Simulation of the Registration and the Base Exchange Protocols in HIP Layer with Estimation of the Handover Latency

Y B Jhansi
Dept of Computer Science and Engineering
S V University College of Engineering
Tirupati – 517 502, India

Ch D V Subba Rao
Dept of Computer Science and Engineering
S V University College of Engineering
Tirupati – 517 502, India

ABSTRACT
In the current Internet, hosts are identified using IP addresses that depend on the topological location of the hosts. In other words, the IP addresses are semantically overloaded since they identify both the hosts and topological locations. These dual operations of the IP address causes problems when the host has to change its IP address due to mobility. The location information changes, but it should not affect the identity information of the host.
The Host Identity Protocol (HIP) is rather new concept that separates the identity and location information. The separation is done by introducing a new layer between the transport and network layers of TCP/IP stack called HIP Layer that maps host identifiers to network locators.
In this paper, we will discuss how the mobility problem is addressed by HIP, the simulation of the Registration and the Base Exchange protocols in HIP Layer along with the estimation of the Handover Latency. And we will compare TCP/IP with HIP over TCP/IP. The Handover Latency is the Metric we are used to compare between the two.

Keywords
Mobility, Host Identity Protocol, Base Exchange, Registration Protocol, Handover Latency, OMNeT++.

1. INTRODUCTION
The Internet was originally designed with the assumptions that hosts are static and trusted. These assumptions are not correct anymore, because users are getting more mobile everyday. Their computers are moving with them wherever they go, for example to school, the library, work, etc.
Because of this growing mobility there has been a demand for a more flexible and mobile internet protocol suite. In the current Internet, hosts are identified using IP addresses that depend on their topological location. The IP addresses are thus overloaded since they identify both the hosts and their topological location. When the typical IP protocol suite is used, whenever a host moves to another location it has to break down all its connections and build them back up again with the new IP address. This problem is addressed as Mobility problem.
The IETF (Internet Engineering Task Force) and IRTF (Internet Research Task Force) for example are working on multiple similar solutions to meet the new requirements (mobility, multi-homing and security). The Host Identity Protocol (HIP) [1] [4] is one of these efforts. The HIP tackles the problem at the root. It decouples the name of the location of the host from the name the applications use to communicate with the host. The Host Identity Protocol (HIP) provides a method of separating the end-point identifier and locator roles of IP addresses [2]. HIP allows consenting hosts to securely establish and maintain shared IP-layer state connections, allowing separation of the identifier and locator roles of IP addresses, thereby enabling continuity of communications across IP address changes. By doing this the HIP ensures a solution to the mobility problem found in the typical IP suite.
The paper is organized as follows. Section 2 gives details related to HIP. Section 3 discusses the Base Exchange Protocol. Section 4 discusses the Registration Protocol and RVS extension. Section 5 presents how HIP handles the mobility with single secure association pair. Section 6 describes the simulation of the Registration and the Base Exchange protocols. Section 7 presents the outcome of the simulation. Section 8 gives the details related to the simulation of the TCP/IP. Section 9 gives the comparison of the TCP/IP with HIP against TCP/IP. Section 10 presents the conclusion and recommends future work.

2. HOST IDENTITY PROTOCOL
The current Internet architecture, though hugely successful, faces many difficult challenges. The most important ones are the incorporation of mobile terminals (hosts) and an overall lack of protection against Denial-of-Service attacks and other lacking security mechanisms. Most existing approaches are point-solutions that patch support for a subset of the required improvements into the current Internet architecture, but do not cleanly integrate with one another and do not present a stable base for future evolution. The HIP [1] [4] is a promising new basis for a secure mobile Internet. The cornerstone of HIP is the idea of separating a host's identity from its present topological location in the Internet [2]. This simple idea provides a solid basis for mobility [3] feature.

2.1 Identifier–locator split
IP address has two roles: a topological information string and an identifier of a host.

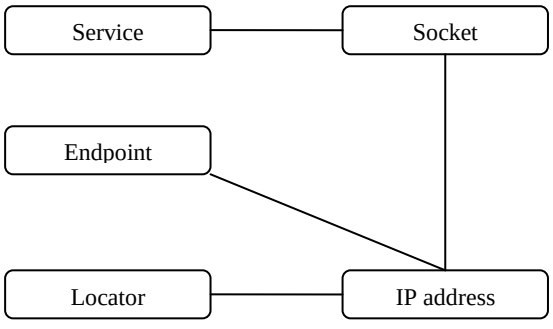

Figure 1 Logical entities connected to IP address.

