

# Which Layer for Mobility? - Comparing Mobile IPv6, HIP and SCTP

Mika Ratola

Helsinki University of Technology

Telecommunications Software and Multimedia Laboratory

mratola@cc.hut.fi

## Abstract

The current Internet is based on an architecture created decades ago. Today however, the use of mobile devices and wireless networks present new challenges for location management and security. Therefore many alternative solutions have been engineered. This paper introduces and compares three mobility implementing protocols, each from a different layer. The purpose of the comparison is to determine which layer - three, three and a half, or four - would be best suited for mobility. The chosen protocols are Mobile IPv6 (MIPv6), Host Identity Payload (HIP), and Stream Control Transmission Protocol (SCTP) respectively. I want to emphasize that there is really no straightforward solution to the choice of layer for mobility. On the contrary, approaches used in different layers often complement rather than exclude each other.

KEYWORDS: MIPv6, HIP, SCTP, mobility comparison

## 1 Introduction

The Internet's current addressing scheme follows the design decisions made in the 1970's. Back then, the Internet was a quite static network, and all hosts were connected to it through one specific interface. Any computer could be easily identified with its unique IP address. The location directly identified the node in the network. Many things have changed since then, including computer networks.

The main revolution has been the deployment of mobile devices. With wireless interfaces giving ease of use, these devices have become increasingly popular. The underlying Internet, however, does not support the needed features and architectural structures for mobility. Because of that, the existing general mobility support solutions in the IP world have tried to hide the dynamic change of IP addresses from the higher layer protocols [20].

Another major change concerns security: Today's Internet is accessible for practically anyone, and this unfortunately opens a chance for misuse as well. Both individuals and businesses use the Internet for sending and receiving important messages, and these transactions must be properly secured. Therefore, when considering choices for mobility, security issues have to be taken into account.

Mobility can be implemented in different layers of the Internet architecture but this paper focuses on layers three, three and a half, and four. Since these layers have quite different tasks, one protocol from each of these layers is chosen

to represent the tasks of the layer. Mobile IPv6 represents layer 3, as it is currently well studied and implementations for it exist. SCTP, a new challenger for TCP and UDP, is selected for layer 4. Finally, between these two traditional layers mentioned above, a new solution for mobility is situated, namely HIP. It is selected for layer 3.5 because it offers new ideas and is an interesting design.

The purpose of this paper is to find some benefits and drawbacks when using one of the layers mentioned above as a place for mobility. To accomplish this, an overview of the three protocols with a short description of their functioning is given, and a comparison of the layers is done through the protocols.

The rest of the paper is organized as follows. Section 2 gives some background to current addressing, mobility, and security issues. In Section 3, the OSI-reference model is examined and some views for the location for the mobility are presented based on the model. Section 4 presents the three selected protocols giving an insight for comparison, which takes place in Section 5. Finally, Section 6 concludes this paper.

## 2 Background

As discussed above, there is a need for change from address orientation to host orientation. This means that the original situation of using only well-known single persistent IP addresses is no longer a viable solution. A setup of this kind was valid in the early days of the Internet, when four issues were considered invariants [13]:

- An address received was the address sent,
- Addresses were stationary (non-mobile),
- Source and destination were reversible, and
- All hosts knew to which address they should send packets to reach the wanted host.

According to Henderson, these assumptions cause four fundamental problems in the network layer [3]. The first concerns addressing: As IP routing and addressing are hierarchically defined for scalability, the mobile hosts usually have a topologically incorrect interface address when they attach to a new network. Secondly, when changing network, the mobile host may become unreachable to the rest of the network unless the new address is somehow mediated to other nodes (location management). The third problem is

related to session management: as the current transport protocols use the IP address as part of the connection identifier, the change of address breaks active connections. Finally, the mobile hosts must be able to authenticate themselves to their peers upon moving and maintain or re-establish network-level security associations.

The main issue to be resolved in the current Internet addressing scheme is the separation of the concepts *address* and *identifier*. Currently the devices connected to the network are identified by their IP addresses. When the mobile device moves between networks, its IP address changes and so does its identifier. The device has two choices to continue the ongoing communication with its peer: The new identifier is mediated to the peer or alternatively the device makes itself reachable via the original identifier. The three protocols discussed in this paper delve into this mobility challenge in their own way.

Mobility raises yet another issue: A host cannot in the truest sense "own" a location name because bits can be duplicated [20]. The only way to accomplish "ownership" in the Internet is to keep the item in question secret. Therefore, the identifiers must also be based on some cryptographic method.

### 3 OSI-model

In this section, the OSI-reference model is briefly introduced and then the focus is moved to layers 3 and 4. Finally, an analysis is made of the location for the mobility based on the OSI-model.

#### 3.1 Location of different functionalities

The OSI model is based on a proposal developed by the International Standards Organization as a first step toward international standardization of the protocols used in the various layers of communication architecture. The model was created in 1983 so it can be considered quite old. [18] Nevertheless, practically all network models developed after it follow the same principles, so it is used here to give insight on into which layer each task is generally placed. It is also worth noting that the OSI model itself was developed in an era when no actual mobility yet existed.

