*I.J. Wireless and Microwave Technologies*, **2017, 3, 49-57**
Published Online May 2017 in MECS(http://www.mecs-press.net)
DOI: 10.5815/ijwmt.2017.03.05

***Available online at*** ***http://www.mecs-press.net/ijwmt***

# A study on Need of Adaptation Layer in 6LoWPAN Protocol Stack

Ruchi Garg[*], Sanjay Sharma

*Department of Mathematics & Computer Applications Maulana Azad National Institute of Technology M.P., India*

**Abstract**

Emergence of wireless embedded applications is resulting in the communication among sensor nodes connected in a wireless personal area network. Sensor nodes gather the real time information and transmit it to the desired application. This requires transmission of IPv6 packets over Low-power wireless personal area network and is called 6LoWPAN. IPv6 is resource intensive protocol whereas 6LoWPAN is resource constraint due to small packet size, limited device memory, short transmission range, and less data rate of sensor nodes. Also these nodes in 6LoWPAN are mainly battery operated hence minimum power consumption is also a major constraint.

To make the efficient transmission of information in such a resource constraint network, an adaptation layer was suggested and implemented by Internet Engineering Task Force (IETF). The placing of this additional layer is in between network layer and data link layer of TCP/IP protocol stack. This paper contributes in the detailed analysis of need of adaptation layer in 6LoWPAN protocol stack. The necessity of this additional layer is justified by explaining the major functions like header compression, fragmentation and reassembly of packets and packet routing handled by it.

**Index Terms:** 6LoWPAN, adaptation layer, fragmentation, header compression, mesh addressing header



## 1. Introduction

Decades before, the advent of Internet and Communication Technology (ICT) had shrunk the world by drastic change in communication scenarios. Since then, the need of communication is increasing which is not only human to machine but moving more towards machine to machine and machine to human. Plethora of applications are emerging which needs billions of devices to communicate with each other in the physical world. Various monitoring systems like health care, street lights, vehicles, agricultural fields, fire detection, home automation, factory automation, etc. are just a few of them which needs large number of sensor nodes to

* Corresponding author.
E-mail address: ruchibhartigarg@gmail.com;ssharma66@rediffmail.com

collect the real time information and communicate it to a location where needed. For communication between these sensor nodes, each one of it should have an IP address in the network. With constant increase in network devices and their IP addresses, IPv6 addressing is now desirable rather than IPv4 as the former can provide up to $2^{128}$ unique addresses [1] [2]. Transmission of IPv6 packets over Low-power wireless personal area network is technically abbreviated as 6LoWPAN [3]. The Internet Engineering Task Force (IETF) 6LoWPAN working group was formed in 2004. Its prime purpose is to handle the challenges of enabling wireless IPv6 communication over the newly standardized IEEE 802.15.4 low-power radio for devices with less transmission range, power and memory. These resource constraint devices are mainly sensor nodes. 2007 was the year when IETF has first released the standards and specifications of 6LoWPAN [4].

The packets in 6LoWPAN have small data transmission range from 10m to 30m at the rate of 20 kbps to 240 kbps, with very constraint node memory of 16 kb RAM and 128kb ROM [5] [6]. Permissible packet size, short transmission range, limited memory, constraint energy are some compatibility issues between IPv6 and LoWPAN (IEEE standard 802.15.4). To remove these compatibility issues for efficient transmission of packets, an adaptation layer is placed between network and data link layer of TCP/IP stack.

This paper emphasizes on the presence of this additional layer i.e. adaptation layer in 6LoWPAN protocol stack. This layer not only manages the compatibility issues but also reduces the transmission overhead of packets as 6LoWPAN is highly resource constraint. The rest of the paper is organized as follows: 6LoWPAN protocol stack is shown in section 2. Need of adaptation layer is explained in section 3. Section 4 covers various functions handled by adaptation layer at sender and receiver nodes. Conclusion and future work is given in section 5.

