

Addressing the Bandwidth issue in End-to-End Header Compression over IPv6 tunneling Mechanism

Dipti Chauhan, Sanjay Sharma

Maulana Azad National Institute of Technology, Department of Computer Applications, Bhopal, 462003, India

Abstract—One day IPv6 is going to be the default protocol used over the internet. But till then we are going to have the networks which IPv4, IPv6 or both networks. There are a number of migration technologies which support this transition like dual stack, tunneling & header translation. In this paper we are improving the efficiency of IPv6 tunneling mechanism, by compressing the IPv6 header of the tunneled packet as IPv6 header is of largest length of 40 bytes. Here the tunnel is a multi hop wireless tunnel and results are analyzed on the basis of varying bandwidth of wireless network. Here different network performance parameters like throughput, End-to-End delay, Jitter, and Packet delivery ratio are taken into account and the results are compared with uncompressed network. We have used Qualnet 5.1 Simulator and the simulation results shows that using header compression over multi hop IPv6 tunnel results in better network performance and bandwidth savings than uncompressed network.

Index Terms—Bandwidth, Compression, Decompression, Multihop, Profile, Tunnel.

I. INTRODUCTION

Internet is growing at an alarming rate and with the advent of new devices and applications it's not possible to sustain with the IPv4 internet protocol. IPv4 is a 32 bits addressing protocol [1], which can address up to 232 devices. But the internet users are much more than this number. So, the need of a new addressing protocol was felt in 1998 by IETF, and this is known as the next generation internet protocol IPv6 [2], which is a 128 bits protocol which can address up to 2128 devices, such a huge number. It means each and every particle on the earth will be addressed; still we are left with a huge number of IP addresses. Despite of numerous advantages of IPv6 over IPv4 the adoption of IPv6 is still very slow worldwide. IPv6 is not backward compatible with IPv4 and IPv4 hosts and routers will not be able to deal directly with IPv6 traffic and vice-versa [3]. However the migration from IPv4 to IPv6 is a long term strategy and currently the main issue is the intercommunication between these two protocols. Different migration techniques like dual stack, tunneling & header translation exists to assist the transition towards the IPv6 networks.

In reality IPv4 is there for a long time and till then we have to deal with a network in which both the protocols be operating side by side. In this paper we are dealing with tunneling techniques to assist the migration towards the IPv6 network.

Tunneling techniques are used when an IPv6 sender wants to communicate with an IPv6 receiver, but the backbone network is based upon IPv4 routers [4]. In order to enable this type of communication, the IPv6 packet is encapsulated inside an IPv4 packet and is sent across the network, at the receiver side the IPv4 packet is striped off and the IPv6 packet is delivered to the intended destination. Figure 1 shows the tunneling mechanism where IPv6 packet is encapsulated inside an IPv4 packet.

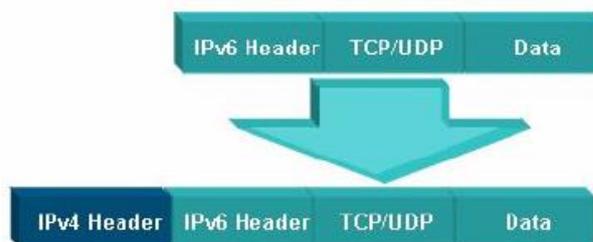


Fig.1. Tunneled IPv6 Packet

The use of tunneling comes with several shortcomings like high header overhead as a result of adding several protocol headers in a packet which results in low efficiency and performance degradation, especially over wireless links where resources are scarce [5]. Using header compression techniques we can improve the efficiency of tunneling mechanism. Header compression deals with compressing the excess protocol headers over the link and decompressing it at the other end of the link [6]. Most of the information in the header is static, like source address, destination address etc, or varies in a specific pattern, like identification field, TTL etc. These packet headers are very important over end to end communication but of very importance from one hop to another. So, it's better to use header compression, which would result in many cases more than 90% savings, and thus save the bandwidth and use the expensive resources efficiently [7]. Using header compression we are increasing the computational complexity, but the compression gains are so high, so that we can

compromise on the complexity. 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 [8]. Table 1 show the header compression gains which can be achieved using header compression over different protocol headers [9].