An application bound to a socket by an IP address is also bound to the topology. If a host changes the IP address, because of moving to another network, an application must break the upper layer bindings and restart the communication via the new IP address. This is very often the case with mobile hosts. When a host moves in the Internet topology from a network to another it is said to be mobile.

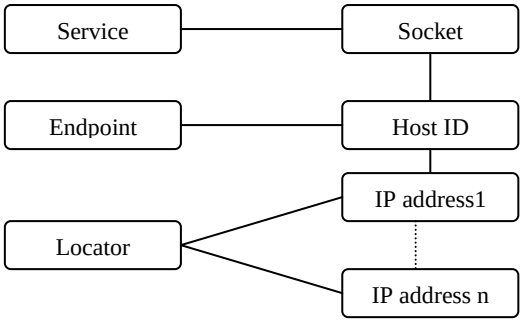

Figure 2 Separating location and identity of Internet hosts.

To solve the above problem the roles of IP addresses are splited, this is known as Identifier–locator split [2]**.** The separation is done by introducing a new layer between the transport and network layers of TCP/IP stack called HIP Layer that maps host identifiers to network locators. IP addresses remain only geographical locators i.e., they purely identify the location of the nodes and no longer take care of the identification. Identification is done by using Host Identifier.

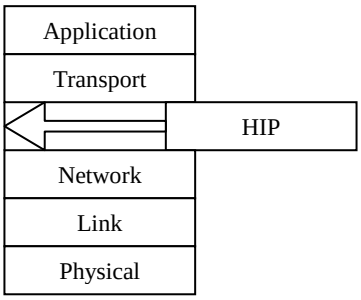

Figure 3 New TCP/IP suite

Two main protocols running in HIP layer are Base Exchange Protocol [6] and Registration Protocol [5]. In the next section we will discuss briefly about the Base Exchange protocol.

3. BASE EXCHANGE PROTOCOL

A HIP association is set up between hosts using the HIP Base Exchange [6]. This cryptographic exchange is used to establish an IP-layer communications context between the hosts. The context is needed to store all information related to the connection, such as the Host Identity used during the connection. The Base Exchange is a four packet exchange. The party which starts the Base Exchange is called the Initiator, and consequently the other party is the Responder.

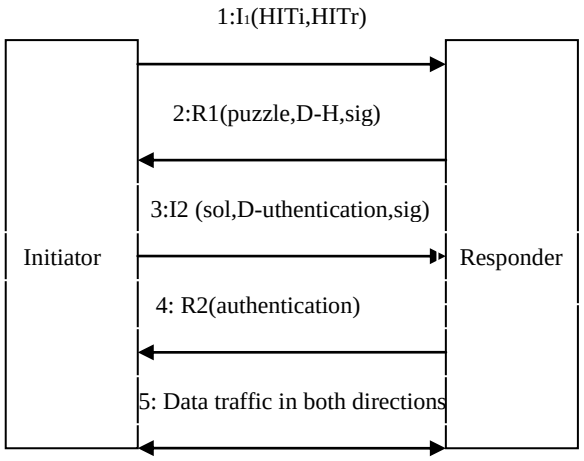

Figure 4 The Base Exchange

Base Exchange [6] is quite similar to the TCP connection establishment procedure. When the Initiator wants to initiate the HIP Base Exchange, an I1 packet is sent to the Responder. The packet contains only the HIT of the Initiator and possibly the HIT of the Responder, if it is known (if the Responder HIT is not known, this is called as opportunistic mode). I1 packet might be spoofed, so Responder does not perform any time consuming operations on this packet. Second packet sent by the Responder, R1, is sent as a reply to the I1 packet. R1 contains the public value of Diffie-Hellman key of the Responder, connection related information such as supported encryption algorithms of the Responder, and finally a signature covering part of the packet data to a void packet replay attacks. Additionally, R1 contains a puzzle which the Initiator must solve in order to make a successful base exchange. Puzzles are designed in such a way that solving them requires significantly more time than creating them. If Responder has many simultaneous connection attempts, it can simply create a harder puzzle. Responder could also set puzzle difficulty based on its level of trust of the Initiator.

