CHAPTER 1

INTRODUCTION

1.1 BRIEF INTRODUCTION TO IPv6

The Internet has evolved to be one of mankind’s largest engineering structures. The underlying protocols which constitute the Internet network have had to scale to the dimensions of the present network, and the diversity of applications and physical layers. The fact that the Internet actually works, despite the rapid growth and change, is a tremendous tribute to the Internet Protocol (IP). The increasing demands of applications are creating motivation to re-examine the fundamental mechanisms of the Internet. The Internet will continue to grow, both in size, capacity and demands of applications.

Internet Protocol Version 6 (IPv6) is a new version of the internetworking protocol designed to address the scalability and services shortcomings of the current standard, IPv4 (Marc et al (1998), Afifi and Toutain 1999). It is an Internet layer protocol for packet-switched internet works. It is designated as the successor of IPv4, the current version of the Internet Protocol, for general use in the Internet.

Unfortunately, IPv4 and IPv6 are not directly compatible; hence programs and systems designed to one standard cannot communicate with those designed to the other. However IPv4 systems are ubiquitous and are not about to go away “overnight” as the IPv6 systems roll in. Consequently, it is
necessary to develop smooth transition mechanisms that enable applications to continue working while the network is being upgraded.

The main change brought by IPv6 is a much larger address space that allows greater flexibility in assigning addresses. The extended address length (Nakajima and Kobayashi 2004) eliminates the need to use network address translation to avoid address exhaustion, and also simplifies the aspects of address assignment and renumbering when changing providers.

It is common to see examples that attempt to show that the IPv6 address space is extremely large. For example, IPv6 supports $2^{128}$ (about $3.4 \times 10^{38}$) addresses (Microsoft Corporation, 2006). IPv6 address space must be managed for the good of the internet community. The large number of addresses allows a hierarchical allocation of addresses that may make routing and renumbering simpler. With IPv4, complex CIDR techniques were developed to make the best possible use of a restricted address space. Renumbering, when changing providers, can be a major effort with IPv4, as discussed in (Ferguson and Berkowitz 1997).

1.1.1 Features and differences from IPv4

To a great extent, IPv6 is a conservative extension of IPv4. Most transport and application layer protocols need little or no change to work over IPv6; exceptions are applications protocols that embed network-layer addresses (such as FTP or NTPv3). Applications, however, usually need small changes and a recompile in order to run over IPv6.
1.1.2 Larger address space

The main feature of IPv6 that is driving adoption today is the larger address space: addresses in IPv6 are 128 bits long versus 32 bits in IPv4, (Deering and Hinden 1998) The larger address space makes administration of medium and large networks simpler, by avoiding the need for complex subnetting schemes. Subnetting will, ideally, revert to its purpose of logical segmentation of an IP network for optimal routing and access, (Atkinson 1995).

1.1.3 Stateless Address Auto Configuration (SLAAC)

IPv6 hosts can be configured automatically when connected to a routed IPv6 network using ICMPv6 router discovery messages, (Thomson et al 1996). When first connected to a network, a host sends a link-local multicast router solicitation request for its configuration parameters; if configured suitably, routers respond to such a request with a router advertisement packet that contains network-layer configuration parameters.

If IPv6 auto configuration is not suitable, a host can use stateful configuration (DHCPv6) or be configured manually. Stateless auto configuration is only suitable for hosts routers must be configured manually or by other means.

1.1.4 Multicast

Multicast is part of the base specifications in IPv6, unlike IPv4, where it was introduced later. IPv6 does not have a link-local broadcast facility; the same effect can be achieved by multicasting to the all-hosts group (FF02::1).
Most environments, however, do not currently have their network infrastructures configured to route multicast; multicast on single subnet will work, but global multicast might not.

1.1.5 Link-local addresses

IPv6 interfaces have link-local addresses in addition to the global addresses that applications usually use. These link-local addresses are always present and never change, which simplifies the design of configuration and routing protocols.

