NETWORK OPTIMIZATION OF IPV6 NETWORKS USING TUNNEL HEADER COMPRESSION

Dipti Chauhan[1], Sanjay Sharma[2]

[1]*Research Scholar, Mathematics & Computer Applications, Maulana Azad National Institute of Technology, M.P., India*
[2]*Professor, Mathematics & Computer Applications, Maulana Azad National Institute of Technology, M.P., India*

Abstract
IPv6 is the successor internet protocol which will eventually replace IPv4. These two protocols are not compatible with each other and it will take time to migrate towards IPv6, until then both the protocols need to coexist for a long time. The main overhead involved with both the protocols is header length of 20 bytes in case of IPv4 and of 40 bytes in case of IPv6. This overhead will affect the network performance specially over tunneling mechanism where one header is encapsulated inside another. Tunneling is widely deployed over the network for various purposes like network security, mobility and transition mechanism. Header compression can be applied to compress the excess protocol headers to improve the performance of network. In this paper we want to use header compression in context of 6 to 4 tunneling transition. Using header compression over 6 to 4 tunnels would result in better response times reduced packet size and reduced packet losses. We want to simulate this algorithm using EXata Cyber 1.1 simulator.

Keywords: Header Compression, IPv4, IPv6, ROHC, 6 to 4 tunneling.

--***---

1. INTRODUCTION

IPv4 is the most dominating protocol for both wired and wireless medium, and this choice is leading towards the convergence of communication technology. IPv4 address is 32 bits which can address approximately 4.3 billion devices [1]. This was a huge number during its inception, but with the advent of new services and applications, this address pool is expected to run-out in 2011 [2]. The problem of IP address exhaustion has observed in the early 90's and to overcome from this problem different solution like Network Address Translation (NAT), Dynamic Host Configuration Protocol (DHCP) and Classless Inter-Domain Routing (CIDR) [3].Despite of these short term solutions it is impossible to deal with the problem of IPv4 exhaustion. IPv6 is the next generation internet protocol which will eventually replace IPv4. IPv6 uses 128 bits and can address up to 2^{128} devices which is 2^{94} times more than IPv4. IPv6 offers several benefits like larger address space, improved header format, Built -in security, Multicast support, Mobility and Better support for QoS. Despite of several advantages of IPv6, both the protocols are incompatible and the adoption of IPv6 is very slow. Different transition techniques have been proposed for the smooth interoperation of the two protocols.

The main issue involved with IPv4 and IPv6 is Header overhead. The overhead involved with IP is header length of 20 bytes in case of IPv4 and that of 40 bytes in case of IPv6. This overhead will eventually increases in case of Tunneling where one header is encapsulated inside another header. This overhead would decrease the overall network performance, especially over wireless links where resources are scarce. These headers are of very less importance over

hop to hop but serve extremely important purpose over end to end links [4]. So it's better to send compressed header at one end and decompress it at the other end. Header Compression is a method of compressing excess protocol headers while transporting over a link and uncompressing it at the other end of the link [5]. This compression can be done because most of the fields within a header consist of information which remains static throughout transmission or it may vary in a specific pattern. These header fields can be categorized as static, dynamic, static known, and inferred [6]. Figure 1 show the header fields for IPv6 and IPv4 Headers along with their classification. So in order to save network resources it is beneficial to send the compressed packet rather to send uncompressed one.

Fig: 1

Bandwidth is the most costly resource in cellular links. Processing power is very cheap in comparison. Implementation or computational simplicity of a header compression scheme is therefore of less importance than its compression ratio and robustness [7]. The compression gains which can be achieved through header compression are much more valuable over which we can sacrifice our computational complexity.

Rest of the paper is structured as follows: section II discusses about the Tunneling Mechanism specifically about 6 to 4 tunneling. Section III describes the about the existing header compression schemes. Section IV describes the related work in the direction of header compression. Section V describes the proposed methodology. Section VI concludes the paper.

