Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 7, 2011

D. Zhang
X. Xu
Huawei Technologies Co.,Ltd
J. Yao
CNNIC

March 6, 2011

# Investigation in HIP Proxies
draft-irtf-hiprg-proxies-02

## Abstract

HIP proxies play an important role in the transition from the current Internet architecture to the HIP architecture. A core objective of a HIP proxy is to facilitate the communication between legacy (or Non-HIP) hosts and HIP hosts while not modifying the host protocol stacks. In this document, the legacy hosts served by proxies are referred to as Legacy Hosts (LHs). Currently, various design solutions of HIP proxies have been proposed. These solutions may be applicable in different working circumstances. In this document, these solutions are investigated in detail and compare their effectiveness in different scenarios.

## Requirements Language

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 RFC 2119 [RFC2119].

## Status of this Memo



## Copyright Notice

Table of Contents

# 1. Introduction

The Host Identity Protocol (HIP) is a ID/Locator separating technology for Internet Protocol (IP) networks. It introduces a new host identifier layer in the middle of the network layer and the transport layer so as to comprehensively address the issues of mobility, multi- homing. Compared with other ID/Locator separating technologies, HIP is security integrited. The Host Identities (HIs) of HIP enable hosts are verifiable by using cryptographic methodology. Particularly, HIP provides a handshaking process for HIP hosts to authenticate each other and distribute symmetric keys for securing subsequent communication. Therefore, a HIP host cannot directly communicate with a legacy host.

As core components of HIP extensional solutions, HIP proxies have attracted increasing attention from both the industry and the academia. A HIP proxy is a middlebox located between a legacy host and a HIP enabled host for protocol transition. Under the assistance of a HIP proxy, a legacy host can communicate with its desired HIP host without updating its protocol stack.

Currently, multiple research work is engaged in both design and performance assessment of HIP proxies. In this document, we attempt to investigate several important designing solutions and compare their effectiveness in different scenarios. Actually, there has been a detailed discussion of HIP proxies in [SAL05]. This document can be regarded as a complement of that paper. Some new topics (e.g., the asymmetric path issues occurred in the load-balancing mechanisms for HIP proxies and the necessity of extending the HIP RR for HIP proxies) are discussed in the draft. Theoretically, LHs and the HIP hosts which the LHs intend to communicate with can be located anywhere in the network. However, in this document, without mentioning otherwise, it is assumed that legacy hosts are located within a private network and HIP hosts are located in the public network, since this is the most important scenario that HIP proxies are expected to support [SAL05].

The remainder of this document is organized as follows. Section 2 lists the key terminologies used in this document. In section 3, the essential functions of HIP proxies are indicated, and a classification of HIP proxies is proposed to benefit subsequent analysis. In section 4, we analyze the issues that HIP proxies have to face in constructing a Load Balancing Mechanism (LBM) which facilitates communication initiated by LHs. Section 5 analyzes the issues that HIP proxies in a LBM have to face if they also need to support communication initiated by HIP hosts. In section 6, we investigate the issues that HIP proxies have to deal with in supporting dynamic load balancing and redundancy. Section 7 provides a brief discussion about the influence brought by DNSSEC [RFC4305] to the deployment of HIP proxies. Section 8 is the conclusion of the entire document. Section 9 is the security consideration.

## 2. Terminology

BEX: HIP Base Exchange

DI Proxy: DNS Inspecting Proxy

HA: HIP association

LBM: Load Balancing Mechanism

N-DI proxy: Non-DNS Inspecting Proxy

LHs: Legacy Hosts which are made up as HIP hosts by HIP proxies.

## 3. HIP Proxies

### 3.1. Essential Functionality of HIP Proxies

A primary function of HIP proxies is to facilitate the communication between legacy (or Non-HIP) hosts and HIP hosts while not modifying the host protocol stacks. In order to achieve this, a HIP proxy needs to intercept the packets transported between LHs and HIP hosts before they arrive at their destinations. Upon capturing such a packet, a HIP proxy needs to transfer it into the format which can be recognized by the host which the packet aimes to.

Assume a proxy captures a packet sent out by a LH. If the packet is destined to a HIP host, the proxy first tries to find out whether there is an appropriate HIP association (HA) in its local database to process the packet. If the HA exists, the proxy then uses the key maintained in the HA to encrypt the payload in the packet, transfer the packet into the HIP compatible fromat, and forwardes it to the HIP host. However, if there is not a proper HA, the proxy needs to use the HI and HIT assigned to the LH to carry out a HIP Base Exchange (BEX) and generate a new HA with the HIP host. The newly generated HA is then maintained in the local database.

Similarly, when processing a packet from a HIP host, the proxy needs to find a proper HA and use the keying material in the HA to decrypt the packet, and thus the packet is transferred into an ordinary IP packet and forwarded to the legacy host.

