# Host Identity Payload in Home Networks

Miika Komu
Helsinki University of Technology
Department of Computer Science and Engineering
47521c
Miika.Komu@hut.fi

**Abstract**

Host Identity Payload (HIP) is a new protocol layer that can help in adding more security, mobility and multihoming features into the current internet architecture. HIP would be an appealing alternative for home consumers because HIP tries to be as transparent as possible. End users can have varying requirements, such as wireless roaming, quality of service, and anonymity. This paper tries to address the needs of home consumers and to speculate whether or not HIP can satisfy these needs. A brief introduction of the reasons behind mobility difficulties is included, together with an overview of HIP architecture.

## 1 Introduction

Over time, the Internet has evolved from a computer science testbed into a world wide networking infrastructure. World Wide Web (WWW) was one of the things that awoke the interest of ordinary home consumers and led them to start using the Internet. As the amount of Internet users begun to grow exponentially, there was also growth in criminal activity: viruses spread out more easily and numerous hosts were intruded. Computer engineers and scientists begun to develop protocols and tools to help decrease the amount of criminal activity in the networks.

The demands for the network were also changed: originally the hosts in Internet were stationary and fixed, but nowadays support for mobility and multihoming is required. It is becoming clear that the current architecture is not optimal to support mobility. Therefore some extensions are called for.

This paper examines the Host Identity Payload (HIP) as one of the many attempts to redesign the current Internet architecture into a more secure environment, providing support for mobility at the same time. The paper focuses on the consumer side, especially the needs appearing in home computing and communications. The terms “home environment” and “home network” are generalized in this paper, and they refer to consumer homes and their fixed or wireless networks. Consumers will also be referred as “end users” in this document.

Before a discussion of HIP is started, an introduction on mobility and multihoming limitations in the current internet architecture is given in section 2. Host Identity Payload

architecture is overviewed in subsections of section 3 and especially the security issues are overviewed in the subsection 3.3. Section 4 is the most important part of this whole paper: various aspects of end user needs for HIP are discussed. Section 5 lists the conclusions of this paper.

# 2 Endpoint identifiers

The concept of an endpoint identifier has been long neglected [6] by the internet community in network and transport level protocols. Endpoint identifier means the ultimate name of the end of a transport level connection, no matter which route the network level packets are routed. Neglection of endpoints identifiers does involve everyone using the Internet, even the ordinary home end users. Before discussing more about HIP architecture, we will discuss the background motivation behind endpoints from the point of ordinary home end users, because endpoints are one of the most important motivations for using HIP.

To illustrate the problem with endpoint identifiers, let us consider two different scenarios: multihoming and mobility from the view of an ordinary home end user. The home end user acts as a traditional "client" using a light-weight terminal or a personal computer (PC). The home end user accesses Internet through a home network, which is some kind of IP-based residential or wireless network.

Generally, multihoming means that there are two or more different routes to a destination host. The redundant paths can be used in protecting against network failures, enabling load sharing and tuning performance up [16]. Typically, clients are not multihomed, but servers and routers are quite often multihomed.

Multihoming can further be divided into two different types: host multihoming and site multihoming. Host multihoming means that the host a two or more interfaces to the network, as in figure 1. Site multihoming is out of the scope of this paper because it is not interesting from the point of a ordinary consumer. An interested reader should see [16] for more information about site multihoming.

Mobility means that the client changes its topological location in the network (figure 2). This usually means that the client has to change its IP address. Mobility means also that existing connections should not be torn apart, although a small delay is usually involved in changing the IP address in a real world situation.

Transport level connections are formed commonly with Transmission Control Protocol (TCP) [17] or User Datagram Protocol in Internet (UDP) [18]. Both TCP and UDP use IP addresses as a part of their endpoint identifiers (see 3.2.1), that is, they share the same address space. The reason for sharing the same address space was a design decision based on the requirements of networks when Internet was still highly under development: the hosts were quite static and there was no need add a new address space for transport level connections. The new address space would have been redundant, because the hosts were static, and it would have involved an additional burden for routing. Routing would have been more complex because a mapping from transport level identifiers to network level (and vice versa) have had to been introduced.

