Comparison Between Ipv4 And Ipv6 In Adopting Differentiated Services.

Mohammed Al-Zobbi

Abstract: IPv4 is an old protocol and suffers from addresses shortage, redundancy and other problems. IPv6 is a new protocol, which is not been widely adopted, but it supports Quality of Service (QoS) Features and a huge range of unique addresses. IPv6 needs to be tested and evaluated regarding to QoS support. The Differentiated Services "DiffServ" is a QoS framework; it could be implemented in either IPv4 or IPv6. DiffServ requires 6-bit value to be assigned to IPv4/IPv6 header; this value is called DSCP (DiffServ codepoint). DSCP is set in a "ToS" field in IPv4, and in a "Class" field in IPv6. There is no any implementation difference between assigning DSCP in either IPv4 or IPv6. In this research, a comparison between IPv4 and IPv6 in adopting DiffServ framework is presented. DiffServ represents the QoS frameworks; the impact of each version of IPs on QoS, affects the DiffServ framework, and as a result, affects the whole network performance.

Index Terms: Adopting DiffServ, CodePoint in DeiffServ, Comparison between IPv4 and Ipv6, DiffServ and QoS, IPv4 and IPv6, IPv6 and QoS, Network protocol

1. Introduction

The problem of the limited IPv4 addresses could be solved in different alternative technologies such as: subnetting, Network Addresses Translation (NAT), or Classless Inter-Domain Routing (CIDR). However, with NAT, the external people see the entire subnet as one computer, and this inherits problems [26]. IP addresses might be solved for a while, but they will no longer be able to handle the fast growth of Internet. Moreover, some more problems are hardly to be solved relating to the current structure of IPv4. More problems are discussed in the following sections. Routers need to lookup the longest match in the conventional IP routing list; routers then decide the right path to send the datagram through [6]. The limited amount of network would cause any delay problem for the routers; in fact, the computer networks are increasing continuously, which could reach to a level of causing a serious delay for the routed data. In such case, the future network may witness more delay in the network. This routing issue affects the performance of the networks, and as a result it affects the Internet's growth [6]. Subnetting is a technique used to save the IP addresses by assigning as much addresses as possible [3]. Three main classes of addresses are available, these are A,B and C. Class A provides 3 empty digits of addresses, e.g. 122.x.x.x. While Class B provides 2 empty digits of addresses, e.g. 122.211.x.x. and Class C provides 1 empty digit of addresses, e.g. 122.211.101.x. By subnetting, a class A, B, or C network addresses can be divided into different subnets. Each subnet is used to identify a branch of the organizations [3].

Gai (1998) replays that the use of subnets can increase the efficiency of the network to a certain level. In fact, users or organizations may abuse the IP addresses system, by purchasing class B addresses, while the used addresses are not more than 100. In this case, more than 65,500 addresses are potentially wasted. However, Loshin (1999) claims that it's not only the user's fault, but also the nature of IP addresses is a waste, because organizations have to buy their IP for no more than one or two terminals; this means tying up 254 host addresses. The subnetting causes IP addresses wastage of resources. The size of the organization should match the chosen IP class, and this rarely happens.

2. IPv6

IPv6 addresses are 128 bits long or 16 octets. IPv6 addresses are four times longer than IPv4 addresses. The hexadecimal system is used to assign the addresses, instead of the decimal system in IPv4. Eight groups of numbers, separated by a colon (:), are used, each group contains pair of hexadecimal numbers, for example:

FEDC: BA88:45DF:9810:0008:417A:0000:0FB6

In this example, we can see the difficulties of remembering or managing these addresses. This explains the need for DHCP and DNS servers, in order to assign these addresses automatically. A new version of DHCPv6 is designed to provide auto configuration to client-server nodes [4]. Compression methods or shortcuts can be applied to IPv6 addresses by neglecting zeros; for example, we can type 0 instead of 0000, or 20 instead of 0020 [4]. More compression can be applied, by replacing (:) by a series of zeros: as an example: the address 0050:0:0:0:0:0:0:CA2B could be re-written as 50: :CA2B. Some reserved addresses use the principle of shortcut, such as: multicasting: **FF01: :43**, loop back address **: :1**, and unspecified address **: :** [9].

