In-Depth Look at the Host Identity Protocol (HIP): Providing Agile Mobility, Multi-Homing, and Security

Dr. Pekka Nikander

Ericsson Research Nomadic Lab
Hirsalantie 11
FI-02420 JORVAS, Finland

pekka.nikander@nomadiclab.com

Helsinki Institute for Information Technology
Metsänneidonkuja 4, P.O.BOX 9800
FI-02015 TKK, Finland

pekka.nikander@hiit.fi

ABSTRACT

The Host Identity Protocol (HIP) is an experimental architecture and protocol, being developed at the IETF since 1999 and reaching its first stable version in 2007. It enhances the original Internet architecture by injecting a new thin layer between the IP layer and the transport protocols. This new layer introduces a new name space consisting of cryptographic identifiers, thereby implementing the so-called identifier / locator split. In the new architecture, the new identifiers are used for naming application level end-points, thereby taking the prior identification role of IP addresses in applications, sockets, TCP connections, and UDP send and receive system calls. IPv4 and IPv6 addresses are still used, but only as names for topological locations in the network. At the same time, due to the backwards compatibility mode, no changes are needed in applications.

The architectural enhancement implemented by HIP has profound consequences. A number of the previously hard problems become suddenly much easier. Mobility, multi-homing, and baseline end-to-end security integrate neatly into the architecture. The use of cryptographic identifiers allows enhanced accountability, thereby providing a base for easier build up of trust. With some privacy enhancements, HIP allows good location anonymity, assuring strong identity only towards relevant trusted parties. Finally, the HIP protocol has been carefully designed to take middle boxes into account, providing for overlay networks and thereby helping to reduce the currently prevalent problems with bad traffic and routing scalability.

This presentation provides a more in-depth look at HIP, discussing its architecture, design, benefits, and potential drawbacks. While the presentation concludes with a brief description of a demonstration running during the breaks, this written version does not describe the demonstration.

1.0 INTRODUCTION

The Host Identity Protocol (HIP) is a new piece of technology that may change the nature of the Internet within the next few years. The original ideas were formed through discussions at some IETF meetings during 1998 and 1999. Since then, it has been developed by a group of people from Ericsson, Boeing, universities, and other companies, first as an informal activity close to the IETF and later within the IETF HIP working group.

From a functional point of view, HIP integrates IP-layer mobility, multi-homing and multi-access, security, NAT traversal, and IPv4/v6 interoperability in a novel and simple way. The result is much simpler than trying to implement these functions separately, using technologies such as Mobile IP, IPsec,

Nikander, P. (2007) In-Depth Look at the Host Identity Protocol (HIP): Providing Agile Mobility, Multi-Homing, and Security. In *Emerging Wireless Technologies* (pp. 3-1 – 3-14). Educational Notes RTO-EN-IST-070, Paper 3. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int.

ICE, and Teredo. In a way, HIP can be seen as restoring the lost end-to-end connectivity across various IP links and technologies, this time in a way that is secure and supports mobility and multi-homing. As an additional bonus, HIP provides new tools and functions for future network needs, including the ability to securely identify previously unknown hosts and the ability to securely delegate signalling rights between nodes.

From a technical point of view, the basic idea of HIP is to add a new layer to the TCP/IP stack. Roughly speaking, this new layer is injected between the IP layer and the transport layers (TCP, UDP, SCTP, etc). At this new layer, hosts (i.e. computers) are identified with new identifiers, Host Identifiers. These new Host Identifiers are public cryptographic keys. As a result of adding this new layer to the stack, when applications open connections and send packets, they no longer refer to IP addresses but to these public keys. Hence, for example, when an e-mail client opens a connection to the e-mail server hosting the mailbox, the e-mail client hands over a reference to the public key to the operating system, denoting that it wants the operating system to open a secure connection to the host that holds the corresponding private key. The resulting connection is kept open even if both of the hosts, i.e. the client and the server, move. If the hosts have multiple connections, HIP allows these multiple connections to be used for load balancing or as back ups, invisible to the applications.

HIP has been designed in such a way that it is fully backwards compatible to applications and the IP infrastructure. That is, to deploy HIP in a limited environment, all that is required is to update the operating system of the involved nodes to support HIP. No changes are required to applications or the IP routing infrastructure. For full HIP support, a piece of new infrastructure is needed, in order to support HIP rendezvous services. If it is impossible to upgrade the operating system of some particular node, e.g., a legacy mainframe, it is also possible to add a front-end processor to such a system. The front-end processor acts as a HIP proxy, making the legacy host to appear as (a set of) HIP node(s) to the rest of the network.

HIP is currently in the final phases of being accepted as an experimental IETF standard, with the publication of the RFCs describing the protocols, main extensions, and infrastructure support in 2007. The architecture itself is described in RFC 4423 [1], published in 2006. There are three independent open source versions of HIP, and a small but active user community.

In the rest of this presentation we explore the rationale behind HIP, its details, and various extensions and options that are currently slowly maturing. In the end of the presentation, we discuss the current maturity status of HIP.

2.0 EVOLVING ENVIRONMENTS

As discussed in some length in the other accompanying presentations [2][3], the original TCP/IP design was created for an environment where the end-users were assumed to be mutually trusting, at least to a minimal degree, and where the network is assumed to be inherently unreliable due potential attacker activity [4]. Since then the environment has grossly changed, creating a need to device a communication architecture provides the following functions:

- Ability to operate over all kinds of underlying networks, including ad hoc, commercial, and dedicated; this implies the ability to dynamically pay for the services on-line, the ability to hide the real identities of communicating parties from the underlying networks, etc.
- Ability to survive in a partially hostile environment where some of the underlying networks may be only partially co-operating, competing, or even outright antagonistic to each other; this implies the ability to isolate underlying networks from each other, when needed.

