Network Working Group
Internet-Draft
Expires: January 13, 2005

P. Nikander
Ericsson Research Nomadic Lab
T. Henderson
The Boeing Company
July 15, 2004

Considerations on HIP based IPv6 multi-homing
draft-nikander-multi6-hip-01

Status of this Memo

Copyright Notice

Abstract

The Host Identity Protocol implements the identifier locator separation by introducing a new name space and a new layer to the IP stack. The new structure insulates the transport layer protocols from the internetworking layer, thereby allowing transport sessions to remain unaffected even if the underlying IP addresses change. That, in turn, seems to make it easier to solve the so called site multi-homing problem than without introducing such an indirection layer.

This document discusses how the proposed HIP mobility and multi-homing solution, described separately, would fit to the IETF

multi6 working group requirements.

Table of Contents

1. Introduction

The IETF multi6 working group is currently calling various alternative solutions as components for an architectural analysis. The aim of that work is to try to understand the architectural design choices and their tradeoffs.

This document discusses how a Host Identity Protocol (HIP) based approach could solve the multi6 site multi-homing problem. The draft also presents some ideas on how the HIP architecture could be split into components, some of which could be applied to the multi6 problem without adopting all of the current HIP proposal.

1.1 Background

The Host Identity Protocol (HIP) is a proposal for changing the TCP/IP stack architecture by introducing a new name space and a new protocol layer to the stack. The overall design is discussed in the HIP architecture document [3]. The actual protocol details are defined in the HIP protocol specification [2], and the mobility and multi-homing related extensions in the HIP mobility and multi-homing specification [4]. It is expected through this document that the reader is at least superficially familiar with the architecture document and the protocol specifications.

The proposed HIP multi-homing mechanism [4] is primarily aimed to be a host multi-homing solution. Basically, it allows two end hosts to inform each other of their IP connectivity. That is, an end-host sends a set of IP addresses to its peer, and the peer makes sure that the end-host is reachable through (some of) these IP addresses.

In the baseline HIP solution, a HIP base exchange protocol is run each time a new pair of hosts starts to communicate. This protocol is a four-way handshake, requiring public key cryptographic operations. While such a heavy exchange makes sense for applications where the hosts have a fairly long term relationship, e.g. for e-mail, disk access, etc., it may be too heavy for short term transactions, such as some forms of web browsing etc. It is definitely unsuitable for DNS (see Section 5.2.4.1). Therefore, the architecture has been defined in such a way that each application can be configured either to use HIP or to use legacy IP, without the HIP overhead (and, of course, without any benefits from HIP).

While the redundancy (Section 5.3.1.1) and especially transport layer survivability (Section 5.3.1.6) make sense mostly for long term transport sessions, where the state setup may be amortized over a longer period of time, it would be benficial to avoid such a large state setup cost. Therefore, in Section 3, we also describe how some

components of HIP could be applied to the multi6 site-multihoming problem without adopting all of HIP.

1.2 Current venues for HIP work

HIP is being developed in an IETF Working Group within the Internet Area, and a parallel IRTF Research Group. The charter of the IETF Working Group is to develop a minimal set of specifications that can enable HIP experimentation on a wide scale, including the base protocol, DNS resource record definition, and basic mobility and host multihoming. The charter of the IRTF Research Group is to study the potential effects on the Internet of wide scale HIP deployment, and more broadly, the consequences of wide scale adoption of any type of separation of the identifier and locator roles of IP addresses.

It is currently expected that if the HIP mobility and multi-homing solution, or some aspects of it, are selected for further work at the multi6 working group, then the resulting work will be chartered at the multi6 WG. In that case, some co-operation with the HIP WG and the IRTF HIP RG is needed.

1.3 Baseline HIP multi-homing mechanism

The baseline HIP multi-homing mechanism is specified in [4]. Here we briefly summarize the mechanism, giving an outline to those impatient readers that don't have cycles to read the full HIP specifications.

As mentioned above, when two HIP hosts start to communicate, they run the HIP base exchange and create an HIP association. As a part of this association, a multi-homed host MAY send a list of IP addresses that it believes to belong to itself. The recipient of these addresses stores them as potential addresses of the peer. Before using any of new addresses, it SHOULD verify that the peer is indeed reachable through the address. This verification MAY be skipped if the peer host is fully trusted; see [4] for details. According to the current specification, an end-host MAY perform such reachability test at any time, subject to its local policy. Such a reachability test requires only the tested address to work, meaning that such a test can be delayed until the other address(es) become unreachable.