## 2. 6LoWPAN Protocol Stack

6LoWPAN has wireless sensor nodes (WSN) having IP communication capabilities. There are various functional incompatibilities between these two as explained in section 1. To make them compatible there were two options. First option is to make changes in existing layers of TCP/IP protocol stack. Other option is to add another layer in TCP/IP protocol stack without disturbing its layers functionalities. The second option suited well as it does not affect existing TCP/IP layer architecture.

Fig. 1 shows the protocol stack of 6LoWPAN. It has an additional layer as compare to TCP/IP stack. The extra layer is Adaptation layer [3] whose need is explained in section 3.

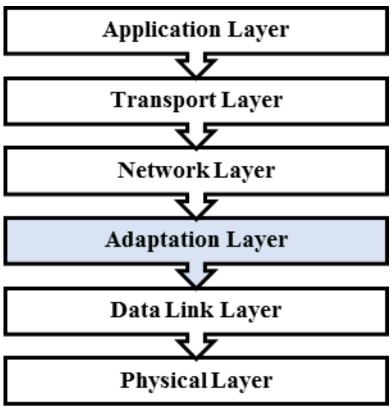


Fig.1. 6LoWPAN Protocol Stack

Similar to TCP/IP stack, 6LoWPAN also has various protocols to support successful communication between nodes [7]. Physical layer provides the basic communication in the network. It transmits the data bits over 802.15.4 by converting them into signals. The task of Data link layer is to detect and correct the errors which may occur during transmission of data bits. Medium access layer or MAC layer is present in data link layer. This layer senses the medium for collision free transmission of frames using various protocols like CSMA/CD or CSMA/CA. Adaptation layer is present between network layer and data link layer. Large sized headers like IPv6, UDP and ICMP are compressed by this layer. Also it handles the fragmentation and reassembly of packets. Above adaptation layer there is network layer which provides routing of the packets. It uses networking protocol i.e. IP (Internet Protocol) for providing addresses to the nodes. Communication between nodes could be connectionless or connection oriented for which transport layer is responsible. The protocols used are TCP, UDP and ICMP. UDP is preferred over TCP because of its less complex and small header size. Also 6LoWPAN is used in applications which needs real time data to be transmitted hence connection oriented is not much desired which again makes UDP more preferred protocol. For the provision of security, UDP based DTLS (Datagram transport layer security) protocol is used [8]. In 6LoWPAN, COAP (Constrained application protocol) and MQTT (message queue telemetry transport) protocols are used in application layer. COAP is preferred over HTTP (commonly used in TCP/IP) because of its support for devices in sleepy mode [8]. The sensor nodes are required to be in sleepy mode to save its battery. The comparison of protocols used in TCP/IP and 6LoWPAN stack is given in Fig. 2.

| TCP/IP Protocol Stack | Layers | 6LoWPAN Protocol Stack |
|---|---|---|
| HTTP, FTP, DNS, SMTP | Application | COAP, MQTT |
| TCP, UDP, ICMP | Transport | UDP, ICMP, DTLS |
| IP | Network | IPv6 with LoWPAN |
| Ethernet, PPP | Data Link | IEEE 802.15.4 MAC |
| Ethernet | Physical | IEEE 802.15.4 PHY |

Fig.2. Comparison of Protocols among TCP/IP and 6LoWPAN Stack

## 3. Need Of Adaptation Layer

As explained in section 1, usage of IPv6 in transmission of packets over LoWPAN (IEEE standard 802.15.4) is not a natural fit. Hence an adaptation layer is proposed by IETF to make IPv6 and 802.15.4 compatible with each other [5]. This layer is placed between network layer and data link layer in 6LoWPAN protocol stack as shown in Fig.1. There are three main functions of this layer. First main function is header compression and decompression. This layer compresses the IPv6 and UDP header. Various techniques have been suggested to perform this function. Second function of adaptation layer is fragmentation and reassembly of packets. Third major task of this layer is Routing. Usually routing is considered as the main task of network layer but it can also be handled by adaptation layer. When routing decision takes place at adaptation layer it is called mesh under routing and when routing is done at network layer it is route over routing. [7] [8]. The border nodes of the WSN should be able to route IPv6 packets into the WSN nodes from outside and route inside packets to outside IP network. Besides these three major functions, there are other functions of the adaptation layer on networking related things like neighbour discovery and multicast support. All functions of Adaptation layer are explained in section 4.