Puzzles are used in an attempt to diminish the effect of Denial of Service (DoS) attacks. DoS attacks can cause huge performance hit to the target, so it is very important to at least try to avoid them. Pre-computed puzzles and re-using R1 packets for a short period of time provide additional security for the Responder. Third packet, I2, contains Initiator's public value of Diffie-Hellman key, Initiator's selection of encryption algorithm supported by the Responder, Host Identity of the Initiator (encrypted using the selected encryption algorithm), Security Parameter Index (SPI) for Responder's ESP traffic, and the calculated answer to the puzzle. Signature of the packet is appended to the packet. The fourth and the last packet is R2, which contains Initiator's SPI and signature over the packet. The rest of a successfully finished base exchange is that the hosts have authenticated themselves to each other and created bidirectional IPsec ESP Security Associations for the HIP related traffic. Right after the Base Exchange packets carrying user data can be sent securely over the ESP channel.

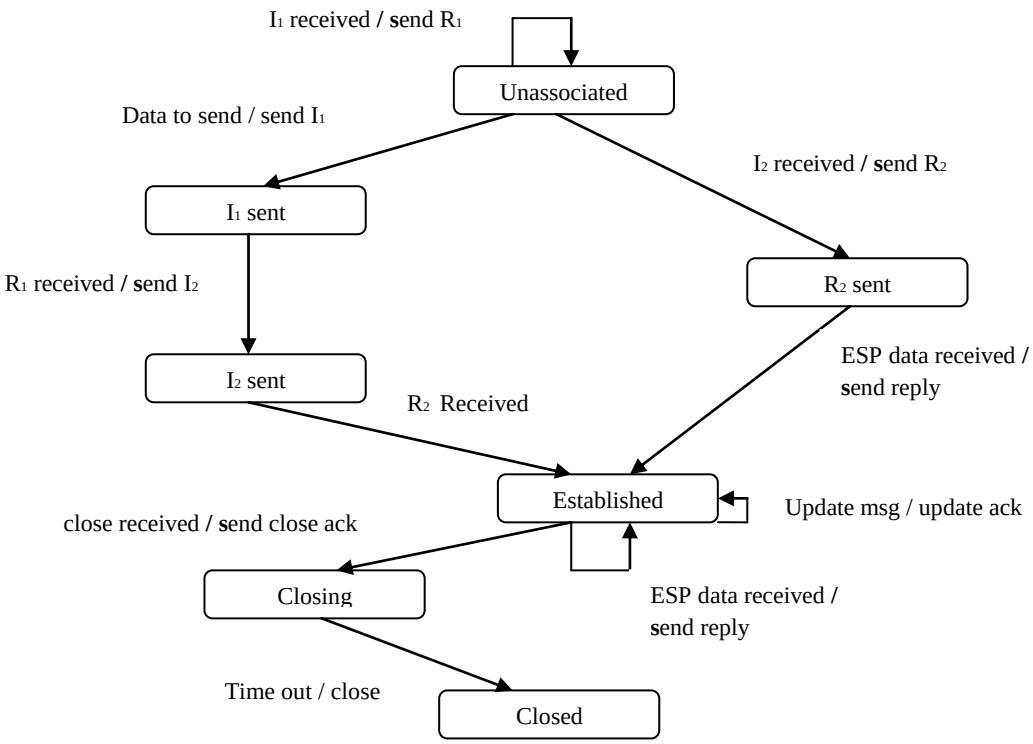

Figure 5 State Transition Diagram of HIP

4. REGISTRATION PROTOCOL

There are situations where the simple end-to-end readdressing functionality is not sufficient (e.g. the initial reachability of a MN, simultaneous mobility of both nodes). In these situations there is a strong need for some extension of the normal HIP architecture [9]: a new network entity was introduced called the HIP Rendezvous Server (RVS) [7]. If a HIP aware MN enters the network and becomes reachable, it should register its new IP address in a network directory, which is known by all the potential Peer Nodes. Basically this is a kind of DNS functionality. However a traditional DNS is not prepared handling frequent address changes. Therefore it is a better solution to use a special entity for tracking the changes of IP addresses. This is the HIP Rendezvous Server [7]. From this section we assume that the MN knows at least one RVS (i.e. the MN knows the IP address and HIT of the RVS.). The MN entering the network registers [5] it's new IP address at the RVS (Figure 6) and reports the IP address of the RVS at the DNS.