- Ability to support application, host, and sub-network level mobility and multi-access as prime design elements and not as extensions.
- Ability to support for full location privacy, especially against any foreign network infrastructure and other third parties.

The goals above can be seen as a new incarnation of the original IP design goal, adopted to the contemporary needs, where the underlying communication network is more diverse than then, sometimes even hostile in addition to being unreliable, and where a fraction of the co-users must be assumed to be either egregiously selfish or plainly malicious.

In addition to revising the original goal to meet today's need, it has also become clear that the operational costs of the current network are becoming quite high. Consequently, there is a need for a network that can self-organise itself, including functions such as infrastructure discovery and the ability to find reasonably functioning communication paths among multiple alternatives.

The HIP architecture and base protocols aim towards these goals. While they do note as such provide for all of the functionality that is needed to fulfil the above mentioned goals, they appear to make a pretty good job in creating a new inter-connectivity layer, relying heavily on the availability of IP infrastructure but, in theory at least, being also able to run over non-IP links and media. It provides baseline protection for communication, including optional functionality to fully protect the identity of the communicating parties from outsiders. It contains a set of basic mechanisms to support host mobility and multi-homing, allowing these mechanisms to be adopted and extended to better fit to differing environments. Finally, it aims to provide a similar or higher level of flexibility than the original IP architecture did, allowing the protocols and mechanisms to be easily extended.

3.0 HIP ARCHITECTURE

The HIP architecture has been carefully crafted to meet the current requirements and simultaneously provide for enough of flexibility for future adoptions. It is inherently build upon the the current IP-based routing infrastructure (IPv4 and IPv6), a huge asset that is unlikely to disappear any soon. At the same time, HIP supports the current binary APIs with sufficient semantic compatibility.

More technically speaking, the current division of the functionality into specific layers appears to be partially wrong or at least suboptimal from security, mobility, and scalability points of view. For example, from a functional point of view, the upper parts of IP (up from and including IPsec) apparently belongs more tightly to transport than the lower part, i.e., the routing and forwarding part of IP. In other words, the current division between IP and transport seems to make a number of network functions, including mobility and multi-homing support, harder than necessary. At more architectural level, one can question the whether the very division between IP and transport makes sense at all. As discussed in the previous presentation [2], there are proposals for recursive architectures where some kinds of routing / forwarding and transport-like end-to-end functions appear on each layer.

HIP attempts to provide a partial fix for that. In particular, it attempts to restore, in an enhanced form, the four "classic" network layer protocol invariants:

- Non-mutable: The source and destination identities sent are the identities received.
- Location independent: The identities does not change during the course of an "association".
- Reversible: A return header can always be formed by reversing the source and destination identities.
- Omniscient: Each host knows what identities a partner host can use to send packets to it.[1]

[1] Actually, the fourth can be inferred from 1 and 3, but it is worth mentioning for the sake of completeness.

In the current world, we have been forced to give up all but reversibility; furthermore, we suspect that the only reason reversibility has been preserved is that the Internet would stop working without it.

Consider now Domain Name System (DNS) names. From the practical, functional point of view are references to IP addresses. As the majority of current applications are based on some variant of the so called socket API, the application itself (directly or within a library) resolves the DNS to an IP address, and uses the IP address, in the socket API, to identify the destination host and application.

The Host Identity (HI) name space, introduced by HIP (see below), fills an important gap between the IP and DNS name spaces. By creating a new inter-networking layer on the top of the existing IP networks, it restores the classic invariants at the identity layer while freeing the network layer from them. That is, HIP direclty allows the the underlying communications to give up all but the 3rd network-layer invariant, and when provided with a global rendezvous service (see Section 6.1), even that can be dropped.

3.1 Naming architecture

In traditional IP networks a host has an IP address that serves two different purposes: it is both a locator, describing the current topological location of the node in the network, and an identifier, describing the identity of the host. This dual role of an IP address has drawbacks when more interfaces are added to the host (a multi-homed host) and when a host is allowed to change its topological location (a mobile node). In case of a mobile node, the locator information is changed each time the node changes its location in the network. The identity of the host, however, still remains the same, which makes it impossible to use the same IP address for both purposes.

A solution to this problem is to separate the identity and location information from each other. HIP separates the location and identity roles of IP addresses by introducing a new name-space, the Host Identity (HI). In HIP, the HI is basically a public cryptographic key of a public-private key-pair. A host possessing the corresponding private key can prove the ownership of the public key, i.e. its identity. The separation of the identity and locator makes it also simpler and more secure to handle mobility and multi-homing in a host.

Figure 1 shows how HIP is located in the current stack. On the layers above the Host Identity layer, the locator of the host is not shown. Only the HI (or its 128-bit representation, Host Identity Tag, HIT) is shown. The Host Identity layer maintains mappings between identities and locators. When a mobile host changes location, HIP is used to transfer the information to all peer nodes and this dynamic mapping on other hosts is modified to contain the new locator information. Upper layers, e.g. applications, are unaware of this change.