### 3.2. A Taxonomy of HIP Proxies

In practice, there are various design alternatives for HIP proxies. To benefit the analysis, in this document HIP proxies are classified into DNS lookups Intercepting Proxies (DI proxies) and Non-DNS lookups Intercepting Proxies (N-DI proxies). As indicated by the name, a DI proxy needs to intercept DNS lookups in order to correctly process the follow-up communication between LHs and HIP hosts, while N-DI proxies do not have to.

Note that a DI proxy implementation may also be able to intercept the lookup between a LH and a resolution server other than DNS. However, currently DNS is the only resolution mechanism detailed analyzed and extended to support HIP communication. Hence, DNS is only resolution mechanism considered in this document.

To avoid confusion, in the remainder of this document, the terms "lookup" and "answer"are used in specific ways. A lookup refers to the entire process of translating a domain name for a legacy host. The answer of a lookup is the response from a resolution server which terminates the lookup.

### 3.3. DI Proxies

Usually, before a legacy host communicates with a remote host, the legacy host needs to consult a DNS server for the IP address of its destination. On this premise, a DI proxy can effectively identify the HIP hosts which legacy hosts may contact in near future by intercepting DNS lookups.

In practice, it is common to deploy one or multiple DNS resolvers for a private network. A host in the private network can thus send its queries to a resolver instead of communicating with authoritative DNS servers directly. If the resolver does not cache the inquired RRs, it will try to collect them from authoritative DNS servers. In a lookup process, a resolver may have to contact multiple authoritative DNS servers before it eventually gets the desired DNS RRs.

On the occassions where a resolver is located out of a private network, a HIP proxy located at the border of the network can intercept the DNS queries from LHs and then use the FQDNs obtained from the queries to initiate a new DNS lookup to the resolver to inquire about the desired information (AAAA RRs, HIP RRs, and etc.). If the host that the legacy host intends to communicate with is HIP enabled, the DNS resolver will hand out a HIP RR associated with an AAAA RR to the proxy. After maintaining the needed information (e.g., HITs, HIs, and IPs addresses of the HIP hosts) in the local database for future usage, the proxy returns an answer with an AAAA

RR to the legacy host.

When the resolver is located inside the private network, conditions are a little more complex. If a proxy is deployed on the path between LHs and the resolver, it can operate as same as what is illustrated in the above paragraph. The proxies which can be deployed in this way are introduced in the remainder of this sub-section. However, if a proxy is located at the border of the network (i.e., in the middle of the resolvre and authoritative DNS servers), the proxy has to intercept the DNS lookups between the resolver and authoritative DNS servers. Because the resolver may have to contact multiple authoritative DNS servers to get a desired answer, for efficiency, the proxy can only inspect the responses from DNS services and find out DNS answers. Because the answer of a DNS lookup does not contain any NS RR, it can be easily distinguished from the intermediate responses. After identifying a DNS answer, a DI proxy can locate the DNS server maintaining the desired RRs from the packet header and identify the FQDN of the inquired host from the packet payload. Then, the proxy initiates an independent lookup to the DNS server to check whether the host is HIP enabled. If it is, the proxy maintains the information of the host for future usage and returns an answer with an AAAA RR to the resolver.

Besides intercepting DNS lookups, some kinds of DI proxies also modify the contents of the AAAA RRs in DNS answers to ehance the efficiency or deploying flexibility. For instance, [RFC5338] indicates that a HIP proxy can returns HITs rather than IP addresses in DNS answers to LHs. Consequently, the data packets from LHs to HIP hosts will use the HITs of the HIP hosts as destination addresses. The HIP proxy can then advertise a route of the HIT prefix to attract the packet for HIP hosts. [PAT07] also proposes a proxy solution which requires a HIP proxy to maintain an IP address pool. When sending a DNS answer to a LH, the proxy select an IP address from its pool and inserts it in the answer. The legacy host will regard this IP address as the IP address of the peer it intends to communicate with. In the subsequent communication, when the host sends a packet for the remote HIP host, it will use the IP address assigned by the proxy as the destination address. Therefore, the HIP proxy can intercept the packets for the HIP hosts by advertising the routes of the IP addresses in its pool. In the remainder of this document, these two types of proxies are referred to as DI-HIT proxies and DI-NAT proxies respectively, and the DI proxies which do not modify the contents of DNS answers (i.e., return the IP addresses of HIP hosts in answers) are referred to as DI-transparent proxies.

Compared with DI-HIT and DI-NAT proxies, DI-transparent proxies show their limitations in multiple aspects. For instance, it is a practical solution for a LH to publish the IP address of its proxy in its DNS AAAA RR so that the packets for the host will be directly forwarded to the proxy. Therefore, when a LH served by a DI-transparent proxy attempts to communicate with two remote LHs served by a same HIP proxy, it is difficult for the host to distinguish one remote host from the other as they both use the same IP address. In addition, DI-transparent proxies cannot work properly in the circumstance where HIP hosts renumber their IP addresses during the communication due to, e.g., mobility or re-homing. For DI-HIT or DI-NAT proxies, these issues can be largely mitigated as the IP addresses of HIP hosts will never be used by DI-HIT or DI-NAT proxies to identity hosts.