Table 1. Compression Gains

Protocol headers	Total header size (bytes)	Minimum Compressed header size (bytes)	Compression gain (%)
IPv4/TCP	40	4	90
IPv4/UDP	28	1	96.4
IPv4/UDP/RTP	40	1	97.5
IPv6/TCP	60	4	93.3
IPv6/UDP	48	3	93.75
IPv6/UDP/RTP	60	3	95

Rest of the paper is structured as follows: section II describes about the Literature survey, section III discusses about the proposed methodology, section 3 describes about the Simulation parameters and scenario. Results are discussed in section 4, section 5 concludes the paper.

II. LITERATURE SURVEY

[10] Proposed a new technique called as Routing-Assisted Header Compression (RAHC) for end-to-end header compression technique for multi-hop wireless ad hoc networks. RAHC works in conjunction with conventional on-demand routing techniques and relies on the information provided by the routing algorithm. Here partial flooding is used for updating the context information for the nodes that lie between source & destination. The failure of intermediate node may result in de-synchronization of the source and destination nodes. In this case routing algorithm initiates route discovery and the context information has to be updated in all the nodes that have been newly introduced into the path. [11] Proposed a new approach for header compression in conjunction with IP security framework. Here the IPv6 protocol header is compressed and the benefit is a reduction of the overhead caused by IPsec tunnel mode in terms of enlarged datagram's. [12] Proposed the approach of end to end header compression over wireless mesh networks. Here packet aggregation scheme is used in cooperation with the header compression mechanism. Simulations shows that only using a suitable header compression scheme can optimize the bandwidth usage in VoIP applications over wireless mesh networks, impacting positively in the loss packet. [13] Proposed the use of header compression in context of IP based ITS communication. Here our kernel implementation of an

open source RoHC library is described. This integration of a RoHC library inside the kernel may also be an advantage in terms of dynamic adaptation. [14] Proposed a new header compression mechanism which can be deployed in end-to-end nodes using the Software-Defined Network concept. Using this mechanism can reduce both packet size and time delay. In addition to utilize the use of the network, it also benefits in time factor for an application that requires low latency and small packet size such as VoIP. [15] Proposed the usage of traffic optimization techniques within the context of the LISP (Locator/Identifier Separation Protocol) framework. These techniques use Tunneling, Multiplexing and header Compression of Traffic Flows (TCMTF) in order to save bandwidth and to reduce the amount of packets per time unit. Using this approach bandwidth can be drastically reduced. [16] Proposed a new technique for header compression scheme ROHC+ for TCP/IP streams in a wireless context, such as the 3G platform, characterized by relatively high loss. Results show that the new header compression provides an efficient use of radio resource with direct benefit on the economic feasibility and on the quality of the service.

III. PROPOSED METHODOLOGY

In this paper we are compressing only the IPv6 header of the packet, as it is of largest length of 40 bytes. We have classified the header fields of IPv6 packet as STATIC, DYNAMIC, and INFERRED [17].

STATIC: Static fields are the header fields which remain unchanged during the life time of a header. These fields are sent only with uncompressed packets.

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.

The following table: 2 classify the header field in the IPv6 header:

Table 2. Header Classification for IPv6 Header

Protocol Field	Classification
Version	STATIC
Flow label	
Next Header	
Source IP Address	
Destination IP Address	
Traffic Class	DYNAMIC
HOP Limit	
Payload Length	INFERRED

At Sender side:

At the network layer we have added a new parameter called tunnel algo to use. It takes the value 0 or 1. Based on this value we are deciding which mechanism to use,

either compresses or uncompressed. It tunnel algo to use is 0: specifies normal tunneling mechanism is used. If it is set to 1: specifies compressed tunneling mechanism to use. Along with this, we are taking a value n, where n is number of uncompressed packets to send. Here initially we are sending first n packets uncompressed in the tunnel, and adding two extra bytes, in these uncompressed packets. This extra byte represents the C_ID and P_ID of the packet and is known as context header. The structure of this context header is shown in figure: 2.



Fig.2. 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.

Context ID (C_ID) represents the context on the basis of which we can identify different flows in router.

Adding this context header we are adding two extra bytes and increasing the header overhead, but this context header is added only to n uncompressed packet, we call it context packets which are needed to establish context between the edge routers. The format for context header packet is shown in figure: 3. We are sending n context packets to the destination edge router.

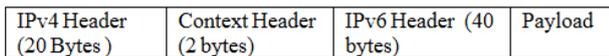


Fig.3. Context Packet

Once n packets are sent, then we remove the IPv6 header from the packets and add the new compressed header before encapsulating inside the IPv4 packet. The format of compressed header is shown in figure: 4.