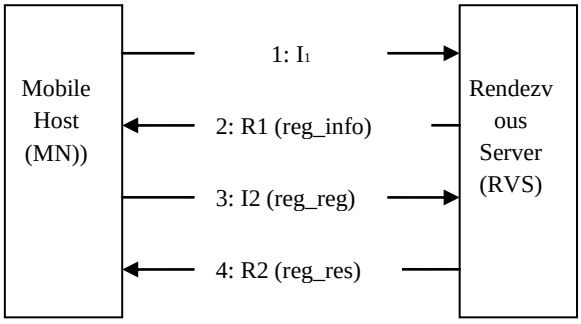

Figure 6 Registration of MN with RVS

If the MN moves to another attachment point while changing its IP address, the node updates its entry at the RVS. Now if another node wants to connect the MN, it performs a lookup at the DNS for the IP address of the MN. The DNS answers with the IP address of the MN's RVS. The Peer Node now initiates the HIP connection by sending the I1 packet to the RVS with the HIT of the MN. The RVS forwards the packet according to the containing HIT of the MN. Furthermore the RVS adds a FROM parameter to the packet representing the IP address of the Peer Node. The MN answers with the R1 packet sending it directly to the Peer Node. The MN adds a VIA_RVS parameter to the packet, which contains the IP address of the RVS [7]. Finally the two nodes finish the Base Exchange [6] in the regular way.

5. HIP MOBILITY WITH SINGLE SA PAIR

If one of the nodes moves form an attach point to another, its IP address usually changes. Since the Host Identity layer is responsible for mapping HIs to IP addresses, the moving node must report its new IP address to its Peer Node. To make the protocol capable maintain this situation some extensions were added: A new HIP parameter was defined to enable a mobile HIP node to update an existing HIP association i.e. to report its new IP address to the Peer Node. This is the LOCATOR parameter. If the mobile node (MN) moves to another attach point, it sends an UPDATE packet with a LOCATOR parameter in it. The LOCATOR holds the new IP address and some other information e.g. the SPI associated with the new IP address. The MN may optionally send an ESP_INFO parameter to create a new inbound SA (rekey). In this case the LOCATOR contains the new SPI to use. Otherwise, the old SPI is identified in the LOCATOR parameter, and the node waits for its UPDATE to be acknowledged.