Moreover, it is difficult for DI-transparent proxies to advertise routing information to attract the packets transported between LHs and remote HIP hosts. Consequently, they can be only deployed at the borders of private networks. DI-HIT (or DI-NAT) proxies, however, can easily attract the packets for HIP hosts to themselves because the packets destined to HIP hosts use HITs (or the IP addresses selected from pools) as their destination addresses. Hence, they can locate inside the networks. Therefore, in private networks which resolvers are located inside, DI-HIT or DI-NAT proxies can be deployed on the path between the resolvers and LHs and reduce the overhead on the proxies imposed by intercepting DNS lookups.

It is recommended to use DNSSEC [RFC4305] to prevent attackers from tampering or forging DNS lookups between resolvers and DNS server. This solution may affect the deployment of HIP proxies. For instance, DI-HIT and DI-NAT proxies need to modify the contents of DNS answers, and thus they should be only deployed on the path between legacy hosts and their resolvers if DNSSEC is deployed. Therefore, a DI-HIT proxy (or a DI-NAT proxy) cannot not be deployed in the middle of DNSSEC-enabled resolvers and authoritative DNS servers.

### 3.4. N-DI Proxies

Unlike DI proxies, an N-DI proxy does not try to intercept DNS lookups transported between LHs (or resolvers) and DNS servers.

In the circumstances where the HIP hosts that LHs intend to contact are predicable, an N-DI proxy can maintain a list of the information of the HIP hosts [SAL05]. After intercepting a packet from a LH, the proxy can ensure the packet is for a HIP host if the destination address of the packet is maintained in the list.

In the circumstances where it is difficult to predicate the HIP hosts that LHs intend to contact in advance, an N-DI proxy needs to contact DNS servers to find out whether the destination IP address of a packet belongs to a HIP host or a legacy host. The information obtained from the DNS servers can be maintained within two lists. One list is for the information of HIP hosts; the other is for the information of legacy hosts. When intercepting a packet, the N-DI first compares the destination address of the packet against the addresses in the lists to find out whether the destination of the packet is HIP enabled. If the address is not maintained in the lists, the proxy then has to consult resolution systems and maintains the information of the host which the packet is aimed for in the correspondent list, according the answers from resolution systems.

## 3.5. Distributed Implementation of DI Proxies

As discussed above, DI proxies have to intercept the DNS lookup packets between legacy hosts and DNS servers in order to facilitate the communication between LHs and HIP hosts. This requires a DI proxy be deployed on the boundary of the private network or on the path where LHs and the DNS resolver tranport their lookup packets. In the circumstance where DNSSEC is deployed, a DI proxy cannot even be deployed on the boundary of the private network either, if the proxy needs to modify DNS lookup packets. Such inflexibility may be undesired under certain circumstances.

In this section, we analyze a design option of DI proxies which improves the deployment flexibility of DI proxies and avoids the issue brought by DNSSEC by separating the DNS related functionality ( i.e., intercepting and modifying the DNS communication) from DI proxies. The component performing the DNS lookup interception is referred to as the DNS lookup inspector while the component performing the protocol transfermation is referred to as the proxy component. A DNS lookup inspector is located in a place where it can intercept and modify DNS lookups. In practice, a DNS resolver can also be extended to act as an inspector.

### 3.5.1. Distributed DI-HIT Proxies

The DNS lookup inspector of a distributed DI-HIT proxy returns HITs in DNS answers to LHs. Therefore, the associated DI-HIT proxy can advertise routing information inside the private network to attract the packets using HITs as destination addresses. Additionally, the inspector needs to transfer other information (e.g, IP addresses of the HIP hosts and RVSes) of the HIP hosts contained in DNS RRs to the DI-HIT proxy component so that the proxy can perform BEXes with the HIP hosts on behavior of LHs.

A DI-HIT proxy component can be associated with multiple DNS lookup inspectors. It is possible for a DI-HIT proxy component to be deployed in public networks to support multiple private networks.

This property is useful when Internet services providers (ISPs) intend to facilitate the legacy hosts in the private networks without HIP proxies to communicate with HIP hosts.

### 3.5.2. Distributed DI-NAT Proxies

A DNS lookup inspector of a distributed DI-NAT proxy needs to not only return the IP addresses in the address pool of the DI-NAT proxy component but also transfer the mapping information of the IP addresses and the correspondent HIP hosts to the DI-NAT proxy component. Moreover, the resolver needs to maintain the mapping information so as to assign an IP address for multiple HIP hosts concurrently.

Similar with Distributed DI-HIT Proxies, a DI-NAT proxy component can also be deployed in a public network. In this case, the addresses in the address pool must be routable in the public network.

### 3.5.3. Distributed DI-transparent Proxies

A DNS lookup inspector of a distributed DI-transparent proxy needs not to modify DNS answers, but it needs to transport the IP addresses and HIs of queried HIP hosts to the DI-NAT proxy component. In this case, a DI-transparent proxy component must be deployed on the boundary of the private network in order to guarantee that it can intercept packets.