The model consists of seven layers. The two lowest ones handle issues concerning the physical medium and the transmission of raw data. Layers five, six and seven cover problems associated with sessions, presentation and application issues. In this paper, the main interest, however, lies in the middle of the model: in the network and transport layers.

According to the OSI-model, the *network* layers main functionalities concern the controlling of the subnet. Key issues here are routing of packets, congestion control, and the interconnection of different networks. On the other hand, the *transport* layers main function is to accept data from the layer above, split it up into smaller units if needed, pass these to the network layer, and ensure that the pieces arrive correctly at the other end. Related issues here are handling multiple connections, multiplexing, flow control, and creating an error-free point-to-point channel that delivers data in the order it was sent. [18] The key differences between these

two layers is that the transport layer is end-to-end while the network layer is chained between routers. In practice, this means that the transport layer must have some sort of naming mechanism to enable a conversation between processes over an established connection.

#### 3.2 Analysis based on the OSI model

As it was mentioned in the previous section, the OSI model does not take into account mobility issues. Even so, based on the basic functionalities suggested by the model for each layer, preliminary conclusions are made in this section into which layer the mobility would fit the best.

In the light of the model, it is quite obvious that both network and transport layers are already rather heavily loaded with functionality. Over the years, the functionalities have been optimised for different purposes in different domains in the form of well established protocols. However, this optimisation comes with a price: adding any new features or functionalities, such as mobility, is very difficult. Therefore, one solution for this problem is to add a new level, where the mobility is implemented. In respect to this paper, the solution would be the deployment of the layer 3.5 that operates between the network and transport layers.

There is a practical drawback, however, when using the new layer between network and transport layers. As all solutions provided so far have followed the structure of the old layers, a binding has formed between them. It really is not very simple to add a layer in between, since the functioning of the network and transport layers is in a way optimised to work together. An almost symbiotic relationship exists. Another issue is that nearly all of the modern implementations rely on the "old way" of viewing the protocol stack. If the layer 3.5 is used as the place for mobility, the adding of the layer should to some extent be transparent and easy. On the other hand, the current layers may be too loaded with functionality to allow the adding of the mobility into them in a sensible manner. In the following chapters, the three rivaling layers are viewed in more detail using a protocol to represent a mobility solution for the corresponding layer.

### 4 Current mobility solutions

In this section a solution for mobility is presented for each of the three layers discussed above. In Section 4.1, Mobile IPv6 is presented. An overview of HIP is given in Section 4.2, and, finally, SCTP is introduced in Section 4.3.

#### 4.1 Layer 3 - Mobile IPv6

##### 4.1.1 Overview

Mobile Internet Protocol version 6 (MIPv6) is a network layer protocol, and it is the current Internet Engineering Task Force (IETF) proposal for a standard for the mobility problem. The protocol relies on IPv6 [2], which was designed from the start to support mobility. MIPv6 enables a mobile device to maintain its IPv6 address and transport layer connections while its point of attachment to the network changes. However, it is worth noting that MIPv6 does

not attempt to solve all general problems related to the use of mobile computers or wireless networks, such as mobile routers [4]. Instead, the protocol offers transparent movement of a mobile node to transport and higher-level protocols and applications. In addition, MIPv6 is suitable for both homogenous and heterogeneous media, for example from an Ethernet segment to another, or from Ethernet to wireless LAN.

Although MIPv6 has come a long way, it is still under ongoing development by the IETF Mobile IP Working Group. The protocol is documented in the Internet draft Mobility Support in IPv6 [4], and the following chapters sum up the protocols operation.

### 4.1.2 Architecture

Each Mobile Node (MN) is identified by its Home Address (HoA). The address is given by a Home Agent (HA), which is a router supporting mobility services in the nodes home network. For discovering a HA, MN uses Dynamic Home Agent Address Discovery protocol.

If MN operates in its home network, conventional mechanisms are used to route packets addressed to it. When the node moves to another network, it acquires a new address called a Care-of Address (CoA) through either stateless or stateful automatic Address Autoconfiguration. The mobile node then informs the Home Agent of its current address. The association between MN's Home Address and Care-of Address is known as a "binding" for the node. Using this information, the Home Agent forwards any packets addressed to MN into the new location. This registration procedure is called a binding update (BU). MIPv6 enables nodes to cache these address bindings into HA's binding cache. [4]

Any node communicating with MN is known as a corresponding node (CN). CN may itself be a stationary or a mobile node. In a basic situation, all traffic between MN and CN are tunnelled through the Home Agent. Figure 1 illustrates this situation.

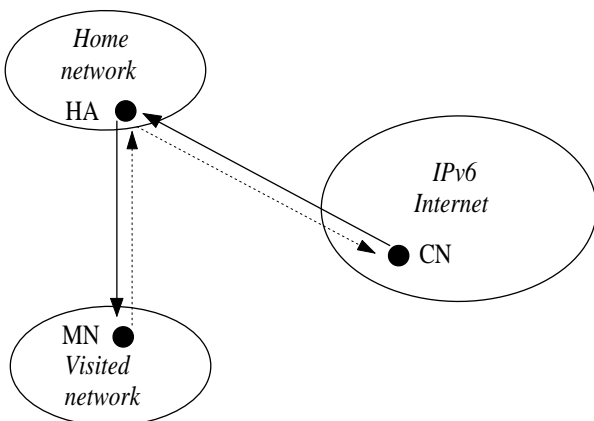


Figure 1: Tunneling traffic between the MN and CN

In Figure 1, data sent by MN to CN is illustrated with a dotted line and the opposite transmission made by CN with a solid line. All traffic goes through HA; this mode of communication is known as bidirectional or reverse tunnelling. IPv6

encapsulation is used in the tunnelling. The nice thing about this mode is that CN does not require to support Mobile IPv6 at all.