The hosts are free to change the information about their addresses at any time. However, note that especially in the case of site multi-homing, one of the addresses may become unreachable while the other one still works. In the typical case, however, this does not require the host to inform its peers about the situation, since even the non-working address still logically exists.

Establishing the initial multi-addressing situation, and all changes

to that, are protected with strong cryptography. There are no known vulnerabilities in the specified mechanisms. (Note, however, that the fact that there are no _known_ vulnerabilities does not mean that there are no unknown ones. There might be, and given the freshness of the specifications, there probably are.)

The current specification outlines a method where one of the peer's addresses is considered as a primary address. By default, all traffic is sent using this address. This practise is similar to how SCTP multi-addressing works, and is designed to work well with current transport layer congestion control. However, the HIP architecture itself would allow multiple addresses to be used in parallel, even for one transport session. Experimentation of such practise is assumed to take place at the IRTF HIP RG.

2. HIP as a site-multi-homing solution

As mentioned above, HIP multi-homing is primarily designed as a solution for multi-homed end-hosts. As such, it offers a multi-address based baseline solution, similar to other multi-addressing based multi6 proposals. Efforts to adopt the approach to site multi-homing, especially in the case where some hosts within the site and outside of the site may be legacy non-HIP hosts, has been fairly minimal.

It is expected that most of what applies to other multi-addressing based multi6 proposals apply also to HIP.

Since the multi-homing aspects of HIP do not seem to considerably differ from other multi-address based proposals, the focus in this section is on the factors that differentiate HIP from the other solutions. Multi-homing aspects are covered in Section 5.

2.1 Hiding of underlying IP version

HIP hides the underlying IP version from applications. That is, an IPv4 legacy application can be run over IPv6, and vice versa, HIP acting as an insulation layer. This also means that the HIP mobility and multi-homing solution allows existing transport sessions to change their underlying connectivity from IPv4 to IPv6 and vice versa, as long as both end-hosts remain reachable (either directly or through a gateway).

2.2 Integrated mobility

The HIP multi-homing mechanism is fully integrated with mobility. In fact, the two mechanism are so integrated that it would be very hard to make them separate.

2.3 Architectural support for multi-realm connectivity

The HIP architecture allows HIP associations to be routed through layer 3.5 middle boxes, thereby extending the associations across multiple IP realms. In other words, HIP would allow controlled NAT-traversal that does not have the ill effects of the current NAT practise. However, fully realising such service requires more work, and is subject to study at the IRTF HIP RG.

2.4 Integrated, mandatory end-to-end security

HIP has its origin as a security solution, aiming to simplify IPsec administration and to address mobility at the same time. Hence, HIP as defined today, requires that all payload traffic is protected with

IPsec ESP. By default, this adds the overhead of carrying the ESP header and trailer in all packets. Additionally, the current IPsec specifications mandate that either encryption or integrity protection of ESP MUST be used, i.e., it is not allowed to use IPsec without encryption and without integrity protection. Hence, all HIP packets are subject to the cost of symmetric crypto processing on both sending and receiving ends. While this cost is fairly minor in most modern architectures, it may have negative effects on small devices, such as PDAs, and large scale servers.

In addition to the header overhead and computational cost, ESP breaks some middle box functionality by making it impossible to inspect and/or modify the packet contents.

It should be noted that when end-to-end security is desirable, HIP adds no additional overhead compared to using standard IPsec mechanisms. Hence, for applications were IPsec based security is adequate and desirable, HIP looks like an optimal or near-optimal multi-homing solution.

In Section 3.1, below, we discuss how ESP could be replaced with other mechanisms for the case where end-to-end security is not needed for payload traffic.

2.5 High state setup cost

The HIP base exchange is a four-way cryptographic authentication protocol, implementing a sigma [11] authenticated Diffie-Hellman exchange, with state-space and CPU-exhaustion denial-of-service protection. The initiator performs one public-key (DSA) signature and two signature verifications, while the responder performs one or two signatures and one verification. A single protocol run requires a few long integer exponentiations, taking a fraction of a second on modern CPU architectures.

2.6 Comparison with other Group-F multi6 proposals

HIP has been classified into one of a set of multi6 proposals, known as "Group F", that propose a "Wedgelayer 3.5 / Fat IP" solution as shown in Figure 1.