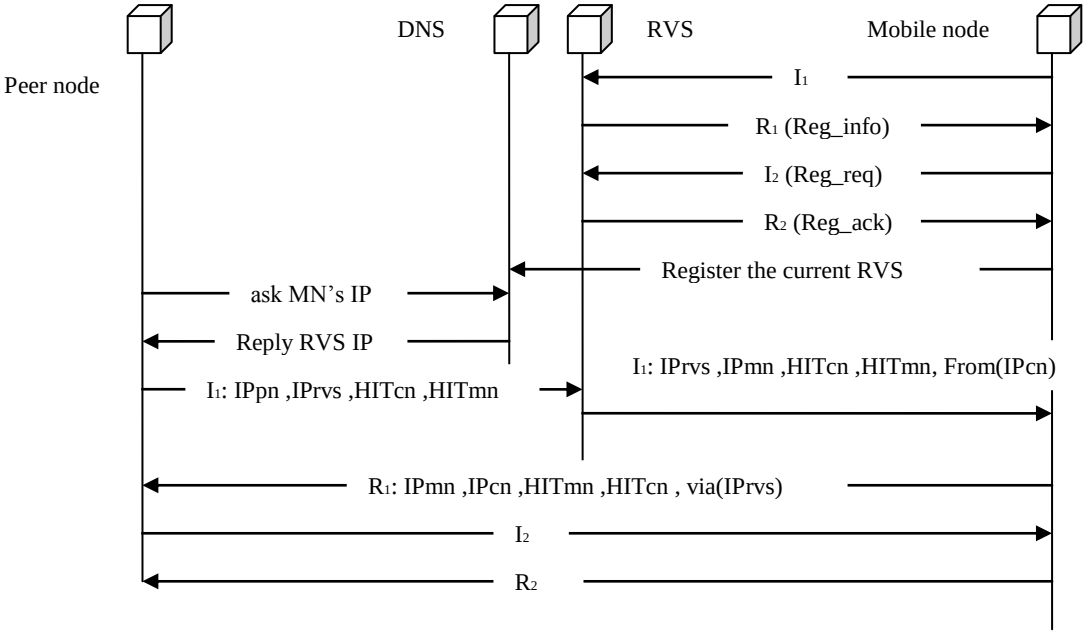

Figure 7 Base Exchange with Rendezvous Mechanism

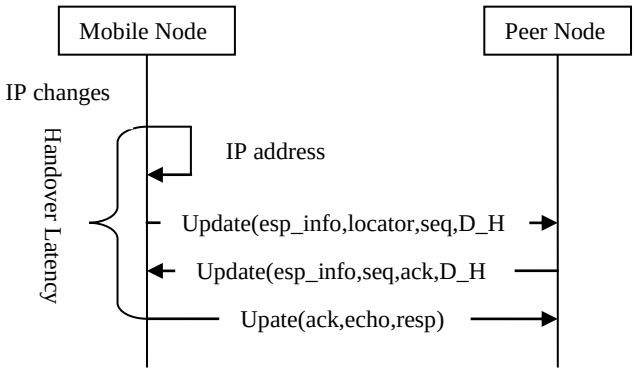

Figure 8 Update Mechanism

6. SIMULATION OF REGISTRATION AND BASE EXCHANGE PROTOCOLS

6.1 Introduction to OMNeT++

In this section the basics of OMNeT++ [10] are introduced in order to give an overview about the fundamentals of the simulation environment we used to develop our Host Identity Protocol simulation.

OMNeT++ is an open-source, component-based simulation package built on C++ foundations. It offers a C++ simulation class library and GUI support (graphical network editing, animation).

The simulator can be used for traffic modeling of telecommunication networks, protocol modeling, modeling queueing networks, modeling multiprocessors and other distributed hardware, systems, validating hardware architectures, evaluating performance aspects of complex software systems, modeling any other system where the discrete event approach is suitable.

An OMNeT++ model consists of hierarchically nested modules. They are simple module, compound module and system module.

Simple modules are the lowest level of the module hierarchy. Simple modules contain the algorithms in the model. The user implements the simple modules in C++ by using the OMNeT++ simulation class library. Compound Modules are module contains submodules, which can also contain submodules themselves. It connects internal simple and compound modules. The top level module is the system module.

Modules communicate with messages thus message sending and receiving are the most frequent tasks in simple modules. Messages contain common attributes (like timestamps) and also arbitrary ones (i.e. any other kind of user data). Simple modules typically use gates (input and output interfaces of modules which can be linked with connections) for sending messages, but direct send to destination modules (using an invocation from the OMNeT++ simulation kernel) is possible as well. OMNeT++ messages can be easily defined by specifying the fields and other possible message content in .msg files and by letting OMNeT++ to take care of creating the necessary C++ classes from the .msg definitions.

The topology of a model is specified using the NED language. Files containing network descriptions generally have a .ned suffix. It is used to define the structure of simulation models in OMNeT++. The simulation models contain modules and their interconnections. A typical .ned description file consists of simple module declarations (i.e. description of the module's interfaces), compound module definitions (i.e. declaration of the module's external interfaces and definition of submodules and their interconnection) and network definitions (i.e. compound modules that are self-containing simulation models). In this way model behavior and model topology are separated: behavior is defined in C++ code, while topology is determined by the NED language. Simulation parameters (i.e. initial parameters of simulation runs which are independent both from the C++ and the NED codes) are specified in .ini files. Separating initial inputs in this way enable users to run simulations for each one of the

interested parameter combination without modifying the existing codes.

6.2 Main Modules of HIP

6.2.1 HIP module

For every new HIP session in HIP layer, HIP module creates a daemon instance called HIPSM. And it is responsible for all mechanisms of the HIP State Machine (HIP SM) described in [8], e.g. for handling HIP Base Exchange and HIP mobility functions. One such daemon instance cares of one SA, which will be identified by the local SPI. HIP SM daemons are registered by destination and source HITs (and SPIs) in the *HIP module*. HITs have to be provided by the applications (or rather the transport layer), therefore HIP-capable DNS extensions are also integrated into HIPSim++. It is also responsible for managing change of state occurring in addresses of host interfaces.

6.2.2 HIPSM module

The *HIPSM module* implements the main functions of the HIP State Machine (shown in Figure 3.2). In our model transitions of HIP State Machine assume that packets are successfully authenticated and processed. One instance of *HIPSM* represents and manages one HIP connection with one Security Association. *HIPSM* handles transitions occurring during HIP Base Exchange, RVS registration, UPDATE mechanism, etc. and generates HIP messages according to the state transitions. *HIPSM* module also handles changes in partner IP addresses (sets the locators by receiving and processing UPDATE messages), but the actual storage happens in the main HIP module's *hitToIpMap* structure.