Even so, bidirectional tunnelling is not always efficient, especially if the MN is close to CN, and therefore communicating through the Home Agent creates an unnecessarily long path. MIPv6 offers a solution for this sort of situation through route optimization: Only the first packet is tunnelled through HA. Then MN can register its current location by sending its current binding information to a CN (a BU message). After this, the packets from CN can be routed directly to the Care-of Address of MN with the help of CN's home address in the routing header. Similarly, MN sends all packets to CN directly, using the Home Address destination option. Route optimization is presented in Figure 2.

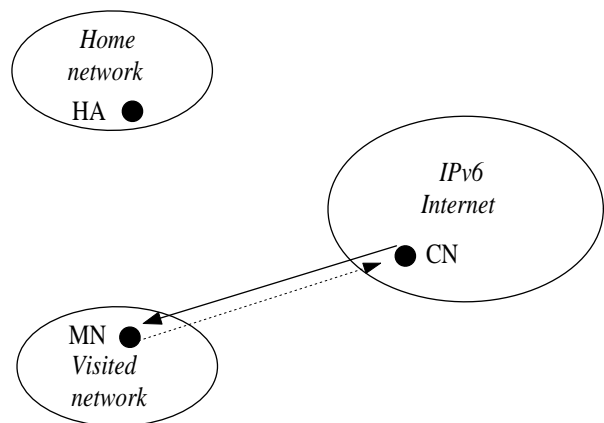


Figure 2: Route optimization for traffic between MN and CN

As previously, the data sent by MN to CN is marked with a dotted line, and the traffic from CN to MN is marked with a solid line in Figure 2. The shortest communication path is used when packets are routed directly to MN's Care-of Address. This also eliminates congestion around HA. In addition, in case of a failure in home network or in the path to it, the impact is reduced. [4]

### 4.1.3 Security

Binding updates are one of the key factors in the functioning of MIPv6. Therefore, the binding messages must be authenticated and protected against replay attacks to prevent malicious nodes from corrupting the binding caches with invalid addresses. [4]

Before using any binding updates, the Mobile Node must register to the Home Agent. This is done in order to create an IPsec Security Association (SA) between the two entities. If manual keying is used, SA is pre-installed. Internet Key Exchange (IKE) can also be used, if it is supported by both parties.

When the Security Association is created, it is used to authenticate the binding update messages between MN and HA. To achieve this goal, MIPv6 uses the IPsec framework [1]: either Authentication Header (AH) or Encapsulated Security Payload (ESP) can be used with a non-null authentication algorithm.

When authenticating the binding update between MN and CN, a return routability procedure is used instead of SA. The procedure uses cryptographic tokens in verification. Basically, CN sends test messages as a challenge, and MN responses. After this, MN constructs from random data and data gathered from the procedure a binding management key, K<sub>bm</sub>, that is used in the binding procedure. The Binding Updates are then protected against replay as the messages used have sequence number, and with a Message Authentication Code (MAC), tampered messages can be detected. The MACs are created with HMAC-SHA1 algorithm. [4]

#### 4.1.4 Problems

Although many security related issues have been dealt with in MIPv6, some problems still exist. One of these is the possibility of a Denial of Service (DoS) attacks if a malicious peripheral deploys false binding updates. This can happen when a new Care-of Address is acquired and the Neighbor Discovery is used. Internetworking and transition between Mobile IPv4 and IPv6 is yet another issue that probably will cause problems. Thirdly, the design principle of making the mobility invisible to the applications has resulted in a very complex architecture and extremely heavy protocol. Especially route optimization procedure suffers from a large overhead. Finally, as the registering procedure may take a long time, many packets will be lost. This may result in an unacceptable level of quality of service.

## 4.2 Layer 3.5 - HIP

### 4.2.1 Overview

Host Identity Payload (HIP) [13] is a solution that locates the mobility between the network and transport layers. It is actually a new namespace protocol, and it is mainly defined in three drafts [9, 10, 11] by Robert Moskowitz. An IETF Working Group exists for it as well.

HIP introduces a new Host Identity layer (layer 3.5) between the IP layer (layer 3) and the upper layers. The reason for this is to avoid the situation where binding sockets to IP addresses forces the address into a dual role of endpoint and forwarding identifier. In HIP, upper layer sockets are bound to Host Identities (HI) instead of IP addresses. In addition, the binding of these host identities to IP addresses is done *dynamically*. The purpose of HI is to support trust between systems, enhance mobility, and greatly reduce the DoS attacks.