## 4. Functions Of Adaptation Layer

6LoWPAN is a communication protocol for wireless connectivity in sensor based applications with restrictive resources. It enhances the scalability and mobility of sensor networks. The challenge of 6LoWPAN is that the IPv6 network and IEEE 802.15.4 network are totally different. Placing an adaptation layer between the IP layer and the Data link layer is the solution to transport IPv6 packets over IEEE 802.15.4 links. It intensely reduces IP transmission overhead by performing various functions as explained below.

### *4.1. Fragmentation & Reassembly of packets*

Maximum transmission unit (MTU) of IPv6 packets is of 1280 bytes whereas in 802.15.4 the permissible packet size is only 127 bytes. In 6LoWPAN, IPv6 packets are to be transmitted over 802.15.4 which is not possible without fragmenting them to support MTU of it. Hence to encapsulate the IPv6 large sized packets onto 802.15.4, they need to be fragmented, transmitted and reassembled after reaching the destination [9]. At sender node when an IPv6 packet size exceeds the available link layer payload size, the 6LoWPAN fragmentation mechanism takes place. It will treats the IPv6 packet as a single data field and iteratively break this data into fragments. The size of the fragments will be according to the maximum frame size at the data link layer. Each fragment will then include a fragment header before it is transmitted. The task of fragmentation and reassembly of packets is performed by adaptation layer. From network layer of source node, packets will come to adaptation layer which will first check its size. If packet size is greater, then it will fragment it and forward to MAC layer. Fragment header as shown in Fig. 3, is send with each fragment when it is transmitted [9] [10]. The fields of fragment header are Datagram Size, Offset and Tag.

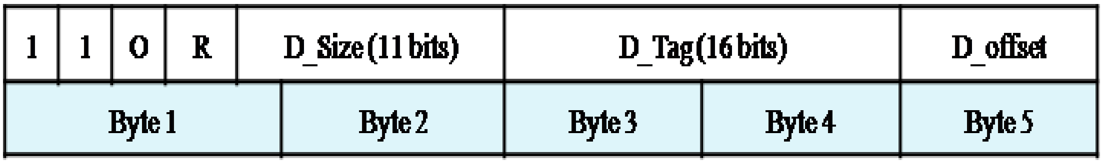


Fig.3. Fragment Header

The first two bit of byte 1 are ‘11’ that means this header is a fragment header. D_size indicates the size of datagram before fragmentation. The size of un-fragmented datagram is send with every fragment so that sufficient buffer is allocated on receiver side as datagram may reach out of order. D_Tag is a unique number which is attached with each fragment belonging to same datagram. D_offset indicates the placing of the fragment in an un-fragmented datagram. It is useful for arranging the fragments in order during reassembly of datagram. Bit number 3 of byte 1 represents offset (O) whose value could be 0 or 1. The offset value of first fragment will be 0 and for rest of the fragments it will be 1. The fragment header of first fragment is of 4 bytes as there is no need to send the offset value in it. For rest of the fragments, the fragment header is of 5 bytes. Bit numbers 4 and 5 are reserved bits (R) for future use.

Reassembly process takes place at adaptation layer of receiver node. Each fragment carries information about how much buffer need to be reserved at receiver node so that during reassembly of complete packet there is no memory crunch. The datagram offset field indicates the position of the current payload within the original IPv6 packet. Fragment reassembly time is usually 60 sec. If all fragments belonging to same packet does not arrive and misses then reassembly buffer is drained. In this case all fragments need to be transmitted again by the sender. When fragments reach the adaptation layer of receiver then they are reassembled to form a complete packet and then passed to the upper layer.

