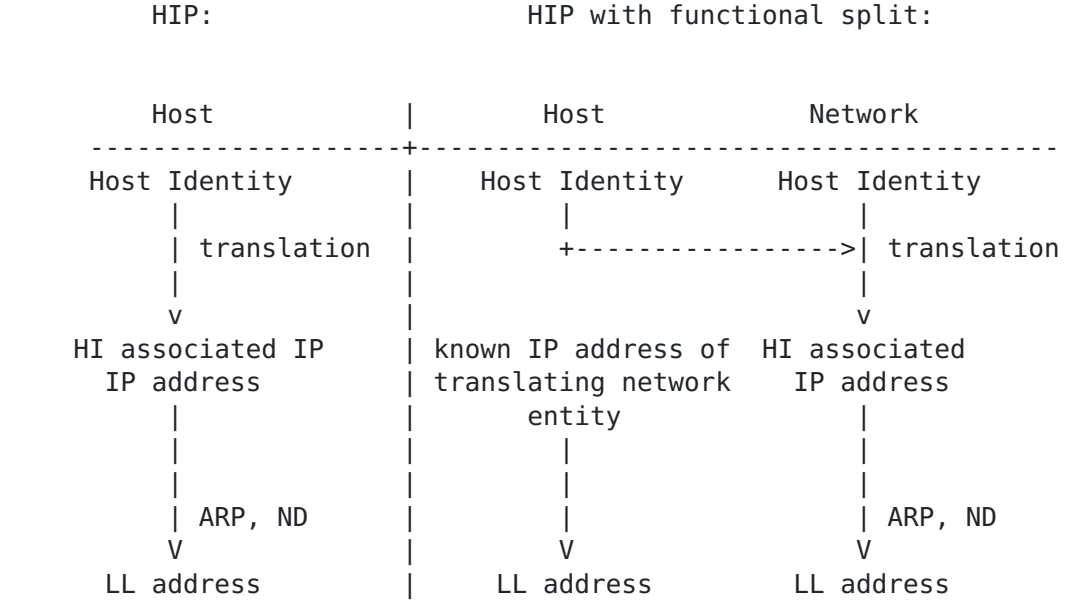

Figure 8: Relocating HIP functions into the network

The proposed functional split separates the HI-to-IP address resolution from the end host's HIP layer and relocates it to an entity inside the network. Instead of addressing their peers directly, hosts now address a network entity that then resolves the HI to the IP address apparently associated with the remote host. ("Apparently", because that address may in fact belong to another network entity that will deliver to the remote host.) Consequently, the IP addresses used by the end host's HIP layer is that of a "well-known" network component. Various possibilities for the relocation of the HIP lookup exist. The following sections describe one approach that relocates the resolution function to an enhanced rendezvous server that hosts have previously registered with. To avoid ambiguities with the previously described rendezvous servers and brokers, the discussion will use the term "rendezvous agent" for this new entity. (This terminology may change in future revisions of this document.)

Two separate scenarios for communication involving a rendezvous agent (RVA) exist. First, hosts may address all traffic to the RVA. Hence, the RVA address is the well-known address that the host's HIP layers use according to the functional split in Figure 8. In the second case, hosts address all packets to their current access routers, which act as relays between the hosts and their RVAs.

The following figures extend the notation used in the beginning of this document. Initiator and responder hosts are still I and R,

respectively. The local IP address of host X is IP_L(X) whereas its global IP address, maintained by the RVAs, is IP_G(X).

6.1.2 Communication with Local Rendezvous Agents

This scenario assumes that mobile hosts are allowed to communicate directly with their associated local RVAs. They have previously registered with the RVAs as described in Figure 8.

Figure 9 shows the operation of the protocol when a host directly communicates with its RVA. It also illustrates the case where I and R are located in different domains.