Using the same address space in network level and transport level has introduced problems

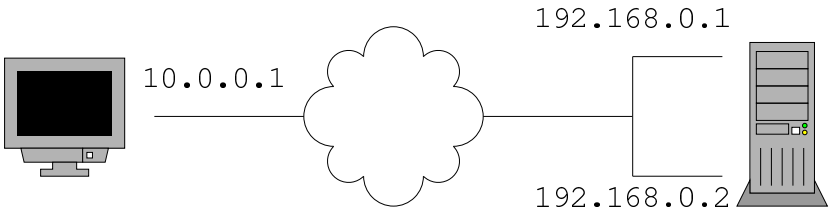


Figure 1: Simple end-host multihoming example

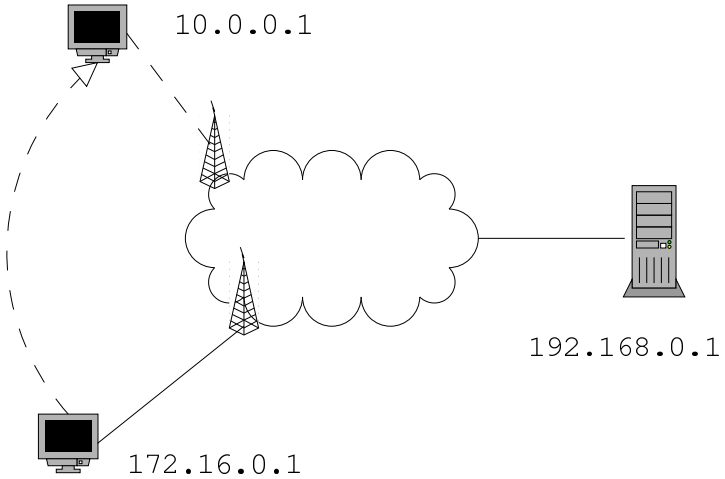


Figure 2: Simple mobility example

in both multihoming and mobility [6]. Here are three examples from the IPv4 world to further illustrate the problems:

**Example 1: End-host multihoming.** There is a client (marked with a monitor) with one network interface (10.0.0.1) and a server (marked with a PC-tower) with two network interfaces (192.168.0.1 and 192.168.0.2) in figure 1. Consider a situation where the client has established a transport level connection with the server and the connection is routed from 10.0.0.1 to 192.168.0.1. Now, the interface on 192.168.0.1 breaks down for some reason. The obvious solution would be to reroute the connection via 192.168.0.2, but this would break the connection in IPv4 or IPv6 without some kind of mobile-IP support.

**Example 2: Mobility.** If a host moves into another network (figure 2), it has to change its IP address. Again, all transport level connections will be torn down when the IP address changes, because the connection is bound to a static IP address.

All of the three previous examples have a common problem: changes in network level routing affect also end-to-end transport level connections. The reason for this is that transport uses the same identifiers (IP addresses) for hosts as network does. This causes a problematic dependency for transport level into the network level and it is one of the major reasons for multihoming and mobility problems.

There are many solutions and work-arounds for multihoming and mobility problems. Mobile IP [12] and Stream Control Transport Protocol (SCTP) [11] have been engineered to tackle the problems, just to mention a few alternatives. Host Identity Payload (HIP) is one

of the many alternatives and offers a new name space for identifying hosts independently of network level routing thus solving many of the mobility and multihoming problems. HIP also uses public key cryptography to reduce various network attacks to make internetworking safer.

# 3 Host Identity Payload Architecture

HIP architecture is described briefly in the following sections because currently HIP is not a widely known topic in the internet community. Internet drafts [3], [2] and [1] are the most valuable resource on HIP, since there are currently no RFCs on HIP. This section is basically an overview of the drafts, so no references to the drafts are given. If another source of information is used or a specific HIP draft needs to be pointed out, an explicit reference is given.

## 3.1 Host Identity

Endpoint identifier issues were shortly introduced in section 2 and identifying endpoints is in the core of HIP. Endpoint identifier in HIP is called a Host Identity (HI). HI is not just a plain name for an end-host as IP addresses are in the current TCP/IP architecture. HI is permanently integrated with security, because HI is a public key pair. A HIP implementation must support at least Digital Signature Algorithm (DSA) [13] as a cryptographic algorithm.