1.1.6 Jumbo grams

In IPv4, packets are limited to 64 KB of payload. When used between capable communication partners and on communication links with a maximum transmission unit (MTU) larger than 65,576 octets (65536 + 40 for the header), IPv6 has optional support for packets over this limit, referred to as jumbo grams which can be as large as 4 GB (Borman et al 1999). The use of jumbo grams may improve performance over high-MTU networks.

1.1.7 Network-layer security

IPSec, the protocol for IP network-layer encryption and authentication (Kent and Atkinson 1998), is an integral part of the base protocol suite in IPv6 unlike IPv4, where it is optional (but usually implemented). IPSec, however, is not widely used at present except for securing traffic between IPv6 Border Gateway Protocol routers.

1.1.8 Simpler processing by routers

IPv4 has a checksum field that covers the entire packet header. Since certain fields such as the TTL field change during forwarding, the
checksum must be recomputed by every router (Ferguson and Berkowitz 1997). IPv6 has no error checking at the network layer but instead relies on link layer and transport protocols to perform error checking, which should make forwarding faster.

So far the salient features of IPv6 are explained. In the following divisions an introduction to the research work is explained.

Section 1.2 describes transition mechanisms where three methods are explained.

Section 1.3 briefs about VLSI implementation of routing table.

Section 1.4 gives an introduction about the mobility implementing protocols.

1.2 TRANSITION MECHANISMS

This work presents a comprehensive explanation of the various transition mechanisms and brings about the prospects of a novel Address Translation Technique and a Tunneling Mechanism.

Actually two different sets of problems arise when IPV6 is introduced. The first one is related to having IPv6 communication among two or more IPv6 islands which are isolated in IPv4 world. The second set is related to the establishment of communication between the existing IPv4 world and the new IPv6 world.

Solutions for the first set of problems are generally based on dual stack routers and IPv6 in IPv4 tunnels. Mechanisms to solve the second set of problems rely on network address translation technologies and IPv4 in IPv6 tunneling. During the time of migration from IPv4 to IPv6 networks, a
number of transition mechanisms have been proposed by IETF to ensure smooth, stepwise and independent changeover, (Gilligan and Nordmark 1996).

Mainly, there are three existing mechanisms (Forouzan 2003) Dual-Stack, Tunneling and Translation. (Oliver and Oliver 2006) (Chen et al 2004) as shown in Figure 1.1.

![Figure 1.1 Transition Mechanism](image)

Some of the transition mechanisms work by encapsulating IPv6 packets in IPv4 packets then transporting them via an IPv4 network infrastructure and others work by operating dual IPv6/IPv4 stacks on the hosts or edge routers to allow the two versions of the Internet Protocols (IPv6 and IPv4) to work together.

1.2.1 Dual Stacks

The most promising transition strategy has been to consider all IPv4 machines to have an IPv6 address and be able to run on both networks. Referring to "protocol stacks" for IPv4 and IPv6, the term "dual-stack" (Hoog 2007) is used to describe this. The premise of dual-stack is that if a device is configured to run on IPv4 and then IPv6 is added in parallel, the device has a choice. When every IPv4 machine is dual stacked with IPv6, IPv4 can be
removed and the entire IPv4 protocol stack can be removed, resulting in a completed transition. Dual-stack approach is illustrated in Figure 1.2.

![Dual Stack Mechanism](image)

**Figure 1.2 Dual-stack Mechanism**

In general, both IP versions (IPv4 and IPv6) in the dual IPv6/IPv4 stack node are enabled; but if one stack in this node is disabled for operational reasons, this node may operate in one of the modes, (Deering and Hinden 1998)

- It operates like IPv4-only node, if the IPv4 stack is enabled and IPv6 stack is disabled.
- It operates like IPv6-only node, if the IPv4 stack is disabled and IPv6 stack is enabled.

### 1.2.2 Tunneling mechanism

Tunneling mechanism will be used when two hosts that are located in two different IPv6-only zones want to communicate with each other by passing their packets through IPv4-only zone, (Charles Perkins 1996). In this case the IPv6 packets will be encapsulated in the IPv4 packets to be passed
through the IPv4-only zone (Figure 1.3). This mechanism enables the island IPv6 end systems and routers to communicate through an existing IPv4 infrastructure, (Conta et al 1998).

