ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Enhancing the Efficiency of IPv6 Tunnelling Mechanism by using Header Compression over IPv6 Header

Dipti Chauhan[1], Sanjay Sharma[2]

Research Scholar, Department of Mathematics & Computer Applications, MANIT, Bhopal, India[1]

Professor, Department of Mathematics & Computer Applications, MANIT, Bhopal, India[2]

Abstract: Internet is growing at a tremendous rate and with this rate of growth it's not possible to sustain with the IPv4 and therefore the solely alternative is to adopt IPv6, the new internet protocol. Despite of numerous advantages that IPv6 offers over IPv4 the adoption rate of IPv6 by the users and is very slow. The main overhead involved with IPv6 protocol is header overhead of 40 bytes, and this overhead is even more when we are using tunneling mechanism, where one header is encapsulated inside the other. This overhead may have an effect on the performance particularly over wireless links where resources are scarce. In this paper we want to improve the efficiency of tunneling mechanism over IPv6 networks by using Header Compression technique. Here we are compressing the IPv6 Header of the packet as IPv6 header is of largest length and the results are analyzed on the basis of different packet sizes, different parameters like throughput, jitter, end-to-end delay and packet delivery ratio are calculated. These results are the compared with uncompressed network. Simulations are carried out over Qualnet 5.1 Simulator. These results show that the using header compression over IPv6 tunneling mechanism performing better than uncompressed network.

Keywords: Context id, Header Compression, Header Decompression, IPv4, IPv6, Profile id, Tunneling.

I. INTRODUCTION

This document is a template. An electronic copy can be downloaded from the conference website. For questions on paper guidelines, please contact the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website.

IPv6 [1] is the successor internet protocol, which offers several benefits and advantages over IPv4 [2]. Instead the adoption of IPv6 by the people is very slow, <1% part of the world population is using IPv6. The reason behind slow adoption is that both the protocols are not compatible with each other. A host or a router which supports only IPv4 does not forward and IPv6 packet, similarly IPv4 only hosts cannot communicate with IPv6 hosts & routers [3]. This incompatibility of the two protocols can break the internet connectivity which results in overall performance degradation.

There is still lot of work to be done till every network is switched to IPv6. In fact the reality is that IPv4 will be there for a long time and there is a lack of IPv6 devices. Various transition techniques have been developed to assist the migration to IPv6 like Dual stack, Header Translation and Tunneling [4]. Dual stack deals with maintain both the protocol stacks IPv4 as well as IPv6 on the devices. Header Translation deals with translating the headers through CG NAT (Carrier Gateway Network Address Translator) device. Tunneling deals with encapsulating IPv6 packet inside an IPv4 packet. In this paper we are dealing with tunneling techniques to assist the migration towards IPv6 networks.

1.1 Tunneling

Tunneling can be defined as encapsulating an IP packet inside another IP packet, by forwarding the packet to its destination through intermediate networks that do not support the packets protocol [5]. If an IPv6 host wants to send a packet to IPv6 host, but the underlying network is based on IPv4, then tunneling comes into the play. Here IPv6 packet is encapsulated inside an IPv4 packet and is send across the network at the destination this IPv4 header is striped off and the IPv6 packet is delivered to the intended destination. Tunneling techniques are used in various contexts like security, mobility and transition purposes. However the use of tunneling comes with various disadvantages like header overhead, packet reordering, in efficient routing, and Quality of service. Header Compression techniques can be used to improve the efficiency of tunneling mechanism. Here in this paper we are using tunneling technique and Header Compression is applied over IPv6 header which results in better performance of the network. Figure 1 demonstrates the tunneling concept where an IPv6 packet is encapsulated inside an IPv4 packet.