HI is a location independent identifier and the public key part of HI should be stored in a directory. HI public keys could be stored using Light Weight Directory Access Protocol (LDAP) [14] or Domain Name System (DNS) [15]. HIP supports also anonymous Host Identities which should not be stored in the DNS (otherwise they would not be anonymous).

## 3.2 Host Identity Tag and Local Scope Identity

A HIP protocol design using Host Identities would be inefficient because usually public keys tend to be long and inserting a long public key into a packet causes too much overhead. A public key could also be of variable length and the support for variable length public key identifiers would be harder to implement. Instead, a hash over the public key is used, producing a 128-bit field called Host Identity Tag (HIT).

HIT identifies HI in an efficient way and it can be used for further negation between end-hosts. HIT should be statistically unique but collisions are still possible. HIT should be interpreted as a hint of the correct public key in a collision situation.

Local Scope Identity (LSI) is even shorter than a HIT. LSI is a 32-bit localized representation of a HI. LSI exists mainly to support backwards compatibility with IPv4 Application Programming Interface (API).

The fixed lengths of LSI and HIT correspond exactly with IP address lengths in IPv4 (32-bit) and IPv6 (128-bit). This is not at all a coincidence but a careful design choice

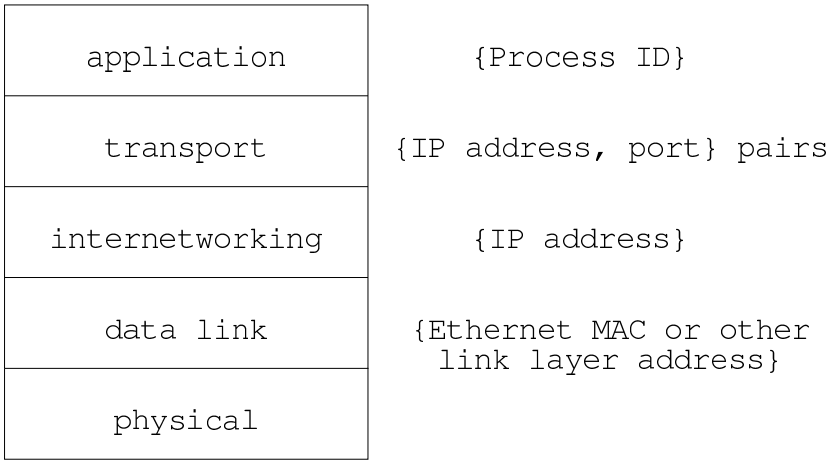


Figure 3: Current layering model

to support existing TCP/IP architectures. For example, to make a TCP connection to a host in IPv6, one must create a socket, bind the socket to the source and destination port and connect to the host IP address. The connection mechanism and API calls could be transparently the same in HIP aware end-hosts, but the IP-address would be interpreted as a HIT. This means that the semantic meaning of the API call is changed in HIP. API changes will be discussed more in the following section.

### 3.2.1 Host Identity Protocol Layering Architecture

The simplest way to begin the introduction of HIP layering architecture can be started by comparing the current internet layering architecture with HIP layering architecture as in [5].

Current internet layering architecture is shown in figure 3. Processes have process identification numbers (PIDs) to be distinguished from each other. A process can make a transport level connection (lower level connections, "raw sockets", are also possible) to an another process on the same or another host. Transport level connection identifier is a {*source IP address*, *source port*}, {*destination IP address*, *destination port*} pair [17]. IP address suffices for an identifier in internetworking level. Data link layer uses hardware addresses, such as Ethernet Media Access Control (MAC) addresses.

HIP drafts state that a new layer is needed in the current architecture to handle Host Identities. The new layer is placed between transport and internetworking layers as shown in figure 4. If the current and HIP layering architecture are compared, it can be seen that there are more changes in the HIP layering architecture than just an additional level: transport level connections are now identified with a pair of {*HI*, *port*} instead of a pair of {*IP address*, *port*}. This affects the semantics of API calls, which was brought up in the previous section.