There are two ways to accomplish tunneling. One is automatic tunneling which uses IPv4-compatible IPv6 addresses to add a route to a special IPv6 prefix which points to a tunnel destination. Any packets destined for a v4-compatible address will be sent through the tunnel.

In “configured tunneling,” the address of the tunnel exit point is configured on the tunnel entry point and similar encapsulation is used. A combination of automatic and configured tunneling can also be used to route IPv6 packets across a v4 network. Teredo, ISATAP, 6 to 4 and IPv6 over IPv4 are other tunneling mechanisms.

Teredo encapsulates IPv6 packets over User Datagram Protocol (UDP), which allows them to pass through NAT nodes. ISATAP can be used by v6 hosts on a v4 network without any IPv6 routers using a specially constructed ISATAP address. 6 to 4 also uses a special prefix for tunneling.

The IPv6 header contains the address of the final destination and the IPv4 header contains the address of the tunnel endpoint. The commonly used tunneling mechanisms include IPv4/IPv6 configured tunnel (Gilligan and Nordmark 2000), 6 to 4, (Carpenter and Moore 2001) ISATAP (Templin et al 2004), Silkroad (Liu et al 2004) / Teredo (Huitema 2005), Tunnel Broker/TSP (Blanchet and Parent 2005), DSTM (Bound et al 2002), etc.
Figure 1.3 Tunneling mechanism

i. IPv6 over IPv4 Tunnel is applied when IPv6 hosts/islands inside native IPv4 network need to communicate with native IPv6 network, but there is no direct IPv6 link between them. Typical mechanisms include IPv6 configured tunnel, 6 to 4 and ISATAP.

ii. IPv4 over IPv6 tunnel can be used for the IPv4 hosts Netzwerks attached with IPv6 network to get IPv4 connectivity or transmit IPv4 traffic via IPv6 network. Available mechanisms include IPv4 configured tunnel and DSTM.

iii. Tunnel traversing NAT
It is difficult to provide IPv6 connectivity for the users behind the IPv4 NAT with common IPv6 over IPv4 tunneling mechanisms. It is proposed to use IPv6-in-UDPin-IPv4 technology to traverse the NATs, in which IPv6 packet is encapsulate in IPv4 UDP packet, (Elliston.B., Pouffary.Y. and Young .A., 1997). The commonly used mechanisms include Teredo, Silkroad and TSP.
1.2.2.1 Tunnel broker

The Tunnel servers, called Tunnel Brokers automatically manage tunnel requests coming from the users (Samad et al 2002). This approach is expected to be useful to stimulate the growth of IPv6 interconnected hosts and to allow early IPv6 network providers to provide easy access to their IPv6 networks. The main difference between the Tunnel Broker and the 6 to 4 mechanisms is that they serve a different segment of the IPv6 community.

The Tunnel Broker fits well for small isolated IPv6 sites, and especially isolated IPv6 hosts on the IPv4 Internet, that want to easily connect to an existing IPv6 network. The 6 to 4 approach has been designed to allow isolated IPv6 sites to easily connect together without having to wait for their IPv4 ISPs to deliver native IPv6 services (Bradner S 1996). This is very well suited for extranet and virtual private networks. In addition, the Tunnel Broker approach allows IPv6 ISPs to easily perform access control on the users enforcing their own policies on network resources utilization.

1.2.3 Translation Mechanism

This mechanism is necessary and will be used if the source host (sender) wants to use IPv6, but the destination host does not understand IPv6. So, the translation method will be needed in order to convert the IPv6 header into IPv4 header to be understood by the destination host (Ra’ed AIJa’afreh et al 2007). In this mechanism, the translator will be located between the two different network areas in order to perform a translation for the following two cases:

1. When the IPv4 packet that is coming from IPv4 zone should be translated into IPv6 packet that is going to IPv6 zone.

2. When the IPv6 packet that is coming from IPv6 zone should be translated into IPv4 packet that is going to IPv4 zone.
Among the variety of translation mechanisms that are proposed the dominant ones include NAT-PT (Tsirtsis and Srisuresh 2000) BIS (Tsuchiya et al 2000), TRT (Hagino and Yamamoto 2001), Socks 64 (Kitamura 2001), Stateless IP/ICMP Translation (SIIT) (Nordmark 2000) and BIA (Lee et al 2002), all of which can be grouped under three categories: Network Layer Translation, Transport Layer Translation, and Application Layer Translation.