If datagram is small and a single frame is sufficient enough to carry the payload than there is no need to perform fragmentation. In that case no fragment header is attached with the packet.

### *4.2. Header Compression*

IPv6 packet size is of 1280 bytes whereas in 6LoWPAN the Maximum Transmission Unit (MTU) of a packet is 127 bytes. The frame format of 802.15.4 is shown in figure 4. Out of its 127 bytes, 23 bytes is for link layer header, 21 bytes for security header, 5 bytes for fragment header and 2 bytes for footer [7]. This leaves only 76 bytes for upper layer headers and payload, as illustrated in Fig. 4.

| Header | Security Header | Fragment Header | IPv6 Header | UDP Header | Payload | Footer |
|---|---|---|---|---|---|---|
| 23 Bytes | 21 Bytes | 5 Bytes | 40 Bytes | 8 Bytes | 28 Bytes | 2 Bytes |
| | | | 76 Bytes | | | |

Fig.4. 802.15.4 Frame Format

The upper layer headers are IPv6, TCP and UDP of 40, 21 and 8 bytes in length respectively. If transmission of IPv6 packets takes place using TCP then only 16 bytes are left for payload and in case of UDP, only 28 bytes for actual payload. Hence compression of IPv6 and UDP/TCP header is needed so as to increase the bytes available for payload. Fig. 4 shows the use of UDP along with IPv6 header.

Header compression is performed at adaptation layer of 6LoWPAN protocol stack. Various techniques have been suggested and in implementation to perform compression of header so as to create effective space for payload in 6LoWPAN frame. HC1 [5], HC1g [12], and IPHC [11] [13] are the techniques for compressing IPv6 header. In the process of compression, the header fields whose values could be derived again at some other layer are elided from main header. When compressed header along with frame reaches the destination, the elided header fields are derived by adaptation layer. These fields are mainly derived from the link layer header which is carried in an 802.15.4 frame. In hop by hop communication, the process of compression and decompression takes place at every hop. IPv6 header and UDP header is compressed by Adaptation layer. HC1 and IPHC are explained in next sections.

*4.2.1. IPv6 Header compression by HC1*

HC1 was first header compression technique for 6LoWPAN suggested in RFC 4944 [5] in year 2007 for compressing IPv6 header. HC1 is acronym for Header Compression 1. In place of 40 bytes of IPv6 header, 2 bytes are used which indicates the way IPv6 header is compressed and from where its values can be recovered [5] [15]. The 2 bytes in this encoding are dispatch header and HC1 as illustrated in Fig. 5.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | SA | | DA | | T | NH | | HC2 |
| Dispatch Header (8 Bits) | | | | | | | | HC1 (8 Bits) | | | | | | | |

Fig.5. 6LoWPAN_HC1 Encoding

Dispatch header indicates which header will be coming next. Bit numbers 0-1 in dispatch header indicates IPv6 header will be followed after it. Bit numbers 6-7 of first byte has value 10 are indicative of the presence of compressed IPv6 (HC1) header. Source address (SA) and destination address (DA) fields in HC1 header are fetched from link layer address. Field T represents traffic class and flow label which is 0 in HC1. Field next header (NH) indicates which header will be followed by compressed IPv6 header. NH field can have 4 values i.e. 00, 01, 10 and 11 representing uncompressed header, UDP, TCP and ICMP respectively. HC2 field indicates the way next header arrives. If HC2 is 0, then next header is uncompressed, otherwise compressed. Hop limit is uncompressible field in this technique hence it has to be carried inline.