## 4. Issues with LBMs in Supporting LHs to Initiate Communication

If there is only a single HIP proxy deployed for a private network, the proxy may become the cause of a single-point-of-failure. In addition, when the number of the users increases, the overhead imposed on the proxy may overwhelm its capability, which makes it the bottleneck of the whole mechanism. A typical solution to mitigate this issue is to organize multiple proxies to construct a LBM. By sharing overhead of processing packets amongst multiple HIP proxies, a LBM can be more scalable and failure tolerant.

### 4.1. LBMs adopting Load Balancers

Load balancers have been widely utilized in the design of LBMs. When adopted in a HIP proxy LBM, a load balancer needs to pool all proxy resources and be located in a position where it can intercept DNS lookups or modify DNS lookups if necessary. Also, the load balancer needs to distribute the information it learned from the DNS lookups to the appropriate proxies it manages. Therefore, a load balancer can be regarded as a DNS lookup inspector which distributes overheads to different DI proxy components according to certain policies. The policies adopted by a load balancer can be various. For instance, a load balancer may require all the packets from a LH be processed by a same HIP proxy while other balancers expect all the packets for a HIP host to be processed by a same HIP proxy.

#### 4.1.1. Load Balancer Supporting DI Proxy Components

In a LBM where a load balancer manages multiple DI-HIT proxy components, the load balancer must be able to intercept, and forward the information about the HIP hosts being queried to the appropriate proxy components. Additionally, the load balancer needs to modify DNS lookup packets and returns HITs in DNS answers to LHs (or resolvers). In order to intercept the packets sent from LHs to HIP hosts, the load balancer may need to advertise a route of HITs.

In a LBM where a load balancer manages multiple DI-NAT proxy components, the load balancer must be able to intercept, and forward the information about the HIP hosts being queried to the appropriate proxy components. Additionally, the load balancer needs to modify DNS answers and returns IP addresses in the address pools of the assigned DI-NAT proxies in DNS answers to LHs (or resolvers). DI-NAT proxies can advertise the routes of the IP addresses in the pools so that the load balancer does not have to intercept the packets between LHs and HIP hosts.

In a LBM where a load balancer manages multiple DI-transparent proxy components, the load balancer must be able to intercept, and forward the information about the HIP hosts being queried to the appropriate proxy components. The load balancer does not modify DNS answers, but it needs to be located in a place( e.g., the egress of the private network) where it is able to intercept the packets sent to HIP hosts and forward them to the assigned proxies.

#### 4.1.2. Load Balancer Supporting N-DI Proxies

When the HIP proxies that a load balancer manages are N-DI proxies, the load balancer must be able to intercept and modify DNS lookups packets. Additionally, the load balancer must be located in a place ( e.g., the egress of the private network) where it is able to intercept the packets sent to HIP hosts and forward them to the appropriate proxies. In this solution, the load balancer does not forward the information about the HIP hosts being queried to the appropriate proxies. The N-DI proxies need to consult resolution systems themselves.

## 4.2. LBMs without Load Balancers

Generally, in a LBM without a load balancer, there are two methods to distribute communication between LHs and HIP hosts among different HIP proxies. The first solution is to divide the LHs in the private network into different groups (e.g., according to their IP addresses), and the LHs in different sections are taken in charge of by different HIP proxies. The second solution is to divide the HIP hosts in the Internet into multiple groups (e.g., according to their HITs or IP addresses), every HIP proxy serves all the LHs in the private network but only take in charge of the packets to and from the HIP hosts in a group. Abstractly, the two solutions are identical. However, the first solution requires a private network to be divided into multiple sub-networks, and each of them is served by a HIP proxy. This may introduce additional modification to the topology of the private network, which is not desired in many cases. Therefore, in the design of existing LBM solutions, the second type of solution can be more preferred. In the remainder of this document, we mainly consider the second one.

### 4.2.1. Issues Caused by Intercepting DNS Lookups

```
+--------------------+           +------------------+
|                    |           |                  |
|                +---+-------+   |                  |
| +-----------+  |HIP proxy 1+---+      +---------+ |
| |Legacy Host|  +---+-------+   |      |HIP Host | |
| +-----------+      |     .     |      |  (HH1)  | |
|                    |     .     |      +---------+ |
|                +---+--------+  |                  |
|                |HIP proxy n +--+                  |
|Private Network +---+--------+  | Public Network   |
|                    |           |                  |
+--------------------+           +------------------+
```
Figure 1: An example of LBM

Figure 1 illustrates a simple LBM. In this mechanism, n proxies are deployed at the border of a private network. If such proxies are DI-HIT proxies, in order to share the overheads in processing data packets, each proxy needs to advertise a route of the HIT section it takes in charge of. In addition, each proxy also needs to advise a route of a section of IP addresses (or a default route for the entire IP address namespace) inside the private network to intercept DNS lookups. A problem occurs when the DNS lookups and the data packets sent by a legacy host are intercepted by different proxies. In such a case, the proxy intercepting a data packet will lack essential knowledge to correctly process it. If the proxies adopted in Figure 1 are DI-transparent proxies, then each proxy only needs to advertise a route of a section of IP addresses which is adopted to intercept both DNS lookups and data packets. On this occasion, if a HIP host and the DNS server maintaining its RR fall into two different IP sections, the DI-transparent proxy intercepting the lookups for the HIP host will not be the one intercepting subsequent data packets.

