Name: Ioan Raicu

Course: CS590C

Project Supervisor: Dr. Douglas Comer

Date: 1/3/2003

Project Title: A Performance Evaluation of IPv4 vs. IPv6 Forwarding

ABSTRACT

Although the Internet Protocol known as IPv4 served its purpose for over 20 years, its days are numbered. With IPv6's maturity increasing, there is a need to evaluate the performance benefits or drawbacks for the end user that the new IPv6 protocol will have in comparison to the well established IPv4 protocol. Theoretically, the overhead between the two different protocols should be directly proportional to the difference in the packet's header size, and therefore the performance of IPv6 should be similar to IPv4. However, according to our findings, the empirical performance difference at the application layer between IPv4 and IPv6 when utilized in a realistic setting is much larger and inconsistent than anticipated. Network processors, through their ease of programmability, offer an ideal testbed to investigate IPv6 behind the scenes and isolate the performance bottlenecks that caused IPv6 to perform worse at the application layer. I will be using Agere's 2.5G PayloadPlus chip set simulated by System Performance Analyzer (SPA). I plan on implementing IP forwarding for both the IPv4 and IPv6 protocols; once that is completed, I plan to use the new testbed to evaluate IPv6 again. I hope to gain insight to some of my previous results obtained from the application layer.

1.0 INTRODUCTION, BACKGROUND INFORMATION, AND RELATED WORK

The background information is covered in [1, 3, 4, 5, 6, 12] while the related work is covered in [8, 9, 10, 11].

My previous work related to this topic is [2, 7]. Essentially, I evaluated the performance of IPv4 vs. IPv6 at the application layer. According to our findings, the empirical performance difference between IPv4 and IPv6 when utilized in a realistic setting is much larger and inconsistent than anticipated. The metrics utilized were throughput and latency, the most common metrics when testing network performance. Our testbed consisted of two dual stack (IPv4/IPv6) routers and two end nodes running both Windows 2000 and Solaris 8.0; each OS had their particular implementation of a dual IPv4/IPv6 stack. Our goal was to perform an unbiased application layer empirical performance evaluation between the two protocol stacks (IPv4 and IPv6), and at the same time, compare two different implementations (Windows 2000 and Solaris 8) on identical hardware and under identical settings.

1.1 IPv4 and IPv6 Architecture

Internet Protocol was first developed in the early 1980s. Its intent was to interconnect few nodes and was never expected to grow to the size of the Internet has become today. IPv4 was initially designed for best-effort service and only scaled to today's Internet size because of its state-less design. One of the few things that the creators of the Internet Protocol never envisioned was the exhaustion of a 32 bit address space. In the early 1990s, it became pretty evident that if the Internet will continue to grow at the exponential rate of doubling every eighteen months, the IPv4 address space would be depleted by the turn of the millennium. Some temporary solutions were offered, such as NAT (Network Address Translator) or CIDR (Classless InterDomain Routing) [**14**], however work began on a new Internet Protocol, which was first called IPnG from Internet Protocol Next Generation, but later became known as IPv6, Internet Protocol version 6 (IPv6). IPv6 is the main focus of our work and hence this thesis.

The most evident reason for a new version of an IP was to increase the address space; IPv6 was designed with a 128 bit address schema, enough to label every molecule on the surface of the earth with a unique address ($7x10^{23}$ unique IP addresses per square meter) [**14**]. Even in the most pessimistic scenario of inefficient allocation of addresses, there would still be well over 1000 unique IP addresses per square meter of the earth. There were other reasons that were a bit more subtle, such as better support for inelastic traffic and real time applications, and without doubt will most likely drive the deployment of IPv6 just as hard as the address space depletion problem. Twenty years ago, the only kind of traffic that existed on the internet was elastic traffic, such as emails or file transfers. Elastic traffic enjoys having high bandwidth and low latency, however if the network can only deliver a small percentage of its capacity, than the transmission will still deliver the data just as good, but just at a later time. On the other hand, inelastic traffic has much more stringent restrictions in which bad network performance can render the data useless. In the past five years, multimedia applications have

emerged and have mostly dominated the Internet's growth and demand for more bandwidth and processing power. IPv6 was designed for both elastic and inelastic traffic in its vision scope. That does not mean that IPv6 is not a best effort service anymore, but merely that it has the potential to interoperate much easier with Quality of Service (QoS) architectures such as RSVP, Integrated Services (Intserv), and Differentiated Services (Diffserv) in order to make end-to-end QoS over IP-based networks a reality. These features of IPv6 are outside the context of this paper, so please refer to Chapter 5 in regards to future work. [**14**]