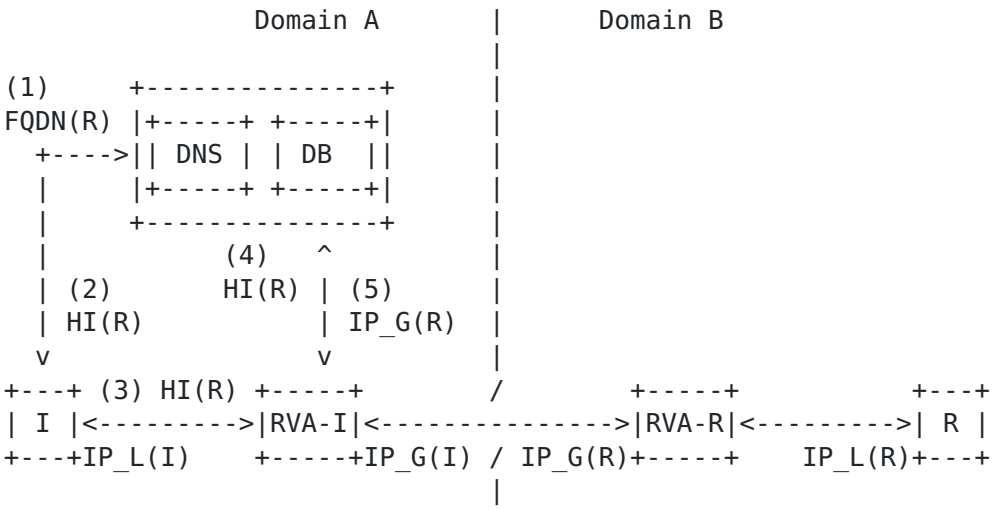

Figure 9: Direct communication between host and rendezvous agent

When I wants to establish communication with R, it first resolves FQDN(R) into the associated HI(R), possibly via the DNS. When rendezvous agents are used, the DNS MUST NOT return R's IP addresses IP(R). I then sends its packets to R to its RVA (RVA-I) and includes R's HI(R). When RVA-I receives the packets destined for R, it resolves HI(R) into R's global IP address IP_G(R) with the help of a currently unspecified database (DB). This database may or may not be collocated with the DNS.

>From then on, RVA-I serves as a proxy between I and R's RVA (RVA-R). RVA-I will never expose IP_G(I), much less IP_L(I), and RVA-R does the same. Communication between the RVAs occurs with the hosts'

global IP addresses IP_G(I) and IP_G(R), while packets forwarding between a host and its RVA uses the local IP addresses IP_L(I) and IP_L(R).

One disadvantage of this approach is that it places a high load on

the RVAs caused by relaying of all traffic. To limit load on a particular RVA, multiple RVAs may serve the registered hosts of a domain to distribute load. Domains may coordinate load distribution when hosts first attempt to register with an RVA. The specifics of such mechanisms are out of the scope of this document.

Some service providers may have policies that forbid mobile hosts to communicate directly with network components and require them to use controlled edge relays. This provides some measure of protection by hiding the actual location and IP addresses of the network components. Under such a policy, HIP relays or attendants might be collocated with access routers. The next section briefly discusses this case.

6.1.3 Communication with First-Hop Attendants

This section assumes that a policy prohibits mobile hosts to communicate directly with RVAs, but requires them to use attendants. These attendants may be collocated with a domain's access routers and serve as relays for IP packets between a host and its associated RVAs.

Hosts usually know the IP address of their access router. A default router's IP address can thus serve as the well-known IP address used by the host's HIP layer. The access router's relaying function may also support RVA discovery by processing hosts' discovery or registration request and assign them an alias address to identify their assigned RVA. This approach uniquely identifies RVAs without revealing their IP address to the mobile hosts. Access routers map these aliases into the associated RVA's IP address. (This description simplifies the process for the purposes of discussion. The specifics of attendant functionality are out of the scope of this draft.)