A candidate solution to the problem that DI-HIT-proxy-based LBMs and DI-transparent-proxy-based LBMs face is to propagate the mapping information obtained from DNS lookups amongst HIP proxies. Therefore, after intercepting a DNS conversation, a proxy can forward the gained information to the proxy expected to process the subseq uent data packets. Alternatively, a proxy can attempt to collect required information from resolution systems after intercepting a data packet. This approach, however, imposes addition overheads to DI-proxies in communicating with resolution servers.

If the proxies in Figure 1 are DI-NAT proxies, the problem can be eliminated. In a DI-NAT-proxy-based LBM, each DI-NAT proxy needs to advertise two routes, one of the IP addresses in the pool and one of a section of IP addresses for intercepting DNS lookups. After intercepting a DNS lookup, a DI-NAT proxy will return an IP address within the pool in the answer to the requester (a LH or a resolver), which can ensure the subsequent data packets will be transported to the same proxy.

If a DNS resolver supporting DI proxies can forward the mapping information obtained from DNS lookups to appropriate HIP proxies, the issue can be easily addressed. In this case, the DNS resolver actually acts as a load balancer.

### 4.2.2. Issues with LBMs in Capturing and Processing Replies from HIP hosts

Theoretically, when representing a LH to communicate with a HIP host in the public network, a HIP proxy can use either an IP address it possesses or the IP address of the LH as the source address of the packets forwarded to the HIP host. However, in practice, the later option may cause an asymmetric traffic issue in the load balancing scenarios where multiple HIP proxies provide services for a same group of LHs. Assume there are two HIP proxies located at the border of a private network. If the proxies adopt the later solution, they need to advertise the routes of the LHs in the public network respectively. As a result, it is difficult to guarantee the packets transported between a legacy host and a HIP host are stuck to a same HIP proxy, and thus after a proxy intercepts a packet it may lack the proper HIP association to process it.

A possible solution to address this problem is to share HIP state information (e.g., HIP associations, sequence number of IPsec packets) amongst the related HIP proxies in a real-time fashion. However, during communication, some context information such as the sequence numbers of IPsec packets can change very fast. It is infeasible to synchronize the IPsec message counters for every transmitted or received IPsec packet, since such operations will occupy large amounts of bandwidth and seriously affect the performances of HIP proxies. [Nir 2009] indicates that this issue can be partially mitigated by synchronizing IPsec message counters only at regular intervals, for instance, every 10,000 packets.

An issue similar with the one mentioned above is discussed in [TSC05], and an extended HIP base exchange is proposed. But the proposed solution only tries to help HIP-aware middle boxes obtain the SPIs used in a HIP base exchange and cannot be directly used to address the issue mentioned above.

When adopting the preceding option, proxies need to advertise the routes to their addresses in the public network respectively, and so the packets transported between a LH and a HIP host are intercepted by the same proxy. The issue discussed above can thus be addressed. In the following discussions, without mentioning otherwise we assume that a HIP proxy uses one of its IP addresses as the source IP address of a packet which it sends to a HIP host.

## 5. Issues with LBMs which also Support HIP Hosts to Initiate Communication

Apart from the basic functions (i.e., supporting LHs to communicate with HIP hosts), in many typical scenarios, HIP proxies may also need to facilitate the communication initiated by HIP hosts. In this section, we attempt to analyze the issues that a HIP proxy has to face in the conditions where HIP hosts proactively initiate communication with legacy hosts.

In order to support the communication initiated by HIP hosts, the HIP proxies of a private network should have the knowledge essential to represent the LHs to perform HIP BEXs. Such knowledge consists of the IP addresses of the legacy hosts, their pre-assigned HITs, the corresponding HI key pairs, and any other necessary information. In addition, such information of the LHs should be advertised in resolution systems (e.g., DNS and DHT) as HIP hosts. Otherwise, a HIP host has to obtain the HIT of the LH in the opportunistic model which, however, should only be adopted in secure environments.