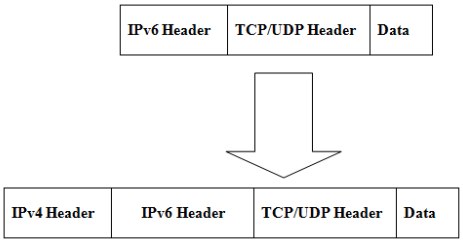

Figure 1: Tunneling

DOI 10.17148/IJARCCE.2015.44102

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

1.2 Header Compression

Header Compression is the process for compressing the excess protocols headers before transmitting over a link and decompressing it at the other end of the link [6]. It is possible to compress the protocol headers due to the redundancy in header fields of the same packet as well as consecutive packets of the same packet stream. Header can be compressed because most of the information in a header remains static or vary in a specific order. The information in the header serves very useful purpose for delivering a packet from source to destination, but is of less importance from one hop to another [7]. So to improve the network performance header compression can be applied over the packet header and resources can be efficiently utilized. Resource utilization is a major issue in networks, especially over wireless links where there is always scarcity of resources. Moreover in mobile networks like UMTS resources vary due to radio conditions. Figure 2 shows the header compression over IP/UDP/RTP header.

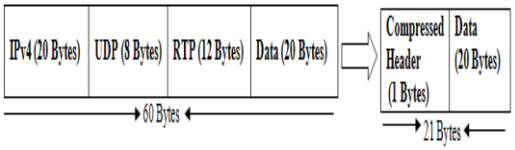

Figure 2: Header Compression over IP/UDP/RTP Header

Various header compression schemes have been proposed from the past over the internet, table: 1 shows the different standards for header compression along with their features [8].

Table 1: Different standards for header compression

IETF Standard	VJHC[9]	CTCP [10]	CRTP [11]	ROHC [12]
Developed	1990	1999	1999	2001
Headers	IPv4/TCP	IPv4(including options and fragments), IPv6 (including extension headers), AH, Minimal Encapsulation header, Tunnelled IP headers, TCP (including options), UDP, ESP	IPv4, IPv6 (including extension headers), AH, Minimal Encapsulation Header, tunneled IP headers, UDP, RTP	IPv4 (including options and fragments), IPv6 (including extension headers), AH, Minimal Encapsulation headers, GRE, Tunnelled IP headers, UDP, RTP,ESP
Headers Compressed to Minimum	3-6 Bytes	2 Bytes	2 Bytes	1 Byte
Link Type (BER,RTT)	Dial-up (Low-short)	Dial-up and wireless (Low to medium, Short to medium)	Dial-up and wireless (Low to medium, Short to medium)	Wireless (High, Long)
Encoding Mechanism	Differential	Differential	Differential	Window-based Least significant bit.
Error Recovery (Feedback)	TCP Based (No)	TWICE (yes)	TWICE (yes)	Local Repair (Yes)
Recommended in standards	—	UMTS Release 99 onwards CDMA2000 Release B onwards	—	UMTS Release 4 onwards CDMA2000 Release B onwards

1.3 Classification of Header fields

Most of the information contained in the header is static or vary in a specific pattern, we can classify this information as STATIC, DYNAMIC, STATIC KNOWN and INFERRED [13]. Figure 3 specify this classification for IPv4 and IPv6 Headers.

Static: Static fields remains unchanged during the lifetime of a header i.e. source and destination address. There is no need to sent static entries when we are sending compressed packets. They are only sent with uncompressed packets.

Static Known: These are the fields which are constant in any packet header, i.e. Header length in IPv4 packet.

Dynamic: These are the fields which change in a specified pattern or randomly. These fields are compressed efficiently, i.e. Identification field in IPv4 header.

Inferred: These fields are never sent within a packet and they are inferred from the lower layers in the protocol stack, like Total length in IPv4 packet.

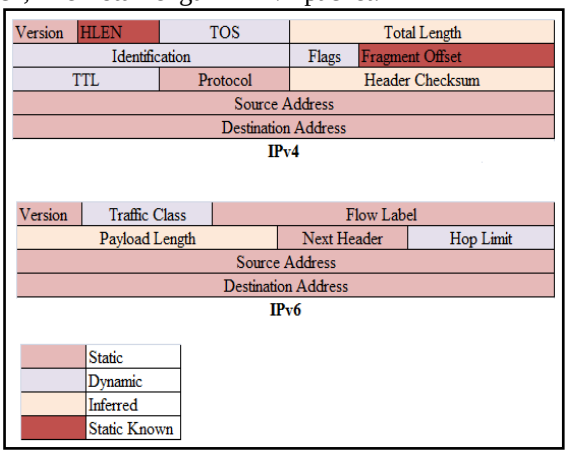

Figure 3: IPv4/IPv6 Header Classification

Rest of the paper is structured as follows: section II describes about the proposed methodology, section III describes about the Simulation parameters and scenario. Results are discussed in section IV, section V concludes the paper.