1.2.3.1 Network Address Translation

Network address translation or transition strategies from IPv4 to IPv6 include assignment of IPv4 addresses to IPv6 Hosts, IPv6/IPv4 network address and protocol translator, ISATAP, 6 to 4, Teredo, (an IPv6 transition technology that allows IPv6 connectivity across the IPv4 Internet between hosts that are located behind network address translators (NATs) and others. Such address translation schemes gain importance from the fact that developers porting applications from IPv4 to IPv6 rely on these valuable tools. For instance, a server application ported to IPv6 can be tested without having to port the client as well (Zsako J 1999).

1.2.3.2 Network Address Translation and Protocol Translation

Network Address Translation and Protocol Translation (NAT-PT) is a transition mechanism to provide transparent routing between IPv4 and IPv6 end nodes. NAT-PT uses a combination of network address and protocol translation. (Tsirtsis and Srisuresh 2000) The difference between NAT-PT and NAT in IPv4 (Srisuresh and Egevang, 1994) is that translation does not happen between private and global addresses, but between IPv4 and IPv6 addresses. The aim of NAT-PT is to provide translation between IPv6-only and IPv4only nodes (Tsirtsis et al 2000).
1.2.3.3 Address Translation method

Address translation is trivial when using IPv4-mapped and IPv4-compatible IPv6 addresses. For the IPv6-to-IPv4 direction, the translator simply extracts the lower 32-bits of an IPv6 address to obtain an IPv4 address. For the opposite direction the translator sets the lower 32-bits of the IPv6 source/destination addresses to the IPv4 source/destination addresses, and sets the upper 96-bits of the IPv4 source and destination addresses to the IPv4-mapped and IPv4-compatible prefix, respectively. However, it is considered to be an inefficient way to use IPv4-mapped address as it has the drawback of requiring IPv6 routers to contain routes to IPv4-mapped addresses. The alternative is to use IPv6-only addresses to refer to IPv4 nodes, which requires a translator (Figure 1.4) to maintain an explicit mapping between IPv4 and IPv6 addresses. This translator is designed to support all of the scenarios just described.

Figure 1.4 Address Mapping
1.3 VLSI IMPLEMENTATION OF ROUTING TABLE

Having given an introduction about IPv6, a need to design a router table in VLSI arises.

Nowadays, an access to on-chip memory takes 1-5ns for SRAM and about 10ns for DRAM. One access to off-chip memory takes 10-20ns for SRAM and 60-100ns for DRAM. This figure shows that the development of high-speed IP lookup algorithms which can be implemented on chip is of great demand. The trend is to use a chip to exclusively perform the IP lookup task.

The entire look-up process is done in hardware using Field Programmable Gate Arrays (FPGA) technology, which allows hardware programming. The important advantage of FPGA is the possibility of a field reconfiguration of the accelerator by reprogramming the FPGA, i.e. without changing the hardware. This way, new function can be added or existing one modified without any hardware upgrade. The entire solution will be based on the Xilinx target architecture (Virtex II chip) and VHDL will be used as a language for simulation and design synthesis. The two main memories used in this proposal are CAM (Content addressable memory) and SRAM(Static Random Access Memory). Both of them store the look-up program for the router. It does 50 million searches per second. The main part of the look-up processor is CAM.