### 5.1. DNS Resource Records for ML Hosts

In practice, the AAAA RR of a LH can consist of either the IP address of the LH or the address of its HIP proxy. In the preceding approach, the routing infrastructure will try to forward the packets for the LH to the host directly. Therefore, in this case, HIP proxies must be located on the path of such packets to intercept them. In the later approach, the packets for a legacy host are firstly forwarded to the associated HIP proxy. Compared with the preceeding approach, the later case enable a proxy then to be deployed more flexibly and to be more efficient in private networks where legacy hosts and HIP hosts are deployed in an intermixed way, since the HIP proxy will not have to intercept the packets transported between HIP hosts. However, the later approach may cause problems when processing packets sent by legacy hosts in the public network. Normally, a HIP proxy needs to serve a number of LHs. When using the later approach, the packets destined to these LHs will have a same destination address (i.e., the IP address of the proxy). Therefore, when receiving a packet from a legacy host located in the public network, the proxy may find it difficult to identify the LH which the packet should be forwarded to.

A simple approach which combines the advantages of the above two solutions but avoids their disadvantages is to extend the RDATA field in HIP RRs [RFC5205] with a new proxy field, which contains the IP address of a HIP proxy. In the extended HIP RR of a LH, the proxy field consists of the IP address of its HIP proxy, while the proxy field in the RR of an ordinary HIP host is left empty. Therefore, a HIP host intending to communicate with the LH can obtain the IP address of the proxy from DNS servers and set it as the destination address of the packets. The packets are then routed to the proxy. When a non-HIP host intends to communicate with the legacy host, it can obtain the IP address of the legacy host from the AAAA RR as usual and set it as the destination address of the packets; the packets are then transported to legacy host directly.

It is also possible to use the RVS field in a HIP RR to transport the information of a HIP proxy. However, in certain scenarios, a special proxy field can bring additional benefit in security. For instance, it is normally assumed that the BEX protocol is able to establish a security channel for the hosts on the both sides of communication to securely exchange messages. However, this presumption may be no longer valid in the presence of HIP proxies, as the messages between legacy hosts and proxies can be transported in plain text. With the Proxy field, it is easy to distinguish the legacy hosts made up by HIP proxies from the ordinary HIP hosts. Therefore, a HIP host can assess the risks of exchanging sensitive information with its communicating peers in a more precise way.

## 5.2. An Asymmetric Path Issue

In a load balancing scenario where multiple HIP proxies are deployed at the border of a private network, the packets transported between a legacy host and a HIP host may be routed via different HIP proxies. Therefore, when a packet is intercepted by a HIP proxy, the proxy may find that it lacks essential knowledge to appropriately process the packet. Hence, an asymmetric path issue occurs.

In order to explain the asymmetric path issue in more detail, let us revisit the LBM illustrated in Figure 1. In addition, assume that the HIP proxies are DI-HIT proxies and their IP addresses are maintained in the DNS RRs of the LHs. When a HIP host (e.g., HH1) looks up a legacy host at a DNS server, the DNS server returns the IP addresses of all the HIP proxies in an answer (see Figure 2). Upon receiving the answer, HH1 needs to select an IP address and sends an I1 packet to the associated HIP proxy. Assume the HIP proxy 1 is selected. Then after a base exchange, HIP proxy1 and HH1 establish a HIP association respectively. Upon receiving the first data packet from HH1, the HIP proxy uses the HIP association to de-capsulate the packet and forwards it to the legacy host. In the forwarded packets, the HIT of HH1 is adopted as the source IP address, and thus the HIT of HHI is adopted as the destination address in the reply packets sent by the legacy host. Assume that the HIT of HH1 is within the section managed by HIP proxy n. According the routes advertised by the proxy n, the packet is forwarded to the HIP proxy n which, however, does not have the corresponding HIP association to deal with the packet. Similarly with DI-HIT proxies, DI-transparent proxies and N-DI proxies also suffer from the asymmetric path issue in the load balancing scenarios, since they cannot guarantee the data packets which are transported between a legacy host and a HIP host stick to a single HIP proxy too.

```
+----------------------+         +---------------------------+
|                      |         |                           |
|                  +---+-------+ | (3)                       |
|            (4)  -|HIP proxy 1+-+<-                         |
|                / +---+-------+ |  \ +--------+ (1)+------+|
|+-----------+< -      |     .   |   -|HIP Host|--> |  DNS ||
||Legacy Host|-        |     .   |    |  (HH1) |<-- |Server||
|+-----------+ \   +---+-------+ |    +--------+(2) +------+|
|           (5) - >|HIP proxy n+-+                           |
| Private Network  +---+-------+ |    Public Network         |
|                      |         |                           |
+----------------------+         +---------------------------+
```

Figure 2. An example of the asymmetric path issue

As we mentioned in section 3.3.1, the approach of synchronizing HIP associations and IPsec associations amongst HIP proxies can be used to address this issue. However, this issue will introduce additional communication overhead on HIP proxies. Here, we discuss several other alternative solutions.

The simplest solution is to allow a HIP proxy to discard the I1 packets it receives if they are not original from HIP hosts which the proxy takes in charge of. In addition, the proxy can inform the senders of the incidents using ICMP packets. Therefore, after waiting for a certain period or upon receiving a ICMP packet, a HIP host will try to select another HIP proxy from the list in the DNS answer and send an I1 packet it. In the worst case, this process needs to be recursive until all the HIP proxies in the list have been contacted. Because a HIP host may have to send the multiple I1 packets in order to communicate with a LH, this solution may yield a long delay. Note that in some DNS based load balancing approaches, a DNS server only returns one HIP proxy in an answer. On such occasions, HIP hosts have to communicate with DNS servers repeatedly, and the negative influence caused by the communication delay can be even exacerbated.