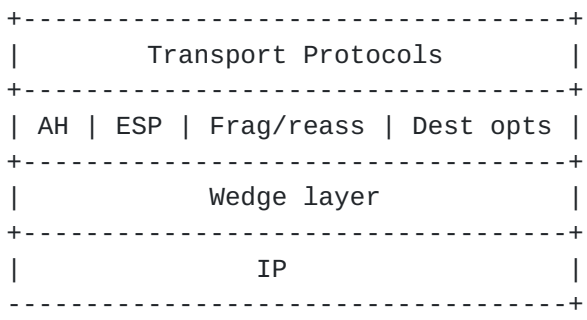

Figure 1: Protocol stack

Recently on the multi6 mailing list, a number of these Group F proposals have been contrasted with respect to how they make use of Application Identifiers (AIDs), Context Identifiers (CIDs), Context Identification Tags (CIDTs), and IP layer locators. As presently specified in [2], HIP uses Host Identity Tags (HITs), which are hashes of the host identity public keys, as both AIDs and CIDs, and uses IPsec SPIs as CIDTs.

Compared to other multi6 proposals, the state setup cost of HIP seems to be largest. In Section 3.2 we discuss how this state setup cost might be delayed to a later moment, allowing two hosts to start communicating and creating the state only if they later determine that they want security or want to change active IP addresses. In Section 3.2.1 we discuss how the security properties of such delayed state setup might be improved with hash chains.

In HIP, the AIDs have strong cryptographic properties but are not routable, which can cause problems for application level referrals. In Section 3.3, we discuss the potential for making HIP AIDs routable.

3. Using components of HIP or modified HIP instead of full HIP

In this section, we first briefly describe how HIP could be modified so that it would impose less computational overhead, and then how some HIP-like ideas could relate to other multi6 proposals, and vice versa.

3.1 Using HIP without IPsec ESP

As mentioned above, some applications may have long lasting connections that would benefit from redundancy and transport layer survivability, but would not need end-to-end security. DRM protected video streams using application level encryption might be one such example. In cases like that, it would be beneficial to use HIP without ESP.

There seems at least to be two different ways how HIP could possibly be used without ESP:

1. One possibility is to replace the ESP header with a simple header that carries the SPI, similar to the M6 header proposed in the SIM proposal [9]. In the HIP case, the context tag (CIDT) would be the SPI.

2. Another possibility would be to allocate a single bit in the IP header, indicating whether in the receiving end the source and destination locators should be rewritten into identifiers or not. This would be somewhat similar to the NOID [7] and CB64 [8] proposals. When ESP is not used, there is no replay protection, and therefore there is no need for multiple parallel SAs. In this case, the CIDTs would be based on the IPv6 flow IDs and the locators.

Adding support for non-ESP communication would add a need for policy into HIP. For each connection, the end-points would need to decide whether to use ESP or not. Since one of the current HIP goals has been and still is simplicity, this feature has not been added to the current HIP specifications. From a functional point of view, the possibility of not using ESP is a mere performance optimization.

3.2 Delaying HIP state setup

It might be possible to delay the actual HIP state setup. However, it would not be possible to use ESP before the state has been established. The main benefit from this kind of optimization is to initially avoid the computational cost.

This variant is tentatively called LHIP, for Lightweight HIP.

The idea goes as follows:

1. The Initiator sends an I1, just like in the current specification. However, the I1 packet would contain an extension to indicate that it wants to delay state setup. It could also piggyback the initial payload, e.g., TCP SYN.

2. The Responder checks it policy to see if it allows delayed state setup for the HITs and IP addresses in I1. If it doesn't, it sends an R1 as usual, and forgets about the packet. On the other hand, if it does allow delayed state setup, it generates a new nonce for lightweight state set up. The lightweight state will consist of the HITs, the IP addresses, and the nonce. The recipient will remember these, either by storing them or algorithmically, for a limited delta period. The Responder replies with a new packet (LR1), which contains the HITs and the nonce. A TCP SYN ACK may be piggybacked on the LR1.

3. When the Initiator receives the LR1, it stores the nonce, and sends an LI2, containing just the HITs and the nonce.

4. When the Responder receives the LI2, it knows that some node is reachable at the given IP address. However, it has no assurance that the host is actually the one identified by the HIT. Hence, the HIT cannot be used for access control or any other security purposes -- any node might have claimed to be identified by the HIT.

If the Initiator later wants to move or use ESP, it must update the lightweight state to a full HIP association. Similarily, if the Recipient later wants to move, use ESP, or open a new transport session in the reverse direction, it must update the state. Note that the notion is asymmetric here: the Recipient must update the state also in the case of opening new transport sessions, since it has no assurance that the other host actually "owns" the given HIT.

It is noteworthy that this lightweight setup is completely insecure, allowing the initator to use any HIT as an identifier for itself. On the other hand, it adds very little overhead to the setup of an initial connection, allowing the TCP three-way handshake to be piggybacked on the protocol.