1.3.1 Design Approach

In the Internet Protocol version 6 (IPv6) the destination address of an Internet Protocol (IP) packets is 128 bits long. There are two addressing schemes in use, namely the Classful addressing scheme and a Classless Inter
Domain Routing scheme (CIDR). Three different network sizes are defined in the classful address scheme namely class A, B, and class C. In CIDR the 128 bit IP address is broken down into the network address part and the host address part. IP routers forward packets based only on the network address until the packets reach the destined network. Typically an entry in the forwarding table stores the address prefix and the routing information. The address lookup operation amounts to finding an exact prefix match in the forwarding table. In order to allow more efficient use of the IP address space and to avoid the problem of forwarding table explosion, arbitrary length prefix are allowed in the CIDR scheme. With CIDR, routers must find out the best matching prefix for IP packet forwarding.

This work presents a mechanism to perform fast longest-matching prefix route lookups in hardware for an IP router. Since the advent of CIDR in 1993, IP routes have been identified by a <route prefix, prefix length> pair, where the prefix length is between 0 and 128 bits. The routing table in a router is used to lookup an IP address in an array of entries, each consisting of a network address that is the prefix of a group of IP address and the corresponding next hop number to the network (Figure 1.5).

For every incoming packet, a search must be performed in the router’s forwarding table to determine which next hop the packet is destined for (Gupta et al 1998). With CIDR, the search may be decomposed into two steps. First, the set of routes with prefixes that match the beginning of the incoming IP destination address are found. Then, among this set of routes, the longest prefix are selected. This is the route to use for identifying the next hop.
This work is motivated by the need for faster route lookups; in particular, a fast, hardware implementable lookup algorithms. In this research a lookup mechanism has been developed that achieves the following goals.

The lookup procedure should be easily implementable in hardware using simple logic. Ideally, the route lookup procedure should take exactly one memory access time.

If it takes more than one memory access, then (a) the number of accesses should be small, (b) the number of accesses should be bounded by a small value in all cases, and (c) the memory accesses should occur in different physical memories, enabling pipelined implementations.

In general, for implementation the overhead to update the forwarding table should be small. The technique presented here is based on the assumption that the memory is cheap. The route lookup mechanism will be used in routers where speed is a premium; for example those routers that need to process at least 10 million packets per second. On backbone routers there are very few routes with prefixes longer than 48 bits. This is verified by an examination of the MAE-EAST backbone routing tables (Merit Networks).
1.3.2 Synthesis of Routing Table

Xilinx FPGA is a very effective platform to realize routing table in hardware. Content Addressable Memories (CAMs) that do exact matching can be used to implement best matching prefix.

The complexity of Xilinx FPGAs and their ability to be reconfigured in the field turns out to be valuable assets that facilitate the implementation of flexible network applications.

This algorithm has been run in practical routing tables. This routing table designed in VLSI is synthesized by Xilinx and downloaded into Spartan 2E. There are many devices (Altera, Actel, cypress) supporting the IP lookups downloading. However the Spartan 2E systems are easily available and all kinds of software supports this Xilinx. So in this research the Xilinx series Spartan has been preferred for verification.

1.4 MOBILITY IMPLEMENTING PROTOCOLS

This research introduces and compares three mobility implementing protocols in order to determine the best suitability of a protocol for mobility. The chosen protocols are Mobile IPv4, Host Identity Payload (HIP) and Mobile IPv6. The Mobile IP (MIP) allows the user to keep their own IP addresses even though they move from one network to another.

1.4.1 Mobile IPv4

Mobile IPv4 uses triangular routing (i.e., tunnel) as shown in Figure 1.6 In triangular routing, data packets sent from the Correspondent Node CN (a fixed terminal) to the Mobile Node (MN) is sent to the MN’s HA (Home Agent) first using standard IP routing. The HA encapsulates the data packets and tunnels the data packets to the MN’s CoA (Care of Address) (Jie Li Hsiao-Hwa Chen 2005). At the associated FA (Foreign Agent), the
data packets are de tunneled and sent to the MN. Although triangular routing is simple and easy to use, it is inefficient since it takes a route from a CN to an HA and then to an MN (Nikander 2001).

Mobile IP supports both wired and wireless networks. To eliminate Triangular Routing, Route Optimization technique is used (Charles Perkins 1998). Figure 1.8 shows the route optimization technique.