A solution which is able to avoid the delay issue is to endow DNS servers with the capability of returning the IP address of an appropriate HIP proxy in an answer according to the HIT (if the proxy is a DI-HIT proxy or a N-DI proxy) or the IP address (if the proxy is a DI-transparent proxy) of a requester. That is, the HIP proxy described in a DNS answer should take in charge of the namespace section which the requester belongs to. In order to achieve this, DNS servers need to 1) maintain the information about the sections of the namespaces that HIP proxies take in charge of, 2) locate the appropriate HIP proxy according to the HIT or the IP address of a HIP requester. These requirements result in modifications to current DNS servers in the implementation of the DNS server applications and the conversation protocols between requesters and DNS servers. For instance, a HIP host may need to transport its HIT in DNS requests in order to help DNS servers locate an appropriate HIP proxy. An negative impact of this solution is to introduce additional complexity and overhead to DNS servers.

Another solution is to extend RVS servers as load balancers. After receiving an I1 packet from a HIP host, the load balancer then select an proper HIP proxy and forward the packet to it. Using this solution, a DNS server only needs to reply a record on receiving a query from a HIP host, which reduce the traffic transported between DNS servers and HIP hosts.

The asymmetric path issue can be eliminated when DI-NAT proxies are adopted. A DI-NAT proxy located at the border of a private network maintains a pool of IP addresses which are routable in the private network. After receiving a packet from a HIP host, the DI-NAT proxy processes the packet and forwards it to the destination legacy host. In addition, an IP address selected from the pool is adopted as the source address of the packet. Therefore, when the legacy host sends responding packets to the HIP host, the packets will be transported to the same HIP proxy. The asymmetric path issue is thus eliminated.

# 6. Issues with LBMs in Supporting Dynamic Load Balance and Redundancy

In practice, there are requirements for LBMs to support dynamic load balance and redundancy. That is, when a proxy in a LBM is not able to work properly or the overheads imposed on it surpass a threshold, the proxy can delegate all of (or a part of) its job to other proxies. A proxy provide backup sevice is called a backup proxy, and the proxy served by a backup proxy is called a primary proxy. Note that two proxies can be backup proxies for each other on different jobs. In this section, we analyze the performance of different types of HIP proxies in supporting dynamic load balance and redundancy.

If there is a load balancer intercepting and distributing traffic among different proxies, the balancer can flexibly forward traffic to other proxies when a proxy cannot work properly. However, if there is no load balancer deployed, in order to provide backup services, a backup proxy has to advertise the same routes as those advertised by the primary proxy in both the private and the public networks. To avoid affecting the normal operations of the primary proxy, the routes advertised by the backup proxy have a much higher cost than that of the routes advertised by the primary proxy. When the abnormal conditions mentioned above occurs, the primary proxy can withdraw the routes it previously advertised so that the packets supposed to be processed by the primary proxy will be forwarded to the backup proxy. We refer to the routes advertised by a proxy for backup purposes as the backup routes of the proxy. In contrast, we refer to the routes advertised by a proxy to achieve its primary job as the primary routes of the proxy. In practice, the proxies in a LBM can provide backup services for one another. Therefore, a proxy in such a LBM may needs to advertise both primary and backup routes.

The synchronization of state information between primary and backup proxies is also very important. Without proper HIP associations, a backup proxy cannot correctly take place of the primary proxy to process the packets. The state synchronization problem has been discussed above. If there is no state synchronization, a backup proxy may select to send signaling packets to HIP hosts to initiate new HIP BEXs.

In the remainder of this section, we discuss the operations of different types of HIP proxies in achieving dynamic load balance and redundancy without the assistance of load balancer.

### 6.1. Application of DI-HIT proxies in supporting dynamic load balance and redundancy

As mentioned in section 3.1, a DI-HIT proxy needs to at least advertise two primary routes in the private network, a route of a section of HITs for intercepting data packets, and a route of a section of IP addresses for intercepting DNS lookups. When the proxy cannot work properly, it can withdraw both routes to enable a backup proxy to take over its job.

In some cases, a DI-HIT proxy may only want to delegate a part of its job to others so as to reduce the overloads it undertakes. To achieve this objective, the proxy can divide its routes into multipe more detailed routes. When the overload on the proxy is high, it can only withdraw a subset of the routes. For instance, a DI-HIT proxy can selectively only delegate a part of the responsibility in processing DNS lookups to a backup proxy by withdrawing one of its lookup intercepting routes.