2. TUNNELING

Tunneling is the transition method used when we want to send an IPv6 packet from an IPv6 source to IPv6 destination and the backbone network is based upon IPv4. Here in this case tunneling mechanism is used. Tunnels are used to carry an IPv6 packet through an IPv4 infrastructure or vice versa. This is the transition period and tunnels are the most widely used transition mechanism for the smooth interoperation of the two protocols. Different types of tunnels are deployed for different scenarios like host to host tunnels, router to router tunnels, host to router and router to host tunnels. Classification of tunneling mechanism is usually done by the encapsulating node which determines the address at the end of the tunnel. In Router-to-Router or Host-to-Router tunnels, the IPv6 datagram is tunneled to a router while in host-to-host or router-to-host tunnels, the IPv6 datagram is to its final destination. In tunneling at the source, an IPv6 packet is encapsulated inside an IPv4 packet, i.e. IPv4 is the carrier for IPv6 packet and at the destination the IPv4 header is striped off and the IPv6 packet is delivered to the intended destination. In the current scenario a large variety of tunnels are being used over network in several contexts like IP transition, mobility support, and security.

6 to 4 is an automatic tunneling mechanism that allows connecting IPv6 networks over IPv4 infrastructure to another IPv6 network [9]. Border router is used to configure automatic 6 to 4 tunnels and the tunnel end point is determined by the IPv4 address of the border router which is extracted from the IPv6 address prefix 2002::/16. This prefix is reserved for 6 to 4 address and the format is 2002: border-router-IPv4-address ::/48. To configure 6to4 tunneling, a tunnel interface is created on each dual-stack edge router and three key components relevant to 6to4 configuration are: The tunnel mode (6to4), The tunnel source (IPv4 interface or address), The 6to4 IPv6 address (within 2002::/16). Figure 2 shows the scenario for 6 to 4 tunnel. Here Host A on IPv6 LAN wants to communicate to another Host B on IPv6 LAN via IPv4 backbone.
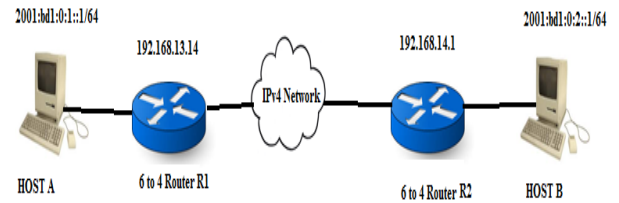

Fig: 2

3. EXISTING HEADER COMPRESSION SCHEMES

Different Header Compression Techniques have been studied and proposed from the past over the internet protocols. In 1990 Van Jacobson invented the original header compression scheme VJHC [10] is used to compress TCP/IP packets over slow-speed serial links. This mechanism can compress 40 bytes of IPv4/TCP header up to 3-6 bytes. The main issue with VJHC is that it does not support UDP and IPv6 Headers, so it is not considered as a good option for wireless channels. It also disables the TCP fasts retransmit and fast recovery algorithms in the presence of errors. IP header Compression (IPHC) [11] augments VJHC, and can compress IPv4, IPv6, AH, Minimal Encapsulation header, Tunnelled IP headers, TCP , UDP, ESP headers as well. It uses TWICE algorithm to compute TCP checksum to determine the context information. Using this the TCP fast retransmit and fast recovery algorithms are enabled. IPHC can reduce the IP header up to 2 bytes for non TCP session and 4 bytes for TCP session. The CRTP [12] compressing IP/UDP/RTP header compression is a detailed specification for RTP compression. Studies [13, 14] shows that the performance of these header compression techniques is quite poor in wireless links, as wireless links subject to high bit error rate (BER), high round trip time (RTT) and error prone. So there is a need of next generation header compression technique that can be used efficiently over wireless links. Robust Header Compression ROHC [7] is designed to improve the efficiency and robustness over wireless links, and can compress different protocol headers. ROHC can compress IP/UDP/RTP headers to just over one byte, even in the presence of severe channel impairments. This compression scheme can also compress IP/UDP and IP/ESP packet flows.