3.2.1 Securing LHIP state setup

Based on some initial discussions, it may be feasible to bring some security to the LHIP state setup using hash chains. However, futher analysis would be needed. For an example of how hash chains could be used for securing multi6 related state setup, see [12].

If such hash chains (or something similar) was added to the creation of the lightweight state, the state could probably be used for securing changes in the mobility and multi-homing state of the hosts. However, it would not be sufficient for creating ESP SAs.

3.3 Using HIP with routable AIDs

As mentioned above in Section 2.6, HIP completely decouples upper layer protocols (ULP) from IP layer locators. Specifically, HITs are used in the transport layer pseudoheader for checksum computations, and HITs may be passed to the application, depending on the implementation. Because locators are not used in the checksum, the transport layer entities cannot communicate until a protocol exchange establishes the AIDs to use for the session. Because HITs may be passed to the application, application level referrals may cause a HIT, treated as a locator by the application, to be passed to another non-multi6-aware host.

It seems possible to modify HIP to use routable AIDs rather than HITs as the AIDs. One such possibility would be to use the lower-order bits of the HIT as the interface ID of the primary locator of the host, which is then combined with the subnet prefix to form a routable AID. This type of AID has cryptographic authentication properties, although weaker than those of full HITs, and somewhat more susceptible to collisions. Such a change does not seem to detract from the cryptographic properties of a full HIP base exchange, which could be conducted later or upon detection of collision, or initially as presently defined in [2]. In combination with a lighter-weight initial exchange, this could make the protocol very similar to CB64 [8]. Note also that there is no requirement to use a crypto-based locator; if an initial exchange establishes context that prevents session theft, such as described in the WIMP [12] proposal, any type of AID may be used. This type of operation may have certain privacy advantages.

The change required to the HIP base specification would be to use the initial locators as the AIDs of the transport layer checksums. Note that HITs may still be passed to applications for HIP-aware applications, but HITs would no longer be passed in place of locators to non-HIP-aware applications.

4. Discussion

4.1 Default router and source address selection

Source address selection must be based on first selecting the outgoing router, based on the current reachability state, and then source address to be used, not vice versa.

4.2 Selecting primary destination address

Section 8.4 of [4] briefly discusses how the hosts should select the primary destination address to use for their peers. It should be noted that the currently presented discussion is probably not the optimal way of solving the problem. Further engineering and maybe some research is required on the topic.

4.3 Reacting to addresses becoming unreachable

In the case of site multi-homing one of the addresses may become unreachable while the other one still works. In the typical case, however, this does not require the host to inform its peers about the situation, since even the non-working address still logically exists.

Brian Carpenter: It's just as well you don't require this notification. The last node to know that an address is unreachable is the node that address belongs to. Unreachability is discovered at the other end of the multihomed session.

5. Evaluation against of RFC 3582 and MULTI6 Solution Questionaire

5.1 Approach

5.2 Answers to MULTI6 Solution Questionaire

5.2.1 Routing

As HIP is implemented on top of IP, it does not directly affect basic IP routing. Routing within any IP realm is performed just as today. The end-hosts maintain a binding table that maps Host Identifiers into a set of IP addresses, and the HIP mobility and multi-homing protocol [4] is used to update these bindings.

5.2.1.1 Primary multi-homing solution idea

The HIP mobility and multi-homing protocol [4] allows an end-host to send information about all of its IP addresses, both IPv4 and IPv6, to its peers. The peers have to check reachability of these addresses prior to using them for sending large amount of traffic. This mechanism allows interacting HIP hosts to establish multi-addressing based multi-homing state.

The exact mechanisms on how a host is supposed to perform address and path selection are not defined in the current HIP specifications. However, the required practise is assumed to be similar to any other multi-addressing based multi-homing solutions. Hence, it is expected that the multi6 WG (instead of the HIP WG) will define the required mechanisms.

Some of the above described variations of HIP allow delayed establishment of the full HIP association. However, the details such practise are currently undefined and there is no implementation experience on the aspect.

5.2.1.2 Uniqueness

[I don't understand what the title of this section refers to.]

5.2.1.2.1 Mobility

HIP addresses mobility. A mobile host sends HIP readdressing information to all of its peer hosts, allowing them to update addressing information.

Initial rendezvous is planned to be started with DNS. An initiating host that wants to contact a mobile host is supposed to look up the Host Identifier and a set of current IP addresses from the DNS. The

set of current IP addresses may include real active addresses of the mobile host, addresses of a Rendezvous server, or both.

Once the initiating host has a tentative set of addresses, it sends an HIP I1 packet to an address. If the address is a real address of the mobile host, the mobile host will directly answer with an R1 packet, and the rest of the HIP base exchange is run between the used addresses. At the end the hosts inform each other about their multi-addressing state.