A great advantage in this mobility solution is that the hosts can easily have both the current IPv4 and the new IPv6 addresses. Furthermore, there is no need to change the current routing methods. Multi-homing, NAT-traversal, anonymity, and avoiding Man in the Middle (MitM) -attacks are other features the HIP has to offer.[12]

In the following sections, the operation of HIP is presented in more detail. The protocol is documented mainly in three drafts (architecture in [9], protocol in [10], and implementation in [11]) which are used as the main source of information in presenting HIP in this paper.

### 4.2.2 Architecture

HIP is similar to MIPv6 in the sense that the main goal for both of them is to make mobility transparent to the applications. In HIP, the hosts are identified with public keys, not IP addresses. A typical host identity is a public cryptographic key of an asymmetric key-pair. Each host will have at least one HI that can either be public or anonymous.

The HIs can be different in sizes depending on the used public key method. Therefore, the HI is represented via its 128 bit (SHA-1) hash, called Host Identity Tag (HIT), or 32 bit Local Scope Identity (LSI). The HIT identifies the public key that can validate the packet authentication, and HITs should be unique in the whole IP universe. They are stored in some public address directory (e.g. DNS) with the exception of anonymous identities.

LSIs are 32 bit localized representations of a HI. Each host selects its communicating partners LSI, and the value must be random. Even so, collisions between different LSIs may easily occur, and therefore they should only be used in local scope according to local policies. The main reason for LSIs is to make the use of HIP possible with existing protocols such as IPv4. The LSI's advantage over HIT is its size; On the other hand, the LSI's disadvantage is its local scope.

One of HIP's features is authentication during connection establishment. To achieve this, the HIP-protocol (or Host Layer Protocol) [10] is used. However, normal packets cannot be used, and so HIP presents a new packet structure: The transport layer packet (e.g. TCP) must be enclosed with a HIP header, which contains the HIT. Figure 3 illustrates the situation. For simplicity, any extension headers are omitted from the figure.

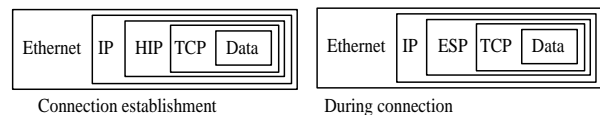


Figure 3: The HIP packet structure

HIP could be carried out in every datagram throughout the connection but alternatively the HIP payload can be compressed into an ESP payload (in IPv6) after the HIP exchange (see Figure 3). Thus, HIP packets are only needed to establish an authenticated connection.

As mentioned above, the HIP protocol is used to authenticate the connection. In addition to authentication, the procedure establishes Security Associations for a secure connection with IPsec ESP. The HIP-protocol uses a four-way handshake with Diffie-Hellman key exchange. The entity that wants to establish a connection is referred to as initiator and the other party as responder. Before the actual exchange takes place, the initiator has fetched the responders IP address, HI, and HIT from an address directory (e.g. DNS). Figure 4 illustrates the exchange, and the four packets used in it are explained below.

**I1 packet** is sent by the initiator to see if the responder speaks HIP. The packet contains the HITs of the both parties.

**R1 packet** is sent back as a reply by the responder. As the responder cannot yet trust the initiator, it initiates a three-

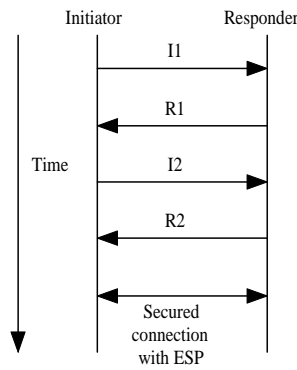


Figure 4: The HIP exchange

way cookie exchange. Packet R1 holds the responders public Diffie-Hellman key, HI, and information about the supported ESP modes as well as a challenge. The impact of a DoS attack is minimized as the responder is the one giving the challenge.

**I2 packet** contains the initiators public Diffie-Hellman key and a computed response to the challenge. The computation makes the DoS attack unprofitable for the initiator. The ESP options are also sent with the packet.

**R2 packet** completes the handshake. The responder sends it if the initiators response to the challenge was correct. After the sending of the R2 packet, the ESP encrypted datagrams (see figures 3 and 4) can be used to secure the whole connection.

During the secured connection, mobility in HIP is quite straightforward. As HIs are used to identify the mobile node instead of IP addresses, the *location* of the node is not bound to the *identifier*. Therefore only a simple signalling protocol (the HIP protocol discussed above) is needed to take care of the dynamic binding between the node's IP address and HI. When one of the communicating peers changes location, it simply sends a HIP readdress (REA) packet through the secured ESP channel. The SAs are bound to the HITs and not to addresses, and thus the connection continues uninterrupted.

However, if the responder changes location before the connection has been properly established or if both of the peers change location at the same time (the double jump problem), a *rendezvous server* is needed. It is a packet forwarding agent which simply temporarily forwards the initial HIP packet to the responder. All further packets are handled normally between the initiator and responder.