2.1 IPv6 Extension Header

IPv4 header contains an option header, while IPv6 assigns the options in separateextension headers. This design improves the routers performances; the routers don't need to examine extra fields, especially if they were unused. The extension header is almost 64-bit size, and they can't be

- *The author has completed a master degree in Computing, from the University of Western Sydney / Australia. This research was supervised by Dr. Seyed Shahrestani. Author email: malzobbi@yahoo.com* empty; they are used only on demand. Figure (2.3), illustrates an extension headers examples [1]. The extension headers are processed in the order they appear in the packet [4], RFC 1883 recommends that these headers appear as the following: IPv6 header, Hop-by-Hop option header, Destination Options headers, Routing headers, Fragment header, Authentication header, Encapsulating Security Payload header, Destination header, Upper header [4,23]. The Hop-by-Hop and Destination extension header are encoded by using TLV format. Hop-by-Hop extension header handles the extra-large IP packets, which is over 65,535 bytes; routers are unable to handle such these jumbo packets. In IPv6, these packets are discarded without adding the Hop-by-Hop optional extension header. Figure (2.4) shows the format of option header in TLV format [23].

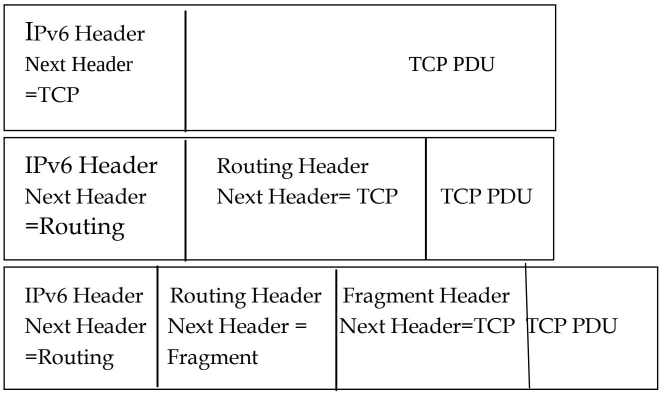

Figure (2.3), *Extension Header Source [4]*

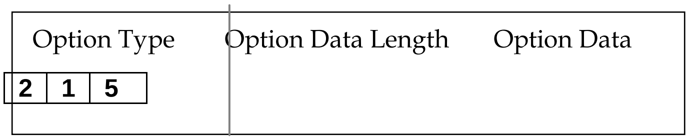

Figure (2.4), *TLV options*

Figure (2.4) shows three fields in the extension header, these fields are divided as the following: IPv6 header notifies the optional header by using the next-header field, for example; Hop-by-Hop option header is notified by (0), and fragment option is notified by (44) [3]. The Jumbo Payload Option is essential to the packets that exceed the 65,535 bytes. The Payload Length field in IPv6 header assigns 16 bits only, while the Jumbo Payload option assigns (4n + 2), whereas n: any Integer. However, the maximum limit of data is 4,294,967,295 octets [23]. The Jumbo Payload Option is included in Hop-by-Hop option header, the Option Type value = 194, and the Option Data Length value = 4, which indicates the following data. The Payload Length field in IPv6 header is set to zero to indicate the Hop-by-Hop option header.