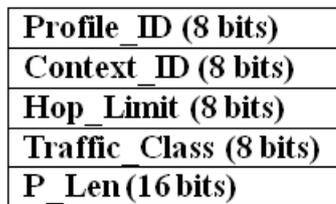


Fig.4. Compressed Header

Here Profile_ID and Context_ID are derived from the context header and remaining fields are the dynamic and inferred fields which are sent with every compressed packet. The format for compressed packet is shown in figure: 5.

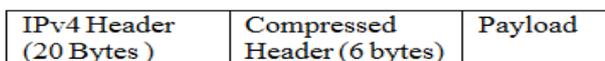


Fig.5. Compressed Packet

We are sending compressed packet until the simulation ends. Figure 6 specifies the working at the sender side:

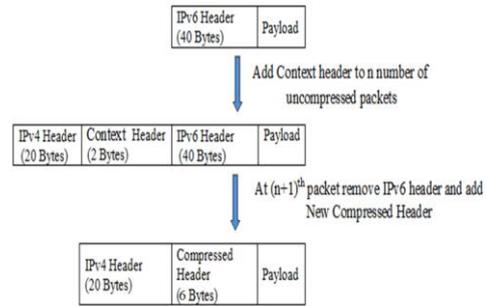


Fig.6. Handling packet at sender side

At Destination side:

At the dual stack router (edge router) receives the IPv6 packet whose destination address is its own address it does the following:

IPv4 header is stripped off from the packet.

For n uncompressed packets

First remove the context header after reading the values of p_id and c_id.

Read static info from the IPv6 packet and stores the information for corresponding c_id and send the packet to the intended destination.

At n+1 packet

Read compressed header to get p_id and c_id.

Check the static entry for this corresponding c_id.

Make a new IPv6 header based on this static and dynamic information.

Add this new IPv6 header in front of the packet.

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

The following figure 7 and figure 8 specifies the handling of packet at the destination:

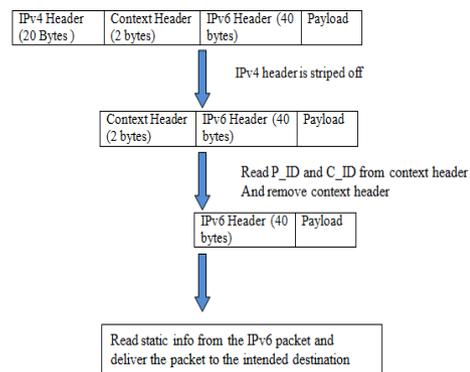


Fig.7. Handling n packets

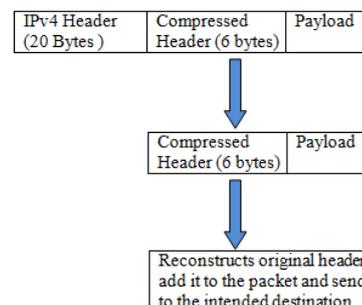


Fig.8. Handling n+1 packets

5.1. Throughput

Throughput can be defined as the number of packets delivered per unit time. More the bandwidth greater the throughput will be for any network. It is measured in bits/sec. Here we are analyzing throughput for varying bandwidth of wireless network. The formula for throughput is given as:

$$\text{Throughput (T)} = \frac{8 * \text{Total No. of Bytes Received}}{\text{time last packet sent} - \text{time first packet sent}}$$

Figure 10 depicts the graph for throughput. We are getting highest throughput when the bandwidth is 5 mbps, since more the bandwidth greater the throughput will be. Throughput is very less in case when bandwidth is 500 kbps. From the graph it is clear that every time we are getting better results is case of compressed network. Here we can say that bandwidth is proportional to throughput, i.e. as bandwidth increases, throughput increases considerably.

$$\text{Bandwidth (B)} \propto \text{Throughput}$$

Table 4. Throughput

Throughput (bits/s)		
Bandwidth	Uncompressed	Compressed
500 KBPS	9362	13169
1 MBPS	30451	33847
2 MBPS	122506	164379
3 MBPS	281281	298649
4 MBPS	389871	409606
5 MBPS	407710	409603