### 4.2.3 Security

The HIP security is quite good. Firstly, as discussed in the previous section, the connection establishment is well authenticated with the help of IPsec. During this procedure, the Security Associations needed for a secure ESP connection are obtained. Secondly, the HIP identifiers are public keys, and therefore they can be used to authenticate the HIP packets as well as to protect them from most Man-in-the-Middle attacks [9]. In addition using public keys as identifiers means that no explicit Public Key Infrastructure (PKI)

is needed. Thirdly, the impact of DoS attacks is decreased as the *responder* is the one giving the challenge and deciding its difficulty. If DoS attacks are attempted using multiple I1 packets, the responder can to some extent reuse the R1 packets. Finally, HIP supports anonymity as HITs can be anonymous. This is appealing for many users but on a governmental level it can be seen as a threat.

Having said all this about HIP, there are some security issues, that must still be addressed. The challenge mechanism used in HIP creates a possibility for new DoS attacks. Since interpreting the I2 packet takes some resources from the responder, a flood of those packets could overwhelm the responder. The solution to this problem is to accept only a fixed number of I2 packets from the same HI (i.e. node).

There are also a number of MitM attacks that can be used against HIP. The resolution to most of these attacks is to use secure and authenticated connections. In addition, the HIs can be fetched from a signed DNS zone so that these signed HIs are used to validate the HIP packets. Other problematic issues with HIP are presented in the following section.

### 4.2.4 Problems

The initial problem that HIP was designed to solve is the separation of location and identity of a node in order to achieve mobility. The architectural decision made was to add a new layer into the existing world wide communicational model. Although the solution has many benefits, the choice of using a new layer has a serious drawback: Any current node wanting to use HIP has to make changes in the operating system kernel. This argument should be taken seriously as it means updating practically all applications that in some form use the Internet. In addition, a change of this magnitude has never before been attempted, and the architecture of the layers has remained the same for decades. To sum it up, the main problem with HIP may not be a technical one but rather a marketing issue. After all, IPv6 and Universal Mobile Telecommunications System (UMTS) have not taken off as hoped, since no single party has applied enough pressure to the market or taken the role of a leader in using the new technology.

A much more modest problem is related to the features of HIP: the current specification of HIP does not support multicasting. Including this new feature of course means modifications to the protocol. With current trends moving towards the increased usage of multicasting, this issue almost inevitably has to be addressed if HIP wants to be one day the de facto mobility solution.

Another problematic issue with HIP is that in the end it does not *force* the use of security, even though the security can be achieved when HIP is configured and used properly. Having said that, this sort of approach has also an advantage. Currently HIP uses public keys, DSA for encryption, DNS as a sort of storage facility, and ESP to secure connections but in the future better solutions may be invented. The fact that HIP does not force any specific security methods gives it flexibility to be able to change along with new times.

The build in security considerations raise yet another problem for HIP. As the HIP namespace is cryptographic in nature and the public keys are used in the connection es-

establishment, heavy computations are needed. This presents a problem especially for mobile devices with limited CPU power. The impact is reflected as slower connection establishment.

## 4.3 Layer 4 - SCTP

### 4.3.1 Overview

Stream Control Transmission Protocol (SCTP) [15] is an IETF proposed standard protocol for the transport layer. It is designed to eventually replace TCP and perhaps also UDP. Like TCP, SCTP is reliable but offers new features such as *multi-streaming* and *multi-homing*. In particular, the multi-homing feature of SCTP enables it to be used for mobility support, without any special router agents in the network. Other features included in SCTP are error-free and non-duplicated data transfer, network-level fault tolerance through supporting of multi-homing, and resistance to flooding or masquerade attacks.

As mentioned earlier, the multi-homing ability enables SCTP to support mobility. A host is called multihomed if it has multiple network layer addresses (e.g. IP addresses). A transport protocol supports multi-homing if the endpoint can have more than one transport layer addresses, as is the case with SCTP. The mobility comes here from the ability to change the endpoints (e.g. IP addresses) while keeping the end-to-end connection intact.

The problem in SCTP is to perform these address reconfigurations dynamically. The solution is to use the Dynamic Address Reconfiguration (ADDIP) [16] extension for SCTP, which enables the SCTP to add, delete, and change the IP addresses during an active connection. The SCTP with the ADDIP extension is called mobile SCTP (mSCTP) [6], and it provides a seamless handover for mobile hosts that are roaming between IP networks.

The SCTP was originally created by the IETF Signalling Transport Work Group (SIGTRAN WG) but found a new home in 2001 with the Transport Area Work Group (TSV WG). The SCTP itself is an RFC [15] but mSCTP exists only as a draft [6]. However, in this paper the main operational focus is placed on mSCTP, and the following sections summarize its functioning.

### 4.3.2 Architecture

Originally SCTP was designed to provide a general-purpose transport protocol for message-oriented applications, as is needed for the transportation of signalling data (e.g. in the Public Switched Telephone Network). A key difference to TCP is the concept of several streams within a connection which are known as *associations*. An SCTP stream represents a sequence of *messages* as opposite to a sequence of bytes. Messages can be *bundled* together, which means that they are multiplexed into the same SCTP packet.

SCTP can be used as the transport protocol for applications where monitoring and detection of loss of session is required. For such applications, the SCTP peer and path failure detection mechanisms, especially the *heartbeat*, will actively monitor the connectivity of the session.