HC1 technique works well when addresses are link local but doesn't support compression when global unicast & multicast are there [12]. Hence, global multicast address is carried inline (128 bits of address without compression). Hop limit is uncompressible field in this technique [10]. As HC1 technique doesn't support compression of global addresses hence HC1g, extension of HC1 was suggested [5]. HC1g is acronym for header compression of global address. It can compress global unicast address. In this technique, 6LoWPAN network is assigned a single global address prefix of 64 bits in length. When source or destination address matches this assigned default value, it can be compressed by eliding the prefix [12]. To compress global unicast and multicast addresses, a better approach was suggested i.e. IPHC [13].

### *4.2.2. IPv6 Header compression by IPHC*

Header compression technique which can compress link local, global unicast and multicast IPv6 addresses is Internet Protocol Header Compression (IPHC) [13]. This encoding could be of 2 bytes (in link local communication), 3 bytes (when additional context encoding is present). IPHC encoding [12] is shown in Fig.6.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | **1** | **1** | **TF** | | **NH** | **HLIM** | | **CID** | **SAC** | **SAM** | | **M** | **DAC** | **DAM** | |
| **Byte 1** | | | | | | | | **Byte 2** | | | | | | | |

Fig.6. IPHC encoding

Bits numbers (0-2) is dispatch header and its value 011 indicates a compressed IPv6 header using IPHC encoding. The bit value 1 in next header (NH) field indicates that next header is compressed using IPHC_NHC compression technique [13]. HLIM is Hop limit and can have the values 00, 01, 10 or 11 representing hop limit either carried inline, 1, 64 or 255 respectively. Source address compression (SAC) and destination address compression (DAC) fields represents whether compression is stateless or context based when value is 0 or 1. Source address mode (SAM) and destination address mode (DAM) indicates the number of address bits which are to be carried inline. The destination address is unicast when M=0 and multicast when M=1.

When CID=1 then byte 3(Context Identifier Extension) [13] [14] will follow the DAM bits. Context identifier extension of IPHC is shown in Fig. 7. Byte 3 identifies the pair of contexts to be used when the IPv6 source and/or destination address is compressed. There are 4 bits each for source and destination context identifier which makes possibility of total 16 contexts. The context used for encoding source address and destination address need not to be same. SCI and DCI are used to identify the prefix which is used when source and destination address are state full & compressed.

| 0 1 2 3 | 4 5 6 7 |
|---|---|
| Source context Identifier (SCI) | Destination context Identifier (DCI) |
| Byte 3 | |

Fig.7. Context identifier extension of IPHC

IPHC encoding can have maximum of 16 contexts. As mentioned above, when communication with external network takes place then CID =1 and one extra byte for context extension is to be carried which is an overhead on the header. Context extension field carries SCI and DCI, where the value of SCI is redundant value as its value can be derived from edge router address because prefix is same for source address and edge router [15]. Comparison among various techniques of header compression is well explained in [14].

*4.3. Mesh Under Routing*

In TCP/IP protocol stack, routing is considered as one of the main functions of network layer but in 6LoWPAN stack, it is also handled by adaptation layer. In 6LoWPAN when routing decisions takes place at network layer it is called route over routing and when routing is at adaptation layer it is mesh under routing. Packets reach the destination through multi hopping as transmission range of nodes is less as mentioned in section I. In mesh under routing, with every packet mesh addressing header is attached. This helps in transmission of packets over multiple hops. Mesh addressing header as shown in Fig. 8 has mainly three fields which are Hop limit, Source address and Destination address [10].

| 1 | 0 | S | D | HL | Source Address | | Destination Address | |
|---|---|---|---|---|---|---|---|---|
| Byte 1 | | | | | Byte 2 | Byte 3 | Byte 4 | Byte 5 |

Fig.8. Mesh Addressing Header

The size of this header ranges between 5 and 17 bytes depending on the addressing modes in use. The third and fourth bits in byte 1 indicate which addressing mode (16 bit short or 64 bit extended address) to use for the source and destination addresses. Hop limit (HL) is 4 bits in length and it indicates the maximum number of hops a packet can go through. Its value will decrement at every hop. When value of hop limit becomes 0, the node will discard the packet. In multi hoping, the fragments belonging to same packet may take different routes. When all fragments reaches successfully, they are reassembled by adaptation layer and an IP packet is formed. Otherwise source will have to retransmit the fragments.