6.2.3 RvsHIP module

To extend the basic HIP capabilities with the RVS functions, *RvsHIP module* is derived from the *HIP module*. The main purpose of it is to handle the incoming registration messages according to [5] and by forwarding I1 messages [7] to the appropriate HIP responder chosen from the registered ones.

6.2.4 DnsBase module

The *DnsBase module* is a simple UDP applicastion. It realizes basic DNS server functionality for name resolution of HIP hosts and implements the new Resource Record (DNS HIP RR). The module resolves domain names to HITs and IP addresses and in case of mobile HIP hosts also provides RVS information.

7. OUTCOME OF THE SIMULATION

A simulation in OMNeT++ [10] can be run in two different ways: visual and text-only. The visual simulations are shown in the following graphics. This way of running the simulation is particularly useful when first running the simulation, or to get acquainted with the protocols or networks the program simulates. It shows all the messages that are exchanged between the modules in an animation. Also, with larger simulations, you can look deeper into each module, to see what messages are exchanged internally.

Figure 9 shows the main window of the simulation. This screen includes several controls to run the simulation. Simulations can be run step by step, normal (shows every message as an animation), fast (show animations, but faster) and express (which doesn't shows any animation). It is also possible to run the simulation until some point.

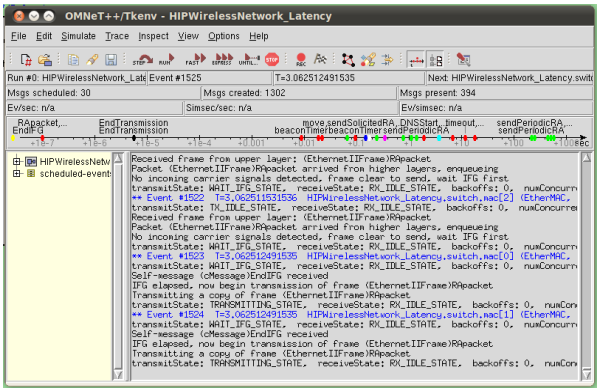

Figure 9 Main Window of the simulation.

Figure 10 show the actual network that is simulated, which has only twelve components (rvs, dnssrv, hipsrv, switch, 3 Access points, 4 routers and mobilehiphost).

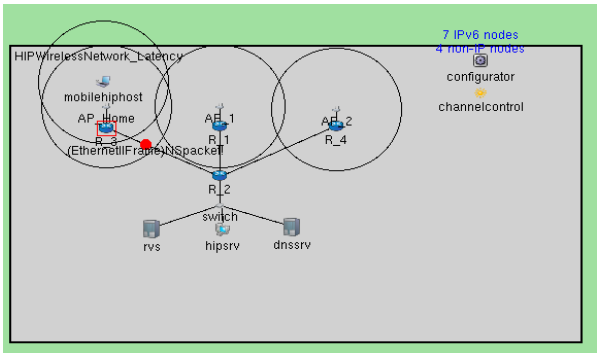

Figure 10 Network topology for HIP handover.

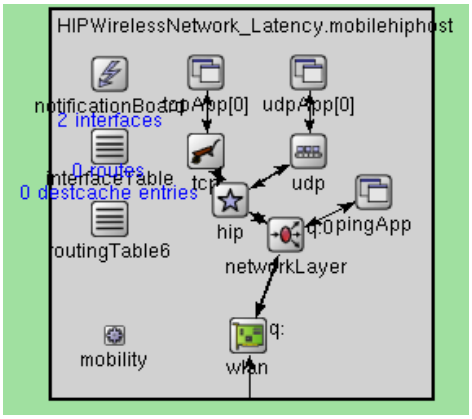

Figure 11 Internal structure of Mobile HIP node.

In the simulation, first mobile HIP node and wired HIP node is registered at RVS [7] by using Registration protocol. Next Base Exchange [6] is performed between the registered mobile HIP node and the wired HIP node. After this the data transfer is done between the two nodes. Meanwhile the mobile HIP node changes its location and associated with new access point. At that time UPDATE parameter is sent by the mobile HIP node to the wired HIP node giving information about its new IP address. Without breaking the connection handover is performed between the two nodes.