If the I1 destination address is an address of a Rendezvous server, the Rendezvous server will forward the packet to the currently registered address of the mobile host. The mobile host will send an R1 directly back to the initiating host, and the rest of the HIP base exchange is run directly between the mobile host and the initiating host.

If a mobile host changes its active address while the HIP base exchange is going on, there will be a timeout and the initating host needs to start again, either using another address from the set of addresses received from the DNS, or remaking the DNS query if necessary.

All hosts that use rendezvous servers are assumed to include the rendezvous server address in their active address sets. Hence, if two interacting mobile hosts move at the same time so that the readdressing indications cross each other in the network and get lost, the host will fall back to the rendezvous server address after a timeout. (The length of the timeout is currently unspecified, and subject to local policy of the hosts.) Hence, provided that the hosts have updated their current location to the rendezvous server, the hosts will be able to continue communications.

The HIP mobility mechanism is expected to replace Mobile IP for all communication taking place between HIP enabled hosts. When a HIP host is communicating with a legacy host, it may use Mobile IP, provided that the host stack includes both HIP and Mobile IP implementaitons.

5.2.2 Identifiers and locators

5.2.2.1 Split identifiers and locators

HIP is based on the idea of splitting identifiers and locators. Public cryptographic keys are used as identifiers. IP addresses are used as locators. From the routing point-of-view, IP addresses are used just like today. From the applications and transport layer point of view, identifiers (in the form of HITs and LSIs [3]) replace

IP addresses, unless a change as described in Section 3.3 is made to HIP.

5.2.2.2 Binding lifetime

The lifetime of the binding from an identifier to a locator is defined in the protocol messages. Typically it is equal to the lifetime of the locator. The host creating the binding state simply accepts the lifetime from the sending host.

5.2.2.3 Update of bindings

The bindings are updated by a host sending HIP readdressing paramters, typically in a HIP UPDATE packet. A single packet may update several sets of bindings.

Whenever a new address is associated with an identifier, the hosts must verify the reachability of the address before using the address for payload traffic. This procedure is required in order to block flooding attacks [6].

Updating the bindings have no direct effect on transport connections, which will remain up. Changes in the actual paths may have effects on transport connections, such as changes in QoS.

5.2.3 On the wire

HIP, as currently defined, consists of two protocols. One is a new protocol, the HIP protocol, run directly on the the top of IP. The other one is IPsec ESP. Using telecom terminology, the HIP protocol forms a control plane, and all user plane traffic is encapsulated in ESP.

As discussed above, it might be possible to use HIP without requiring all user plane traffic to be ESP encapsulated. However, such practise has not been defined in detail and there are no implementation experience.

5.2.3.1 Solution layer

HIP can be consider to be a layer 3.5 solution.

HIP is applied to every packet, in the form of encapsulating them into ESP envelopes. The ESP SPI field is used to associate the packet with the right end-point identifiers in the receiving end.

As described above in Section 3.1, it ESP was not used, a single bit in the IP header might suffice to allow the receiving host to

associate HIP and non-HIP traffic with the appropriate sockets. In that case the source and destination IP addresses would be used to associate the packet with the right end-points. This practice has the drawback that does not allow multiple host identifiers to be hosted on a single node.

If the single bit approach is deemed infeasible, it would be possible to create a new extension header that would contain a new demultiplexing field. From the HIP demultiplexing point of view, the contents of the field would be similar to ESP SPI.

5.2.3.2 Correctness of the selected layer

Multi-homing is a phenomenon that clearly appears between hosts, not between applications or transport sessions. Hence, a multi-homing solution should be located at a layer that has host granularity, and not any finer granularity. This leaves out transport and higher layer solutions.

Multi-homing can be considered to be either as an end-to-end or a routing level phenomenon. In the case of end-host multi-homing, where a single host has multiple accesses to the Internet, the situation seems to be best modelled as an end-to-end one. Respectively, the case of intra-transit-provider connectivity, an extreme form of site multi-homing, is probably best modeled as a part of the overall routing topology. Various types of end-site multi-homing (soho...multinational) fall on different locations on this axis.

The IP layer contains both end-to-end and routing functions. Hence, IP layer could implement both end-to-end and routing based multi-homing solutions.

Since HIP introduces a new name space, Host Identifiers, it is best described as a shim or 3.5 layer solution [10]. In other words, it is end-to-end in nature, affecting some of the current IP layer end-to-end functionality, but relies clearly below the transport layer.

A layer 3.5 solution has a number of good properties:

- It is possible to continue using unmodified TCP and UDP.