If a packet is fragmented by adaptation layer then mesh addressing header is send along with every fragment [10]. The router may route the fragments of same packet through different routes. In mesh under routing when all fragments reach the destination node, then they are reassembled by adaptation layer present there. If any of the fragments belonging to a packet is missing, then all fragments are to be retransmitted [16] [17].

In route over routing the reassembly of fragments is done at every hop. If any of the fragments is missing then all are to be retransmitted to that hop. Adaptation layer forms the complete packet and checks the destination node of it. If packet is for that node then it is forwarded to the network layer which further forwards it to the upper layer i.e. transport layer. Otherwise the network layer forwards that packet to the next hop [16]. Same procedure is repeated until packet reaches its destination node whose address is given in mesh addressing header. When source node and destination node are connected directly and data can be received by receiver without any hop then there is no need to attach mesh addressing header. It can be elided from the 6LoWPAN frame.

In 6LoWPAN, there exists number of routing protocols like 6LoWPAN Ad-hoc On-Demand Distance Vector (LOAD), Multipath based 6LoWPAN Ad-hoc On-Demand Distance Vector (MLOAD), Dynamic MANET On-Demand for 6LoWPAN Routing (DYMO-Low), Hierarchical Routing (Hi-Low) and Extended Hi-Low [7] [17]. As 6LoWPAN is light weight protocol due to restrictive resource usage need hence its incorporating routing protocols are also simplified to support this requirement. 6LoWPAN LOAD routing protocol is a simplified On-demand routing protocol based on AODV [RFC3561] for 6LoWPAN. The Adhoc On Demand Distance Vector (AODV) routing algorithm is a routing protocol designed and used for ad hoc mobile networks. Both unicast and multicast routing can be handled by AODV. It is an on demand algorithm, which means routes between nodes are formed only as required by source nodes. The routes are maintained till need is not complete. The Dynamic MANET On-demand (DYMO) for 6LoWPAN routing protocol is used by mobile nodes in wireless networks especially where multi-hoping is required. It determines unicast routes between nodes within the network hence it is adaptive to the mobile network. Hierarchical routing protocol is based on 16-bit short address of a 6LoWPAN. Details of these routing protocols are beyond the scope of this paper.

## 5. Conclusions

6LoWPAN has emerged to transmit the IPv6 packets over low power wireless personal area network. For performing the same, some changes have been done in TCP/IP protocol stack. The prominent change is the addition of adaptation layer in between network layer and data link layer of TCP/IP stack. In this paper we have reviewed and analysed the need of the presence of adaptation layer. We have also presented the various functions handled by adaptation layer to make its presence in 6LoWPAN protocol stack as a necessity. Simulation is the best way to show adaptation layer functions i.e. IPv6 header compression, fragmentation and reassembly of packet. Our future work includes simulation of wireless network topology with low power sensor devices in NS3. In that we will show the comparative results using different header compression techniques.

## References

[1] R. Hinden and S. Deering. Internet Protocol Version 6 (IPv6) Addressing Architecture. RFC 3513. [Online]. Available: http://tools.ietf.org/ html/ rfc3513

[2] B. A. Forouzan, "Network Layer: Internet Protocol", in *Data Communication & Networking*," 4th ed. New York: McGraw-Hill, 2007, ch. 20, pp. 582-597.

[3] Z. Shelby and C. Bormann, "Introduction," in *6LoWPAN: The Wireless Embedded Internet*, 1 ed. United Kingdom: Wiley, 2009, pp. 3-11.

[4] N. Kushalnagar, G. Montenegro and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," RFC 4919, IETF network working group, Aug. 2007.