An SCTP packet is composed of a 12 byte *common header* and *chunks*. In the header, a 32-bit checksum is used to detect transmission errors. SCTP packets with an invalid checksum are silently discarded. A randomly created 32 bit *verification tag* allows a receiver to verify that the SCTP packet belongs to the current association and not to an old one. The chunk on the other hand may contain either control information or user data. Chunks have variable length and there are currently 13 types of them in standard use.

The association establishment in SCTP, as in HIP, uses the four-way handshake. The passive side is called a *server* and the other is a *client*. The handshake procedure is as follows: First, the server receives an INIT chunk. Using its data, the server generates a secure hash of these values and a secret key. These values along with a MAC are put into a COOKIE, and returned in an INIT-ACK chunk. The client using the received COOKIE assembles a COOKIE-ECHO chunk and returns it to the server. Finally, the server verifies with the MAC, that the COOKIE is the same as it sent, and replies with a COOKIE-ACK chunk. Now the association is established. When one of the communicating parties wants to end the association, it can be done in two ways: Either by graceful shutdown, ensuring that no data is lost, or hard termination (abort), not taking care of the peer. Unlike TCP, when either endpoint performs a shutdown, both of the endpoints stop accepting data.

During association startup, a *list of transport addresses* (i.e. IP address-port -pairs) is provided between the communicating entities. These addresses are used as the endpoints of different streams. SCTP regards each IP address of its peer as one "transmission path" towards this endpoint. The association spans transfers over all of the possible source/destination combinations. Also one of the addresses is selected as *initial primary path*, which may be changed later if needed. The ADDIP extension used in mSCTP aids in this dynamic address reconfiguration. Below the whole soft handover with mSCTP is described [7] with the help of Figure 5.

In the figure the mobile node (MN) initiates an SCTP association with the corresponding node (CN). The resulting association consists of IP address 2 for MN and IP address 1 for CN (the primary path). After a while, MN decides to move (the thick arrow) from network A to network B. The following steps [6] are repeated every time MN moves into a new location.

*Step 1: Obtaining an IP address for new location.* As MN is moving towards network B, at some point it reaches the overlapping region (see Figure 5). Then MN obtains the new IP address 3 from the Access router B with the help of DHCP or IPv6 address auto-configuration.

*Step 2: Adding the new IP address to the SCTP association.* MN informs CN of the new address by sending an Address Configuration Change (ASCONF) chunk. As a reply the ASCONF-ACK is sent.

*Step 3: Changing the primary IP address.* While MN further continues to move towards Access router B, it needs to set the new address as its primary address. The changing of addresses is done according to specific rules, for example as soon as a new IP address is detected. However, the configuration of this change triggering rule is a challenging issue for

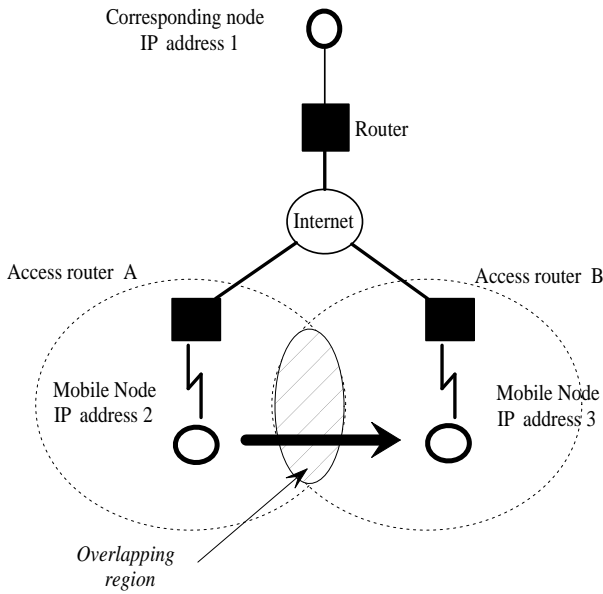


Figure 5: mSCTP soft handover

mSCTP.

*Step 4: Deleting the old IP address.* As MN has moved to network B, the old IP address becomes inactive, and it is deleted from the address list. The knowledge from underlying layers can be used to determine when the address becomes inactive.

It is worth noting that if CN is initiating the association towards the MN, a *location management* scheme is needed. Mobile IP can be used, for example, for CN to find the current location of MN and to establish an SCTP association. After the association is successfully setup, the mSCTP will be used for providing seamless handover as discussed above. [8]

### 4.3.3 Security

Being a transport layer protocol, SCTP has the advantage to use services provided by the layers below it. One of these services is security, offered by the network layer. For instance, IPsec can be used to achieve data integrity (with AH) and data confidentiality (with ESP). [19]

Still, SCTP offers some own security measures, such as the four-way handshake mentioned in Section 4.3.2 and resistance to flooding or masquerade attacks (the "cookie" mechanism discussed in Section 4.3.2). The use of Verification Tags prevents the insertion of extraneous packets into the flow of an established association (e.g. blind attacks). Some vulnerabilities still exist to MitM attacks.

### 4.3.4 Problems

The main problems protocols nowadays have mostly relate to security. As mentioned in the previous section, SCTP can rely on the layers below to handle these issues. However, the mobility presents its own minor problems to mSCTP. The protocol is mainly targeted for client-server services, in which the *client initiates* the session with a fixed server.