4. RELATED WORK

[15] Discloses a method for compressing 6 to 4 tunnels, by compressing IPv4 and IPv6 headers by downsizing and sharing the fields. Using this reduce the header length in transmission, is able to obtain MTU value approximately equivalent to non tunnel encapsulation, and improves transmission efficiency. [16] Proposed a new approach using packet aggregation to cooperate with the header compression scheme, over wireless mesh networks. Using this approach over VoIP application can decrease significantly the packet loss, impacting positively on the speech quality. [17] Proposed a new protocol TuCP for tunneling over IP networks. This protocol is used for compression the outer tunnel headers in conjunction with

ROHC for compressing inner headers. [18] Suggested two approaches to optimize the use of ROHC profile in NEMO (Network Mobility) networks. In the first approach it uses only the IP profile of ROHC as IP is the largest header. This approach suggests using a smaller number of profiles, and results show that using ROHC compression the VoIP and video flows from 40/60 bytes for IPv4/IPv6 into 6 bytes and TCP flows are compressed from 52/72 bytes into 8 bytes. In the second approach called ROHC Selected-Profile Compression. In this approach, the ROHC profiles of various IP flows are classified on the basis of resources used at MR and HA. Results show that the second approach is more effective in terms of bandwidth savings than first approach. [19] Highlights a header compression algorithm that works end to end rather than node to node. No header compression algorithm support end-to-end compression for multimedia flows over ad hoc wireless networks. This could be a cost effective technique to raise the efficiency and resource utilization in end-to-end wireless networks. To measure the performance of SEEHOC algorithm it proposed an RTP/UDP/IP compressed header format over wired and wireless networks. Results prove that this algorithm improves the resource utilization without introducing overheads and improves the overall performance of the network. [20] Addresses the de-synchronization issues which lead to poor network performance. The authors argue that there is no need to preserve the header context while compressing or decompressing with ARHC integration in MANETs. If a packet is lost or corrupted, it will not affect the subsequent packets due to independence of successive compressed headers.

5. PROPOSED METHODOLOGY

Application layer along with the transport layer protocol either TCP or UDP, along with network layer protocol IP consists of a packet header. Most of the information in the packet remains static, or vary in a specific pattern or can be obtained from other fields. So it's better to compress such information and reduce the size of packet, so that the link can be efficiently utilized. Here in our proposed approach a sub layer is added between Network layer and Link layer. The task of this sub layer is to add profile id (p_id) and context id (c_id), in the form of header information. Profile Id will be used for compression and decompression pattern. Context Id is used to identify multiple flows in the tunnel. Different profiles are used like IPv6 only profile, IPv6/UDP, IPv6/TCP. Initially few packets are send uncompressed with p_id and c_id. Then this sub layer fetches uncompressed packet from the router queue at the network layer, apply compression algorithm compress the packet according to the particular profile and place it to the router queue.

Figure 3 shows the position of sub layer along with the compressed header. At the other end of the tunnel, this compressed packet is decompressed according to the profile id mentioned in the compressed header.
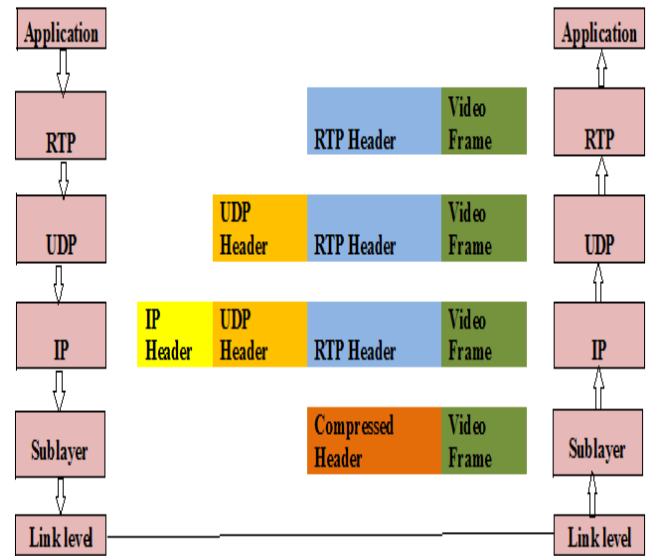