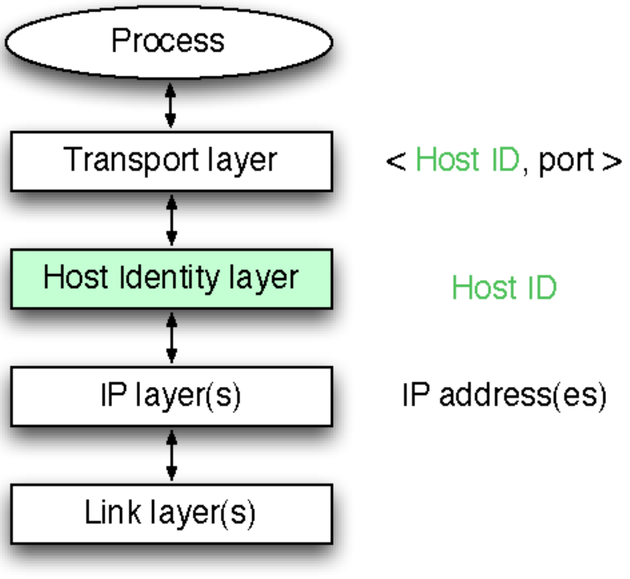

Figure 1 A New Layer

During the connection initialisation between two HIP nodes, a four-way handshake, a so-called Base Exchange, is run between the nodes. During the exchange, hosts identify each other using public key cryptography and exchange Diffie-Hellman public values. Based on these values, a shared key can be generated which further is used to generate keying material to be used in other cryptographic operations. During the Base Exchange hosts negotiate used cryptographic protocols, and establish also an IPsec Encapsulated Security Payload (ESP) Security Association (SA) between them. The ESP keys are retrieved from the generated keying material and all further data traffic is sent as encrypted over the ESP SA.

3.2 A more detailed look

Considering the situation more carefully from an architectural point of view, it becomes clear that HIP pretty much provides the same base functionality in the enhanced architecture than IP provided in the original IP architecture: end-to-end connectivity over different links and media. However, since IP pretty much still provides universal (though limited) connectivity throughout the world, the efforts on HIP have more focused on those aspects of (extended) end-to-end connectivity that today’s IP does not provide that well: support for mobile and multi-homed hosts, security, middle-box support, and connectivity between the two versions of IP.

Figure 2 depicts the closer positioning of the new functionality. Basically, the current IP-layer functionality is divided into those functions that are more end-to-end (or end-to-middle) in nature, such as IPsec, and those that are more hop-by-hop in nature, such as the actual forwarding of datagrams. HIP is injected between these two sets; in practice, architecturally immediately below IPsec, often functionally embedded with the IPsec SA processing.

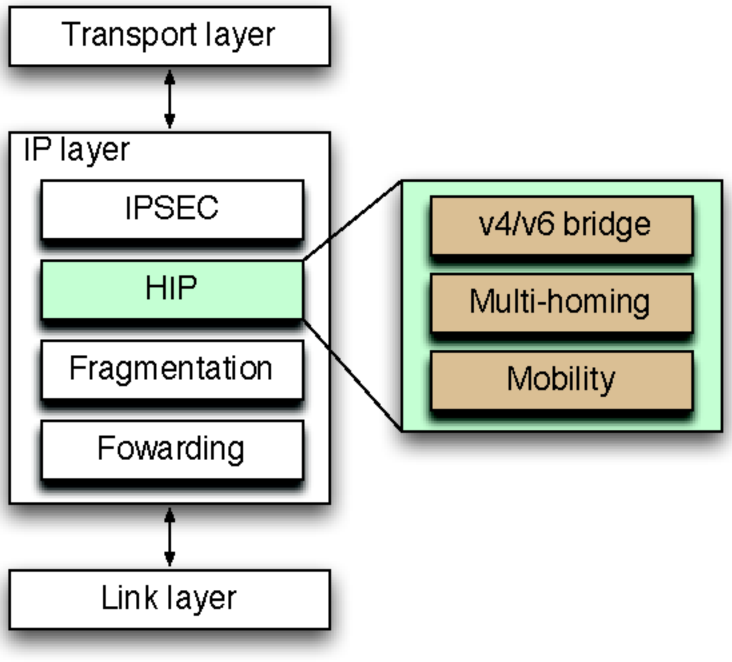

Figure 2. A more detailed look at the layering

With HIP, the separation of the location and identity information disentangles the packet de-multiplexing at the receiving end-host (or middle box) and the forwarding taking place at the routers. From the highest-level architectural point of view, the main issue is a simple addition of a new layer of indirection. However, the host receiving a packet identifies (and verifies) the correct HIP-layer association indirectly, typically by first getting the correct session keys based on the ESP Security Parameter Index (SPI) in the received packet, and then decrypting and verifying the integrity of the packet. Thus, the actual IP addresses that were used for routing the packet are irrelevant after the packet has reached the destination interface.

This is in stark contrast with the prevailing IP practice, where the transport layer identifiers are created by concatenating the IP-layer identifiers (IP addresses) and the port numbers. The main benefit of the current practice is implied security: since the transport identifiers are bound to the actual network locations, the transport connections get automatically bound to the locations. That allows the routing and forwarding system to be used as a weak form of security: binding identity to the location allows reachability to be used as a (weak) proxy for the identity.