For supporting peer-to-peer services, the mSCTP must be used along with an additional location management scheme as discussed in Section 4.3.2. In addition, the ADDIP extension used in mSCTP to achieve seamless handover is only a draft, and therefore more work needs to be done in the test and implementation of this option to make it work as expected [14].

Another problematic issue arises when the underlying network operates on IPv6. Certain address types supported by IPv6 are not routable (i.e. link-local) or reachable outside of specific domains (i.e. site-local). If a peer lists one of these addresses to a peer that has no connectivity to that address, an association could self-destruct and create a black hole effect [17].

Performance in wireless environments can also cause problems for SCTP. The protocol assumes that all losses are caused by congestion but in wireless networks higher bit error rates and more frequent delay spikes are encountered. This will cause SCTP to back-off unnecessarily, and result in poor throughput. [14]

## 5 Comparison

The previous chapters presented three mobility supporting solutions from layers three, three and a half, and four. In this section, these three layers are compared through the presented protocols. It is worth noting that the comparison is not universal for the layer in question but the main aspects are found with this sort of comparing approach. The key issues to consider are security, signalling and other functional overhead, and the effects on both applications and overall architecture.

As all of the three protocols are quite new, the security considerations were taken into account from the start. For example, IPsec can be used together with all of them. Even so, the return routability procedure used by MIPv6 has known security problems [20], and SCTPs security proposals are heavy [13]. HIP on the other hand has been specifically designed with security in mind. It has a simple and fast solution to the key distribution system, which is one of the biggest issues with MIPv6. In addition, HIP unlike MIPv6 enables location changes that do not break ESP secured connections. The cryptographic nature of HIP namespace also increases support for security.

Cryptographic methods used in HIP, however, require heavy computations. This may present efficiency problems at least for mobile devices with limited CPU power. On the other hand, HIP like SCTP functions directly between endpoints whereas MIPv6 may have to use the home network [3]. This naturally increases the distance for packets to travel and is more vulnerable to networking errors. Having said that, the SCTP routing tables may become large when connection loads are extensive, although SCTP is considered as efficient as TCP [5]. SCTP requires heavy signalling as well [13].

In addition to efficiency aspects, the architectural changes have to be considered too. HIP and SCTP both work with current IPv4 and future IPv6 networks but MIPv6 relies only on IPv6. Furthermore, MIPv6 requires changes to

routers whereas the other solutions do not. When comparing from the applications point of view, however, HIP requires changes to the kernel breaking at least some of the applications, and SCTP is incompatible with all old applications [13]. Nevertheless, known implementations for all of the three protocols exist.

When considering the big picture, HIP adds a totally new layer to a well established Internet architecture which increases data traffic and implementation complexity [3]. On the other hand, this approach avoids most of the multi-homing and mobility problems associated with the current architecture (e.g. the concept of identity). The problem with MIPv6 is that it is burdened with backward compatibility issues making it extremely complex, and end-host multi-homing is not yet supported [20]. SCTP offers multi-homing services, and is the best mobility solution for cellular networks. The reason for this is that HIP and MIPv6 suffer from undesired end-to-end latency when readdressing is rapid [3].

To sum it up, each protocol or layer has its benefits and drawbacks. The environment and markets greatly dictate, which aspects are most appreciated at a given time. In the following section I conclude my view for the best location for mobility in the Internet architecture.

## 6 Conclusions

Mobility and security have been an active research area in recent years because the current Internet architecture has been insecure and originally designed to be very static. Old assumptions do not hold anymore as hosts and addresses are no longer equal. Therefore many alternative solutions have been engineered. In this paper, three protocols, each from a different layer, were examined.

Everyone should remember that there is really no straightforward solution to the choice of layer for mobility. On the contrary, approaches used in different layers often complement rather than exclude each other. The difficulty of choosing a layer best suited for mobility is apparently acknowledged in the research community as no one really has before performed this sort of comparison between the layers.

The fact is, however, that almost all current software implementations rely on TCP or UDP. This should be noted when choosing a layer for mobility. Selecting the transport layer will cause problems and reconfiguration but the lower layers do not have as great effect. On the other hand, the network layer is stuffed with functionality, and there really exists no good implementation there that would cover all security aspects. Therefore, in my opinion, the mobility should be implemented in a new layer between network and transport layers. HIP seems to be a good solution for mobility in layer 3.5 as it solves many security, mobility, and multi-homing issues at the same time. After all, if changes have to be made in any case, why not just rethink and renovate the whole architecture while we are at it.