Drafts do not state explicitly why HIP should be placed between transport and network levels. It could be argued that it is a better placement than above transport level, at session layer in Open Systems Interconnection (OSI) reference model. The reason for this argu-

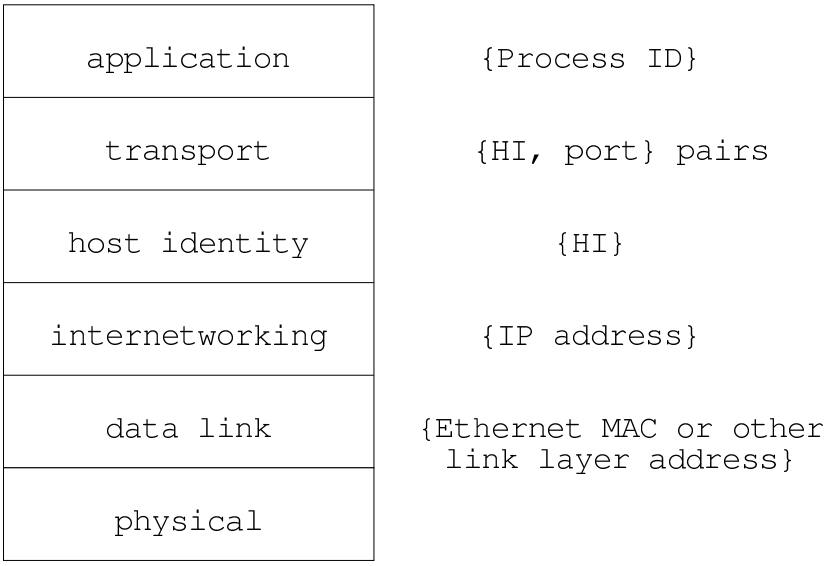


Figure 4: HIP layering model

ment is that connections should be protected at the lowest possible end-to-end level for better security. HIP security features will be discussed more in section 3.3.

HIP must provide a binding from Host Identities to IP addresses and vice versa, because HIP stands between transport and internetworking layers. HIP architecture proposal does not tell exactly how the mapping should be done. However, in [5] it is proposed that an initial binding could be achieved by storing HI and some IP addresses in DNS.

What is the significance of adding a new layer into the current architecture? Clearly it involves penalty pay-offs in form of increased data traffic and complexity in protocol handling. On the other hand, now we can identify endpoints and avoid most of the multihoming and mobility problems associated with current internet architecture, at least in theory. As recalled from section 2, the major reason for problems with multihoming and mobility ascends from neglecting endpoints identifiers. This is not enough yet, because other problems involved with multihoming and endpoints are security related and that is the topic of the next section.

### 3.3 Authentication and Encryption

The primary function of HIP is to bring real end-host identification into the current architecture. HIP uses Host Identities to accomplish this task. Since Host Identities are cryptographically based identifiers instead of just plain names, they can be used to provide some level of security into networking. HIP itself provides only authentication support for connections by means of *base exchange*. Base exchange is examined in some more detail in section 3.4, but basically it is a simple authentication mechanism that is designed to avoid further attacks against the authentication mechanism itself.

Let us take an example of connection establishment and authentication using HIP. In this example, both end-hosts are connected physically into an ethernet network and they are

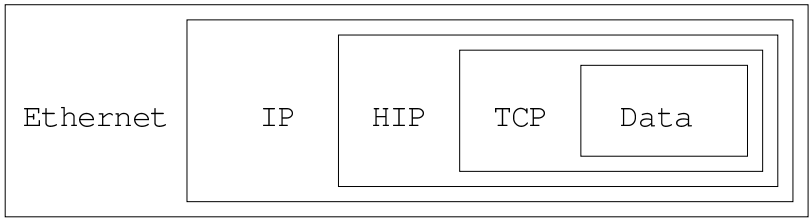


Figure 5: Encapsulation in an example connection establishment

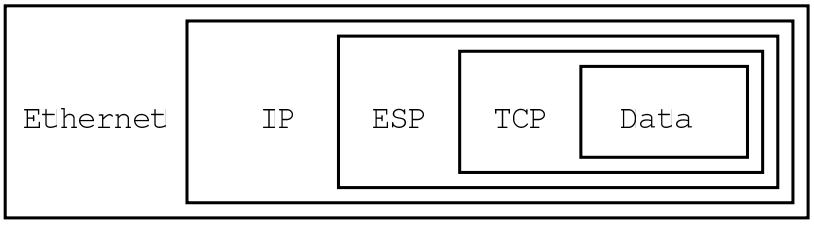


Figure 6: Encapsulation during an example connection

also HIP enabled. One of the hosts wishes to make a TCP connection to the other host. Before a connection can be established, hosts must run the base exchange between them. Base exchange packets are encoded in a HIP header, which is physically located in IP packet payload, as illustrated in figure 5. Any kind of extension headers are omitted for simplicity in the figure.