- It would become possible to move much of the SCTP and DCCP multi-addressing functionality into the new layer. Such functionality would then be shared between them and the legacy transport protocols, TCP and UDP.

The approach would make it easier to collect per-path MTU and RTT information, if seen appropriate from the transport point-of-view.

The approach does not require any changes to the IP layer or the pseudo header. (But see also Section 3.1.)

5.2.3.3 Expansion of packet size

The solution does not cause any expansion of packet size other than that caused by ESP. If ESP is not used, the single-bit solution, outlined above, would allow HIP to be used without any expansion of packet size.

5.2.3.4 Fragmentation

It is expected that HIP solutions report a reduced MTU to upper layer, similar to current ESP practise. Other than that, the standard ESP fragmentation practise is used. The current implementations seem to work, but no-one has performed a detailed analysis.

5.2.3.5 Changes to ICMP error semantics

The current HIP specifications do not create any new ICMP error messages. However, a detailed analysis is needed to see if there are any subtle changes to the current semantics. Such an analysis has not been made.

5.2.4 Names, Hosts, Endpoints, or none of the above?

5.2.4.1 Relationship with DNS

It is expected that the HIT (or the HI) of each HIP host is stored into the DNS, in addition to the IP address(es). When a HIP host starts to connect to another HIP host, it queries for both the HIT/HI and the addresses. If a HIT/HI is received, the initiating host creates a piece of local state, attempts to create a HIP association with the peer upon first connection request. If the association is created, the hosts establish their multi-addressing state directly. The addresses stored in the DNS are not used beyond building the HIP association. If no HIT/HI are received, the initiating host falls back to using legacy IP.

Defining the required new RR type is a working item for the HIP WG.

5.2.4.2 Interactions with 2-faced DNS

Interactions with 2-faced DNS have not been fully analyzed. However, as HIP reduced the applications' dependency on IP addresses, it looks like that HIP would easily allow 2-faced DNS to be used. Furthermore, if there is a proper HIP aware security gateway between the two domains, it should be possible to fully control the creation of HIP associations between the domains.

5.2.4.3 (Non)need for centralized registration

HIP does not require centralized registration. The identifiers are public keys, and typically self-generated.

It is expected that the IRTF HIP RG will study how to provide a service similar to reverse mapping for the public keys.

5.2.4.4 (No) Circular dependencies with DNS

DNS is not expected to use HIP. In a typical implementation, this is accomplised by configuring the DNS proxies and servers to bind/ connect to IP addresses, not HITs.

5.2.4.5 Multihomed DNS servers

Multi-homed DNS servers are expected to continue direct utilization of multiple IP addresses.

5.2.4.6 Application/API changes

Most old code will just work. Multi-party applications doing IP-address-based referrals will break, unless HIP uses routable AIDs as described in Section 3.3. The IRTF HIP RG will study how to support such multi-party applications.

To gain full benefit from HIP, extensions to the current socket API are expected to be needed. However, using such extensions is not required to benefit from the multi-addressing properties of HIP.

5.2.4.7 Backward compatibility and incremental deployment with current IPv6

The current implementations allow full compatibility with the current IPv6, with the exception of using a large but unused part of the IPv6 address space to represent HITs internally. No requirements are placed on non-multihomed, non-mobile legacy hosts.

HIP is designed to be incrementally deployed. It is expected that

HIP capable servers announce their capability to run HIP by listing the new resource record in the DNS. Possibilities to run HIP opportunistically, without DNS, are to be studied at the IRTF HIP RG.

5.2.4.8 Backward compatibility with IPv4

HIP works with both IPv4 and IPv6, even allowing simultaneous use of both IPv4 and IPv6 connections.

It has not been analyzed how HIP interacts with existing 6to4 gateways. Such work is not on the HIP WG charter, but may be pursued at the IRTF HIP RG.

5.2.4.9 Interaction with middleboxes

Firewalls. Since HIP introduces a new control protocol to be run directly over IP, and uses IPsec to secure payload traffic, HIP would break most current firewalls. However, the HIP base exchange and the rest of the control protocol has been carefully designed to be friendly towards future firewalls, allowing HIP aware firewalls to control HIP traffic.

NAT. HIP, as currently defined, does not work with IP-multiplexing NAT boxes. On the other hand, it would be fairly trivial to build HIP aware NAT devices that would allow multiple Host Identities to be NATed behind one IP address.

Web caches. Since HIP encrypts by default all traffic, HIP does not work with existing web caches or other application level middle boxes. If HIP was to be used without IPsec (see Section 3.1), Web proxies, and transparent application layer middle boxes might work. However, that hasn't been analyzed.