3. Differentiated Service (DiffServ)

The Internet uses share bandwidth, the increase demand on the Internet leads to wide varieties of multimedia applications such as: real time voice-over-IP, video streaming, and video conferencing. DiffServ can satisfy quantitative performance (such as peak bandwidth) or relative performance (such as class differentiation). It first assigns bits to IP header, and then it uses these bits to determine how packets are forwarded, and finally uses the marked packet to force QoS policies and conditions at the network boundaries [12]. The PHB provides different treatments to the network traffic. DiffServ requires monitoring, policing, and shaping in order to investigate the satisfaction for the QoS. Traffic analysis and statistical calculation are essential processes to provide the satisfied configuration for PHB. Some difficulties may face the PHB configuration as a reason of non-standardizing the PHB configuration, and the non-fully understanding of traffic rules and polices. Figure (3.1) shows DiffServ diagram. The DS size is 8-bit divided into two fields; the first field is called DiffServ Codepoint (DSCP), the field size is six-bit used to identify the packet. The second field size is two-bit Currently Unused (CU).

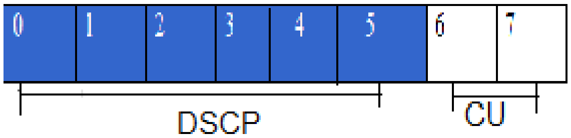

Figure (3.1), *shows DiffServ classification source [20]*

The unused field should be ignored, while the six-bit field can be set as (0,1). There is no standard to assign values, implementers should be aware of six-bit field to create their own table of index, which is used to select a particular packet handling mechanism. The particular codepoint is selected by the following the format “xxx000” or DiffServ is selected as “xxx000|xx”. This packet marking is done on the edge routers, the edge routers examines the packet, and then decides the suitable codepoint to that packet based on the table of index. The core router doesn’t need to assign any marks, it only reads the codepoint and decides the class selector PHB based on it. A default BHP “000000” is used to enable the delivery of packets without any strict conditions or rules. The packet marked with a default BHP may be re-marked again with another codepoint as it passes the boundary to another domain.

3.1 DiffServ and Tunneling

The main principle, of DiffServ tunneling configuration, is setting the inner and outer DSCP properly to avoid transmission faults. Six different statuses are defined for encapsulation and decapsulation processes, these statuses are shown in figure (3.3). The diagram shows [1-before] means the packet before encapsulation, [2-inner] means the inner packet is encapsulated, [3-outer] means the outer packet is encapsulated, [4-inner], and [5-outer] mean the inner and the outer packets are decapsulated, while [6-after] is the packet after decapsulation.

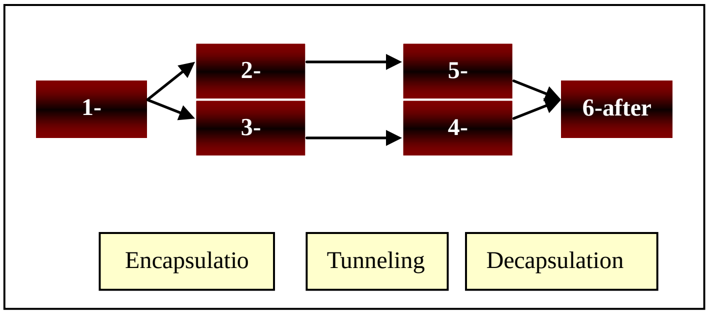

***Figure (3.3)**, shows 6-statuses for encapsulation and decapsulation processes.*

DiffServ architecture permits packet reordering; other technologies such as IPsec, and L2TP are sensitive to packet reordering. If packet reordering is disabled in IPsec, then DiffServ must stop the reordering to avoid wrong interaction. One way can disable reordering in DiffServ; this way is enabling PHB to prevent reordering by setting [2-outer] to a specific value. If reordering method is desirable within the tunnel, multiple tunnels should be used, whereas each single tunnel has different behaviors, and IPsec tunnel is separated from the DiffServ tunnel. However, multiple tunnel modes affect the tunnel security and consume the bandwidth, as a result of packets redundancy. In the DiffServ configuration, the traffic conditioning could be processed on either egress or ingress. The egress recognizes either [5-outer] DSCP or [4-inner] DSCP; otherwise, DSCP value could be lost during the decapsulation process. Network designers must decide which header should traverse the correct value of DSCP. The chosen header is based on the following conditions;