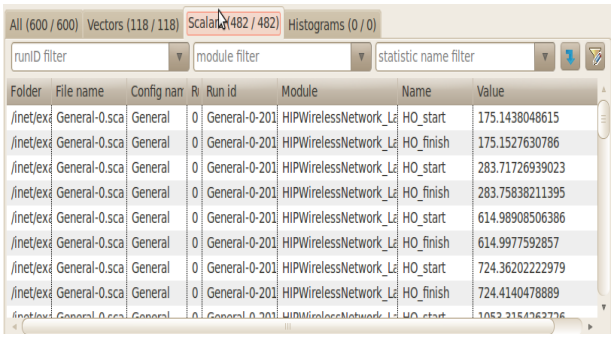

Figure 12 Recorded Handover start time and end time

Handover is defined as the connection established with the first Access Point could be maintained with the next Access Point.

Handover Latency is defined as the time required by the peer node (wired HIP node) to update the new IP address of the mobile node (when the mobile node changes its network attachment point, it get new IP address).

Simulation time is recoded before handover starts and after handover finishes. The difference between these two values is the Handover Latency. The values are shown in the table 1 The average Hanover Latency is 26.665ms.

Table 1 Handover Delay

Handover Start (in Sec) (HO_start)	HandoverFinish (in Sec) (HO_finish)	Handover Delay (in Sec)
175.14381	175.15276	0.00895
283.71727	283.74838	0.04111
614.98908	614.99776	0.00868
724.36202	724.41405	0.05203
1053.31543	1053.32332	0.00789
1164.07794	1164.12905	0.05111
1494.32620	1494.33190	0.00570
1603.21003	1603.25114	0.04111
1933.76817	1933.77703	0.00886
2043.45041	2043.49152	0.04111

8. HANDOVER IN TCP/IP NETWORKS

In this section, we will discuss how handover will happen in TCP/IP networks when mobile node moves from one place to another. Handover delay is estimated in this network and then we will compare it to TCP/IP with HIP Handover.

In TCP/IP network, if a host changes the IP address, because of moving to another access point, an application must break the upper layer bindings and restart the communication with the new IP address. This is very often the case with mobile hosts. When a host moves in the Internet topology from access point to another access point it is said to be mobile.

Figure 13 shows the network topology for TCP/IP Handover. Here we consider two access points which are broadcasting the messages which contain the IP address of its access point. When a mobile IP host enters into the network. It will be associated with the nearest access point. Communication to the destination nodes will be taken place by using this access point. This communication will be lost when the node moves to another location at that time it has to disconnect its communication and then reconnect it with the new access point's IP address.

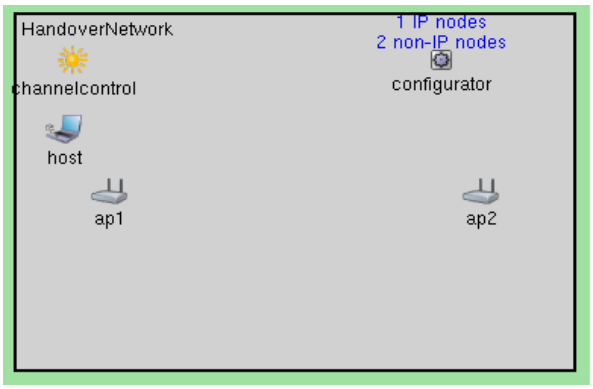

Figure 13 Network topology for TCP/IP Handover.

We simulated the network as shown in fig 13. In this network host is using the TCP/IP that can be clearly shown in the internal structure of the mobile host Figure 14.

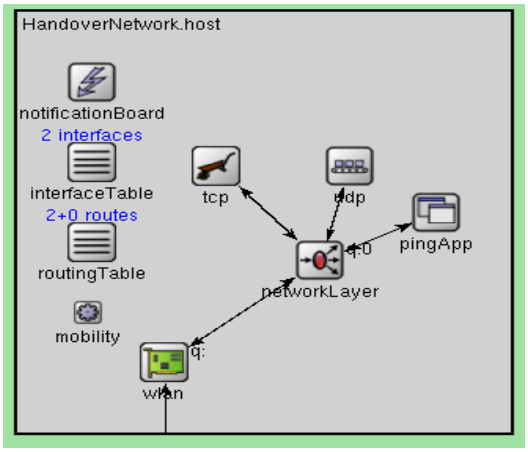

Figure 14 Internal structure of the mobile TCP/IP host.