Fig: 3

The following figure:4 depicts the scenario for tunnel header compression. Here IPv6 sender wants to sends an IPv6 packet to the IPv6 destination via an IPv4 backbone. Router A and B are dual stack routers where compression and decompression takes place.
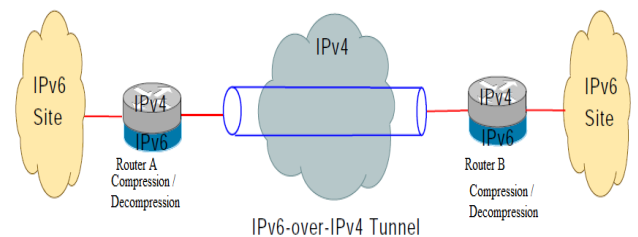

Fig: 4

Algorithm Steps:

Step1: Router A Receives the IPv6 packet:
Step1.1: Determines the next hop on the basis of routing table entries.
Step1.2: Here next hop will be tunnels address.
Step 2: Here at this point IPv6 packet is encapsulated inside IPv4 packet.
Step 2.1: IPv4 Source Address: Tunnels source address.
Step 2.2: IPv4 destination Address: Tunnels end point.
Step 3: Initially send first few packets uncompressed to maintain context and the static information.
Step 3.1: Once context is established then go to step 4.
Step 4: At this point compression algorithm works:
Step 4.1: Before encapsulation, IPv6 header along with the upper layer headers is compressed;
Step 4.2: Now this compressed packet will be e0ncapsulated inside the IPv4 packet.
Step 4.3: This packets will be sent through the tunnel.
Step 5: At the tunnel end point: i.e. at router B
Step 5.1: IPv4 header is striped off.
Step 5.2: The compressed header will be decompressed according to the specific profile.

Step 6: This IPv6 packet is routed onto the IPv6 LAN toward the destination address as specified in the IPv6 packet.

6. CONCLUSION

Tunneling is most widely used solution for the interoperation of the two protocols till the entire networks is switched towards IPv6. In this paper we have a new proposed header compression algorithm for tunneling over IPv6 networks for enabling the smooth transition of the Internet. The usage of tunnels over the network comes with several shortcomings including high header overhead, high bandwidth usage, packet reordering. Using header compression for tunnels over IPv6 network we would like to address the issue of high header overhead which would result in better response time, reduction in packet loss and decrease in packet header overhead. We would like to simulate my algorithm on EXata/Cyber simulator. The expected outcome of this algorithm would definitely result in improved throughput and packet delivery ratio and also decreased end-to-end delay and jitter.

REFERENCES

[1]. Marina Del Rey, California 90291 "INTERNET PROTOCOL, DARPA INTERNET PROGRAM, PROTOCOL SPECIFICATION", RFC 791.

[2]. W. Lehr (MIT), T. Vest (Eyeconomics.com), E. Lear (Cisco), "Running on Empty: the challenge of managing Internet addresses", Internet Assigned Numbers Authority File Version: Internet Address TPRC 10_21_08.doc.

[3]. Dipti Chauhan, Sanjay Sharma, "A Survey on Next Generation Internet Protocol: IPv6", International Journal of Electronics & Industrial Engineering (IJEEE), ISSN: 2301-380X, Volume 2, No. 2, June 2014, Pages: 125-128.

[4]. An introduction to IP header compression, white paper, www.effnet.com.

[5]. M. Degermark , B. Nordgren, S. Pink. Network Working Group: Request for Comments: 2507: IP Header Compression.