1. Partially DiffServ configuration: if the ingress is DiffServ compatible only, then the egress discards the [5-outer] without copying DSCP value to [4-inner] during the decapsulation process. In such case, the [4-inner] DSCP is not recognized by the egress unless it's set to zero.
2. Fully DiffServ configuration, the tunnel carries two DSCP values (i.e. inner and outer). It is necessary to decide the useful value on the egress side, the decision should take two different models into considerations, and these models are:
   - In the Uniform model: the outer header contains the useful DSCP value, therefore, the configuration shouldn't neglect the outer header.
   - In the Pipe mode: the inner header contains the useful DSCP value, therefore, the configuration shouldn't neglect the inner header. IPsec tunnel is usually based on the Pipe model. Choosing the wrong header may break the security tunnel.

The sequence of decapsulation via conditioning is essential, the egress can't reach the [4-inner] to perform the conditioning. Therefore, decapsulation should be done first, and then the conditioning process is performed at [6-after]. In some cases, the configuration could be more complicated and requires more understanding for a wider network. When the tunneled packet travel through many intermediate nodes, both inner and outer DSCP values are useful, the outer value is necessary to maintain traffic conditioning among the intermediate nodes. This type of configuration should be considered carefully, before configuring both ends. It's clear that configuring DiffServ tunnel is a complicated issue; any wrong configuration could cause the tunnel fail and break the security. The configuration should be done manually based on TCA. The network administrator should have a complete knowledge about the rest of the network nodes whereas the traffic route through.

4. IPv6 Advantages in QoS

It is essential to remember that DiffServ framework is a QoS issue, and it's a set of techniques used to support QoS. Let's first discuss IPv6 advantages in supporting QoS level. In general, these advantages reduce the routing time and increase the network performance: IPv6 header is a fixed size of 40-byte; the fixed size increases routers efficiency and reduces the routing delay time. In v4, routers spend more time to look up the IP Header Length field; Header Length field is not available in v6, because v6 header is a fixed size, while v4 is a variable size of 20-60 bytes. Header Length field in v6 is no longer needed due to the fixed header size [37]. IPv6 header contains Flow Label field of 20-bit, this label assists routers to check the datagram ID, and source and destination port. Routers do not need to read the higher layers to classify the packets based on the port and protocol types. This field definitely saves the routers time and supports QoS. DiffServ is one of QoS frameworks that get advantages of the Flow Label field in IPv6 header. The MF classification in the edge routers doesn't need to refer to the transport layer to recognize the 3 out of the 5 tuples; instead, routers read the needed 5 tuples values from the same layer and header during the packet classification process. Fragmentation in IPv6 is left to the host node, rather than consuming more fragmentation time on the router side. IPv6 uses the extension header "Fragment Header" to assign the fragmentation process if needed, unlike v4, which assigns 3 different fields (Datagram ID, Flag, and Fragment Offset), with 32-bit size, in the IP header for the fragmentation process; these fields are always available, even if the fragmentation process is not required. In v4, the routers need to read, and may write, the three fragmentation fields on each routed packet. This process is a time consumer and causes more delay on routing packets.

2 **Security Feature:** one of the main DiffServ problems is the security features, service can be easily stolen by marking packet headers with appropriate DSCP codes [29], this type of stealing could be established on both edge and/or core routers [31]. IETF DiffServ working group proposed two main solutions; these are Auditing and IPsec. Auditing is defined as monitoring any suspicious events in the domain. While IPsec tunnel mode provides an essential security to the DiffServ domain [31]. IPsec was developed for IPv6; also it can be implemented in IPv4. IPsec and other security methods work properly with dual stack v4/v6 or tunneling. While NAT solutions for IPv4 breakthe end-to-end semantics and other security mechanisms such as Kerberos[1], as a resultof rely on end-to-end addresses, whilst these addresses are hidden. IPsec

is easier to be implemented over IPv6 tunneling, simply because NAT is not designed for IPv6 [32]. As mentioned before, IPv6 security is expected to be an essential issue for the IPsec vendors, especially that IPv6 relies on the tunneling mode to process its transmission. IPv6 is expected to be more securing than IPv4, as a result of the commercial issues [22]. The above advantages support QoS in IPv6. More advantages are not mentioned here, such as wireless mobile uses. However, our main interest is QoS features in v6, and the above findings indicate that v6 doesn't provide any QoS features, but it enhances a better and faster traffic route over the network nodes.

5. IPv6 Disadvantages in QoS

The IPv6 disadvantages are due to the current implementation of Internet network. Millions of Internet routers are not IPv6 compatible, the connection between two-end entities requires both-end protocols similarities, otherwise tunneling or translation is essential. The desired Internet plan is to enable the v6 domains over wider geographical networks; this approach investigates the desired QoS advantages over the network, in fact, this approach is not the current picture. As a result of the limited use of v6, three main implementation problems appeared on using v6 in DiffServ; these are Delay Problem in implementation, Security Problem in Implementation, and cost. These problems are expected to last for decades before the new protocol takes a place. Both of edge routers and core routers are essential nodes to implement DiffServ frameworks properly. The edge routers carry the responsibility of applying SLA and SLS as required; these processes are expected to increase the edge routers delay. However, there are no any other solutions available at the time, and the high-speed processors may overcome this delay for a while. Two main parameters in QoS problems are known as delay and congestion, each of which is related to each other's. The delay causes high possibility of increasing congestion, and the congestion causes high possibility of increasing packet droppings. The same delay problem could affect the core routers, which is more desirable to have less delay. The BA in DiffServ drops the less priorities packets, if any congestion occurs, before PHB takes a place. IPv6 increases the possibility of delay, especially, when the transmission behavior varies all the time. IPv6 routers switch between tunneling behavior and NAT-PT translation behavior; each one of these behaviors may encounter unknown delay; this delay affects the whole performance of DiffServ. In this scenario, IPv6 packets couldn't be transferred through a tunnel, the NAT-PT is essential to enable the packet routing through IPv4 routers. The core router should spend an extra time to translate between IPv6/IPv4 addresses, IPv6 is designed to support QoS parameters by saving a considerable routing time; the design objective could be reasonable, if the packet travels through an IPv6 tunnel. In fact, this strategy may not happen. Adding to the objective loss, the core routers may witness more delay as a result of IPv4/IPv6 address translation. NAT-PT is better to be implemented on the edge routers rather than the core router. However, the same delay will be encountered, especially when the edge router is linked directly to the backbone. Both routers (edge and core) are essential nodes in adopting DiffServ and accelerating the routing time regardless to NAT-PT implementation. Another solution to the backbone translator requires installing NAT-PT translator on each entire IPv6 node in the organization. Each time the node needs to communicate with any outside IPv4 node, it translates its IP address before it reaches the core router; this technique reduces the translation delay on the core routers. However, this may be acceptable for small networks, but it's not possible for the large networks. The tunneling process may cause another kind of delay, especially when "Don't Fragment" value is set in IPv4 header (*see section 2.3*), in such case, encapsulator reacts by re-transmitting IP datagram, unless MTU discovery is used, which is also a time consumer. More delay is expected when Dual Stack is deployed. Dual Stack produces two major delays; the first delay is caused by the resolver on filtering DNS, while the tunneling process causes the second delay. The resolver delay time is varied and depends on the right IP recognition by the application, if the resolver replayed both versions, then each version should be tested, until the right version is chosen; this technique causes an extra delay (*see section 2.5*). Investigating the ultimate advantages of IPv6 design objectives require many factors participation. In the above scenario, the edge router doesn't need to consume an extra time for the following processes; fragmentation process, unknown header size process, and transport layer classification. On the other hand, IPv6 forces the edge and the core routers to handle extra functions that are not available in IPv4. Edge and core routers are essential nodes in DiffServ framework functionality. The edge and the core routers have to handle the following extra functions: searching for the longest match prefix in IPv6 address for the In/Out data flow, NAT-PT translation (if implemented), and Dual Stack or tunneling processing (if implemented). The comparison between v4 and v6 in handling these processes derives the following results:

- DiffServ classification in IPv6 (same as v4).
- Searching for the longest match IPv6 address for the In/Out data flow (IPv6 slower)
- NAT-PT translation (IPv6 slower)
- Tunneling or Dual Stack (IPv6 slower).

The first v6 process (classification) consumes the same period of time as that of v4 does, because the classification depends on the DSCP value, which is stored in IPv4/TOS field and in IPv6/Class field, the same period of time is required in both versions. The second process (searching for longest match) requires more time in IPv6; the reason is that IPv6 supports 16-byte addresses, with network prefixes up to 121 bits, while IPv4 supports 4-byte addresses, with network prefixes up to 24 bits (*see section 2.2.2*). The third and the fourth processes are the NAT-PT, and Tunneling or Dual Stack respectively, which are available in IPv6 only. The above results derive a conclusion of longer expected delay on the backbone side. Using QoS frameworks may not give the best possible performance in the network design. With the existing technology, IPv4 performance is better. Both of edge routers and core routers are essential nodes to implement DiffServ frameworks properly. The edge routers carry the responsibility of applying SLA and SLS as required; these processes are expected to increase the edge routers delay. However, there are no any other solutions available at the time, and the high-speed processors may overcome this delay for a while. Two main parameters in QoS problems are known as delay and congestion, each of which is related to each other's. The delay causes high possibility of increasing congestion, and the congestion causes high possibility of increasing packet droppings. The same delay problem could affect the core routers, which is more desirable to have less delay. The BA in DiffServ drops the less priorities packets, if any congestion occurs, before PHB takes a place. IPv6 increases the possibility of delay, especially, when the transmission behavior varies all the time. IPv6 routers switch between tunneling behavior and NAT-PT translation behavior; each one of these behaviors may encounter unknown delay; this delay affects the whole performance of DiffServ.

6. Solutions

It's possible to improve IPv6 speed and security implementation by adopting different technologies or techniques. Here are some suggested solutions to improve the general performance of the network. Using MPLS technology to avoid the longest match prefix search (improving speed). MPLS is a 3-field label addedat the data link layer, it works with both IPs versions and reduces the routing time, and as a result QoS improvement. Using tunneling transmission or Dual Stack methods rather than using NAT-PT (improving speed and security). Tunneling is required for the pure IPv6 entire network, while Dual Stack is needed where the entire network contains both versions. Using RSIP as an alternative to NAT-PT. However, this technique is still not clarified, and it needs to be standardized by IETF (Improving Security). Improving tunneling method to reduce the Internet security threat (improving security). Installing NAT-PT translator on each IPv6 node in the domain, this technique can be applied for small networks to reduce the translation delay on the edge or core routers (improving speed). Improving DiffServ tunneling configuration methods. Many suggested methods could reduce configuration complexity such as reducing number of configuration models, and standardizing TCA between different domains.

7. Conclusion

Investigating the differences between IPv4 and IPv6 in adopting DiffServ requires finding each protocol impact on QoS performance. In this research, a study is proposed on the performances of IPv4 and IPv6. IPv6 is a new protocol, and it is essential to be widely distributed, before it takes a place in the QoS market. While IPv4 is an old protocol, in comparison with IPv6, it is redundant and performs slower over the networks nodes, but it is whole performance remains comparatively faster, regarding to the current Internet network. IPv4 performs better than that of IPv6 does in the core and edge routers; the core and the edge routers interact together to provide a scalable DiffServ performance, if their speed remain steady. The current Internet implementation, forces IPv6 to follow different behavior, IPv6 needs to be translated or tunneled before it is transmitted between two-end nodes. The translation is needed when the destination node is an IPv4 compatible, NAT-PT is used to translate IPv6/IPv4 addresses, and this technique causes more delay in the core routers, and breaks the security mechanisms. Tunneling is more desirable technique, its security performance is the best over IPsec, but it may cause a security threat over the Internet; for this reason, more care should be taken on implementing the tunneling technique. Moreover, IP encapsulation process is a time consumer. The architectural design for IPv4 is less efficient than IPv6 in routing datagram through the network. On the other hand, the current Internet implementation provides IPv4 a better performance over the Internet. IPv6 causes a significant delay, especially, if the transmission variation has occurred between tunneling behavior and NAT-PT translation behavior. The delay occurs on the core and/or edge routers may cause delay symptoms. The DiffServ BA drops the lower priorities packets, if any congestion occurred. IPv6 implementation and management are expensive. Upgrading to IPv6 requires upgrading network nodes to be IPv6 compatible, and the QoS software is limited and still under development. Managing IPv6 in DiffServ tunnel is complicated, more information is required about the traffic nodes, and in somehow it's fault sensitive and requires more care.

8. References

[1]. Black. U, (2000). QOS In Wide Area Networks, Prentice Hall Series.

[2]. Clark. M.P, (2003). Data Networks, IP And The Internet, Wiley.

[3]. Ferguson P & Huston G (1998). Quality of Service: Delivering QoS on the Internet And In Corporate Networks. Wiley Computer Publishing.

[4]. Gai. S, (1998). Internetworking IPV6 With Cisco Routers, McGraw-Hill.

[5]. Held.G & Hundley. K (2000). Cisco IOS IP: Field Guide, McGraw-Hill.

[6]. Loshin. P, (1999). IPV6 Clearly Explained, Morgan kanfmann publishers.

[7]. Ichiro J. & Hagino I. (2000), Socket API for IPv6 traffic class, http://playground.sun.com/pub/ipng/html/presentations/Dec2000/flowinfo.pdf, [Retrieved: 17/12/03].

[8]. Liebeherr J. &Christin N. (2002), Rate Allocation And Buffer Management For Differentiated Services. http://www.informatik.uni-trier.de/~ley/db/journals/cn/LiebeherrC02, Computer Networks Vol. 40

[9]. Partridge. C, (1995), RFC 1809: Using the Flow Label Field in IPv6

[10]. Perkins. C, (1996), RFC 2003: IP Encapsulation Within IP

[11]. Perkins.C, (1996.), RFC 2002: "IP Mobility Support"

[12]. Nichols.K & S. Blake &F. Baker &D. Black (1998), RFC 2474: Definition of The Differentiated Services Field (DS Field) In The IPv4 and IPv6 Headers

[13]. Blake, S & Black, D, (Dec 1998) RFC 2475: An Architecture for Differentiated Services

[14]. Almquist, P (July 1992), RFC1349: Type of Service In The Internet Protocol Suite

[15]. Chen, X, (Nov, 2002), Special Topics: Diffserv Model, http://www.isi.edu/nsnam/ns/ns-tutorial/tutorial-02/slides/diffserv-1.pdf. [Retrieved: 04/01/04] Assured Forwarding PHB Group

[16]. Bake F (June 1995), RFC 1812:Requirements for IP Version 4 Routers

[17]. Binst P V & Vandenbroucke R, (2002), IPv6 Diffserv Study and Test Reporthttp://www.ngnlab.org/diffserv_report.doc, [Retrieved: 01/01/04]

[18]. Wagner S (2002), Seminar 1: Elements of IP Network Design http://www.umiacs.umd.edu/docs/LTS_Sem_021102.pdf, [Retrieved: 22/12/03]

[19]. Jain S R and Hassan M, (2002). Engineering QoS, Artech House,

[20]. Floyd S, (Mar 1995): DARTnet II Meeting: Packet scheduling research, http://www.icir.org/floyd/talks/sf-dartnet-95.pdf [Retrieved: 25/12/04].

[21]. Heinanen J, Finland T & Baker F, (June, 1999), RFC 2597: Assured Forwarding PHB Group.

[22]. Grossetete P, (2001), IPv6 @ Cisco. www.cisco.com/warp/public/732/Tech/ipv6/docs/deployment.ppt, [Retrieved: 09/12/04]

[23]. Deering S, & PARC X & Hinden R, (December 1995), RFC 1883: Internet Protocol, Version 6 (IPv6) Specification.

[24]. Inoue H,Basic Differences Between IPv4 Header And IPv6 Header,http://www.ipv6style.jp/en/tech/20030331/index.shtml, [Retrieved: 01/01/04]

[25]. Conta A & Carpenter B, (2001): The IPv6 Flow Label and Use of IPv6 Flow Labels With Diffserv. http://playground.sun.com/pub/ipng/html/presentations/aug2001/IPv6-flow-label-07.PDF. [Retrieved: 01/01/04]

[26]. Holly Hubbard Preston, Network World: Edge Routers For IPv6 Migration, http://www.itworld.com/Net/4057/NWW010423tech/. [Retrieved: 16/12/03].

[27]. Mark A. and Miller, P.E. (1998): Implementation IPv6:Migration to then Next Generation Protocols, M&T Books.

[28]. Leonidas L& Lupu E & Sloman M (September 2003), An Adaptive policy-Based Framework for Network Services Management. Journal Of Network and Systems Management.

[29]. Jha S. & Hassan M. (2003), Engineering Internet QoS, Atrech House Publishers.

[30]. Aalto T. (1996), IPv6 Authentication Header and Encapsulated Security Payload, http://www.tml.hut.fi/Opinnot/Tik-110.551/1996/ahesp.html, [Retrieved: 18/12/03]

[31]. Striegel A. (2002), Security Issues in a Differentiated Services Internet, http://www.ee.iastate.edu/~gmani/tiw-2002/diffserv-security.pdf, [Retrieved: 10/01/04]

[32]. Fernández J. & Peña S. (2001), Problems Due To Widespread Use Of NAT And IPSEC Considerations, http://www.uninet.edu/6fevu/text/IPSEC-NAT.SGML.html, [Retrieved: 11/01/04]

[33]. Tsirtsis G. & Srisuresh P. (Feb 2000), RFC 2766: Network Address Translation - Protocol Translation (NAT-PT), [Retrieved: 11/01/04].

[34]. Black D. & Brim S. & Carpenter B. (June 2001), RFC 3140: Per Hop Behavior Identification Codes.

[35]. Francis D. (2002), Cisco Helps Service Providers Capitalize on Opportunities at the High-End Edge of the Network, http://www.cisco.com/en/US/products/hw/routers/ps167/products_press_coverage09186a00800c9485.html,[Retrieved: 11/01/04].

[36]. Ginsburg D. & Hatter M. (2002), Implementing IP Services At The Network Edge, Addison-Wesley.

[37]. Chown T. (2002), IPv6 and QoS, www.6init.com/public/iir_qos04.pdf

[38]. M. Kaat, (2000). RFC 2956. Overview of 1999 IAB Network Layer Workshop. [Retrieved: 18/01/04].

[39]. P. Srisuresh, (1999). RFC 2709: Security Model with Tunnel-mode IPsec for NAT Domains. [Retrieved: 15/01/04].

[40]. Tsirtsis G. & Srishuresh P. (1999), Network Address Translation - Protocol Translation (NAT-PT), http://www.ietf.org, [Retrieved: 27/01/04]

[41]. R. Gilligan, & E. Nordmark, (2000). RFC 2893: Transition Mechanisms for IPv6 Hosts and Routers. [Retrieved: 12/01/04].

[42]. Haddad I. (2004), Connecting to the IPv6 Internet, http://linux.oreillynet.com/pub/a/linux/2004/01/22/ipv6.html [Retrieved: 25/01/04]