Communication establishment is similar to the scenario described in Figure 9, but no direct path exist between a host and its RVA. Instead, as shown in Figure 10, the access router terminates the path and sets up a new one towards the host or the RVA, respectively. (The current revision of this document does not yet discuss the details of the require IP addressing schemes and binding maintenance. A future revision may investigate these issues.)

One advantage of this architecture is that it does not reveal the RVAs' IP addresses to the mobile hosts. The attendants that are

collocated with the access routers relay all communication, including signaling and data traffic. A second advantage is that attendants can support discovery of appropriate RVAs before or during a host's registration process.

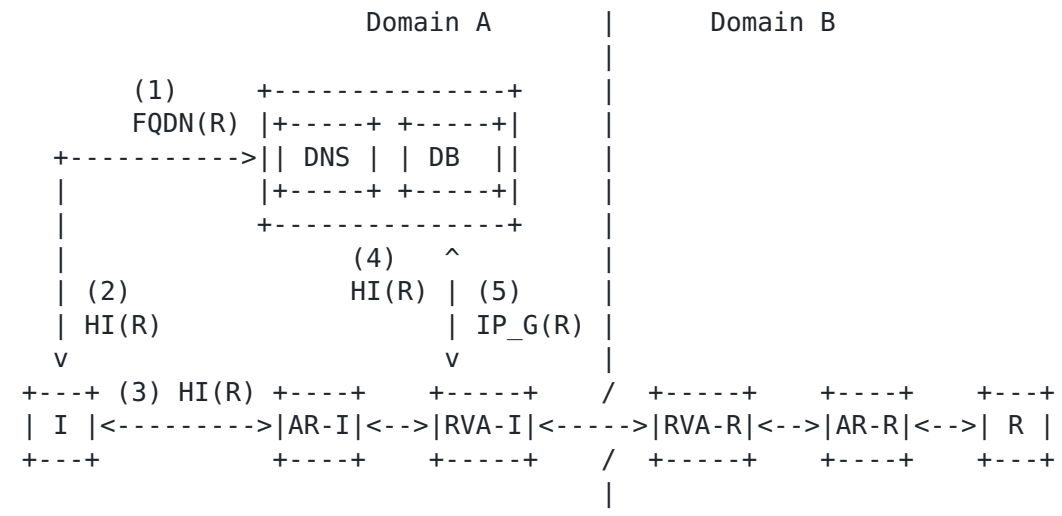

Figure 10: Indirect communication between host and rendezvous agent

A disadvantage of the attendant-based architecture is that two relays per host exist, bringing the total number of relays along the path from I to R to four. This is because access routers now encapsulate and decapsulate packets in addition to the RVAs. This introduces additional per-packet processing overhead and increases forwarding delays.

In addition, this solution is difficult to realize without introducing and maintaining state at the RVAs and attendants. This can affect scalability and performance of the access routers.

6.2 Location Privacy between HIP and Non-HIP Nodes

Extending the proposed approaches for establishing location privacy to include communication with non-HIP nodes is difficult. They require the initiator to transmit the peer's host identity to the translating function inside the network. Including HI(R) in the HIP header easily meets this requirement for HIP nodes. For non-HIP nodes, this is not an option. Furthermore, non-HIP nodes require a static identifier to replace the HI for communication, which some entity in the network must then transparently resolve. A static IP address might be such an identifier, for example, using MobileIP, which offers the peer's "home address" for such a purpose.

One approach to transmit a peer's static IP address to the translating network entity (or RVA) could be IP tunneling. Host I

addresses R's static IP address in the inner, tunneled packet, whereas the outer IP header addresses the RVA (assuming direct communication between a host and its RVA.) The RVA terminates the tunnel, decapsulates the packet and resolves R's static IP address to

R's globally visible IP address IP_G(R).

As above, packets between RVAs use IP_G(I) and IP_G(R) as addresses, respectively. Packets between a host and its RVA use the host's local IP address and the RVA's address in the outer IP header, whereas the inner header must use the static IP addresses of both.

7. Security Considerations

The security aspects of different HIP rendezvous mechanisms are currently being investigated. They will be discussed in a future revision of this document.