II. PROPOSED METHODOLOGY

In this paper we are compressing the IPv6 header of the packet as IPv6 header is of largest length of 40 bytes. At the sender side at network layer we have added a new parameter tunnel algo to use, this uses a binary value of either 0 or 1. If this value is 0 the normal tunneling mechanism is used, it this value is set to 1, our compression mechanism is used. Along with this parameter, we also have added a new value n , where n shows the number of uncompressed packets to send. Based upon the valve of n, we are sending uncompressed packets to the network with adding 2 extra bytes in the packets.

This extra bytes are referred are Context Header, and this is used to establish the context between the edge routers. Doing this we are increasing the size of the header but we are adding this context header to only n number of uncompressed packets and then we are removing the IPv6 header of the packet and adding only the Compressed Header and send it to the network. Figure 4 shows the format of Context Header.

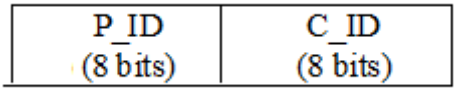

Figure 4: Context Header

Where, Profile ID (P_ID) represents the different profiles and these profiles need to be decompressed according to the profile id specified. Currently the profile specified is IPv6 only profile.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Context ID (C_ID) represents the context on the basis of which we can identify different flows in router.
Our proposed approach is shown in figure 5

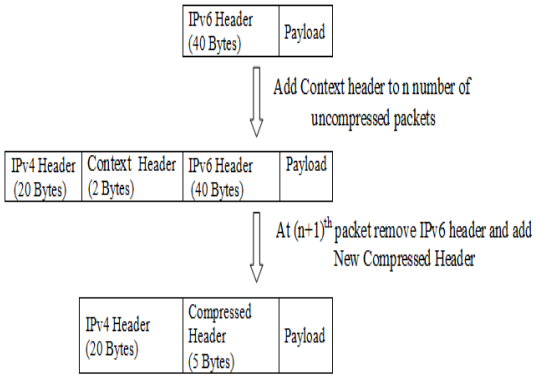

Figure 5: Proposed Approach

Once n no of uncompressed packets are sent then only we are send the dynamic information and eliminating the IPv6 Header. For sending the dynamic information we are adding a new header called compressed in the packet. Figure 6 shows the compressed header format.

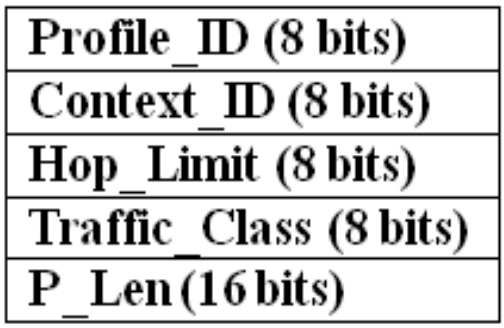

Figure 6: Compressed Header

Where the fields Profile_ID and Context_ID are derived from the modified header and the remaining are the dynamic fields which are always sent along with the packet. Doing this we have compressed 40 bytes of IPv6 header up to 6 bytes and here with increasing the overall efficiency of IPv6 Tunnels.

At the receiver end, the router receives the IPv4 packet, discards the ipv4 header, and for n number of uncompressed packets, the edge router stores the static entries, and at (n+1)th packet it reads the value of p_id and c_id from the ipv6 packet, and on the basis of these entries, gets the static information for this packet and reconstructs the new IPv6 header, and add this new header to the packet, and based upon the destination address in the packet delivers the packet to the intended destination.

III. SIMULATION TEST BED

Simulation allows us to provide an environment for designing, creating and analyzing the performance of our protocol. Variety of simulators is available like NS-2, NS-3, Opnet, GNS 3, Exata/Cyber, Qualnet etc. In order to test the performance of our algorithm we have used Qualnet 5.1 simulator [14]. Figure 7 shows the scenario of our network. This scenario depicts multihop scenario over hybrid network. Here the end users are using wired network where as the intermediate backbone routers are wireless.