4.0 REMEDIES TO THE CURRENT PROBLEMS

In this section, we have a brief look at how HIP relates to the current problems in the Internet, as identified in the accompanying presentation [2]. As such, HIP does not provide any direct remedy to the unwanted traffic or routing problems, but when employed to separate the control and data traffic from each other (see Section 5.1), it enhances architectural flexibility in a way that can be used to make unwanted traffic more expensive and to divide the routing problem into smaller pieces, thereby reducing scalability burdens. HIP does not provide any direct remedies for the resource control and congestion problem, either. On the other hand, the strong identifiers may increase the ability to attribute traffic to specific real-life sources, thereby indirectly contributing to solving those problems. However, such practises lie beyond the scope of this presentation.

With its build-in mobility and multi-homing support, HIP is a direct answer to the mobility and multi-homing problems; see Section 4.1. Through its cryptographic identifiers and the optional BLIND base exchange [5], it directly enhances both privacy and accountability. The identifiers also provide a potential starting point for building explicit infrastructure to represent trust and reputation. Other, more technical remedies are briefly discussed in Section 4.2.

4.1 HIP-based combined mobility and multi-homing

With HIP, the separation between the location and identity information ensures that the packet identification and routing can be cleanly separated from each other. The host receiving a packet identifies the sender by first getting the correct key based on the ESP Security Parameter Index (SPI) and then decrypting the packet. Thus, the actual IP addresses that were used for routing the packet are irrelevant after the packet has reached the destination interface.

The HIP mobility and multi-homing protocol defines a LOCATOR parameter that contains the current IP address(es) of the mobile entity. When the mobile entity changes location and therefore IP address, it generates a HIP UPDATE packet with one or more LOCATOR parameters, signs the packet with the private key matching to the used HI, and sends the packet to the peer node and to one or more rendezvous servers (RVS).

When the peer node receives a LOCATOR parameter, it can start an address verification process for the IP address(es) that are included in the parameter. Reachability verification is needed to avoid accepting false updates. The verification can be skipped in special circumstances, for example when the peer is completely sure that the address is correct.

Because the mobile entities can move between networks using different IP address versions, the address received by the peer may also be of different IP address version than the previous address. It is possible that the peer does not support that particular IP address version at all in which case the peer node needs to use a proxy node that converts the traffic between these two protocol versions.

Enhanced mobility support

The Mobility and Multi-homing specification describes also an enhancement to the standard location update procedure. The Credit-Based Authorisation process allows the peer host to use the new locator already before the reachability has been verified. The peer calculates credit based on the amount of data received from the mobile host and it can send at maximum the same amount of data to the mobile host. The limitation disappears, when the reachability verification is accomplished. This method can enhance the performance of certain real-time applications. Voice over IP applications do suffer from long breaks in the connection but using this method, the break can be made much shorter.

When handovers are done break-before-make, the connection is lost for a while. HIP allows make-before-break style handovers, enhancing the handover performance significantly while the handover can be made even without any packet loss.

Advanced mobility issues

As described earlier, HIP supports mobility between and within different IP address realms. However, all addresses in described scenarios have been from public address base, thus being routable in the Internet. Mobility between a private address space behind a NAT and public address spaces is not supported with the basic HIP when the private address space is behind a legacy NAT device. Using advanced methods, this type of mobility is possible to achieve using an IPsec Security Parameter Index (SPI) based address translation. This, a so-called SPINAT device, is HIP-aware and can take advantage of the passing by HIP packets.

In practice, a SPINAT device uses the HIP Base Exchange and UPDATE packet information and gets all relevant information from them (locators, HITs, and SPI values used in the ESP connection). With this information, the SPINAT device can do required address translations between public and private address spaces.

4.2 Other functionality enabled or made easier by HIP

In addition to enabling different forms of mobility and multi-homing, HIP enables the following other types of functions:

- Due to using public cryptographic keys as identifiers, HIP adds cryptographically secure delegation as a primary architectural element. That, in turn, can be used to implement different forms of proxied functionality; for example, subnet mobility, allowing an intelligent network to signal mobility on mobile hosts' or subnets' behalf, and allowing application level service delegation.
- Due to using a specific IP protocol number for control traffic and careful infrastructure design, HIP can be used to separate control and data traffic on different "planes".
- HIP provides channel binding based security to applications using the IPv6 API, providing them assurance that the peer or service identified by the HIT used at the socket interface is actually the one that the underlying stack has connected to.
- HIP provides mobility and multi-homing transparently over both IPv4 and IPv6, even across IPv4 NATs, and allows most IPv4 and IPv6 applications to be intermixed; i.e., with HIP a typical IPv4 application can communicate directly with an IPv6 application, and vice versa. However, the application-level interoperability does not apply to all applications.
- HIP-enabled firewalls can authenticate passing HIP base exchange and update packets, and punch holes for IPsec ESP traffic selectively.

5.0 A FEW USEFUL FEATURES

In this section we briefly discuss two more advanced pieces of functionality that are becoming much easier than before with the introduction of HIP. First, in Section 5.1, we show how HIP can be used to separate the control and data traffic in the network into separate planes and the potential benefits such a practise provides. In Section 5.2, we discuss HIP-based signalling delegation and especially how it can be applied to provide sub-network mobility and multi-homing, extending the idea in Section 5.3 to the application domain.

5.1 Control / data separation

Under normal conditions, when legacy IPv4 NATs are not used in the network, HIP control traffic is carried in a separate protocol that has its own IP protocol number, distinct from TCP and UDP. This allows easy separation of HIP control and data traffic; in a baseline implementation, all control packets are carried in the HIP protocol and all data packets are carried within IPsec ESP envelopes.