Some of the differences between IPv4 and IPv6 features are outlined in the next few statements. Keep in mind that most of the improvements on IPv6 were done with three things in mind: scalability, security, and support for multimedia transmissions. First of all, the address space is increased from 32 bits to 128 bits. Obviously, this increase in address space means more capacity for nodes, but it also enlarges the header overhead and the routing tables' size. Unlike IPv4, IPSec support has become a requirement in the IPv6 header. This was a much needed improvement to at least offer basic security features. Payload identification for QoS handling by routers is now supported by the flow label field. This was introduced primarily because of the earlier statements about multimedia applications that require more stringent guarantees of data delivery. Fragmentation support has been moved from both routers and sending hosts to just sending hosts. This is an important fact due to the amount of work that the routers have been alleviated by, and therefore it improves scalability. The IPv6 header does not include a checksum and has no options included in the header, but rather introduces extension headers. This allows faster processing at the routers by performing the checksum less often and analyzing only the header information needed. Finally, IPv6 requires no manual configuration or DHCP, which will become more and more important as the number of nodes increases. Overall, IPv6 was carefully thought out and was designed with future applications in mind. [**14**]

Theoretically, taking a close look at the brake-down of the various headers in both IPv4 and IPv6, it is evident that the overhead incurred is minimal between IPv4 and IPv6. As a quick overview of Table 1 found below, the primary difference between IPv4 and IPv6 is that IPv4 has a 20 byte header while IPv6 has a 40 byte header. Although the address space in IPv6 is four times the size of its counterpart, IPv6 has decreased the number of required fields and made them optional as extension headers. Let's take the IPv4 UDP packet as an example to better understand Table 1. The total Ethernet MTU (Maximum Transfer Unit) is 1514 bytes, from which 14 bytes are the Ethernet header, 20 bytes are the IP header, and 8 bytes are the UDP header. The payload for a UDP packet in IPv4 is 1472 bytes, and is computed by Equation 1:

MTU = Payload + TLH + NLH + DLLH

Equation 1: MTU calculation; the formula used in deriving Table 1; payload is the application layer data size; TLH is the transport layer (TCP/UDP) header size; NLH is the network layer (IP) header size; DLLH is the data link (Ethernet) layer header size; MTU is the total Ethernet MTU size that is transmitted on the physical medium.

	IPv4 TCP	IPv6 TCP	IPv4 UDP	IPv6 UDP
TCP/UDP Payload	1460	1440	1472	1452
TCP/UDP Header	20	20	8	8
IP Payload	1480	1460	1480	1460
IP Header	20	40	20	40
Ethernet Header	14	14	14	14
Total Ethernet MTU	1514	1514	1514	1514
Overhead %	3.7%	5.14%	2.85%	4.27%

Table 1: Packet breakdown and overhead incurred by header information; please refer to Equation 1 for obtaining the information above

The difference between IPv4 and IPv6 would most obviously be the IP header, which instead of being 20 bytes, would now be 40 bytes. The overhead that is incurred by having header information can be figured out by taking the total Ethernet MTU and dividing by the TCP or UDP payload. For example, the difference between IPv4 UDP and IPv6 UDP is a mere 1.42 %, while for TCP it is almost the same at 1.44 %.

In theory, the performance overhead between these two protocols is so minimal that the benefits of IPv6 should quickly overshadow the negatives. In Chapter 4 and 5, I will discuss the performance evaluation in reality between IPv4 and IPv6, which proved to be quite a bit larger than the theoretical difference.

In order to better visualize the layering principles, we captured a screen shot of Microsoft Network Monitor as it displays a packet and all its header information and placed in Figure 1 and Figure 2. Figure 1 displays a ping echo (ICMP) message and its header information. Notice that the IP version is 4 and the IP header length is 20 bytes. Notice also the source and destination addresses, as they are all part of the packet header information.