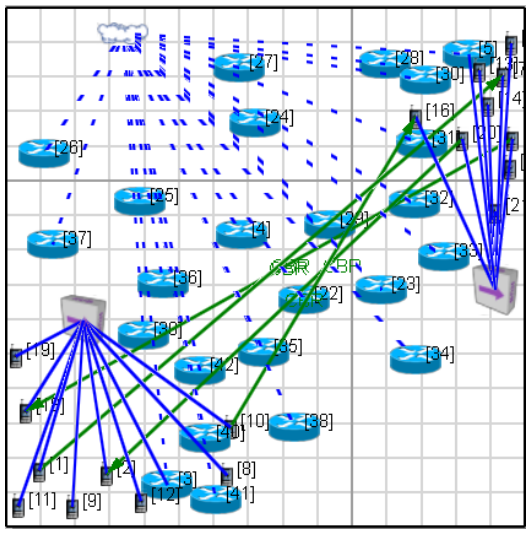

Figure 7: Scenario for Simulation

Here in the scenario there are two IPv6 networks, and Router 3 and Router 5 are dual stack routers. The intermediate routers are IPv4 only routers and are all wireless routers. Here an IPv6 sender wants to send an IPv6 packet to an IPv6 destination but the backbone is based upon IPv4 network. Here a tunnel is created between router 3 and 5 to enable this communication. The following table-2 shows the simulation parameters for this scenario.

Table 2: Simulation Parameters

Parameter	Value
Simulator	Qualnet 5.1
Studied Protocols	Bellman Ford for IPv4 Networks.
	RIPng for IPv6 Networks.
Area	1500m x 1500m.
Total no. of nodes	42 nodes.
Dual Stack Edge Nodes	02
IPv4 only nodes	22
IPv6 only nodes	18
No. of Packet Sources.	04
Type of sources	CBR
MAC protocol	802.11 for Wireless Networks. 802.3 for Wired Networks.
Packet size	64,128,256,512,1024,and 2048 Bytes
Traffic Rate	100 packet per second
Mobility model	None
Simulation time	300 seconds
Channel type	Wired & Wireless.
Antenna model	Omni Directional

IV. RESULTS & DISCUSSIONS

The Qualnet 5.1 Simulator is used to analyse the parametric performance of Compressed and

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Uncompressed Network. The metric based analysis is shown form figure 8 to 11. The simulation is carried upon hybrid network where end users are wired networks and the backbone network is wireless. Here 4 Constant Bit rate (CBR) applications are used on varying packet sizes of 64, 128, 256, 512, 1024, and 2048 bytes. Here comparison is done for compressed and uncompressed network.

4.1 Throughput: Throughput is defined as the number of packets (bytes) received by the destination among the packets in a given time. The unit of throughput is bits/sec. Here we are analysing throughput for varying packet sizes. The formula for throughput is given as:

Throughput (T) = 8*Total No. of Bytes Received/(time last packet sent - time first packet sent)

Figure 8 depict the graph for throughput. Simulation results shows that when packet size is less i.e. the case of 64 bytes we are getting no difference in the throughput of compressed and uncompressed networks being reason is that the size of packet is small so all the packets are delivered to the intended destination. However a significant improvement is observed when the packet sizes increases to 256,512 and 1024 bytes. Here throughput is better in case of uncompressed networks since more packets are delivering in case compressed network. When packet size is increased to 2048 bytes there is significant degradation in throughput as packet loss is more in this case but still we are getting better results in case of uncompressed networks.