The control–data separation can be used to hide the actual IP addresses of most end-hosts. Under such an arrangement, the clients connecting to servers must do so through a separate control plane, by always first contacting a HIP rendezvous server located in the control plane. The rendezvous server can provide the client directly with a cached puzzle and verify the puzzle in the incoming I2 packet. Only messages that have been pre-screened through this DoS-resistance mechanism are passed to the server. The server then still has the opportunity of verifying the clients identity and authorisation before making the decision whether to make a contact with the client and reveal the server's IP address(es) to the client or not

The basic setting is illustrated in Figure 3. The network consists of two “planes”, the data plane (blue) that is a plain IP-based router network, with HIP-enabled firewalls located at strategic points, and the control plane (purple) that is implemented as an “overlay” on the top of the data plane. In practise, the control plane consists of HIP proxies that typically synchronise location information either partially or fully between each other.

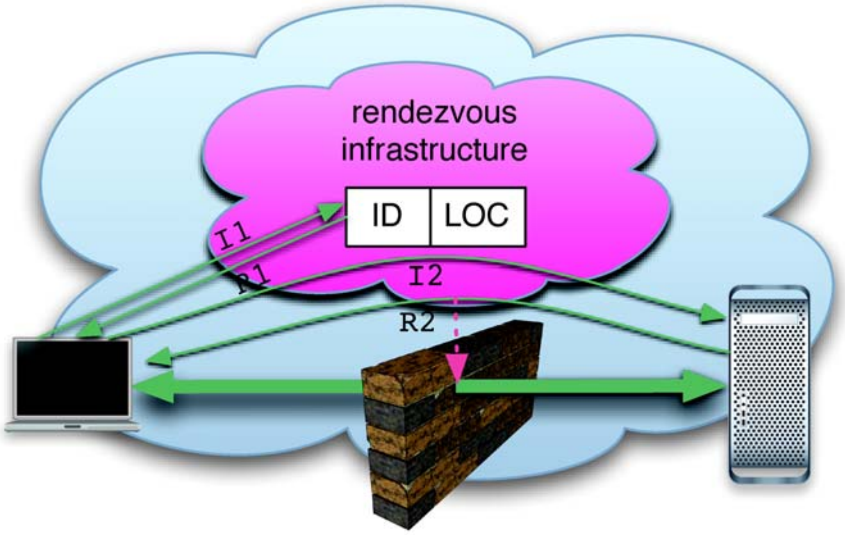

Figure 3 A HIP-based control data separation architecture

The server is located behind a HIP-enabled firewall. Before it can be reached, it must register to the rendezvous infrastructure, creating a binding between the server’s identity (ID) and current reachability (R), or locators. While doing so, it may cache a number of pre-computed R1 packets at the rendezvous infrastructure.

When a client wants to make a contact with the server, it sends the first HIP base exchange message, I1, to the rendezvous infrastructure. The infrastructure looks up a cached R1 packet and passes it to the client. Once the client has solved the puzzle therein, it can send an I2 packet, again to the infrastructure. A node in the infrastructure verifies that the puzzle has been correctly solved, and passes the I2 packet to the server. At this point, the server can verify the client’s identity and authorisation. Only if the client can be positively identified and has proper authority, the server responds to the client with an R2 packet. Depending on the location of the firewall(s), the R2 packet can either be sent directly to the client or it may be necessary to pass it through the overlay, thereby triggering hole punching at the firewall. Finally, the actual data traffic traverses directly through the data plane, through the firewalls.

5.2 Mobile routers and signalling delegation

As mentioned above, public-key-based identification provides a natural facility for delegation. In the context of mobility and mobile sub-networks, delegation can be used to delegate mobility signalling from individual mobile nodes to a mobile router, and further from the mobile router to some infrastructure node at the fixed network side.

Figure 4 illustrates the basic idea. First, individual mobile nodes in a mobile network delegate, to a mobile router, the right to inform their peers about their location. In the next step, the mobile router further delegates that right to a router (or another infrastructure node) within the fixed part of the network (illustrated as a could).