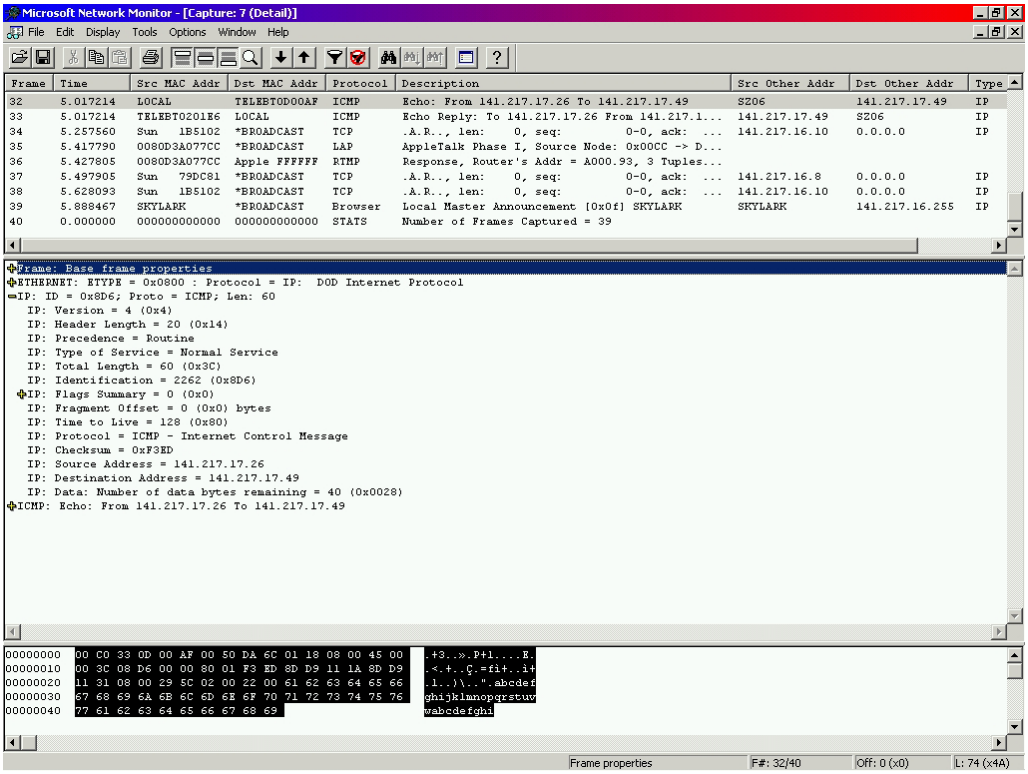

Figure 1: IPv4 Packet as depicted by the Microsoft Network Monitor

Figure 2 will show a similar screen shot, but this time presenting an IPv6 packet. Figure 2 displays a ping echo (ICMP) message and its header information for an IPv6 packet. Notice that the IP version is 6 and the IP header length is 40 bytes. Notice also the IPv6 128 bit source and destination addresses, as they are all part of the packet header information. Some new fields can also be seen, such as priority, flow label, and next header. We will discuss these in more detail later.