## References

- [1] Comer, Douglas E., *Internetworking with TCP/IP Vol I: Principles, Protocols and Architecture*, Fourth Edition, Prentice Hall, 2000, 750 p.
- [2] Deering, S., Hinden, R., *Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, IETF IP Version 6 Working Group*, December 1998, <ftp://ftp.rfc-editor.org/in-notes/rfc2460.txt>
- [3] Henderson, T., R., Ahrenholz, J., M., Kim, J., H., *Experience with the Host Identity Protocol for Secure Host Mobility and Multihoming* In. *IEEE Wireless Communications and Networking*, pp. 2120-2125, 16-20 March 2003, vol.3
- [4] Johnson, D., Perkins, C., Arkko, J., *Mobility Support in IPv6, Internet draft, version 24, IETF Mobile IP Working Group*, June 2003, <http://www.ietf.org/internet-drafts/draft-ietf-mobileip-ipv6-24.txt>
- [5] Jungmaier, A., Schopp, M., Tüxen, M., *Performance Evaluation of the Stream Control Transmission Protocol* In. *Proceedings of the IEEE Conference on High Performance Switching and Routing*, pp. 141-148, 26-29 June 2000
- [6] Koh, S., J., et al., *Mobile SCTP for Transport Layer Mobility, Internet draft, version 3, IETF*, February 2004, <http://www.ietf.org/internet-drafts/draft-sjkoh-sctp-mobility-03.txt>
- [7] Koh, S., J., Jung, H., Y., Min, J., H., *Mobile SCTP for IP Mobility Support in Transport Layer* In. *Proceeding of CIC (Cellular and Intelligent Communications)*, October 2003, Korea <http://pec.etri.re.kr/~sjkoh/pub/2003-cic-sjkoh.pdf>
- [8] Koh, S., J., Xie, Q., *mSCTP with Mobile IP for Transport Layer Mobility, Internet draft, version 3, IETF*, February 2004, <http://www.ietf.org/internet-drafts/draft-sjkoh-mobile-sctp-mobileip-03.txt>
- [9] Moskowitz, R., *Host Identity Payload Architecture, Internet draft, version 2, IETF*, February 2001, <http://homebase.htt-consult.com/~hip/draft-moskowitz-hip-arch-02.txt>
- [10] Moskowitz, R., *Host Identity Payload and Protocol, Internet draft, version 5, IETF*, October 2001, <http://homebase.htt-consult.com/~hip/draft-moskowitz-hip-05.txt>
- [11] Moskowitz, R., *Host Identity Payload Implementation, Internet draft, version 1, IETF*, February 2001, <http://homebase.htt-consult.com/~hip/draft-moskowitz-hip-impl-01.txt>
- [12] Nieminen, K., *Tietoturvaprotokollat, 3.4.2003, Presentation in course 8306500 Tietoturvaprotokollat in Tampere University of Technology* <http://www.cs.tut.fi/kurssit/8306500/HIP-esitelma.pdf>
- [13] Nikander, P., *IPCN 2001, From Address Orientation to Host Orientation* <http://www.tml.hut.fi/~pnr/presentations/IPCN2001slides.pdf>

- [14] Shaojian, F., Atiquzzaman, M., SCTP: state of the art in research, products, and technical challenges In. *IEEE, Proceedings of Computer Communications Annual Workshop (CCW)*, pp. 85-91, 20-21 October 2003
- [15] Stewart, R., et al., Stream Control Transmission Protocol, *RFC 2960, Network Working Group*, October 2000, <http://www.networksorcery.com/enp/rfc/rfc2960.txt>
- [16] Stewart, R., et al., Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration, *Internet draft, version 8, Network Working Group*, September 2003, <ftp://ftp.ietf.org/internet-drafts/draft-ietf-tsvwg-addip-sctp-08.txt>
- [17] Stewart, R., Metz, C., SCTP: new transport protocol for TCP/IP In. *IEEE, Internet Computing* pp. 64-69, Nov.-Dec. 2001, vol. 5
- [18] Tanenbaum, Andrew S., *Computer Networks*, Third Edition, Prentice Hall, 1996, 813 p.
- [19] Tüxen, M., Sctp bakeoff and other topics, 2000 <http://www.ietf.org/proceedings/00jul/SLIDES/sigtran-bakeoff/tsld015.htm>
- [20] Ylitalo, J., Jokela, P., Wall, J., Nikander, P., Endpoint Identifiers in Secure Multi-homed Mobility, <http://www.tml.hut.fi/~pnr/publications/Opodis02-Ylitalo-et-al.pdf>

## Abbreviations

ADDIP - *Dynamic Address Reconfiguration*  
AH - *Authentication Header*  
BU - *Binding Update*  
CN - *Corresponding Node*  
CoA - *Care-of Address*  
DoS - *Denial of Service*  
ESP - *Encapsulated Security Payload*  
HA - *Home Agent*  
HI - *Host Identity*  
HIP - *Host Identity Payload*  
HIT - *Host Identity Tag*  
HoA - *Home Address*  
IETF - *Internet Engineering Task Force*  
IKE - *Internet Key Exchange*  
IP - *Internet Protocol*  
Kbm - *binding management Key*  
LSI - *Local Scope Identity*  
MAC - *Message Authentication Code*  
MitM attack - *Man-in-the-Middle attack*  
MN - *Mobile Node*  
mSCTP - *mobile SCTP*  
OSI - *Open Systems Interconnection*  
PKI - *Public Key Infrastructure*  
SA - *Security Association*  
SCTP - *Stream Control Transmission Protocol*  
TCP - *Transmission Control Protocol*  
UDP - *User Datagram Protocol*  
UMTS - *Universal Mobile Telecommunications System*