After the base exchange has been carried out, there is no need anymore to embed the HIP header in IP packets, since HIP is used just for determining endpoints and establishing authentication. Instead, Encapsulated Security Payload (ESP) [7] can be used in IPv6 to encrypt the rest of the connection as illustrated in figure 6. This saves both bandwidth and computing resources because HIP header does not have to be tagged into every packet sent.

### 3.4 Base Exchange: An Example Scenario

Let us see an example scenario extracted from [1] using the information presented so far. A HIP aware host, which will be referred as initiator, wants to create a transport level connection to another HIP aware host, which will be referred as responder. Before the connection can be established, the end-hosts must be authenticated to each other using HIP "base exchange" protocol. Base exchange involves sending HIP packets encapsulated inside IP packets as in figure 5. The example scenario is shown in table 1.

**Steps 1 and 2**. The initiator looks up the IP address, HIT and HI of the responder from DNS in the first two steps. The remaining packet exchange (four packets) is called "the base exchange".

**Step 3**. The initiator sends a HIP packet marked as I1 in step three which basically means that the initiator is trying to see if the responder is able to speak HIP.

**Step 4**. The responder does not trust the initiator at this point: the responder is aware of various Denial-of-Service (DoS) attacks and tries to minimize the impact of a possible

| step number | packet type | direction of flow | action |
|---|---|---|---|
| 1 | | I → DNS | Lookup R |
| 2 | | I ← DNS | R's address, HI, and HIT |
| 3 | I1 | I → R | Hi. Here is my I1, let's talk HIP |
| 4 | R1 | I ← R | OK. Here is my R1, handle this HIP cookie |
| 5 | I2 | I → R | Compute, compute, here is my counter I2 |
| 6 | R2 | I ← R | OK. Let's finish HIP cookie with my R2 |

Table 1: An example scenario of a HIP base exchange

DoS attack by initiating a 3-way cookie exchange with the initiator in step four. The impact of a DoS attack is minimized because it is the responder, not the initiator, that gives the challenge in R1. The responder sends also its public Diffie-Hellmann key in packet R1.

**Step 5**. Initiator is forced to do computation in step five to produce the response in I2 for the challenge that was in R1. This makes an DoS attack unprofitable for the initiator. Initiator sends also its public Diffie-Hellman key in packet I2.

**Step 6**. When responder has checked that I2 is correct, the connection is verified using R2 from initiator. ESP encrypted datagrams can be sent from now *on because both of the parties have authenticated theirselves using HIP protocol. Figure 6 shows an example of the packet encapsulation when ESP packets are being sent.

The previous example was explained at a very high level of abstraction. A reader interested in more detail should see [3], because it also shows the HIP packet format (which is not represented in this paper) and how the base exchange fits into HIP datagrams at bit level. Ericsson has some explanatory material about HIP, including flow graphs and ascii graphics of all of the exchanged HIP packets [9]. Ericsson has also planned to do some optimization on the base exchange.

### 3.5 Multihoming and Mobility

End-host multihoming can be attained with just the basic HIP features. Consider an end-host with two networking interfaces as in figure 1 and that the end-host has published its HI, HIT and both of the IP addresses in a (DNS) directory. All existing connections are kept running if one of the interfaces goes down because connections are bind into a HI rather than into an IP address of a specific interface.

Mobility can be handled with the HIP mechanisms listed in the previous sections if the initiator changes its IP address, as stated in [2]. If responder changes IP address, a *rendezvous* server is needed to forward the initial HIP packet to the responder. Only the initial packet is forwarded and the rest are handled like initiator packets [2]. A more elaborate scheme of achieving mobility is discussed in paper [5]. The same paper discusses also the double jump problem which basically means that both of the connected end-hosts change their IP

*The specifications of HIP state that ESP encrypted data can be encapsulated already in packets I2 and R2

addresses at the same time.