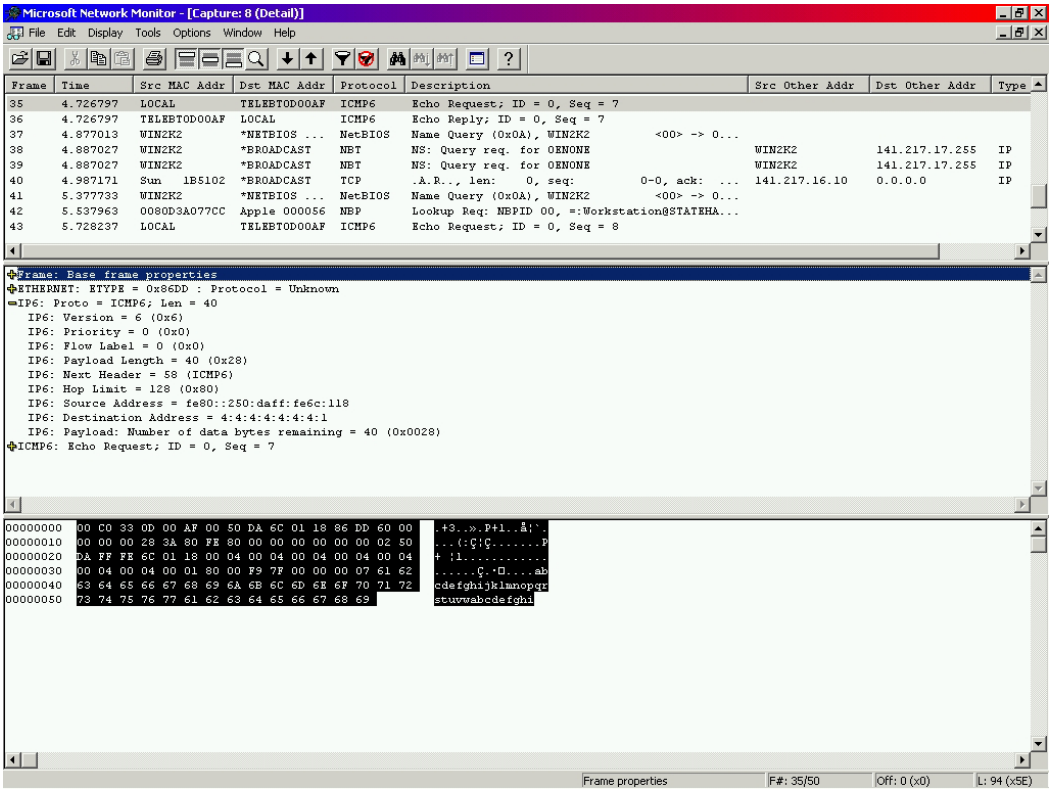

Figure 2: IPv6 Packet as depicted by the Microsoft Network Monitor

1.2 IPv4 Specification

Internet Protocol version 4 is the current version of IP, which was finally revised in 1981. It has a 32 bit address looking like 255.255.255.255, and it supports up to 4,294,967,296 ($4.3x10^9$) addresses. The IPv6 header is a streamlined version of the IPv4 header. It eliminates fields that are unneeded or rarely used and adds fields that provide better support for real-time traffic. An overview of the IPv4 header is helpful in understanding the IPv6 header.

<table>
<tr><td>Version
4 bits</td><td>Length
4 bits</td><td>Type of Service
8 bits</td><td colspan="2">Total Length
16 bits</td></tr>
<tr><td colspan="3">Identification
16 bits</td><td>Flags
3 bits</td><td>Fragment Offset
13 bits</td></tr>
<tr><td colspan="2">TTL
8 bits</td><td>Protocol
8 bits</td><td colspan="2">Checksum
16 bits</td></tr>
<tr><td colspan="5">Source Address
32 bits</td></tr>
<tr><td colspan="5">Destination Address
32 bits</td></tr>
</table>

Figure 3: IPv4 Header format

The "Version" field indicates the version of IP and is set to 4 in the case of IPv4; the size of this field is 4 bits.

The "Internet Header Length" field indicates the number of 4-byte blocks in the IP header. The size of this field is 4 bits. The minimum IP header size is 20 bytes, and therefore the smallest value of the Internet Header Length field is 5. IP options can extend the minimum IP header size in increments of 4 bytes. If an IP option does not use all 4 bytes of the IP option field, the remaining bytes are padded with 0's, making the entire IP header an integral number of 32-bits (4 bytes). With a maximum value of 0xF, the maximum size of the IP header including options is 60 bytes (15*4).

The "Type of Service" field indicates the desired service expected by this packet for delivery through routers across the IP internetwork. The size of this field is 8 bits, which contain bits for precedence, delay, throughput, and reliability characteristics. Unfortunately, this field was not widely utilized, and only recently with the coming of RSVP did it see much activity. For example, RSVP uses the type of service field in order to setup flow labels.

The "Total Length" field indicates the total length of the IP packet (IP header + IP payload) and does not include link layer framing. The size of this field is 16 bits, which can indicate an IP packet that is up to 65,535 bytes long.

The "Identification" field identifies the specific IP packet. The size of this field is 16 bits. The Identification field is selected by the originating source of the IP packet. If the IP packet is fragmented, all of the fragments retain the Identification field value so that the destination node can group the fragments for reassembly.

The "Flags" field identifies flags for the fragmentation process. The size of this field is 3 bits, however, only 2 bits are defined for current use. There are currently two flags: one indicates whether the IP packet might be fragmented, while the other indicates whether more fragments follow the current fragment.

The "Fragment Offset" field indicates the position of the fragment relative to the original IP payload; the size of this field is 13 bits.

The "Time-to-Live" (TTL) field indicates the maximum number of links on which an IP packet can travel before being discarded. The size of this field is 8 bits. The TTL field was originally used as a time count with which an IP router determined the length of time required (in seconds) to forward the IP packet, decrementing the TTL accordingly. Modern routers almost always forward an IP packet in less than a second and are required by RFC 791 to decrement the TTL by at least one. Therefore, the TTL becomes a maximum link count with the value set by the sending node. When the TTL equals 0, the packet is discarded and an ICMP Time Expired message is sent to the source IP address.