[6]. Ms.Devishree Naidu , Ms. Rakhi Tapadiya, Implementation of Header Compression in 3GPP LTE, 2009 Sixth International Conference on Information Technology: New Generations

[7]. C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannq L E. Jonsson, R. Hakenberg, T. Koren, K. Le, *2*. Liu, A. Martensson, A.Miyazaki, K Svmbro, T. Wiebke, T. Yosbimura, and H. Zheng, "Robust Header Compression (ROHC): Framework and four profiles: RTP, UDP,ESP, and uncompressed," RFC 3095, July 2001.

[8]. R. Gilligan, FreeGate Corp. E. Nordmark Sun Microsystems, RFC: 2893 Transition Mechanisms for IPv6 Hosts and Routers, Inc.August 2000.

[9]. B. Carpenter, K. Moore, Connection of IPv6 Domains via IPv4 Clouds, Network Working Group: Request for Comments: 3056, February 2001.

[10]. V. Jacobson, "Compressing TCP/IP Headers for Low-Speed Serial Links," RFC 1144, February 1990.

[11]. M. Degermark , B. Nordgren, S. Pink. Network Working Group: Request for Comments: 2507: IP Header Compression.

[12]. S. Casner and V. Jacobson. Compressing IP/UDP/RTP Headers for Low-Speed Serial Links. Internet Standards Track RFC 2508, IETF, 1999.

[13]. M. Degermark, H. Hannu, L. Jonsson, and K. Svanbro. Evaluation of CRTP Performance over cellular Radio Links. *IEEE Personal Communications*, 7(4):20–25, 2000.

[14]. Lars-Erik Jonsson, Mikael Degermark, Hans Hannu, and Krister Svanbro. Header Compression for IP-Telephony over Cellular Links, 1999. IETF 46, Ericsson Research Group Presentation.

[15]. Patent: Method and device for compressing 6to4 tunnel: Patent No: CN 101645823 B: Cao Zuo Qing , Huang Zhe, Sep 2, 2009

[16].Andrea Giordanna Nascimento, Edjair Mota, Saulo Queiroz, Leandro Galvao, Edson Nascimento: Towards an Efficient Header Compression Scheme to Improve VoIP over Wireless Mesh Networks ,IEEE Catalogue No: 978-1-4244-4671-1/09/$25.00 ©2009 IEEE, Multimedia and Computer Network Research Group – GRCM, Federal University of Amazonas.

[17]. Priyanka Rawat, Jean Marie Bonnin, and Laurent Toutain: Designing a Tunneling Header Compression (TuCP) for Tunneling over IP, IEEE ISWCS 2008.

[18]. Priyanka Rawat, Jean Marie Bonnin, and Ana Minaburo,: Optimising the use of Robust Header Compression profiles in NEMO Networks, 2008 IEEE DOI 10.1109/ICN.2008.35.

[19]. Maher Kaddoura and Steven Schneider: SEEHOC: Scalable and Robust End-to-End Header Compression Techniques for Wireless Ad Hoc Networks, Architecture Technology Corporation Eden Prairie MN USA. IEEE Catalogue No: 0-7803-8798-8/04/$20.00 ©2004 IEEE.

[20]. Hai Wang, Dawei Niu, Wendong Zhao, Yuan Yan, Gup Yan, "A Robust Header Compression Method for Ad hoc Network," in International conference on networking and services. ICNS, 2006.[4]. Reference 4

BIOGRAPHIES

Dipti Chauhan, completed her M.Tech degree from Barkatullah University Institute of Technology Bhopal in 2011. She is having a teaching experience of 6 years and currently she is working as a full time research scholar from MANIT, Bhopal. Her research interests include next generation networks and IPv6.

Sanjay Sharma, completed his Ph.D from Barkatullah university Bhopal in 2004, in the area of compressing large databases. He did his MCA from MANIT, Bhopal in 1990. He is also an IPv6 Certified Gold and Silver Network Engineer from IPv6 forum, University Sains Malaysia. He is having a teaching experience of 22 years, and currently he is Professor and Head of Department in MANIT Bhopal.