Figure 15 shows the window where the recorded results of the simulation, "Handover in TCP/IP networks" are present. From the Figure 15 it is clear that the Handover delay of the IP network is 1.40572622176s.

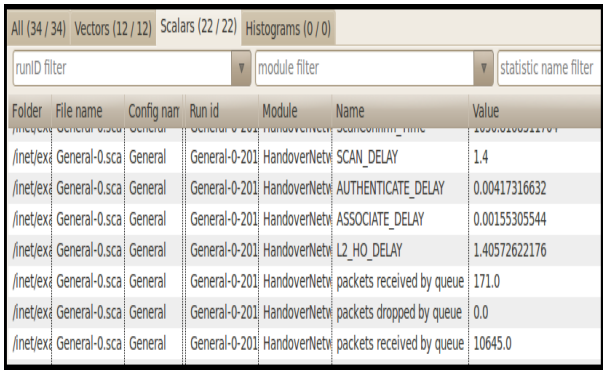

Figure 15 Recorded Handover Delay in TCP/IP network

9. COMPARISION OF HANDOVER BETWEEN HIP NETWORK AND TCP/IP NETWORK

Based on the simulation result of the HIP network(from section 7), the Handover Delay is 26.665ms i.e., 0.026665s. From section 8, it is clear that the handover delay of the TCP/IP network is 1.4057s.

With the HIP, the handover delay is reduced from 1.4057s to 0.026665s i.e., by 98 percent. This is because of mapping between locator and new IP addresses in TCP/IP with HIP system where the communication is not lost but it is redirected to new IP address when mobile node changes its location. But this is not the case in TCP/IP where the communication has to be disconnected and then reconnect the same previous communication with the new IP address when mobile node changes it location.

10. CONCLISUON AND FUTURE WORK

Simulation of the Registration protocol and the Base Exchange protocol in Host Identity Protocol Layer has been carried out by using RVS and also estimated the Handover Latency value. Handover Latency of the TCP/IP is estimated in section 8. Comparison is made between the estimation of Handover Latency in TCP/IP with HIP and the estimation of Handover Latency in TCP/IP. After the comparison we conclude that Handover Latency in TCP/IP with HIP is much less than the Handover Latency in TCP/IP.

As part of the future work, it is planned to make comparisons to the other mobility supported protocols (like mobile IP) with mobile HIP.

11. REFERENCES

[1] Fayez Al-Shraideh,"Host Identity Protocol", Networking Lab, Helsiniki University of Technology. Proceedings of the International Conference on Networking, International conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL '06), IEEE 2006.

[2] Pekka Nikander, "Applying Host Identity Protocol to the Internet Addressing Architecture", Proceedings of the 2004 International Symposium on Applications and the Internet (SAINT'04), IEEE 2004.

[3] Wesley M. Eddy, NASA GRC/Version FNS, "At What Layer Does Mobility Belong?" IEEE Communications Magazine, October 2004.

[4] Petri Jokela, Pekka Nikander, Jan Melen, Jukka Ylitalo and Jorma Wall, "Host Identity Protocol-Extended Abstract", Ericsson Research, Nomadic Lab, Finland.

[5] J. Laganier, T. Koponen, L. Eggert: "Host Identity Protocol (HIP)] Registration Extension", IETF RFC 5203 (http://www.ietf.org/rfc/rfc5203.txt), April 2008.

[6] Tuomas Aura, Aarthi Nagarajan and Andrei Gurtov, "Analysis of the HIP Base Exchange Protocol", Microsoft Research, United Kingdom.

[7] J. Laganier, L. Eggert: "Host Identity Protocol (HIP) Rendezvous Extension", IETF RFC 5204 (http://www.ietf.org/rfc/rfc5204.txt), April 2008.

[8] R. Moskowitz, P. Nikandar, P. Jokela and T. Henderson, "Host Identity Protocol", IEucture for HIPTF RFC 5201 (http://www.ietf.org/rfc/rfc5201.txt), April 2008.

[9] Infrastructure for HIP, http://infrahip.hiit.fi/ .

[10] OMNeT++, a public-source, component-based, modular and open architecture discrete event simulation environment. Community site: http://omnetpp.org/ .