The "Protocol" field identifies the upper layer protocol; the size of this field is 8 bits. For example, TCP uses a protocol value of 6, UDP uses a protocol value of 17, and ICMP uses a protocol value of 1. The Protocol field is used to demultiplex an IP packet to the upper layer protocol.

The "Header Checksum" field provides a checksum on the IP header only. The size of this field is 16 bits. The IP payload is not included in the checksum calculation as the IP payload and usually contains its own checksum. Each IP node that receives IP packets verifies the IP Header Checksum and silently discards the IP packet if checksum verification fails. When a router forwards an IP packet, it must decrement the TTL. Therefore, the Header Checksum is recomputed at each hop between source and destination.

The "Source Address" field stores the IP address of the originating host; the size of this field is 32 bits.

The "Destination Address" field stores the IP address of the destination host; the size of this field is 32 bits.

The "Options" field stores one or more IP options. The size of this field is a multiple of 32 bits. If the IP options do not use all 32 bits, padding options must be added so that the IP header is an integral number of 4-byte blocks that is indicated by the Internet Header Length field.

1.3 IPv6 Specifications

Internet Protocol version 6 is designed as an evolutionary upgrade to the Internet Protocol (IPv4) and will, in fact, coexist with the older IPv4 for some time. IPv6 is designed to allow the Internet to grow steadily, both in terms of the number of hosts connected and the total amount of data traffic transmitted; it will have a 128 bit address looking like 1234:5678:90AB:CDEF:FFFF:FFFF:FFFF:FFFF, and it will support up to 340,282,366,920,938,463,463,374,607,431,768,211,456 (3.4×10^{38}) unique addresses.

<table>
<tr><td>Version
4 bits</td><td>Traffic Class
8 bits</td><td colspan="2">Flow Label
20 bits</td></tr>
<tr><td colspan="2">Payload Length
16 bits</td><td>Next Header
8 bits</td><td>Hop Limit
8 bits</td></tr>
<tr><td colspan="4">IPv6 Source Address
128 bits (16 bytes)</td></tr>
<tr><td colspan="4">IPv6 Destination Address
128 bits (16 bytes)</td></tr>
</table>

Figure 4: IPv6 Header format

The IPv6 header is always present and is a fixed size of 40 bytes. The fields in the IPv6 header are described briefly below.

The "Version" field is used to indicate the version of IP and is set to 6 in the case of IPv6; the field size is 4 bits.

The "Traffic Class" field indicates the class or priority of the IPv6 packet. The size of this field is 8 bits. The Traffic Class field provides similar functionality to the IPv4 Type of Service field. In RFC 2460, the values of the Traffic Class field are not defined. However, an IPv6 implementation is required to provide a means for an application layer protocol to specify the value of the Traffic Class field for experimentation.

The "Flow Label" field indicates that this packet belongs to a specific sequence of packets between a source and destination, requiring special handling by intermediate IPv6 routers. The size of this field is 20 bits. The Flow Label is used for non-default quality of service connections, such as those needed by real-time data (voice and video). For default router handling, the Flow Label is set to 0. There can be multiple flows between a source and destination, as distinguished by separate non-zero Flow Labels.

The "Payload Length" field indicates the length of the IP payload. The size of this field is 16 bits. The Payload Length field includes the extension headers and the upper layer PDU. With 16 bits, an IPv6 payload of up to 65,535 bytes can be indicated. For payload lengths greater than 65,535 bytes, the Payload Length field is set to 0 and the Jumbo Payload option is used in the Hop-by-Hop Options extension header.

The "Next Header" field indicates either the first extension header (if present) or the protocol in the upper layer PDU (such as TCP, UDP, or ICMPv6, etc). The size of this field is 8 bits. When indicating an upper layer protocol above the Internet layer, the same values used in the IPv4 Protocol field are used here.

The "Extension Header" field is utilized for additional functionality that might be needed, such as jumbo packet sizes, security, etc. Zero or more extension headers can be present and are of varying lengths. A Next Header field in the IPv6

header indicates the next extension header. Within each extension header is another Next Header field that indicates the next extension header. The last extension header indicates the upper layer protocol (such as TCP, UDP, or ICMPv6) contained within the upper layer protocol data unit. The IPv6 header and extension headers replace the existing IPv4 IP header with options. The new extension header format allows IPv6 to be augmented to support future needs and capabilities. Unlike options in the IPv4 header, IPv6 extension headers have no maximum size and can expand to accommodate all the extension data needed for IPv6 communication.