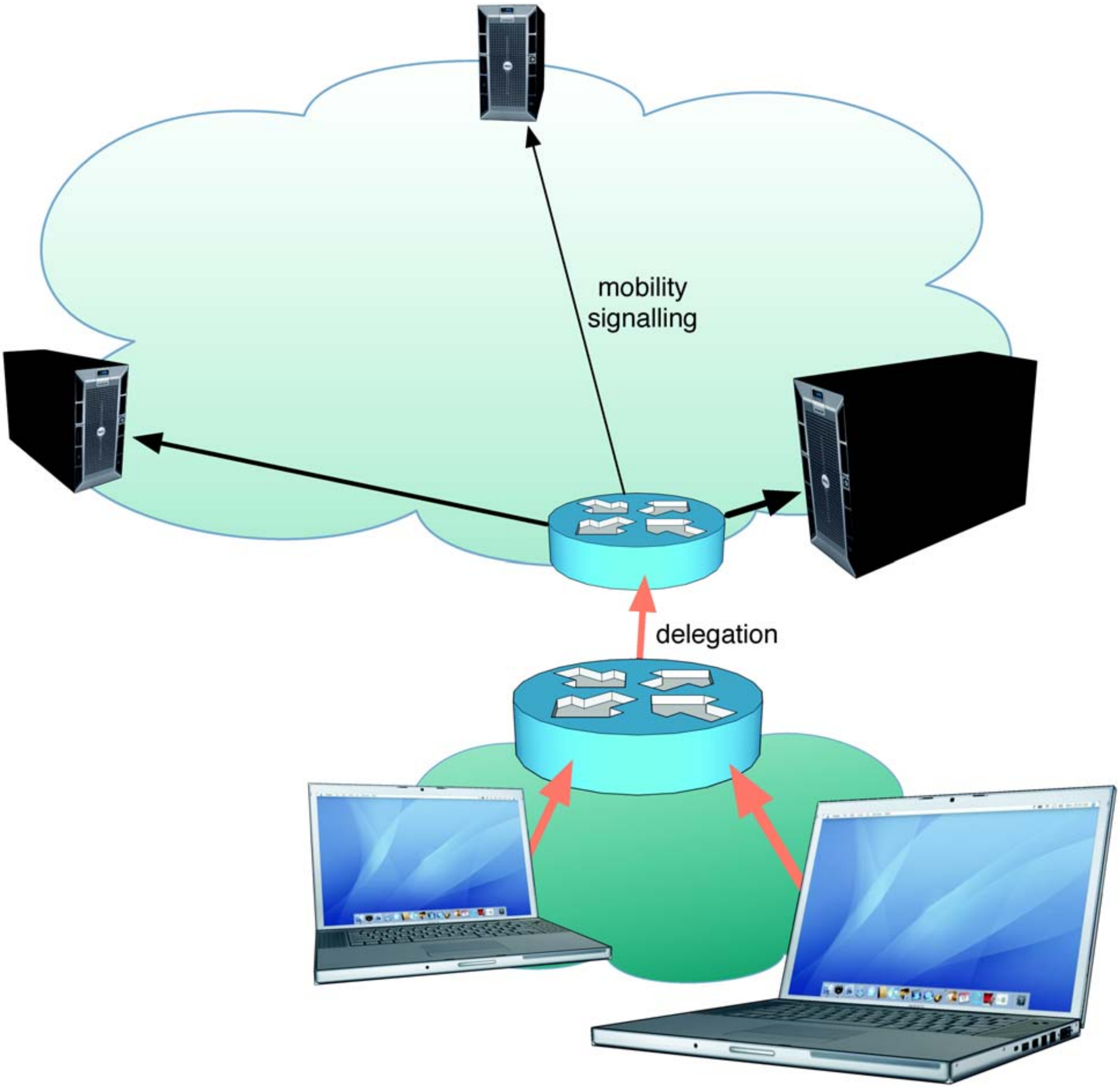

Figure 4 Using delegation to move mobility signalling to the network side

If the underlying IP-layer router mobility functionality is arranged in such a way that the fixed-side router gets informed whenever the mobile router changes its point of attachment, it becomes possible to for the fixed-side router to send the mobility messages directly to the corresponding nodes of all mobile nodes within the mobile subnetwork. Hence, for HIP-layer mobility signalling, no messages need to transmitted over the air interface at the end of the moving entity.

5.3 Application-level service delegation

Another area where delegation can be applied to are applications and services. As illustrated in Figure 5, Host Identities can be allocated to abstract services (leftmost box) and service instances in addition to physical hosts. With that kind of an arrangement, using suitable service resolution infrastructure, a client application can ask for a connection to the abstract service, using the HIT assigned to the abstract service and get delegated and redirected to one of the service instances. Furthermore, for node mobility, the signalling right for mobility signalling can be further delegated from the service instances to the physical node, thereby allowing node mobility to secure update the clients' understanding of the locations of the service instances.

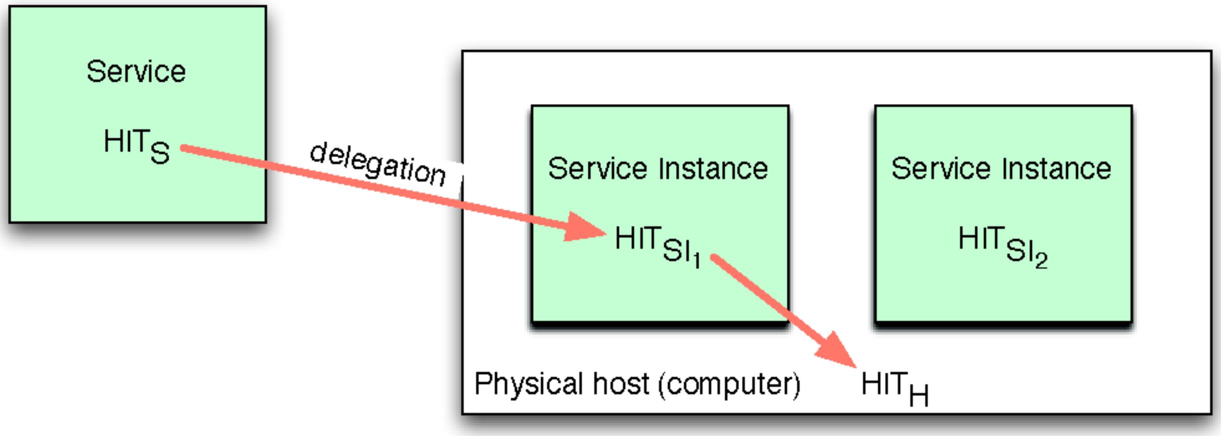

Figure 5 Using delegation to enable abstract services implemented in virtual hosts

6.0 MATURITY STATUS

All basic research and development for the first version of HIP is basically done. The three open source implementations, by Ericsson Research, Helsinki Institute for Information Technology (HIIT), and Boeing Phantom Works, are mature for experimental use. HIP is used by a few people in daily bases (see below), and the initial set of RFCs is expected to be published during 2007.