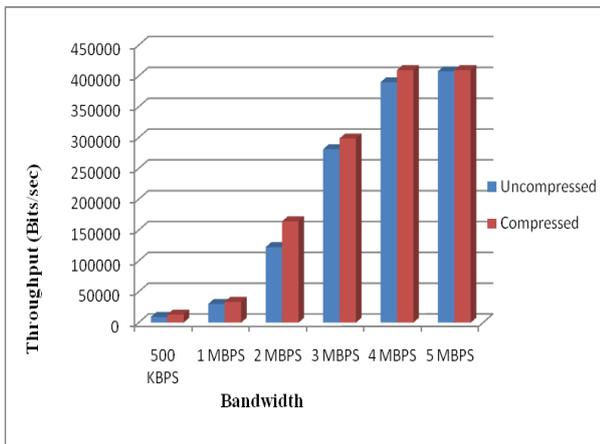


Fig.10. Throughput Vs Bandwidth

5.2. Average End-to-End Delay

Average end-to-end delay is the time interval between the packets is sent by the source node and is received by the destination node. All different delays are included like propagation delay, queuing delays, delay for route discovery, etc. Delay is an important factor for finding the network performance. Delay should be less as the packets must reach from source to destination in very less time. It is calculated in seconds. The formula for delay

calculation is given as:

$$\text{Average end-to-end delay} = \frac{\text{total of transmission delays of all received packets}}{\text{number of packets received}}$$

where,

$$\text{transmission delay of a packet} = (\text{time packet received at server} - \text{time packet transmitted at client}), \text{ where the times are in seconds.}$$

Figure: 11 depicts the graph for Average End-to-End Delay. Result shows that average end-to-end delay is very high for low bandwidth i.e. 500 kbps, but as the bandwidth increases average end-to-end delay decreases considerable and is almost negligible when bandwidth is 5 mbps. But still we are experiencing less delay in case of compressed networks. Here we can say that average end-to-end delay is inversely proportional to bandwidth, i.e. as bandwidth increases, average end-to-end delay decreases considerably.

$$\text{Bandwidth (B)} \propto 1 / \text{End-to-end delay}$$

Table 5. Average-end-to-end delay

Average End-to-End Delay (s)		
Bandwidth	Uncompressed	Compressed
500 KBPS	30.0944	25.4849
1 MBPS	12.6209	12.1964
2 MBPS	3.48059	3.45001
3 MBPS	1.85479	1.86619
4 MBPS	0.865408	0.0178829
5 MBPS	0.0154019	0.014862

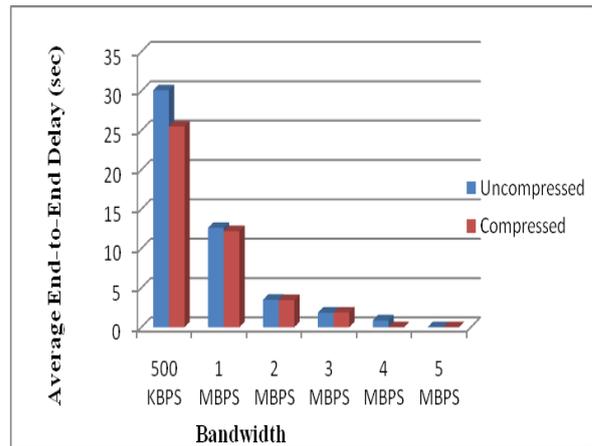


Fig.11. Average End-to-End Delay Vs Bandwidth

5.3. Average Jitter

Jitter is defined as the variation in the arrival time between two consecutive packets. Jitter is measured in secs, and it should be very less. For real time applications jitter should be negligible, for better experience. The formula for Jitter calculation is given as:

$$\text{Average jitter} = \frac{\text{total packet jitter for all received packets}}{\text{number of packets received} - 1}$$

where,

$$\text{packet jitter} = (\text{transmission delay of the current packet}$$

- transmission delay of the previous packet).

Jitter can be calculated only if at least two packets have been received.

Figure: 12 depicts the graph for average jitter. Simulations show that jitter is almost negligible when bandwidth is 5 mbps. But as the bandwidth decreases its impact is shown over the jitter, it increases significantly. We are experiencing less delay in case of compressed network, as we are reducing the overall size of the packet. Jitter is improved in case of our algorithm. Here we can say that bandwidth is inversely proportional to average jitter, i.e. as bandwidth increases, average jitter decreases considerably.