The "Hop Limit" field indicates the maximum number of links over which the IPv6 packet can travel before being discarded. The size of this field is 8 bits. The Hop Limit is similar to the IPv4 TTL field except that there is no historical relation to the amount of time (in seconds) that the packet is queued at the router. When the Hop Limit equals 0, the packet is discarded and an ICMP Time Expired message is sent to the source address.

The "Source Address" field stores the IPv6 address of the originating host; the size of this field is 128 bits.

The "Destination Address" field stores the IPv6 address of the destination host; the size of this field is 128 bits. In most cases the Destination Address is set to the final destination address. However, if a Routing extension header is present, the Destination Address might be set to the next router interface in the source route list.

1.4 IPv4 vs. IPv6

Table 2 shows the highlights in the differences between IPv4 and IPv6 protocols. There are many other differences; however, it depends on what level of detail we wish to examine the matter. There have been entire books written on the IPv6 protocol and all the differences down to the minutest detail from the old IPv4 protocol. One such book is "IPv6 Networks" [**16**] by Marcus A. Goncalves; it offers an excellent in-depth explanation of any material covered here in this chapter regarding IPv6 networks and much more.

An important aspect of the following information is that the facts presented in Table 2 and Table 3 are all theoretical. These are all the proposed changes that have been outline in the various Requests for Comments (RFC) lead by IETF. The actual implementation of all the features is still in the infancy stages of development and there still lacks maturity, as will be presented in our experimental results. Most likely, by the time that IPv6 will be deployed worldwide and will replace IPv4, all the features stated below should be implemented. Most experts predict that in the next five years, most of the Internet will have support for IPv6.

The left hand side of the table represents features of IPv4 while the right hand side represents features of IPv6; they are interrelated and depict how the particular feature of IPv4 was upgraded to support IPv6. Definitions of the terminology or acronyms can be found in Appendix A.

IPv4	IPv6
Source and destination addresses are 32 bits (4 bytes) in length.	Source and destination addresses are 128 bits (16 bytes) in length.
IPSec support is optional.	IPSec support is required.
No identification of payload for QoS handling by routers is present within the IPv4 header.	Payload identification for QoS handling by routers is included in the IPv6 header using Flow Label field.
Fragmentation is supported at both routers and the sending host.	Fragmentation is only supported at the sending host.
Header includes a checksum. Must be computed at every intervening node on a per packet basis.	Header does not include a checksum. It relies on other layers to find erroneous packets.
Header includes options. Potential inefficient use of header bits.	All optional data is moved to IPv6 extension headers.
Address Resolution Protocol (ARP) broadcast ARP Request to resolve an IPv4 address to the link layer.	ARP Request frames are replaced with multicast Neighbor Solicitation messages.
Internet Group Management Protocol (IGMP) is used to manage local subnet group membership.	IGMP is replaced with Multicast Listener Discovery (MLD) messages.
ICMP Router Discovery is used to determine the IPv4 address of the best default gateway.	ICMPv4 Router Discovery is replaced with ICMPv6 Router Solicitation and Router Advertisement.
Broadcast addresses are used to send traffic to all nodes on a subnet.	There are no IPv6 broadcast addresses; a link-local scope all-nodes multicast address is used.

Must be configured either manually or through DHCP.	Does not require manual configuration or DHCP.
Uses host address (A) resource records in the DNS to map host names to IPv4 addresses.	Uses host address (AAAA) resource records in the DNS to map host names to IPv6 addresses.
Pointer resource records (PTR) in IN-ADDR.ARPA DNS domain map IPv4 addresses to host names.	Uses pointer (PTR) resource records in the IP6.INT DNS domain to map IPv6 addresses to host names.

Table 2: Differences between IPv4 and IPv6 protocol [16**]**

Now that the main differences in the protocols are clear, Table 3 will describe the differences between the IPv4 and IPv6 header fields. The left column names the header field while the right side describes the change which IPv6 incurred from its IPv4 predecessor.