6.1 Usage of HIP today

Individual researchers at Ericsson, HIIT, and Boeing use HIP in their every day life, mostly using Linux laptops to access specific HIP-enabled services, such as e-mail. Most of their traffic still flows over vanilla IPv4, though, HIP being used only for the few odd services where the service is HIP-enabled, too.

Two major government organisations in Europe are seriously considering adopting HIP for their internal use. However, at the time of writing (summer 2007), neither of them have made their decisions.

In 2005, Boeing announced their plan to use HIP as a part of their Secure Mobile Architecture, an experimental architecture that they are experimenting with in some of their aircraft assembly halls. An April 2006 Network World article (http://www.networkworld.com/news/2006/050106-boeing-side.html) describes the use of HIP as follows: "Boeing's SMA technology uses public-key infrastructure (PKI) certificates along with Host Identity Protocol (HIP) - an experimental IETF RFC that acts like the IPsec, but has skinnier packets that take up less bandwidth. […]The directory aspect of SMA ties this HIP authentication and packet-marking technique to a Secure Lightweight Directory Access Protocol directory on the back end. The overall architecture will, in theory, allow Boeing to secure its network based on employee or machine identities, instead of IP addresses."

The overall architecture of SMA is depicted in Figure 6 on the next page.

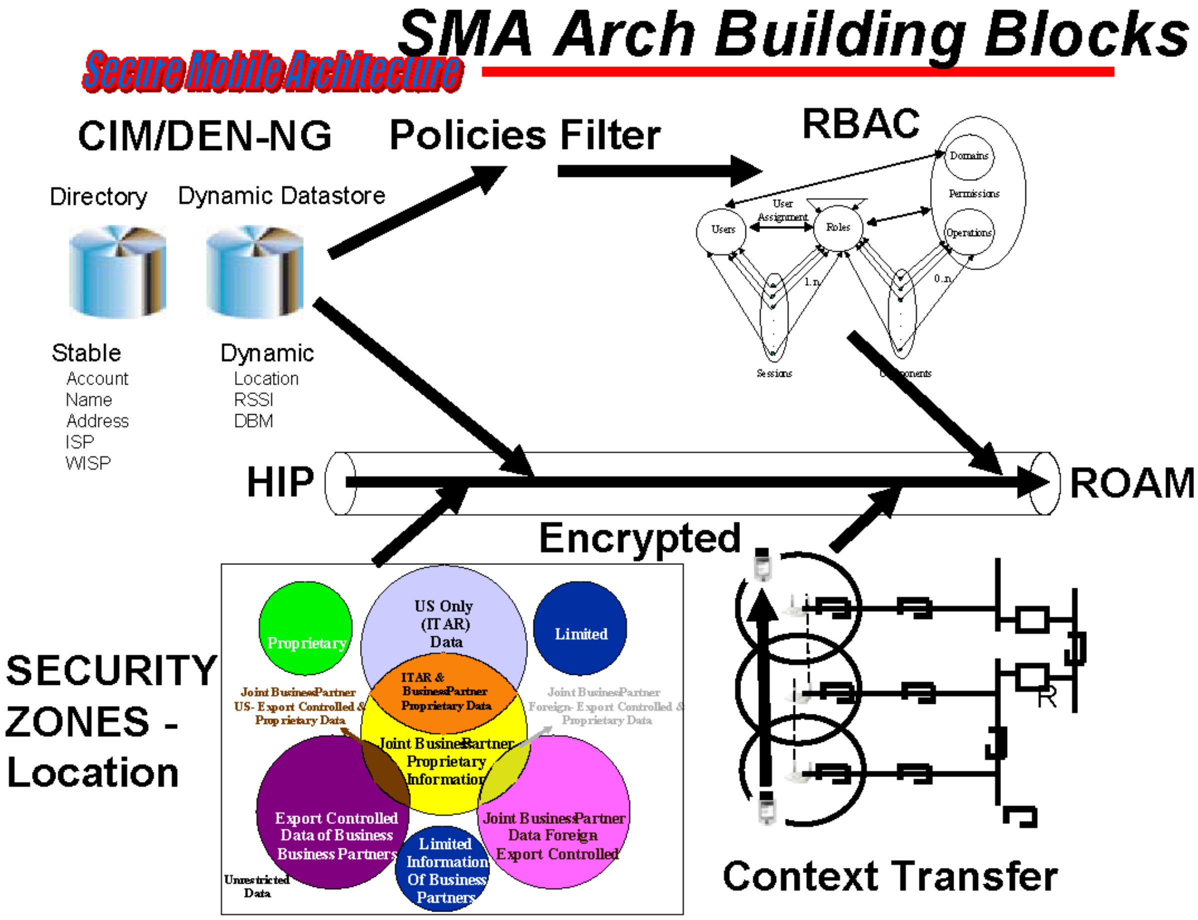
SMA Arch Building Blocks
Secure Mobile Architecture
CIM/DEN-NG
Policies Filter
RBAC
Directory
Dynamic Datastore
Domains
Permissions
Operations
Users
User Assignment
Roles
1.n
0..n
Sessions
Components
Stable
Account
Name
Address
ISP
WISP
Dynamic
Location
RSSI
DBM
HIP
ROAM
Encrypted
SECURITY ZONES - Location
Proprietary
US Only (ITAR) Data
Limited
ITAR & Business Partner Proprietary Data
Joint Business Partner US- Export Controlled & Proprietary Data
Joint Business Partner Foreign- Export Controlled & Proprietary Data
Joint Business Partner Proprietary Information
Export Controlled Data of Business Business Partners
Limited Information Of Business Partners
Joint Business Partner Data Foreign Export Controlled
Unrestricted Data
R
Context Transfer