Bandwidth (B) \propto 1/ Jitter

Table 6. Average Jitter

Average Jitter (s)		
Bandwidth	Uncompressed	Compressed
500 KBPS	0.629625	0.5484
1 MBPS	0.206304	0.19234
2 MBPS	0.0332409	0.0294222
3 MBPS	0.011667	0.0105134
4 MBPS	0.00487796	0.000681807
5 MBPS	0.000647171	0.000631304

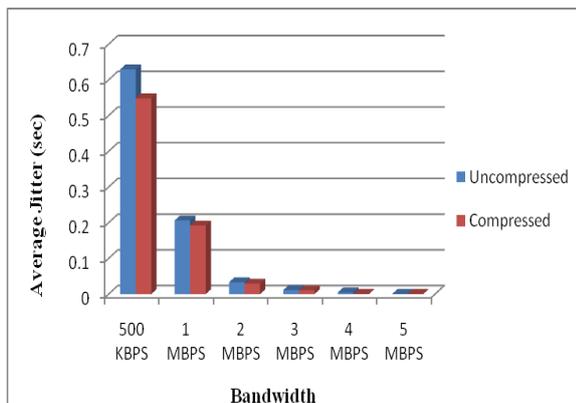


Fig.12. Average Jitter Vs Bandwidth

5.4. Packet Delivery Ratio (PDR)

It is the ratio of number of packets actually delivered from source to destination, to the number of packets sent. PDR should be high for better performance, Higher the PDR, more the throughput. The formula for packet delivery ration is given as:

$$PDR = (\text{Total number of Packets Received} / \text{Total number of Packets Send}) * 100.$$

Figure: 13 depict the graph of Packet Delivery Ratio. From the graph it is clear that PDR is almost 100% when bandwidth is 5 mbps, but as the bandwidth decreases, PDR declines. Because when the bandwidth is high link utilization is effective, as the bandwidth decreases, packet are dropped which degrades the PDR, and its affect is shown over other parameters. Here we are getting better results in compressed networks. Here we can say that bandwidth is proportional to PDR, i.e. as bandwidth

increases, PDR increases considerably.

Bandwidth (B) \propto Packet Delivery Ratio

Table 7. Packet Delivery Ratio

Packet Delivery Ratio		
Bandwidth	Uncompressed	Compressed
500 KBPS	2.284444444	3.213333333
1 MBPS	7.433333333	8.262222222
2 MBPS	29.90666667	40.12888889
3 MBPS	68.66888889	72.90888889
4 MBPS	95.18	99.99777778
5 MBPS	99.53555556	99.99777778

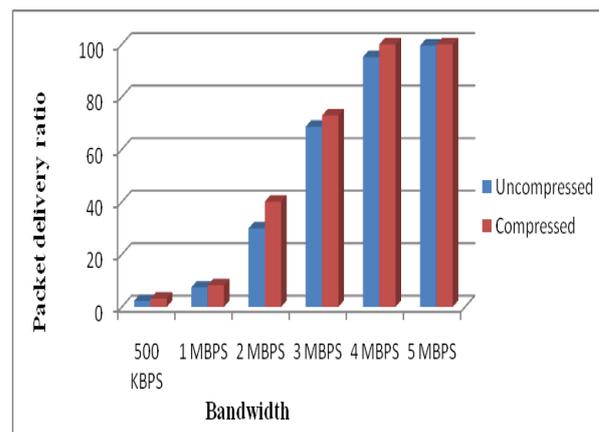


Fig.13. Packet Delivery Ratio Vs Bandwidth

VI. CONCLUSION

IPv6 is the future of the internet without which it's not possible to sustain internet's life. In this paper we have proposed a new approach for improving the efficiency of IPv6 tunneling mechanism which results in the betterment for enabling the smooth interoperation of the two protocols. Using this approach we have compressed the 40 bytes of IPv6 header up to 6 bytes, which is significantly a big improvement. Compressing the IPv6 header we are getting better network deliverables in terms of throughput, average end-to-end delay, Jitter, and Packet delivery ratio. We have compared the result with normal tunneling mechanism. Simulation is carried out over Qualnet 5.1 simulator. Results shows that a compressed network performs better results than uncompressed network. Currently we are simulating it over small networks with limited load, even better results could be achieved if applied to large scale networks. Bandwidth is the most critical resource for a network and it must be utilized efficiently. Here we are better utilizing the bandwidth using header compression. Currently we are having only one profile i.e. IPv6 only profile, later on IPv6/UDP and IPv6/TCP would be added to our work.