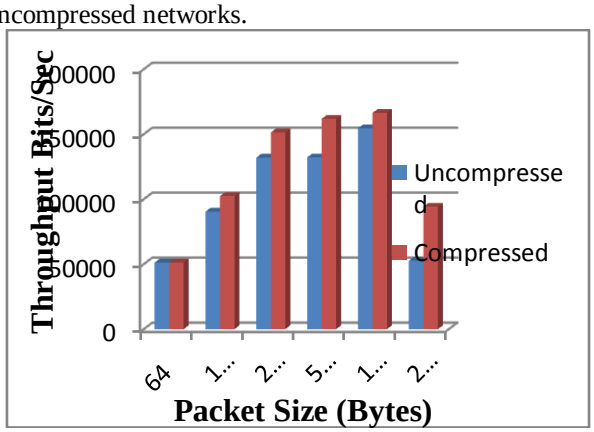

Figure 8: Throughput Vs Packet Size

4.2 End-to-End Delay: It is defined as the time elapsed between the packet sent from the source and is received by the destination node. It includes delays like queueing delay, processing delay, propagation delay, and transmission delay, and delay for route discovery. It is calculated in seconds. The formula for delay calculation is given as:

Average end-to-end delay = (total of transmission delays of all received packets) / (number of packets received),
where, transmission delay of a packet = (time packet received at server - time packet transmitted at client) , where the times are in seconds.
Figure: 9 depict the graph for Average End-to-End Delay.

Result shows that average end-to-end delay is almost negligible when packet size is 64 bytes. But as the packet size increases to 256, 512, 1024 and 2048 bytes delay increases but still we are experiencing less delay in case of uncompressed networks. As packet size increases delay increases because it takes time to send the complete packet from source to destination.

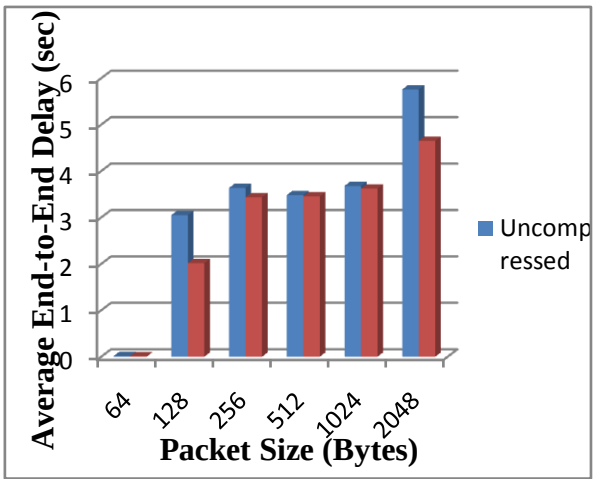

Figure 9: Average End-to-End delay Vs Packet Size

4.3 Average Jitter: Jitter is the variation in arrival time between two consecutive packets. It is observed by network congestion, a sudden network topology change or route changes. It is measured in seconds. The formula for Jitter calculation is given as:

Average jitter = (total packet jitter for all received packets) / (number of packets received - 1)
where, packet jitter = (transmission delay of the current packet - transmission delay of the previous packet).
Jitter can be calculated only if at least two packets have been received.

Figure: 10 depict the graph for average jitter. Results show that jitter is almost negligible when packet is small, but it significantly increases as the packet size increases. Impact of packet size is directly proportional to jitter, as packet size increases, jitter increases. We are experiencing less delay in case of uncompressed network, as we are reducing the overall size of the packet.

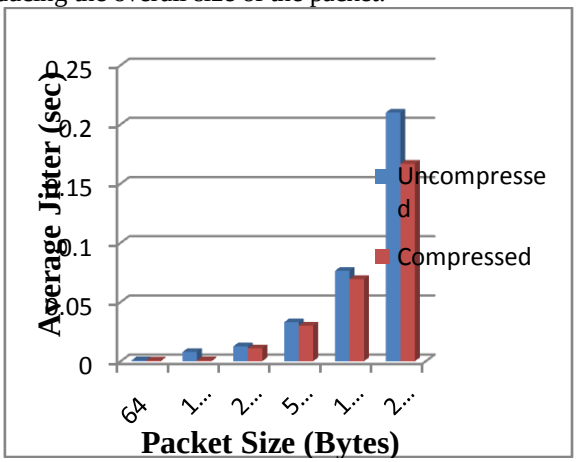

Figure 10: Average Jitter Vs Packet Size

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

4.4 Packet Delivery Ratio (PDR): It is the ratio of total number of packets received by the destination to the number of packets originated by the source. It specifies the packet loss rate, which limits the maximum throughput of the network. The formula for packet delivery ration is given as:

PDR = (Total number of Packets Received / Total number of Packets Send) *100.