5.2.4.10 Implications on scoped addressing

It has not been analyzed how HIP would affect scoped addressing.

Multicast. Not analyzed.

Link local. HIP should not have any effects on link local addresses or using them.

Son-of-Sitelocal. It looks like HIP might reduce need for site local kind of addresses.

5.2.4.11 Layer 2 implications

HIP, as such, does not seem to have any direct implications on layer 2 or neighbor discovery. However, given that HIP introduces a public key per host, it might be possible to further simplify ND and layer 2 security mechanisms.

5.2.4.12 Referrals

As HIP replaces IP addresses with HITs in application data structures, and since HITs cannot be currently resolved into IP addresses, multi-party applications doing IP-address-based referrals will not work. The IRTF HIP RG will study the support of such multi-party applications.

5.2.4.13 Legal stuff / trade marks and name space management

Public keys or their hashes are not mnemonic. The name space does not need to be managed.

5.3 RFC 3582 Section 3 considerations

5.3.1 Multi-Homing capabilities

5.3.1.1 Redundancy

Path redundancy is fully supported, similar to other solutions based on multi-addressing. More specifically, as soon as the hosts have established multi-addressing state by exchanging REA payloads, the hosts may use the different transit providers interchangeably. The current HIP specifications do not specify how a host detects a path failure; such a mechanism is expected to be specified in the multi6 WG.

If a failure occurs before the multi-addressing state has been established, e.g., before the HIP base exchange has been completed, the hosts may try to re-create the HIP state using different IP addresses, if available, e.g., from the DNS. However, the HIP specifications do not currently discuss such a situation, and the actual behaviour depends on local implementation.

5.3.1.2 Load sharing

Load sharing is supported, in the sense that the hosts may use different transit providers interchangeably, similar to other solutions based on multi-addressing.

The current specification does include a feature that allows a host

to control the primary address that it wants its peer to use. If more fine grained control is required, suitable policy mechanisms could be developed on the top of HIP.

5.3.1.3 Performance

By default, HIP based multi-homing does not require intra-transit-provider links to be used. This is similar to other multi-addressing based solutions.

As HIP insulates the transport sessions from the IP addresses, HIP allows more freedom in source or destination address based policy routing.

The baseline HIP solution adds a small delay before the first transport session between a pair of hosts is established. The duration of the delay depends on latency and available CPU resources, consisting of two round trips of latency and requiring hosts to compute Diffie-Hellman shared key, one or two DSA signatures, and verify one or two DSA signatures. Using short keys is allowed by the protocol, subject to local policy considerations.

5.3.1.4 Policy

As of today, HIP does not contain any policy control mechanisms. However, adding such mechanisms seems to be fairly straightforward, not differing from other multi-addressing based solutions.

5.3.1.5 Simplicity

HIP requires both end hosts to be changed. Most applications do not require any changes; applications that use explicit referral may need to be made HIP aware. Incremental deployment is fully supported.

Legacy IPv6 (and IPv4) hosts can be used at a multi-homed site either as such, in which case they do not benefit from multi-homing, or through a HIP proxy, located at the site. If a multi-homed site wants to benefit from multi-homing when communicating with legacy hosts outside of the site, a HIP proxy must be deployed somewhere close to the core network.

The exact details of HIP proxies have not been defined yet.

The current HIP implementations, with limited multi-homing support, have around 10 000 lines of C code. Adding full multi-homing support, as defined in [4] is expected to add less than 3 000 lines of code.

5.3.1.6 Transport layer survivability

The solution provides full re-homing transparency for all transport layer sessions, similar to other multi-addressing based solutions.

Most of the known current implementations do not support transparency for raw IP, as raw IP is considered to be located the host identity layer.

5.3.1.7 Impact on DNS

HIP adds a new DNS resource record for each HIP capable host. The details are to be defined in the HIP WG. This new resource record will be queried along with the IP addresses, thereby adding little overhead.

For redundancy in initial connections, a HIP capable host should list multiple IP addresses in the DNS. However, these addresses are used only for the initial connection. Once the multi-addressing state has been established, the hosts are independent of DNS.

5.3.1.8 Packet filtering

HIP is similar to other multi-addressing based solutions.

5.3.2 Additional requirements

5.3.2.1 Scalability

The solution does not impose new requirements on the routing system.

From a multi-homing point of view, the only required piece of new infrastructure is the new DNS resource record. Adding such records scales approximately as well as the DNS does today.