REFERENCES

[1] Marina Del Rey, California 90291 "RFC: 791- Internet Protocol, Darpa Internet Program, Protocol Specification",

- September 1981. Available: <https://tools.ietf.org/html/rfc791>.
- [2] S. Deering, R. Hinden, "RFC: 2460- Internet Protocol, Version 6 (IPv6) Specification", December 1998, Available: <https://www.ietf.org/rfc/rfc2460.txt>.
- [3] Dipti Chauhan, Sanjay Sharma, "A Survey on Next Generation Internet Protocol: IPv6", International Journal of Electronics & Industrial Engineering (IJEIE), ISSN: 2301-380X, Volume 2, No. 2, June 2014, Pages: 125-128.
- [4] 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.
- [5] Priyanka Rawat and Jean-Marie Bonnin, "Designing a Header Compression Mechanism for Efficient Use of IP Tunneling in Wireless Networks", IEEE CCNC 2010, 978-1-4244-5176-0/10/\$26.00 ©2010 IEEE.
- [6] M. Degermark, B. Nordgren, S. Pink. Network Working Group: Request for Comments: 2507: IP Header Compression, February 1999.
- [7] The concept of robust header compression, ROHC, white paper, www.ietf.org.
- [8] 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. Yoshimura, and H. Zheng, "Robust Header Compression (ROHC): Framework and four profiles: RTP, UDP,ESP, and uncompressed," RFC 3095, July 2001.
- [9] An introduction to IP header compression, white paper, www.ietf.org.
- [10] Ranjani Sridharan, Ranjani Sridharan, Sumita Mishra, "A Robust Header Compression Technique for Wireless Ad Hoc Networks", MobiHoc'03, June 1-3, 2003, Annapolis, Maryland, USA. Copyright 2003 ACM 1-58113-684-6/03/0006.
- [11] Christoph Karg and Martin Lies, "A new approach to header compression in secure communications", Journal of Telecommunications & Information Technology, March 2006.
- [12] André Giordanna O. do Nascimento, Edjair Mota, Saulo Queiroz, Edson do Nascimento Junior, "An Alternative Approach for Header Compression Over Wireless Mesh Networks", 2009 International Conference on Advanced Information Networking and Applications Workshops.
- [13] Nesrine Benhassine, Emmanuel Thierry, Jean-Marie Bonnin, "Efficient Header Compression Implementation for IP-based ITS communications", 12th International Conference on ITS Telecommunications.
- [14] Supalerk Jivorasetkul, Masayoshi Shimamura, Katsuyoshi Iida, "End-to-End Header Compression over Software-Defined Networks: a Low Latency Network Architecture", 978-0-7695-4808-1/12 \$26.00 © 2012 IEEE DOI 10.1109/iNCoS.2012.80.
- [15] Jose Saldana, Luigi Iannone, Diego R. Lopez, Julián Fernández-Navajas, José Ruiz-Mas, "Enhancing Throughput Efficiency via Multiplexing and Header Compression over LISP Tunnels", IEEE International Conference on Communications 2013: IEEE ICC'13 - Second IEEE Workshop on Telecommunication Standards: From Research to Standards.
- [16] Dipti Chauhan, Sanjay Sharma, "Enhancing the Efficiency of IPv6 Tunneling Mechanism by using Header Compression over IPv6 Header", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 4, April 2015.
- [17] G. Boggia, P. Camarda, V.G. Squeo, ROHC+: A New Header Compression Scheme for TCP Streams in 3G Wireless Systems, IEEE International Conference on Communications, 2002. ICC 2002.
- [18] Qualnet 5.1 User's Guide, "Scalable Network Technologies", <http://web.scalable-networks.com/content/qualnet>.

Authors' Profiles



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 in Department of Computer Applications from MANIT, Bhopal. Her research interests include next generation networks and IPv6. She is IPv6 Certified Gold and Silver Network Engineer from IPv6 forum, University Sains Malaysia.



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. His research interests include wireless networks, Next generation networks, Sensor Networks, 6LoWPAN, and Large Databases.

How to cite this paper: Dipti Chauhan, Sanjay Sharma, "Addressing the Bandwidth issue in End-to-End Header Compression over IPv6 Tunneling Mechanism", IJCNIS, vol.7, no.9, pp.39-45, 2015. DOI: 10.5815/ijcnis.2015.09.05