Figure: 11 depicts the graph of Packet Delivery Ratio. From the graph it is clear that PDR is 100% for small packets, but as the packet size increases, Packet Delivery ratio declines. Because when the packet is small, no packets are dropped, but as packet size increases more packets are dropped, which affects the other parameters.

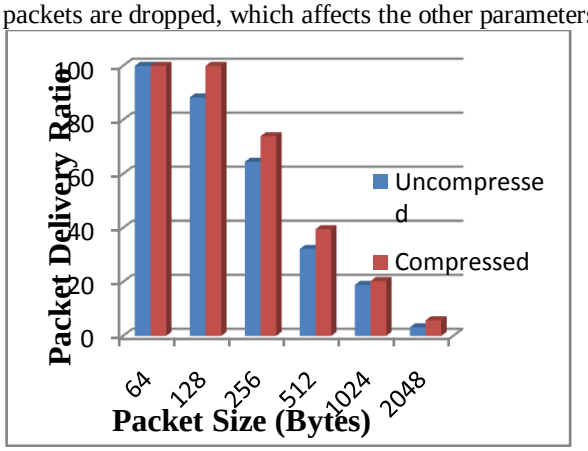

Figure 11: Packet Delivery Ratio Vs Packet Size

V. CONCLUSION

In this paper we have proposed a new approach for compressing the IPv6 header of the packet. We are using this approach over IPv6 tunneling mechanism. We have compressed the 40 bytes of ipv6 header up to 6 bytes. Simulations show that using this approach we are getting better network deliverables in terms of throughput, average end-to-end delay, Jitter, and Packet delivery ratio. Currently we are simulating this in small scale networks with limited nodes, better results would be achieved when applied to large network. In future we want to test this protocol for large scale networks in real time scenario. Also we would like to test this profile over pure wired and wireless network.

REFERENCES

[1]. R. Hinden, S. Deering, Internet Protocol Version 6 (IPv6) Addressing Architecture. RFC 3513. Available: http://tools.ietf.org/html/rfc3513.
[2]. Marina Del Rey, California 90291 "Internet Protocol, Darpa Internet Program, Protocol Specification", RFC 791.
[3]. Omae M. O, Ismail Adeya, "IPV4 to IPV6 - Transition and Benefits", May 2011.
[4]. 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.
[5]. Ioan Raicu, Sherali Zeadally, "Evaluating IPv4 to IPv6 Transition Mechanisms", IEEE International Conference on Telecommunications 2003, ICT'2003, Volume 2, Feb 2003, pp 1091 - 1098, 0-7803-7661-7/03/$17.00©2003 IEEE.
[6]. M. Degermark , B. Nordgren, S. Pink. Network Working Group: Request for Comments: 2507: IP Header Compression.
[7]. Dipti Chauhan, Sanjay Sharma, "Network Optimization of IPv6 Networks Using Tunnel Header Compression" IJRET: International Journal of Research in Engineering and Technology, Volume: 04 Issue: 03 | Mar-2015, Available.
[8]. An introduction to IP header compression, white paper, www.effnet.com.
[9]. V. Jacobson, "Compressing TCP/IP Headers for Low-Speed Serial Links," RFC 1144, February 1990.
[10]. M. Degermark , B. Nordgren, S. Pink. Network Working Group: Request for Comments: 2507: IP Header Compression.
[11]. S. Casner and V. Jacobson. Compressing IP/UDP/RTP Headers for Low-Speed Serial Links. Internet Standards Track RFC 2508, IETF, 1999.
[12]. 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.
[13]. Ms.Devishree Naidu , Ms. Rakhi Tapadiya, Implementation of Header Compression in 3GPP LTE, 2009 Sixth International Conference on Information Technology: New Generations.
[14]. Qualnet 5.1 User's Guide, "Scalable Network Technologies",http://web.scalable-networks.com/content/ qualnet.

BIOGRAPHIES

Dipti Chauhan, completed her M.Tech degree from Barkatullah University Institute of Technology Bhopal in 2011. She completed her MCA from UIT, RGPV in 2006. 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.