HIP uses approximately 120 bit pseudo-random identifiers for identifying hosts. According to the birthday paradigm, as long as the total number of hosts remains considerably lower than sqrt(2^120) = 2^60, the probability of identifier collisions remains low. If the number of hosts is expected to grow larger, the length of the identifier can be doubled with minor modifications to the solution.

5.3.2.2 Impact on routers

The solution does not require changes to IPv6 routers, other than what the multi6 wg has already determined useful for all multi-addressing based solutions.

5.3.2.3 Impact on hosts

HIP provides full backwards compatibility with legacy hosts. Whenever one of the two communicating hosts is not HIP aware, the applications fall back to legacy IP.

HIP does require changes to the host stack. These changes can be classified into two classes:

1. Basic packet processing must be changed to recognize HITs on outgoing packets, and incoming ESP protected HIP packets must replace the IP addresses with HITs. This change is minor, and can be implemented either integral to IPsec, or separate. In a typical implementation, the number of changed and/or added lines of code is a few hundred.

2. A HIP protocol implementation must be added to the stack. This change is a logically separate function. In a typical implementation, the number of lines of code required is in order of 10000-20000.

The solution does not _require_ changes to the socket API or transport layer. However, it is _expected_ that the socket API and the transport layer will be changed in order to gain full benefit from HIP.

As it is currently defined, HIP breaks some multi-party applications that use IP addresses for referral. Solutions to this problem is a research topic, being studied at the IRTF HIP RG.

5.3.2.4 Interactions between hosts and the routing system

HIP does not require interaction between hosts and the routing system, other than what the multi6 wg has already determined for other multi-addressing based solutions.

5.3.2.5 Operations and management

HIP implementations are expected to include a facility that allows an administrator to view HIT to address mapping. There is no HIP MIB or PIB, but it can be expected to be added as a working item for the HIP working group in the future.

5.3.2.6 Co-operationg between transit providers

HIP does not require any co-operation between transit providers. If such co-operation is available, HIP would benefit from it similar to other multi-addressing based solutions.

5.3.2.7 Multiple solutions

HIP should work well with multi-homing solutions that are located solely at the IP layer, i.e., below HIP.

Interoperability with other multi-addressing based solutions depend on many details, and need to be analyzed case-by-case.

5.4 Security considerations

HIP attempts to raise the security baseline in the Internet by employing IPsec ESP protection by default.

6. Security considerations

HIP security has been extensively discussed in [3] and [2]. Mobility and multicast related security issues have been briefly discussed in [4]. As this draft is more a discussion draft and not a protocol specification, security considerations related to using HIP components instead of full HIP are currently not discussed anywhere. Such a discussion is planned to be added at a later stage, if this draft goes forward.

7. Change log

Changes between this version (-01) and -00 draft

- added Section 2.6 comparing HIP with other group F multi6 proposals

- added Section 3.3 describing how HIP could be possibly changed to include routable AIDs

- updated references to HIP WG and HIP RG (Section 1.2)

Informative references

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

[2] Moskowitz, R., "Host Identity Protocol", draft-ietf-hip-base-00 (work in progress), June 2004.

[3] Moskowitz, R., "Host Identity Protocol Architecture", draft-moskowitz-hip-arch-06 (work in progress), June 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] Nikander, P., "Mobile IP version 6 Route Optimization Security Design Background", draft-nikander-mobileip-v6-ro-sec-02 (work in progress), December 2003.

[6] Nordmark, E. and T. Li, "Threats relating to IPv6 multihoming solutions", draft-nordmark-multi6-threats-02 (work in progress), June 2004.

[7] Nordmark, E., "Multihoming without IP Identifiers", draft-nordmark-multi6-noid-01 (work in progress), October 2003.

[8] Nordmark, E., "Multihoming using 64-bit Crypto-based IDs", draft-nordmark-multi6-cb64-00 (work in progress), November 2003.

[9] Nordmark, E., "Strong Identity Multihoming using 128 bit Identifiers (SIM/CBID128)", draft-nordmark-multi6-sim-01 (work in progress), October 2003.

[10] Crocker, D., "CHOICES FOR MULTIADDRESSING", draft-crocker-mast-analysis-01 (work in progress), October 2003.

[11] Krawczyk, H., "The SIGMA family of key-exchange protocols", 2003.

[12] Ylitalo, J., "Weak Identifier Multihoming Protocol Framework (WIMP-F)", draft-ylitalo-multi6-wimp-01 (work in progress), June 2004.

Authors' Addresses

Pekka Nikander
Ericsson Research Nomadic Lab

JORVAS FIN-02420
FINLAND

Phone: +358 9 299 1
EMail: pekka.nikander@nomadiclab.com

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

EMail: thomas.r.henderson@boeing.com

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Acknowledgment

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