8. Acknowledgments

The following people have helped to improve this document through thoughtful suggestions: Marcus Brunner, Tom Henderson, Mika Kousa, Pekka Nikander, Simon Schuetz, Martin Stiemerling, and Juergen Quittek.

9. References

9.1 Normative References

[1] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[2] Moskowitz, R., "Host Identity Protocol Architecture", draft-moskowitz-hip-arch-05 (work in progress), October 2003.

[3] Moskowitz, R., Nikander, P. and P. Jokela, "Host Identity Protocol", draft-moskowitz-hip-09 (work in progress), February 2004.

[4] Nikander, P., "End-Host Mobility and Multi-Homing with Host Identity Protocol", draft-nikander-hip-mm-01 (work in progress), January 2004.

[5] Vixie, P., Thomson, S., Rekhter, Y. and J. Bound, "Dynamic

Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.

[6] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.

[7] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[8] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[9] Hain, T., "Architectural Implications of NAT", RFC 2993, November 2000.

[10] Senie, D., "Network Address Translator (NAT)-Friendly Application Design Guidelines", RFC 3235, January 2002.

[11] Durand, A., Fasano, P., Guardini, I. and D. Lento, "IPv6 Tunnel Broker", RFC 3053, January 2001.

[12] Perkins, C., "IP Mobility Support for IPv4", RFC 3344, August 2002.

9.2 Informative References

[13] Saltzer, J., "On the Naming and Binding of Network Destinations", RFC 1498, August 1993.

[14] Sunshine, C., "Addressing Problems in Multi-Network Systems", IEN 178, April 1981.

[15] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[16] Klensin, J., "Role of the Domain Name System (DNS)", RFC 3467, February 2003.

[17] Touch, J., Eggert, L. and Y. Wang, "TetherNet Anti-NAT - Secure Internet Subnet Rental System", Proc. 3rd DARPA Information Survivability Conference and Exposition (DISCEX-III) 2003, April 2003.

[18] Nikander, P., Ylitalo, J. and J. Wall, "Integrating Security, Mobility, and Multi-Homing in a HIP Way", Proc. Network and Distributed Systems Security Symposium (NDSS) 2003, February 2003.

[19] Perkins, C., "IP Encapsulation within IP", RFC 2003, October 1996.

[20] Levkowetz, H. and S. Vaarala, "Mobile IP Traversal of Network Address Translation (NAT) Devices", RFC 3519, May 2003.

[21] Farinacci, D., Li, T., Hanks, S., Meyer, D. and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[22] Nikander, P., "A Bound End-to-End Tunnel (BEET) mode for ESP", draft-nikander-esp-beet-mode-00 (work in progress), October 2003.

[23] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G. and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.

[24] Hamzeh, K., "Ascend Tunnel Management Protocol - ATMP", RFC 2107, February 1997.

[25] Beijnum, I., "On Demand Tunneling For Multihoming", draft-van-beijnum-multi6-odt-00 (work in progress), January 2004.

[26] Touch, J., "Dynamic Internet overlay deployment and management using the X-Bone", Computer Networks Vol. 36, No. 2-3, July 2001.

[27] Lahey, K., "TCP Problems with Path MTU Discovery", RFC 2923, September 2000.

Authors' Addresses

Lars Eggert
NEC Network Laboratories
Kurfuerstenanlage 36
Heidelberg 69115
DE

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

Marco Liebsch
NEC Network Laboratories
Kurfuerstenanlage 36
Heidelberg 69115
DE

Phone: +49 6221 90511 46
Fax: +49 6221 90511 55
EMail: marco.liebsch@netlab.nec.de
URI: http://www.netlab.nec.de/

Appendix A. Document Revision History

Revision	Comments
00	Initial version.
01	Add discussion on HIP location privacy and rendezvous agents. Minor fixes to figures and their descriptive text. Use boilerplate from RFC 3667.

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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