### 3.6 Implementation

HIP is work in progress and there are no ready implementations of HIP currently. A team of students, that includes the author of this paper, is implementing a prototype of HIP (HIP for Linux, HIPL) as the time of this writing in Helsinki University of Technology. Ericsson has also expressed its interest in prototyping HIP on FreeBSD [9].

## 4 Home Networks

### 4.1 Usage Scenarios

Traditional home networks using a single fixed line have not usually needed support for mobility. Home networks have not needed support for end-host multihoming either, because there was usually only a single personal computer (PC) in the household. However, the needs of ordinary consumers are changing, because information technology is driving the world of today towards an electronically networked infrastructure.

Networking has become easier since the introduction of electronic mail and WWW. They are probably the most common services that consumers tend to use. Consumers also use networks from their workplaces because most of the workplaces are handling or will soon be handling information using computers and networks instead of plain old paper.

Some of the people bring their laptops from work to home. Some might also have hand held gadgets, for example calendars and notebooks, which they bring from work to home for some reason. The reason for carrying these devices to home may be that they need to do some of their office work at home or that they like to use the same device everywhere they go. The consumers may have a family in their home and the children are usually playing some networked games with the stationary home PC in Internet thus keeping the PC reserved from adult use.

The devices that consumers carry around are usually just portable instead of being mobile. Portability means that devices are connected to the network only when the consumer is at work or at home. Outside the workplace or home the device is either turned off or there is no physical network plug or wireless network station nearby. This a very common scenario and even though there are link stations for cellular phones practically everywhere, it is still quite expensive and slow to access networks through cellular phones.

So what the end users really want? That is a good question and the answer can vary depending on who is asked the question. It is obvious that all end users cannot and will not be technically skilled in computer science: ease of use and transparency is needed. End users do not want to use mobile devices if a two-inch user manual must be read before a mobile device can be used everywhere. The device itself should be as smart as possible and hide the details of network from the user. Such devices are hard to implement and the security of such a “plug and play” device can be speculated, but the end users probably care more about usability than security. Discussion of this aspect is continued in section

4.7.

In addition to ease of use and transparency, the end users want also low cost. Pricing is more of a marketing issue and it is excluded from this technically orientated paper. Quality of Service (QoS) is probably what everyone wants and the level of QoS can usually be increased with money. A faster connection could be used where it is needed. In addition to faster connection, an alternative routing mechanism could be associated with QoS. Consider a situation where an end user has to put up with a fixed network ISP that has network breaks often. The end user could have an alternative route (e. g. wireless) to access Internet and the whole process of switching to different route could be totally transparent to the end user.

In the future there could also be wearable devices with network access such as wrist watches or recording eye glasses that would be used by the consumers. Fridge, television and other home equipment could be controlled from the network. Although these devices are stationary and do not need mobility, their network connections should be protected from malicious usage. Somebody could do, for example, expensive water damage by turning the water taps on while the owner of an apartment was on a holiday, if the water system could be controlled from network.

In science fiction, there are often silicon based artificial intelligence entities in movies. This could be real in distant future and these entities could be used as information service agents for human end users. Certainly at least these entities would need the ability to roam in networks?

Let us not get carried too much in science fiction. Next sections contain more realistic discussion about HIP in current home networks.

## 4.2 Shared Connections

One PC computer hardly suffices a modern family with children. The children may need one computer for playing and studying and the parents may need one computer for teleworking or paying bills. In case of multiple computers at home using only one Internet connection, the connection is usually shared with Network Address Translation (NAT). HIP can be used in such an environment, even though [2] states that there should be some changes both in the end-host and in the forwarding NAT host.

There could be some kind of a simple home firewall installed in the NAT host that protects the other hosts behind the NAT host from intruders. Firewalls are yet not common for ordinary users, but various firewall tools are becoming user friendlier and they may become more attractive for end users. One could argue that firewalls would become obsolete in an ideal world where every host could be identified by its public key, like the one provided by HIP. That argument could be theoretically correct, but it will not happen ever in real life. Programmers and users make are still human and are prone to unintentional mistakes that could be misused by intruders (this is discussed further in section 4.7). Therefore it is wise to always have a backup plan, such as a firewall. Besides that, the firewalls would be more powerful than the current ones because the actual identities of hosts could be verified cryptographically instead of just blindly trusting the source IP address, as in current internet design.