Figure 6 SMA Architectural Building Blocks [6]

6.2 Standardisation situation

All the base protocol documents were at the time of writing at IESG evaluation, with a few minor issues still to be resolved. The documents were expected to advance to the RFC Editor during summer 2007, with the expected publication still during 2007. These documents include the following ones:

- Host Identity Protocol
- Using ESP transport format with HIP
- End-Host Mobility and Multihoming with the Host Identity Protocol
- Host Identity Protocol (HIP) Registration Extension
- Host Identity Protocol (HIP) Rendezvous Extension

Another set of three documents are likely to advance to the RFC Editor still during 2007 or early 2008:

- NAT and Firewall Traversal Issues on Host Identity Protocol (HIP) Communication
- Host Identity Protocol (HIP) Domain Name System (DNS) Extensions
- Using the Host Identity Protocol with Legacy Applications

A number of documents are still undergoing working group or research group discussion, and may end up having major changes:

- HIP Extensions for the Traversal of Network Address Translators
- Native Application Programming Interfaces for SHIM Layer Protocols
- HIP Experiment Report

The discussions of the HIP specifications at the IESG has increased the interest of individual IESG members to consider advancing HIP from its current Experimental track to the Standards Track. However, it must be noted that the official IETF consensus has been and remains that the HIP community is expected to conduct a real life experiment on how HIP works in the wider Internet, and only based on a suitable experiment report shall the IETF reconsider its status.

7.0 CONCLUSIONS

In this presentation, we have described the architecture, benefits, and current status of the Host Identity Protocol. The HIP architecture has been designed to restore the “classic” inter-networking variants, allowing hosts to interconnect in the current immensely complex communication environment with IPv4, IPv6, NATs, and other middle boxes. HIP provides built-in, architected support for mobility, multi-homing and multi-access, and baseline security. It enhances the IP architecture by introducing a Host Identity (HI) name space roughly between the IP layer and the transport protocols.

Beside the basic advantage of integrated mobility, multi-homing and security support, HIP provides for a number of potential architectural extensions. The inherent delegation capability can be used to implement subnetwork-level mobility and multi-homing, as well as delegable application names. The architecture allows control traffic to be easily separated from data traffic, providing for enhanced protection against unwanted traffic.

At this writing (summer 2007), the HIP specifications are in the final phases of becoming experimental IETF standards. The protocol is in the daily use of a number of researchers and other early adopters. A few major governmental organisations in Europe are seriously considering the suitability of HIP for their internal use. Furthermore, in 2005 Boeing announced their goal of using HIP as an intrinsic part of Secure Mobile Architecture (SMA), intended to secure the communication needs in their airplane assembly halls.

8.0 REFERENCES

[1] Robert Moskowitz and Pekka Nikander, “Host Identity Protocol (HIP) Architecture,” *RFC 4423,* IETF, May 2006. http://www.rfc-editor.org/rfc/rfc4423.txt

[2] Pekka Nikander, “Evolution of Networking: Megatrends, Current Problems, and Future Directions,” NATO IST-070 Lecture Series on “Emerging Wireless Technologies. Oct 2007.

[3] Pekka Nikander, “ Emerging Inter-networking Technologies: From IPv6 to Host Identity Protocol and Beyond,” NATO IST-070 Lecture Series on “Emerging Wireless Technologies. Oct 2007.

[4] David Clark, “Application Design and the End-to-End Arguments,” presentation at MIT Communication Futures Program (CFP) Bi_annual Meeting, Philadelphia, PA, May 30–31, 2007. http://cfp.mit.edu/events/may07/Slides/CLARK%20Application%20Design.ppt

[5] Jukka Ylitalo and Pekka Nikander, "BLIND: A Complete Identity Protection Framework for End-points", to appear in Security Protocols, Twelfth International Workshop, Cambridge, 24-28 April, 2004.

[6] Bill Estrem et al, “Secure Mobile Architecture (SMA) Vision & Architecture,” Technical Study E041, The Open Group, Feb 2004.

[7] Pekka Nikander, Jukka Ylitalo, and Jorma Wall, “Integrating security, mobility, and multi-homing in a HIP way”, in Proc. *Network and Distributed Systems Security Symposium (NDSS’03),* Internet Society, 2003.

[8] Thomas R. Henderson, Jeff M. Ahrenholz, Jeff H. Kim, “Experience with the Host Identity Protocol for Secure Host Mobility and Multihoming,” in Proc. *Wireless Communications and Networking, WCNC 2003,* Vol 3, 2120–2125, 16-20 March 2003.

[9] Pekka Nikander, Jari Arkko, and Börje Ohlman, “Host Identity Indirection Infrastructure (Hi3),” in Proc. *Second Swedish National Computer Networking Workshop.*

[10] D. Clark, R. Braden, A. Falk and V. Pingali, *FARA: reorganizing the addressing architecture*, Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture (2003), pp. 313-321.