[5] G. Montenegro, N. Kushalnagar, J. Hui and D. Culler,"Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC 4944, IETF network working group, Sep.2007.

[6] Jonathan W. Hui and D. Culler,"Extending IP to Low-Power, Wireless Personal Area Networks," *IEEE Internet Comput.* vol. 12, Issue 4, pp. 37-45, July-Aug. 2008.

[7] N. H. A. Ismail, R. Hassan and K. W. M. Ghazali, "A Study on protocol stack in 6LoWPAN model," JATIT, vol. 41 issue 2, pp. 220-229, 2012.

[8] Jonas Olsson," 6LoWPAN Demystified," Texas Instruments, October 2014.

[9] A. Ludovici, A. Calveras and J. Casademont, "Forwarding Techniques for IP Fragmented Packets in a real 6LoWPAN network," Sensors, pp. 992-1008, 2011.
[10] J. Hui, D. Culler and S. Chakrabarti, "6LoWPAN: Incorporating IEEE 802.15.4 into the IP architecture", Internet Protocol for Smart Objects (IPSO) Alliance, White paper # 3, Jan 2009.
[11] A. Ludovici, A. Calveras, M. Catalan, C. Gómez, and J. Paradells, "Implementation and Evaluation of the Enhanced Header Compression (IPHC) for 6LoWPAN," EUNICE 2009, LNCS 5733, pp. 168–177, 2009. © Springer-Verlag Berlin Heidelberg 2009.
[12] J. Hui and D. Culler, "Stateless IPv6 Header Compression for Globally Routable Packets in 6LoWPAN Sub networks," draft-hui-6lowpan-hc1g-00, June 2007.
[13] J. Hui and P. Thumbert, "Compression format for IPv6 datagrams over IEEE 802.15.4 based networks," RFC 6282, IETF network working group, Sep. 2011.
[14] R. Garg and S. Sharma,"Comparative Study on Techniques of IPv6 Header Compression in 6LoWPAN" Proc. of the Intl. Conference on Advances in Information Processing and Communication Technology - IPCT 2016, Italy, pp. 34-38, ISBN: 978-1-63248-099-6 doi: 10.15224/ 978-1-63248-099-6-33
[15] W. Huiqin and D. Yongqiang, "An Improved Header Compression Scheme for 6LoWPAN Networks," in Ninth International Conference on Grid and Cloud Computing, © 2010 IEEE DOI 10.1109/GCC.2010.74
[16] A. H. Chowdhury , M. Ikram, H. S. Cha, H. R., S.M. Saif Shams, K.H. Kim, S.W. Yoo, "Route-over vs Mesh-under Routing in 6LoWPAN," IWCMC, Germany, pp. 1208-1212, June 2009.
[17] G. K. Ee, C. K. Ng, N. K. Noordin, B. Mohd. Ali, "A Review on 6LoWPAN Routing Protocols," *Proc*. Asia Pacific Advanced Network (APAN) Vol. 30, pp. 71-81, 2010.

**Authors' Profiles**

**Ruchi Garg,** completed her Masters of Computer Applications degree from Barkatullah University, Bhopal in 1999. She is having teaching experience of 12 years. Currently she is a research scholar at Maulana Azad National Institute of Technology (MANIT), Bhopal. Her research interest includes low power wireless sensor networks. Ruchi Garg is member of Institute of Research Engineers and Doctors (IRED).

**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 in MANIT Bhopal. His research interests include Wireless and MANET's, Advanced Computer Networks, Next Generation Networks & IPv6 and Cloud Computing. Prof. Sharma is the member of Computer Society of India (CSI), Indian Science Congress Association and IACSIT.

**How to cite this paper:** Ruchi Garg, Sanjay Sharma,"A study on Need of Adaptation Layer in 6LoWPAN Protocol Stack", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.7, No.3, pp.49-57, 2017.DOI: 10.5815/ijwmt.2017.03.05