During Route Optimization several threats occur. These threats are man-in-the-middle, Denial of service, Attack against secrecy and integrity and Flooding. This work eliminates Triangular Routing for improving security of IPv6 and HIP and to prevent data from threats by applying Return Routability technique.

![Figure 1.6 Triangular Routing in Mobile IP](image)

### 1.4.2 Mobile IPv6 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 (IAB 1995). For discovering a HA, MN uses Dynamic Home Agent Address Discovery protocol (Nikander 2001). 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 Auto configuration.

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. (Johnson et al 2004) 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 tunneled through the Home Agent. Figure 1.7 illustrates this situation.

In Figure 1.7, 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 tunneling. IPv6 encapsulation is used in the tunnelling. The salient feature of this mode is that CN does not require to support Mobile IPv6 at all. Even so, bidirectional tunneling is not always efficient, especially if the MN is close to CN, and therefore communicating through the Home Agent creates unnecessarily a long path.

![Figure 1.7 Tunneling traffic between the MN and CN](image-url)
MIPv6 offers a solution for this sort of situation through route optimization. Only the first packet is tunneled through HA. Then MN can register its current location by sending its current binding information to a CN as a Binding Update 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 1.8.

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. 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.

![Figure 1.8 Route Optimization for traffic between MN and CN](image)

### 1.4.3 Security in Mobile IPv6

Binding updates are one of the key factors in the functioning of MIPv6. Hence the binding messages must be authenticated and protected against replay attacks to prevent malicious nodes from corrupting the binding caches with invalid addresses (Johnson et al 2004).
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 (Kent S. and R. Atkinson 1998). 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 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 that uses cryptographic tokens in verification is used. 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 RSA algorithm (Comer, Douglas E).

1.4.4 Host Identity Payload 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 (HI) 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 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 (HI Tag) from an address directory (e.g. Domain Name Space). HIP work with current IPv4 and future IPv6 networks but MIPv6 relies only on IPv6 (Henderson et al 2003). Furthermore, MIPv6 requires changes to routers whereas the other solutions do not.
1.4.5 Security in HIP

The HIP security is quite good. 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. Since the HIP identifiers are public keys, they can be used to authenticate the HIP packets as well as to protect them from most Man-in-the-Middle attacks (Moskowitz 2001).

In using public keys as identifiers, it means that no explicit Public Key Infrastructure (PKI) is needed. The impact of DoS attacks is also decreased, as the responder is the one giving the challenge and deciding its difficulty. 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. There are also a number of MitM (Man in the Middle) 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.

The key issues considered for comparison were security, signaling and other functional overhead, the effects on both applications and overall architecture.

1.5 OBJECTIVE OF THE RESEARCH WORK

This work presents a comprehensive explanation of the various transition mechanisms and brings about the prospects of an Address Translation Technique and a Tunnel Broker Mechanism. A solution for transition mechanism which includes conversion of IPv4 to IPv6 and vice versa and a mechanism for an efficient router design is proposed which uses small amount of memory.
1.6 ORGANIZATION OF THE THESIS

This thesis work is organized in the following manner.

Chapter 1 gives a broad introduction about IPv6. It also gives an introduction to the research work i.e transition mechanisms, VLSI implementation of routing table and comparison of the mobility protocols.

Chapter 2 deals with the literature survey. It explains the past history, present trend and the future prospects.

Chapter 3 elaborates the transition mechanisms. Address translation techniques are explained in detail. Address notation is given with example. address mapping between 4 to 6 and vice versa is explained under Address translation technique.

Chapter 4 explains various tunneling mechanisms. The implementation and analysis of Tunnel Broker are carried out.

Chapter 5 explains the VLSI implementation of the IPv6 routing table. Design of a routing lookup table to perform fast longest matching prefix is explained. The synthesis using Xilinx software is explained.

Chapter 6 briefs about the comparative study of the three mobility implementing protocols. The three protocols MIPv4, MIPv6 and HIP are studied in detail and various features are compared.

Chapter 7 concludes with the research findings and future enhancements.