### 6.2. Application of DI-NAT proxies in supporting dynamic load balance and redundancy

A DI-NAT proxy needs to at least advertise two primary routes in the private network, a route for its IP address pool, used to intercept data packets, and a route for an IP address section used to intercept DNS lookups. When the proxy cannot work properly, it can withdraw both routes to enable a backup proxy to take over its job. In this case, the delegated backup proxy needs to maintain an IP address pool identical to the one maintained by the primary proxy. Moreover, apart from synchronizing HIP associations, the synchronization of mappings from IP addresses to HITs is also required. Otherwise, the backup proxy cannot translate the received packet correctly.

If a DI-NAT proxy only intends to maintain existing communication between LHs and HIP hosts while not facilitating any more, it can withdraw the lookup intercepting route. As mentioned previously, DI-NAT proxies have the capability to stick the DNS lookups and the subsequent data packets to the same proxy. Therefore, the backup proxy can intercept DNS lookups as well as process the subsequent communication.

### 6.3. Application of DI-transparent proxies in supporting dynamic load balance and redundancy

Unlike DI-HIT and DI-NAT proxies, the routes advertised by a DI-transparent proxy are used for intercepting both DNS lookups and data packets. Therefore, before a DI-transparent proxy withdraws a route, it needs to synchronize the states of the on-going communication affected by the routing adjustment to its backup proxies.

## 7. Conclusions

This document mainly analyzes and compares the performance of different kinds of HIP proxies in LBMs. Amongst the HIP proxies discussed in the document, DI-NAT proxies show its advantages in multiple scenarios. In addition, we argue that the state synchronization among HIP proxies is very important to achieve load balancing and redundancy. There is a topic which is important but not covered in this document is the compatibility among different HIP proxies. The different types of HIP proxies are designed based on different presumptions. The presumptions of different type of HIP proxies maybe conflict with each other. Then how to make a trade-off and enable different types of proxies work cooperatively is an important issue that the designers of HIP extensible solutions have to consider.

## 8. IANA Considerations

This document makes no request of IANA.

## 9. Security Considerations

One design objective of HIP is to provide peer-to-peer security between communicating hosts. However, when a HIP host communicates with a LH under the assistance of a HIP proxy, the security of the communication between the HIP proxy and the LH may not be protected. If the HIP proxy is transparent to the HIP host, the host will believe that it is communicating with a ordinary HIP host and will not realize that the peer-to-peer security between it and the LH is not guaranteed. This may cause potential security risks, especially when the HIP proxy is located in the public network. Therefore, some solutions should be provided for a HIP hosts to detect whether they are actually communicating with HIP proxies.

When sharing HIP state information amongst HIP proxies, the integrity and confidentiality of the state information should be protected. The discussion about the similar issues can be found in [Nir 2009] and [Narayanan 07].

If a HIP proxy is deployed at the border of a private network or within the boundary of a private nework, the security issues with the communcation between the proxy and LHs are not serious. However, if a proxy is deployed in the public network, both the communication between LHs and the proxy and the communication between the proxy and DNS servers should be secured.

## 10. Acknowledgements

Thanks Thomas.R.Henderson for his kindly prove-reading and precious comments.

## 11. References

### 11.1. Normative References

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

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[RFC5205] Nikander, P. and J. Laganier, "Host Identity Protocol (HIP) Domain Name System (DNS) Extensions", RFC 5205, April 2008.

[RFC5338] Henderson, T., Nikander, P., and M. Komu, "Using the Host Identity Protocol with Legacy Applications", RFC 5338, September 2008.

### 11.2. Informative References

[Narayanan 07]
Narayanan, V., "IPsec Gateway Failover and Redundancy - Problem Statement and Goals", 2007.

[Nir 2009]
Nir, Y., "IPsec High Availability Problem Statement", 2009.

[PAT07] Salmela, P., Wall, J., and P. Jokela, "Addressing Method and Method and Apparatus for Establishing Host Identity Protocol (Hip) Connections Between Legacy and Hip Nodes, US. 20070274312", 2007.

[SAL05] Salmela, P., "Host Identity Protocol proxy in a 3G system", 2005.

[TSC05]    Tschofenig, H., Gurtov, A., Ylitalo, J., Nagarajan, A., and M. Shanmugam, "Traversing Middleboxes with the Host Identity Protocol", 2005.

## Authors' Addresses

Dacheng Zhang
Huawei Technologies Co.,Ltd
HuaWei Building, No.3 Xinxi Rd., Shang-Di Information Industry Base, Hai-Dian District
Beijing,   100085
P. R. China

Phone:
Fax:
Email: zhangdacheng@huawei.com
URI:

Xiaohu Xu
Huawei Technologies Co.,Ltd
HuaWei Building, No.3 Xinxi Rd., Shang-Di Information Industry Base, Hai-Dian District
Beijing,   100085
P. R. China

Phone:
Fax:
Email: xuxh@huawei.com
URI:

Jiankang Yao
CNNIC
4, South 4th Street, Zhongguancun
Beijing,   100190
P.R. China

Phone:
Fax:
Email: shenshuo@cnnic.cn
URI: