Innovative Algorithms for the Header Processing Transition from IPv4 to IPv6 and Vice Versa

Basil Al-Kasasbeh[1], Rafa Al-Qutaish[2], and Mohammad Muhairat[2]
[1]Faculty of Information Technology, Applied Science University, Jordan
[2]Department of Software Engineering, Alzaytoonah University of Jordan, Jordan

Abstract: *The huge numbers of computers, devices and networks that connected to the Internet in the networking industry, that require more address space, better Quality of Services support, greater security, and an increasing number of media types and Internet-capable devices have all contributed to drive the development of new IPv6 protocol. The major importance during the development of IPv6 has been how to do the transition away from IPv4 towards IPv6 and vice versa. The work on transition strategies, tools, and mechanisms has been part of the basic IPv6 design effort from the beginning. The transition process from the current IPv4 to the future IPv6 is probably one of the most important subjects during the next generation protocols. This paper reviews the basics of IPv4 and IPv6 headers, and the methods for managing the transformation between IPv4 and IPv6. The proposed algorithms deal with header processing transformation transition between IPv4/IPv6 and vice versa depending on the bi-directional identification and recognition processes of the two distinct headers.*

Keywords: *IPv4/IPv6 transition, IPv4/IPv6 header processing, IP tunneling, IP encapsulation, address mapping, dual stack.*

Received November 9, 2008; accepted April 14, 2009

1. Introduction

When the Internet first came into use nobody was thinking that it will grow this fast and one day we will run out of the IP addresses. Each year the number of the Internet users more than doubled and number of the connection became enormous. The continuous growth of the global Internet requires that its overall architecture evolve to accommodate the new technologies that support the growing numbers of users, applications, appliances, and services. IPv6 is designed to meet these requirements and allow a return to a global environment where the addressing rules of the network are again transparent to the applications. The current internetworking protocol, IPv4 will be unable to adequately support additional nodes or the requirements of new applications because a huge extension of new networks and IP devices attached to the Internet, this given that a large IP address space was needed and hence a new IP protocol would be developed in order to replace IPv4 [6].

IPv6 is a new network protocol that features improved scalability and routing, security, ease-of-configuration, and higher performance compared to IPv4. Most of today's internet uses IPv4, which is now more than twenty years old. IPv4 has been remarkably resilient in spite of its age, but it is beginning to have problems. Most importantly, there is a growing shortage of IPv4 addresses, which are needed by all new machines added to the Internet. Unfortunately, IPv6 is incompatible with IPv4. However, using the new protocol will require changes to the software in every networked device. Consequently, it is necessary to develop transition mechanisms that enable applications to continue working while the hosts and networks are being upgraded. There exist many reasons to make the transition from IPv4 to IPv6: a progressive depletion of the IPv4 address space, a continuous growth of the Internet routing tables, complex IP host and router configuration issues, user requirements for mobility, security and quality of service [21].

The explosive growth of the Internet and its services has exposed deficiencies in IPv4 at the Internet's current scale and complexity. IPv6 was developed specifically to address these deficiencies, enabling further Internet growth and development. IP next generation (IPng) was recommended by the IPng Area Directors of the Internet Engineering Task Force (IETF) at the Toronto meeting on July 25, 1994, and documented in RFC 1752. The recommendation was approved by the internet engineering steering group on November 17, 1994 and made a proposed standard. The improvement from IPV4 to IPV6 comes in the form of simplification of the header format. Even though the IPV6 addresses are 4 times longer than the IPV4 addresses, the IPV6 header is only twice the size of the IPV4 header [5, 15].

This paper reviews the basics of IPv4 and IPv6 headers, and the methods for managing the transformation between IPv4 and IPv6. The proposed algorithms deal with header processing transformation transition IPv4/IPv6 and vice versa depending on the bi-directional identification and recognition processes of the two distinct headers. This paper is organized as follows. Section 2 presents a general overview of IPv4 and IPv6. Section 3 explains the detailed explanation of the proposed algorithms. Finally, section 4 concludes the paper.

2. IPV4 and IPV6: A General Overview

2.1. The Features of IPv4 and IPv6

IPv4 is widely deployed and it is a data-oriented protocol to be used on a packet switched inter-network. IPv6 or IPng is a new version of IP which is designed to be an evolutionary step from IPv4. The main features of IPv6 compared with IPv4 are listed below [13, 14, 16]:

1. Larger IP address space; IPv4 uses only 32 bits for IP address space, which allows only 4 billion nodes to be identified on the Internet. IPv6 allows 128 bits for IP address space, that is, 2^128 nodes to be uniquely identified on the Internet. A larger address space allows true end-to-end communication, without NAT or other short term workarounds against the IPv4 address shortage.
2. Deploy more recent technologies, after IPv4 was specified since more than 20 years ago, we saw many technical improvements in networking. IPv6 includes a number of those improvements in its base specification, allowing people to assume these features are available everywhere, anytime. These technologies include – but are not limited to – the following:
   - Auto configuration: with IPv4, Dynamic Host Configuration Protocol (DHCP) exists but is optional, a novice user can get into trouble if they visit another site without a DHCP server, with IPv6, a stateless host auto-configuration mechanism is mandatory, this is much simpler to use and manage than IPv4 DHCP.
   - Security: with IPv4, IPsec is optional and you need to ask the peer if it supports IPsec. With IPv6, IPsec support is mandatory. By mandating IPsec, we can assume that you can secure your IP communication whenever you talk to IPv6 devices.
   - Friendly to traffic engineering technologies. IPv6 was designed to allow better support for traffic engineering. There are no single standards for traffic engineering yet, so the IPv6 base specification reserves a 24-bit space in the header field for those technologies and is able to adapt to coming standards better than IPv4.
   - Multicast: is mandatory in IPv6, which was optional in IPv4. The IPv6 base specifications themselves extensively use multicast.
   - Better support for ad-hoc networking; scoped addresses allow better support for ad-hoc networking. IPv6 supports any cast addresses, which can also contribute to service discoveries.
3. A cure to routing table growth: the IPv4 backbone routing table size has been a big headache to ISPs and backbone operators. The IPv6 addressing specification restricts the number of backbone routing entries by advocating route aggregation.
4. Simplified header structures: IPv6 has simpler packet header structures than IPv4. It will allow future vendors to implement hardware acceleration for IPv6 routers easier.
5. Allows flexible protocol extensions: IPv6 allows more flexible protocol extensions than IPv4 does, by introducing a protocol header chain. Even though IPv6 allows flexible protocol extensions, IPv6 does not impose overhead to intermediate routers. It is achieved by splitting headers into two flavours: the headers intermediate routers need to examine and the headers the end nodes will examine. This also eases hardware acceleration for IPv6 routers.
6. Smooth transition from IPv4: there were number of transition considerations made during the IPv6 discussions. Also, there are large numbers of transition mechanisms available. You can pick the most suitable one for your site.
7. Follows the key design principles of IPv4. IPv4 was a very successful design, as proven by the ultra large-scale global deployment. IPv6 is new version of IP, and it follows many of the design features that made IPv4 very successful. This will also allow smooth transition from IPv4 to IPv6.

2.2. A General Comparison Between IPv4 and IPv6

Number IPv6 is not meant to be a large step away from IPv4. For this reason, changes in IPv6 can be grouped primarily into many categories. The first area of improvement in IPv6 is expanded routing and addressing capabilities. IPv6 increases the address size from 32 to 128 bits; this is four times as large as IPv4. This expansion supports a much larger number of addressable nodes and should accommodate all reasonable scenarios of future growth. There is also support in IPv6 for simpler auto-configuration of addresses which will help motivate emerging markets to adopt the protocol [20, 22].

A next area of improvement in IPv6 is header format simplification. Although IPv6 addresses are four times as long as IPv4 address, the IPv6 header is only twice the size of the IPv4 header. Some of the IPv4 header fields have been dropped or made optional, decreasing overhead and bandwidth cost. Also, IPv6 includes improved support for options. These options are placed in extension headers which are located between the IPv6 header and the transport layer header. These extension headers can be of arbitrary length and the total amount of options in a packet can be greater than the 40 bytes allowed by IPv4.

IPv6 defines six extension headers. The routing header is used for extended routing similar to IPv4 loose source route. The fragmentation header is used for message fragmentation and reassembly. The authentication header is used for security features like integrity and authentication. The encapsulation header is used for message privacy and confidentiality. The hop-by-hop options header is used for special options that require hop by hop processing. Finally, the destination options header contains optional information that is to be examined by the destination node.

IPv6 header extensions allow for several advantages; forwarding is more efficient, less limitation exists on the length of options in IPv6, and greater flexibility exists for introducing new options in the future. This will be highly important as the Internet evolves to meet the demands of the changing markets of the future.

IPv6 also includes quality-of-service capabilities that were not addressed effectively by IPv4. The Flow Label and Priority fields of the IPv6 header can be used to identify packets which need special handling by routers, such as real-time and multi-media applications. This capability is increasingly important as more applications are being developed that require consistent throughput [19, 24].

Finally, IPv6 includes security capabilities which provide support for authentication, data integrity, and confidentiality. IPv6 includes two mechanisms which address the lack of effective privacy and authentication mechanisms in IPv4: the Authentication header and the Encapsulation header. These mechanisms can be used individually or together to insure varying levels of security.

2.3. Header Format

The IPv6 header format is greatly simplified in comparison to the IPv4 format. This is due to the removal of several fields and the addition of the IPv6 extension headers [12, 13]. Figures 1 and 2 represent the structures of the headers formats for IPv6 and IPv4, respectively.

1. Version (4)	**2.** Traffic Class (4)	**3.** Flow label (24)
4. Payload Length (16)	**4.** Next Header (8)	**5.** Hop Limit (8)
7. Source IPv6 Address (128)		
8. Destination IPv6 Address (128)		

Figure 1. The IPv6 header format.

1. Version (4)	**2.** THL(4)	**3.** Type of Service (8)	**4.** Total L. (16)
5. Identification (16)	**6.** Flags (3)	**7.** Fragment Offset (13)	
8. Time to Live (8)	**9.** Protocol (8)	**10.** Header Checksum (16)	
11. Source IPv4 Address (32)			
12. Destination IPv4 Address (32)			
13. Options + Padding			

Figure 2. The IPv4 header format.

The 'version' field is a 4-bit field that designates the internet protocol version number of the packet. This field is common to IPv4 and IPv6. In the case of IPV4, the 'version' field will be equal to 4. While in the case of IPv6, the 'version' field will be equal to 6. This field is important for routing since IPv6 messages must be handled differently than IPv4 messages. The 4-bit 'priority' field enables a source to specify a desired packet delivery priority with respect to other packets from the same source. This field has two ranges of priority; one range is used for real-time traffic that is sent at a constant rate and does not respond to congestion, and the other range is used for traffic that does respond to congestion. The 24-bit 'flow label' field is used to label packets for which special handling by the IPv6 routers is requested. This special handling is related to real-time service and other non-default quality of service issues. The 'payload length' field contains a 16-bit unsigned integer. This field is common to IPv4 and IPv6. However, in IPv4, this field is called 'Total Length' field. This field specifies the size of the packet following the header in octets. The 'Next Header' field serves as an 8-bit selector. This field specifies the type of extension header that immediately follows the IPv6 header. The values used in this field are the same as the IPv4 Protocol field. The 'Hop Limit' field contains an 8-bit unsigned integer. This field is common to IPv4 and IPv6. However, in IPv4 this field is called 'Time to Live' field. This field is decremented each time the packet is forwarded. If a packet with hop limit zero is encountered, it is discarded. The 128-bit 'source address' field contains the address of the initial source of the packet. The 128-bit 'destination address' field contains the address of the recipient of the packet. The recipient may not be the final recipient of the packet if the routing header is present. These fields are present in IPv4, but they are only 32 bits long. This change in size is due to the changes in addressing in IPv6.

2.4. Transition Strategies

The introduction of IPv6 technology offers many benefits over the existing IPv4, but it is important to continue of applying both technologies until the recent one cover all applications. A number of strategies have been developed for managing this transition from

IPv4 to IPv6, see [1, 4, 12, 18] for surveys and overviews about these strategies. Herein we will explain two of the most common strategies, that is, dual stack backbone and IPv6 over IPv4 tunneling. The most straightforward way to introduce IPv6-capable nodes is the dual stack approach, where IPv6 nodes also have a complete IPv4 implementation as well. In dual-stack backbone deployment, all routers in the network maintain both IPv4 and IPv6 protocol stacks. Dual Stack routing is the preferred deployment strategy for network infrastructures with a mixture of IPv4 and IPv6 applications that require both protocols. This strategy introduced the following disadvantages [10, 23]:

1. All routers must be upgraded to IPv6.
2. Routers require dual addressing scheme.
3. Dual management routing protocols.
4. Sufficient memory for both the IPv4 and IPv6 routing tables.

While in the IPv6 over IPv4 tunneling strategy, the IPv6 node on the sending side of the tunnel takes the entire IPv6 packet, and puts it in the data field of an IPv4 packet. This IPv4 packet is then addressed to the IPv6 node on the receiving side of the tunnel and sent to the first node in the tunnel. IPv6 over IPv4 tunneling encapsulates IPv6 traffic within IPv4 packets, to be sent over an IPv4 backbone. This enables island IPv6 end systems and routers to communicate through an existing IPv4 infrastructure. A variety of tunneling mechanisms are available for deploying IPv6, such as manually configured tunnels, generic routing encapsulation, IPv4-compatible tunnels, 6-over-4 tunnels, intra-site automatic tunnel addressing protocol and multi-protocol label switching [9, 25]. In addition, Chen *et al*. [7, 8] proposed an IPv4/IPv6 transition Mechanism for SIP-based VOIP applications. In their research, they utilized a SIPv4 UA with SLT to communicate with a SIPv6 UA through an open source IPv6 SIP server. On the other hand, they used the SIPv6 UA to communicate with various commercial software and hardware-based SIP UAs and PSTN gateways to exam the functions of the SIPv6 translator.

3. The Proposed Algorithms

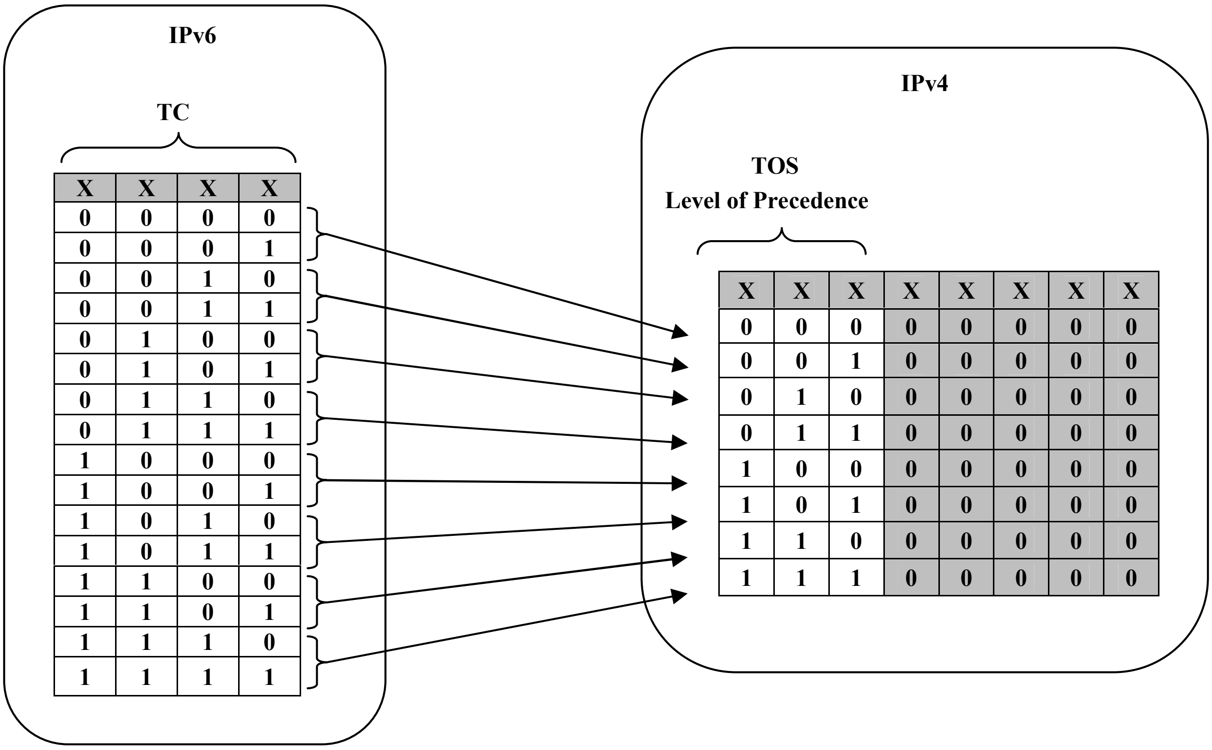
When the Internet environment started in applying the IPv6 technology, two different sets of problems are raised. The first one is related to having IPv6 communications among two or more IPv6 islands isolated in the IPv4 world. The solutions of this set of problems are generally based on dual stack routers and IPv6/IPv4 tunneling approach. The second set is related to the establishment of communications between the existing IPv4 world and the new IPv6 world. The solutions of this set of problems rely on dual stack techniques, application level gateways, Network Address Translation (NAT) technology, or on temporary allocation of IPv4 address and IPv4/IPv6 tunneling. The proposed algorithms depend on understanding the received datagram, capturing the header, identifying the header, verification the header, transformation the datagram to the destination environment, and then transmitting the datagram to the destination address. Furthermore, they are based on the bi-directional operation that leads to converting the received datagram to the destination environment. These proposed algorithms deal with both the deep understanding and analyses of the headers of both technologies (IPv4 and IPv6) and the methods for managing the transformation between these technologies. Moreover, they handle the header processing transition from IPv4 to IPv6 and vice versa. However, this process depends on the bi-directional identification and recognition processes of the two distinct headers. Thus, they depends on the Bi-Directional Intelligent Processing System (BDIPS) [3] and the Bi-Directional Mapping System (BDMS) [2]. The proposed header processing algorithms deal with the in-depth understanding of the two technologies of the header fields, that is, from IPv6 to IPv4 and vice versa.

3.1. Transition from IPv6 Header to IPv4 Header

When make a transition from IPv6 header to IPv4 header, it is necessary to store 0100 in the ‘version’ field of the IPv4 header to indicate that the used IP is of version 4. The contents of the ‘Traffic Class’ (TC) field of IPv6 Header will be mapped to the ‘type of service’ field of IPv4 header, taking into account that the size of TC is 4 bits, for the details of this mapping see Figure 3. In IPv4, the size of the ‘identification’ field is 16 bits, while the size of the ‘flow label’ filed in IPv6 header is 24 bits. However, when copying the contents of the ‘flow label’ filed to the ‘identification’ filed an overflow may be occurred. Thus, to solve this problem, we have used a counter called Packet Counter (PC) which should be initialized to zero when a new packet just started or if the destination address , source address, or the next header have been changed. If this is not the case, that is, no new packet started and the destination address, source address, and the next header have not been changed, then the PC should be increased by one. Later on, the PC will be stored in the ‘Identification’ field of the IPv4 header.

The contents of the rest of the IPv4 header’s field will be based on the contents of the ‘Next Header’ (NH) filed of IPv6. However, if the NH filed has 6 or 17, then save PC in identification field in IPv4 header, copy the contents of NH field to the ‘protocol’ field in IPv4 header, map the IPv6 ‘destination address’ to IPv4 ‘destination address’, map the IPv6 ‘source address’ to IPv4 ‘source address’, use the BDMS as in [25] to perform the mapping of the destination and source addresses. In addition, copy the contents of hope limit field to the TTL field in IPv4 header, compute the header length and save it in the Hdr length in IPv4, compute the payload length, sum the Hdr length and payload length and save the result in the total length filed in IPv4, compute the header checksum, and save the result in the header checksum field in IPv4 header.

Furthermore, if the NH is 43, then copy the EH to the option field in IPv4 header and check the NH. Whereas, if the NH is 44, process the fragment EH to obtain the flags and fragmentation offset values, copy the two values to the flags and fragmentation offset fields in IPv4 header, and check the NH.



X	X	X	X
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

X	X	X	X	X	X	X	X
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

Figure 3. Mapping the contents of the TC to TOS.

For more details on the transition from IPv6 header to IPv4 header, see Figure 4 which illustrates the algorithm of this transition process.

3.2. Transition from IPv4 Header to IPv6 Header

Figure 5 shows the detailed algorithm of the transition process from IPv4 Header to IPv6 Header. However, Some fields of the IPv6 Header will contain the same values from the corresponding fields of the IPv6 Header, that is, the fields which will be copied from IPv4 header to IPv6 Header without any change are: 'Identification' to 'Flow Label', and 'Time to Live' to 'Hop Limit'. Furthermore, the 'Version' field of the IPv6 Header will contain 0110 to denote that the IP version is 6. The contents of the TOS field of IPv4 Header should be mapped to the TC field IPv6 Header, see Figure 6 for the details of this mapping. The 'Source' and 'Destination' addresses fields of IPv4 will be mapped to the 'Source' and 'Destination' addresses fields of IPv6 using the Bi-Directional Mapping System (BDMS), see [25] for the details of the Bi-Directional Mapping System BDMS. The payload length will be computed by subtracting the contents of the 'header length' field of IPv4 header from the contents of the 'total length' field of IPv4, then, the resulted value will be saved in the 'payload length' fields in IPv6 header. The contents of the 'fragment extension header' of IPv6 will be based on the contents of the 'fragment offset' field of IPv4. However, if the 'fragment offset' field of IPv4 is not equal to zero, then copy the contents of the 'flags' and 'fragment offset' fields of IPv4 header to the corresponding fields in the 'fragment extension header' of IPv6. Finally, if the 'protocol' field of the IPv4 header contains six, then we should take in to account that the used protocol is the TCP; otherwise, the protocol is the UDP.

4. Conclusions

Currently the number of available IPv6 applications is very limited compared with huge applications of IPv4; this referred to as an island in the ocean. Therefore the transition algorithms are the best options for transition from IPv4 to IPv6 and vice versa until every host or router is converted to IPv6. The scope of this paper exceeds the encapsulation and tunneling which are nowadays more suitable ways to perform transformation and adaptation between IPv4 and IPv6. However, this paper concentrates on finding an adaptive method for transition between these two versions. The proposed algorithms deal with the intelligent method of transformation and adaptation between IPv4 and IPv6 that called BDIPS. Furthermore, this paper constructs novel algorithms that depend on understanding of the two environment of transmission, that is, received the source packet then converting the information header to be adaptable to the destination end. They are simple and easy to implement as well as they are very efficient and intelligent. In addition, they reduce the packet size effectively rather than encapsulation – which enlarges the packet size due to additional header(s) – and thus reduce the overall transmission time.

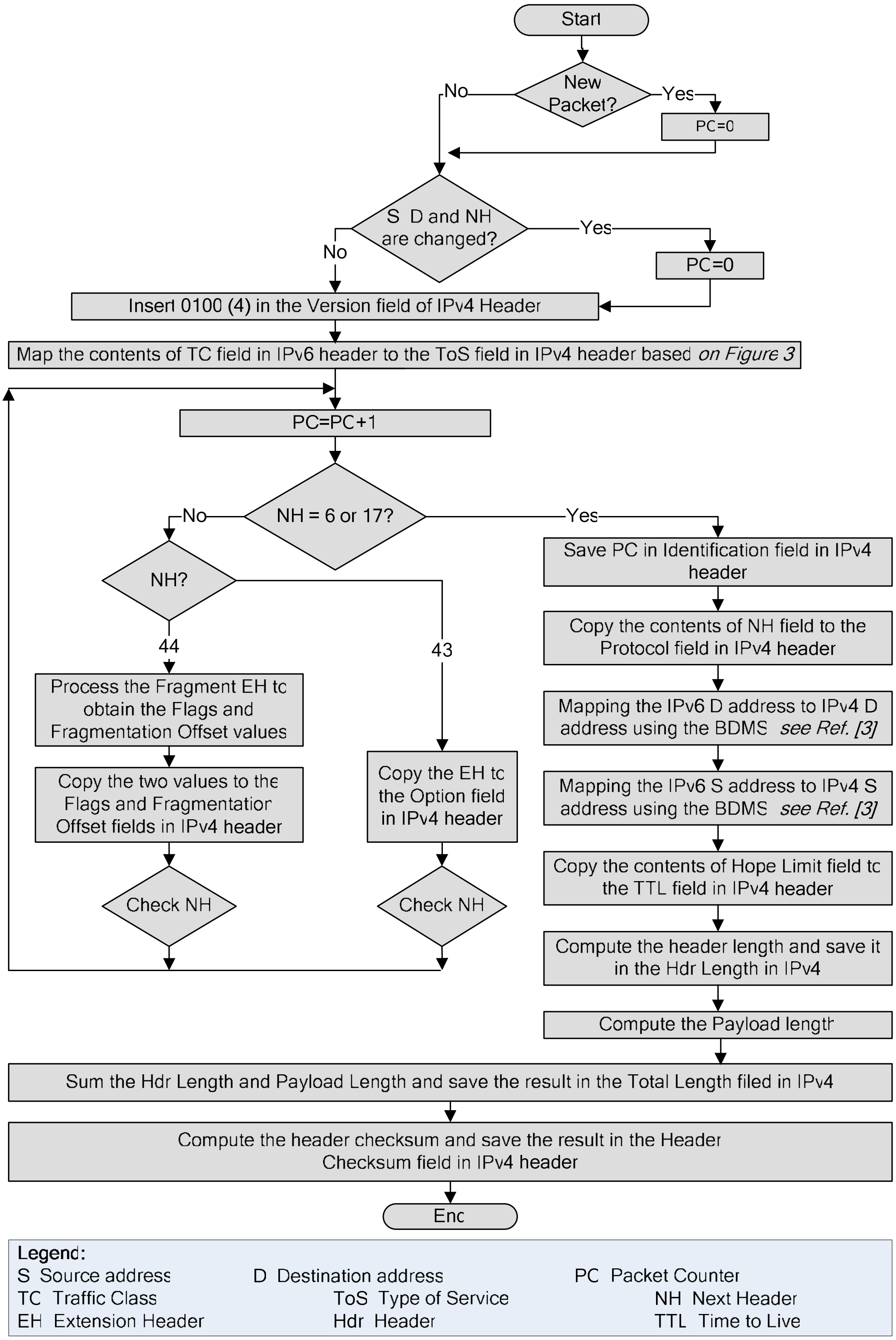

Starl
New Packet?
No
Yes
PC=0
S D and NH are changed?
No
Yes
PC=0
Inserl 0100 (4) in the Version field of IPv4 Headei
Map the contents of TC field in IPv6 header to the ToS field in IPv4 header based on Figure 3
PC=PC+1
NH = 6 or 17?
No
Yes
NH?
44
43
Process the Fragment EH tc obtain the Flags and Fragmentation Offset values
Copy the two values to the Flags and Fragmentatior Offset fields in IPv4 header
Check NH
Copy the EH tc the Option field in IPv4 header
Check NH
Save PC in Identification field in IPv4 header
Copy the contents of NH field to the Protocol field in IPv4 header
Mapping the IPv6 D address to IPv4 D address using the BDMS see Ref. [3]
Mapping the IPv6 S address to IPv4 S address using the BDMS see Ref. [3]
Copy the contents of Hope Limit field tc the TTL field in IPv4 header
Compute the header length and save it in the Hdr Length in IPv4
Compute the Payload lengtr
Sum the Hdr Length and Payload Length and save the result in the Total Length filed in IPv4
Compute the header checksum and save the result in the Headei Checksum field in IPv4 header
Enc
Legend:
S Source address
D Destination address
PC Packet Counter
TC Traffic Class
ToS Type of Service
NH Next Header
EH Extension Header
Hdr Header
TTL Time to Live

Figure 4. The algorithm of the header processing transition from the IPv6 to IPv4.

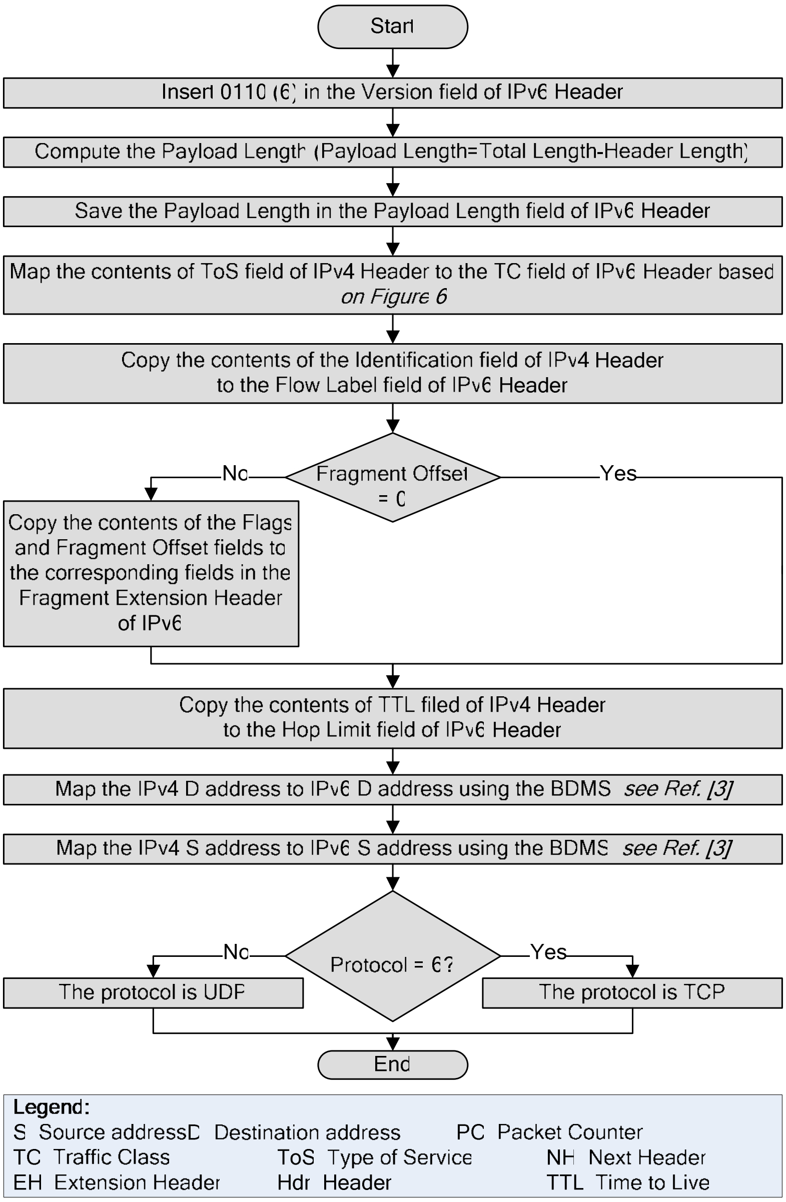
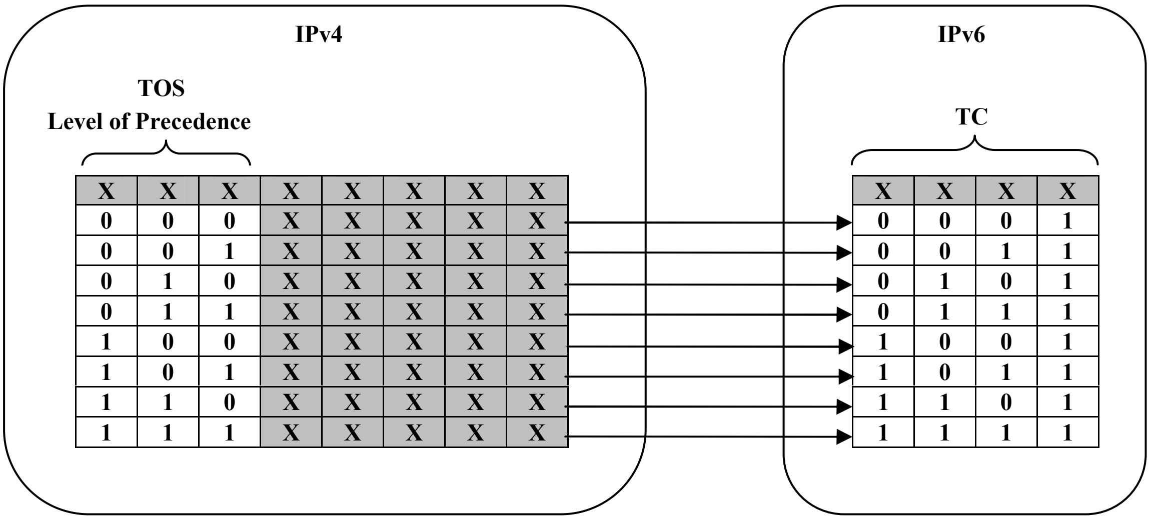

Start
Insert 0110 (6) in the Version field of IPv6 Header
Compute the Payload Length (Payload Length=Total Length-Header Length)
Save the Payload Length in the Payload Length field of IPv6 Header
Map the contents of ToS field of IPv4 Header to the TC field of IPv6 Header based *on Figure 6*
Copy the contents of the Identification field of IPv4 Header to the Flow Label field of IPv6 Header
Fragment Offset = 0
No
Yes
Copy the contents of the Flags and Fragment Offset fields to the corresponding fields in the Fragment Extension Header of IPv6
Copy the contents of TTL filed of IPv4 Header to the Hop Limit field of IPv6 Header
Map the IPv4 D address to IPv6 D address using the BDMS *see Ref. [3]*
Map the IPv4 S address to IPv6 S address using the BDMS *see Ref. [3]*
Protocol = 6?
No
Yes
The protocol is UDP
The protocol is TCP
End
Legend:
S Source address
D Destination address
PC Packet Counter
TC Traffic Class
ToS Type of Service
NH Next Header
EH Extension Header
Hdr Header
TTL Time to Live

Figure 5. The algorithm of the header processing transition from the IPv4 to IPv6.



X	X	X	X	X	X	X	X
0	0	0	X	X	X	X	X
0	0	1	X	X	X	X	X
0	1	0	X	X	X	X	X
0	1	1	X	X	X	X	X
1	0	0	X	X	X	X	X
1	0	1	X	X	X	X	X
1	1	0	X	X	X	X	X
1	1	1	X	X	X	X	X

X	X	X	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Figure 6. Mapping the contents of the TOS to TC.

References

[1] Afifi H. and Toutain L., "Methods for IPv4-IPv6 Transition," *in Proceedings of IEEE International Symposium on Computers and Communications (SCC'08)*, Egypt, pp. 478-484, 2008.

[2] Alja'afreh R., Mellor J., Kamala M., and Kasasbeh B., "Bi-Directional Mapping System as a New IPv4/IPv6 Transition Mechanism," *in Proceedings of the 10th International Conference on Computer Modeling and Simulation (ICCMS*'08), UK, pp. 40-45, 2008.

[3] Alja'afreh R., Mellor J., Kamala M., Kasasbeh B., and Al-Ani M., "A Novel IPv4/IPv6 Mechanism which Supports Transition Connections," *in Proceedings of the 8th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet'07)*, UK, 2007.

[4] Bi J., Wu J., and Leng X., "IPV4/IPv6 Transition Technologies and Univer6 Architecture," *International Journal of Computer Science and Network Security*, vol. 7, no. 1, pp. 232-243, 2007.

[5] Bradaner S. and Mankin A., "The Recommendation for the IP Next Generation Protocol," online: http://www.ietf.org/rfc/rfc1752.txt, accessed on May 15, 2008.

[6] Cannon K. and Caudle K., *CCNA Guide to Cisco Networking*, Thomson Course Technology, Toronto, Canada, 2004.

[7] Chen W., Lin Y., and Pang A., "An IPv4-IPv6 Transition Mechanism for SIP Overlay Network in UMTS All-IP Environment," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 11, pp. 2152-2160, 2005.

[8] Chen W., Su C., and Weng J., "Development of IPv4-IPv6 Translation Mechanism for SIP-based VOIP Applications," *in Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)*, vol. 2, Taiwan, pp. 819-823, 2005.

[9] Cisco., "Implementing Tunneling for IPv6," IOS IPv6 Configuration Guide, Release 12.4, online: http://www.cisco.com/en/US/docs/ios/12_2t/ipv6/SA_tunv6_ps6350_TSD_Products_Configuration_Guide_Chapter.html, accessed on May 15, 2008.

[10] Deering S. and Hinden R., "Internet Protocol, Version 6 (IPv6) Specification," December 1998, online: http://xml.resource.org/public/rfc/html/rfc2460.html, accessed on May 15, 2008.

[11] Gallo M. and Hancock W., *Computer Communications and Networking Technologies*, Thomson Learning, 2002.

[12] Govil J., Kaur N., and Kaur H., "An Examination of IPv4 and IPv6 Networks: Constraints and Various Transition Mechanisms," *in Proceedings of the IEEE Region 3 Southeast Conference*, Huntsville, Alabama, USA, pp. 178-185, 2008.

[13] Halsall F., *Multimedia Communications Applications, Networks, Protocols and Standards*, Addison Wesley, 2001.

[14] Halsall F., *Computer Networking and the Internet*, Addison Wesley, 2005.

[15] Hogg S., "Internet Protocol Version 6: The next generation Protocol," online: http://www.gtri.com/docs/IPv6%20%20the%20Next%20Generation% 20Protocol%20v1-1.pdf, accessed on May 15, 2008.

[16] Kurose J. and Ross K., *Computer Networking: a Top-Down Approach*, Pearson Education, 2003.

[17] Lee D. and Stewart E., "Internet Protocol Version 6 (IPv6) Conformance and Performance Testing," online: http://www.ixiacom.com/library/white_papers/wp_display.php?skey=ipv6, accessed on May 15, 2008.

[18] Mackay M., Edwards C., Dunmore M., Chown T., and Carvalho G., "A Scenario-Based Review of IPv6 Transition Tools," *IEEE Internet Computing*, vol. 7, no. 3, pp. 27-35, 2003.

[19] Mark J. and Zhuang W., *Wireless Communications and Networking*, Prentice Hall, 2003.

[20] Oliver N. and Oliver V., *Computer Networks Principle, Technologies and Protocols for Network Design*, John Wiley and Sons, 2006.

[21] Roese J., "Internet Protocol Version6," online: http://www.enterasys.com/company/literature/ipv6-wp.pdf, accessed on May 15, 2008.

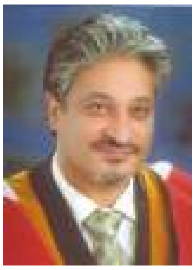
[22] Stallings W., Wireless *Communications and Networks*, Prentice Hall, 2002.

[23] Stallings W., *Computer Networking with Internet Protocols and Technology*, Prentice Hall, 2004.

[24] Tomsho G., Tittel E., and Johnson D., *Guide to Networking Essentials*, Thomson Course Technology, 2004.

[25] Zheng Y. and Akhtar S., *Networks for Computer Scientists and Engineers*, Oxford University Press, 2002.

Basil Al-Kasasbeh received the MSc and PhD degrees in networks, systems and communication devices from Siberian State University of Telecommunications and Informatics, Novosibirsk in 1994 and 2002, respectively. Currenty, he is an assistant professor at the Faculty of Information Technology in the Applied Science University in Jordan. His research interests include mobile and wireless systems, mobile IP, and IPV6.



Rafa Al-Qutaish received the BSc degree in computer science from Yarmouk University, Jordan in 1993, the MSc degree in software engineering from University of Putra, Malaysia in 1998, and PhD degree in software engineering from the School of Higher Technology University of Québec, Canada in 2007. Currenty, he is an assistant professor at the Department of Software Engineering in Alzaytoonah University of Jordan. His research interests include communication software engineering, software measurements, software engineering standardizations, software quality engineering, and applied artificial intelligence.

Mohammad Muhairat received the MS degree in computer engineering from Kharkov State Technical University of Radio Electronics, Ukraine in 1997, and the PhD degree in computer engineering from Kharkov National University of Radio Electronics, Ukraine in 2002. His research interests are in software engineering fields, such as, requirements specification, design & testing, UML notations, and formal methods for requirements specification & design.