IPv4 Header Field	IPv6 Header Field
Version	Same field but with different version numbers.
Internet Header Length	Removed in IPv6. IPv6 does not include a Header Length field because the IPv6 header is always a fixed size of 40 bytes. Each extension header is either a fixed size or indicates its own size.
Type of Service	Replaced by the IPv6 Traffic Class field.
Total Length	Replaced by the IPv6 Payload Length field, which only indicates the size of the payload.
Identification	Removed in IPv6. Fragmentation information is not included in the IPv6 header. It is contained in a Fragment extension header.
Fragmentation Flags	Removed in IPv6. Fragmentation information is not included in the IPv6 header. It is contained in a Fragment extension header.
Fragment Offset	Removed in IPv6. Fragmentation information is not included in the IPv6 header. It is contained in a Fragment extension header.
Time to Live	Replaced by the IPv6 Hop Limit field.
Protocol	Replaced by the IPv6 Next Header field.
Header Checksum	Removed in IPv6. In IPv6, bit-level error detection for the entire IPv6 packet is performed by the link layer.
Source Address	The field is the same except that IPv6 addresses are 128 bits in length.
Destination Address	The field is the same except that IPv6 addresses are 128 bits in length.
Options	Removed in IPv6. IPv4 options are replaced by IPv6 extension headers.

Table 3: Differences between IPv4 and IPv6 headers [**16**]

1.5 RFC2096 Summary – IP Forwarding

RFC 2096, which describes IP forwarding in great detail, discusses the respective variables that must be taken into consideration when implementing IP forwarding. The relevant variables to my implementation are:

- *ipCidrRouteDest*
- *ipCidrRouteMask*
- *ipCidrRouteNextHop*
- *ipCidrRouteIfIndex*

ipCidrRouteDest – The destination IP address of this route. This object may not take a Multicast (Class D) address value.

ipCidrRouteMask – Indicate the mask to be logical AND with the destination address before being compared to the value in the ipCidrRouteDest field. For those systems that do not support arbitrary subnet masks, an agent constructs the value of the ipCidrRouteMask by reference to the IP Address Class.

ipCidrRouteNextHop – On remote routes, the address of the next system en route; Otherwise, 0.0.0.0.

ipCidrRouteIfIndex – The ifIndex value which identifies the local interface through which the next hop of this route should be reached.

1.6 IPv4 Forwarding

The IPv4 forwarding table will have the following entries:

- Destination Address
- Destination Address Mask

- Route Interface Index

The entries in the table will be statically defined since we will not implement routing protocols such as BGP or RIP to automatically update the tables. For every destination address and mask, there will be a next hop and interface index defined. The destination address AND the mask of each incoming packet will be verified against the forwarding table using longest prefix match and forwarded on the appropriate interface. Also, for every incoming packet, the checksum will have to be computed, the TTL field will have to be decremented, the total length field will have to be examined, and the fragmentation field will have to be examined.

The FPP will perform the appropriate processing in the following manner:

1. *Version field*
    a. match against a value of 4 for IPv4 processing
2. *Length field*
    a. determine where the IPv4 header ends and the payload begins
3. *TTL / Hop Limit field*
    a. Match against a value greater than 0 to continue processing
4. *Fragmentation and Protocol field*
    a. IPv4 header processing requires checking the frag bits to determine if the IPv4 packet is fragmented; also the protocol field must be examined to determine higher layer protocol
5. *Checksum field*
    a. *Examine checksum field*
6. *Destination addresses field*
    a. *Examine IPv4 address*
7. Search forwarding table
    a. Search the forwarding table for a pattern matching via the tree look-up in the FPL code
        i. When a match is found, a return value is sent back to the FPL program from which the search was invoked
        ii. Performs simple routing with returned value after steps 8 & 9
8. *TTL field*
    a. Decrement TTL
9. *Checksum field*
    a. Updates the Checksum

The FPP can process 6-8 bits per cycle most of the time. The FPP can make a match for the first 12 bits in 1 clock cycle and thereafter 1 clock cycle for every 4 bits.

1.7 IPv6 Forwarding

The IPv6 routing table will have the following entries:

- Destination Address
- Destination Address Mask
- Route Interface Index

The entries in the table will be statically defined since we will not implement routing protocols such as BGP or RIP to automatically update the tables. For every destination address and mask, there will be a next hop and interface index defined. The destination address AND the mask of each incoming packet will be verified against the forwarding table using longest prefix match and forwarded on the appropriate interface. Also, for every incoming packet, the Hop Limit field will have to be decremented and the next header field must be examined.