## 4.3 Wireless Networks

Wireless networks are inherently less secure than those using fixed lines, because all of the information is transmitted through air. Usually wireless networks are engineered using radio technology and they are less secure than fixed networks, because it is easier to spoof with a radio transceiver outside the residency than to break into the apartment and insert a physical wire-tap into communication wires. A natural counterargument to this would be that spread spectrum technologies would provide enough security, but that is not real security [10]. A wireless network based on infrared beams is safer than one based on radio technology, because infrared beams have a shorter range. Unprotected infrared traffic could still be eavesdropped in the same way as in unprotected radio traffic, even though the transceiver would have to be closer to the building.

Support for mobility at home is not necessarily required for a stationary device with wireless access, because the network access node is in most cases in the same physical location and thereby the IP address need not to be changed. The situation is not the same with a wireless network based on infrared: the current network access node can change if the networking device is just carried out of the range, like to the adjacent room. In such a case, some simple support for roaming would be needed.

A more complex support for roaming would be needed if a device was being carried across different Internet Service Providers (ISPs) during, for example, a long family holiday trip in a car through all of the states in America. This kind of support would be nice to have, but not necessary in the main interests of consumers.

HIP is also optimized to avoid unnecessary traffic and could be used on bandwidth poor wireless networks due to the design features explained in 3.3. Still, the wireless network devices could be physically very small and have a very limited computing capacity and just handling the cookies in base exchange could be too much for these tiny gadgets. This is further discussed in section 4.5.

In any case, HIP combined with ESP would bring security into unsafe wireless networks in a transparent way. It would also ease up wireless roaming, but full support for mobility is still a work in progress even in HIP. [2] provides some pointers on HIP mobility, but nothing really concrete is actually stated. A more elaborate view of mobility in HIP is discussed in [2].

## 4.4 Quality of Service

Quality of Service (QoS) includes at least two important things that are discussed here: performance and reliability. HIP could be used in increasing the QoS in end user hosts by tuning performance up and by adding reliability in the network connections.

Reliability could be improved if the end-host of a consumer has two or more network interfaces and thus multiple redundant paths to Internet. Consider a situation where one of the interfaces breaks suddenly down. The transport level connections of the broken interface are torn down because transport level connections are bound to IP addresses. A HIP aware implementation would not tear the connections down because transport level connections are bound to HITs. The connections would be just rerouted trough a working

interface.

The end-host of a consumer could also want more performance from the network by load balancing the network traffic between two or more interfaces of the end-host. A HIP implementation could ease up setting this kind of load balancing because HIP makes also multihoming possible.

Do the consumers really need load balancing or reliability in their home networks? The answer would probably be "no" because the requirements of consumers are currently low and QoS techniques are immature. However, the needs of the consumers may change and a better QoS could be demanded in the future.

## 4.5 Small Devices

Small devices with network access are becoming increasingly common for home usage. The problem with small devices is the limited computing capacity, but there is way around it: one could use some kind of proxies or routers that would do most of the cryptographic calculation if HIP was being used.

Another vision is that the small devices would be a part of a cluster, that has at least one device with sufficient computing power to support HIP. HIP could also be used in the migration of processes in a cluster system, but there are some fundamental problems associated with it [5].

Cluster systems are not currently appealing to consumers because clusters require a considerable amount of technical knowledge and skill to be successfully configured and maintained. Cluster systems are still evolving and they could be easier to use in the future. A cluster system could be a cheaper alternative in elevating the computing power at home, because the old hardware could still be reused in the cluster.

## 4.6 Anonymity

HIP supports also anonymous HITs and by using packet tunneling and forwarding mechanisms, almost complete anonymity could be achieved [5], so that only the end-hosts would know about each other. Anonymity would be appealing for home end users, but governments would not necessarily be too happy about it because they usually want to exercise control over citizens.

## 4.7 The User Perspective of Security

The security that comes along with HIP seems interesting from an engineering point of view, but do the end users really want security or anonymity? It is more than often seen that end users go around or completely give up security concerns, especially if they have think or spend some to learn how to use new security tools or features [19]. In the future, end users could even get penalty charges of ignoring computer security, because someone might use the penatrated computer of the end user to hack or DoS other hosts in the Internet.

Public key cryptography in HIP requires some kind of security policy. Who makes the decision of accepting a public key: the human or the computer? It is proposed in [20] that as much of the security as possible should be automatized. Automatization can be partly achieved by using certificates which could be supported in a HIP implementation.

It seems to be that HIP could be almost a transparent solution from the point of end users, so that HIP could be a corner stone in adding security into the unsafe world of Internet. However, it remains to be seen how many security problems this kind of "black box security model" would actually solve and how many new problems it would create. In any case, the user perspective should not be ignored when HIP is further developed or otherwise the overall level of security will decrease [21].

# 5 Conclusion

Mobility, multihoming and security have been an active research area in recent years because current internet architecture has been originally designed to be very static and insecure. Many alternative solutions have been engineered to address the problems with the current design and one of them is HIP.

HIP would be an ideal alternative solution from the point of consumers, because it is almost a complete "all-in-one" system: security, mobility and multihoming are included. HIP still needs some auxiliary systems, like rendezvous servers, for full mobility. HIP also needs changes in the kernels of existing end-hosts. There are currently no fully functional prototypes of HIP, so it is more research than reality. More research must be done to see if HIP could someday be run in all computers of regular consumers.

# References

[1] Moskowitz, R., *Host Identity Payload Implementation*, work in progress, an internet draft, January 2001

[2] Moskowitz, R., *Host Identity Payload Architecture* work in progress, an internet draft, January 2001

[3] Moskowitz, R., *Host Identity Payload And Protocol*, work in progress, an internet draft, October 2001

[4] HIP Official Homepage: http://homebase.htt-consult.com/HIP.html

[5] Nikander, Pekka, *A Case for the Host Identity Payload: An Architecture for Multihomed Mobile Hosts*, unpublished manuscript, February 2002

[6] Chiappa, J. Noel, *Endpoints and Endpoint Names: A Proposed Enhancement to the Internet Architecture* work in progress, an internet draft, 1999

[7] Kent, S., Atkinson, R., *RFC 2406: IP Encapsulating Security Payload (ESP)*, November 1998

[8] Host Identity Payload in Linux Homepage: http://gaijin.iki.fi/hipl/

[9] HIP for NetBSD Project: http://hip4inter.net/

[10] Komu, Miika, Nordström, Tero, *Known Vulnerabilities in Wireless LAN Security*, October 1999

[11] Stewart, R., *RFC 2960: Stream Control Transmission Protocol*, October 2000

[12] Perkins, C., *RFC 2002: IP Mobility Support* October 1996

[13] National Institute of Standards and Technology, *Digital Signature Standard (DSS)*, May 1994

[14] Yeong, W., Howes T., Kille S., *Lightweight Directory Access Protocol*, March 1995

[15] Mockapetris, P., *Domain Names - Concepts and Facilities*, November 1987

[16] Black, B, Gill, V., Abley, J., *Requirements for IPv6 Site-Multihoming Architectures*, work in progress, an internet draft, November 2001

[17] Postel, J., *RFC 793: Transmission Control Protocol*, September 1981

[18] Postel, J., *RFC 768: User Datagram Protocol*, August 1980

[19] Whitten, Alma, Tygar, J. D., *Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0*, 1999

[20] Karvonen, Kristiina, *Creating Trust*, referenced in April 2002

[21] Adams, Anne, Sasse, Martina Angela, Lunt, Peter, *Making Passwords Secure and Usable*, referenced in April 2002

## Abbreviations

| | |
|---|---|
| API | Application Programming Interface |
| DNS | Domain Name System |
| DoS | Denial-of-Service |
| DSA | Digital Signature Algorithm |
| ESP | Encapsulated Security Payload |
| HI | Host Identity |
| HIP | Host Identity Payload |
| HIPL | HIP for Linux |
| HIT | Host Identity Tag |
| ISP | Internet Service Providers |
| IP | Internet Protocol |
| LDAP | Light Weight Directory Access Protocol |
| LSI | Local Scope Identity |
| MAC | Media Access Control |
| NAT | Network Address Translation |
| OSI | Open Systems Interconnection |
| PC | personal computer |
| PID | process identification number |
| QoS | Quality of Service |
| SCTP | Stream Control Transport Protocol |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| WWW | World Wide Web |