The FPP will perform the appropriate processing in the following manner:

1. *Version field*
    a. match against a value of 6 for IPv6 processing
2. *Hop Limit field*
    a. Match against a value greater than 0 to continue processing
3. *Next Header field*
    a. Ensure that no further processing must be done on additional headers
4. *Destination addresses*
    a. Examine *IPv6 destination address*
5. Search the forwarding table
    a. Search the forwarding table for a pattern matching via the tree look-up in the FPL code

i. When a match is found, a return value is sent back to the FPL program from which the search was invoked
ii. Performs simple routing with returned value after step 6

6. *Hop Limit field*
   a. Decrement Hop Limit

The FPP has the same processing capabilities; we will examine particular performance bottlenecks in a later section.

2.0 SIMULATION TEST-BED AND ASSUMPTIONS

The testbed for these experiments will be Agere's 2.5G PayloadPlus chip set. I will be using System Performance Analyzer (SPA) that will allow the simulation of the 2.5G chip set.

IP forwarding will be done based on the longest prefix match of the destination address, which is a function facilitated by the 2.5G PayloadPlus network processor. Each routing table will have statically defined forwarding tables and a default route in the event a match is not found.

A complete IPv6 stack implementation should include the support for extension headers, namely: Hop-by-Hop options, Routing, Fragment, Destination Options, Authentication, and Encapsulating Security payload. I do not plan to implement these extension headers for this project due to time constraints. The IP forwarder should be able to function properly in a controlled experiment without the implementation of any extension headers.

3.0 EXPERIMENTAL RESULTS

This section will cover the performance results obtained from Agere's SPA simulation environment and how the results compared to previous work done at the application layer.

4.0 CONCLUSION AND FUTURE WORK

This will conclude the work and draw some conclusions regarding IPv6 based on our findings.

5.0 REFERENCES

[1] IETF IPv6 Transition Working Group, http://www.6bone.net/ngtrans.

[2] I. Raicu. "An Empirical Analysis of Internet Protocol version 6 (IPv6)", Master Thesis, Wayne State University, 2002.

[3] Information Sciences Institute, University of Southern California, "Internet Protocol," Request for Comments 791, Internet Engineering Task Force, September 1981

[4] S. Bradner, A. Mankin, "IP: Next Generation (IPng) White Paper Solicitation," Request for Comments 1550, Internet Engineering Task Force, December 1993

[5] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," Request for Comments 1883, Internet Engineering Task Force, December 1995

[6] S. King, et al. "The Case for IPv6", Internet Draft draft-ietf-iab-case-for-ipv6-06.txt, Internet Architecture Board of Internet Engineering Task Force, December 1999, http://www.6bone.net/misc/case-for-ipv6.html.

[7] I. Raicu, S. Zeadally. "Impact of IPv6 on End-user Applications", 10th International Conference on Telecommunications, ICT'2003, Tahiti Papeete, French Polynesia, February 23, 2003.

[8] R. P. Draves, et al. "Implementing IPv6 for Windows NT". Proceedings of the 2nd USENIX Windows NT Symposium, Seattle, WA, August 3-4, 1998.

[9] P. P. Xie. "Network Protocol Performance Evaluation of IPv6 for Windows NT", Master Thesis, California Polytechnic State University, San Luis Obispo, June 1999.

[10] K. K. Ettikan. "IPv6 Dual Stack Transition Technique Performance Analysis: KAME on FreeBSD as the Case", Faculty of Information Technology, Multimedia University, Jalan Multimedia, October 2000

[11] K. K., Ettikan, et al. "Application Performance Analysis in Transition Mechanism from IPv4 to IPv6". Research & Business Development Department, Faculty of Information Technology, Multimedia University (MMU), Jalan Multimedia, June 2001.

[12] Microsoft Corporation, "Microsoft IPv6 Technology Preview for Windows 2000," December 12, 2000, http://www.microsoft.com/windows2000/technologies/communications/ipv6/default.asp

[13] A. S. Tanenbaum, Computer Networks, Third Edition, Prentice Hall Inc., 1996, pp. 686

[14] A. S. Tanenbaum, Computer Networks, Third Edition, Prentice Hall Inc., 1996, pp. 413-449

[15] William Stallings. High Speed Networks, TCP/IP and ATM Design Principles. Pages 444 - 457.

[16] Marcus A. Goncalves, Kitty Niles. "IPv6 Networks", McGraw-Hill, 1998.

[17] C. Huitema, "IPv6, The New Internet Protocol, Second Edition," Prentice Hall Inc., 1997, pp. 197-221.