Proc. of the Intl. Conference on Advances in Information Processing and Communication Technology - IPCT 2016.

ISBN: 978-1-63248-099-6 doi: 10.15224/ 978-1-63248-099-6-33

Comparative Study on Techniques of IPv6 Header Compression in 6LoWPAN

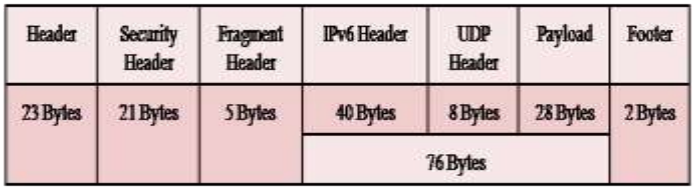
Ruchi Garg, Sanjay Sharma

***Abstract*--Low-power wireless personal area network (LoWPAN) consists of large number of resource constraint devices connected over a wireless link with the aim of gathering real time information and transmitting it to desired application and vice versa. This requires transmission of IPv6 packets over Low-power wireless personal area network and is called 6LoWPAN. The large sized headers like IPv6, TCP, and UDP consume most of the packet space leaving only 28 bytes for actual payload. MTU of 802.15.4(IEEE standard for wireless network) is only 127 bytes much less as compare to IPv6 packet which is of 1280 bytes. Hence Header compression and fragmentation becomes a necessity in 6LoWPAN so as to provide reasonable number of bits for payload. This paper depicts 6LoWPAN protocol stack and various header compression techniques for compressing IPv6 header. It also shows comparison among various techniques on the basis of total number of bits needed in compressed IPv6 header under various scenarios.**

***Keywords*-- 6LoWPAN, header compression, IPv6 header, adaptation layer, fragmentation**

I. Introduction

Internet of Things (IoT) [1] is not a buzz word of present rather it is going to be Internet of Everything (IoE) in near future which will require billions of devices to communicate with each other in the physical world. To cater this need every device in the network has to be provided with its own IP address for which IPv6 addressing is considered best as it can provide up to 2^{128} unique addresses [2]. Transmission of IPv6 packets over Low-power wireless personal area network is technically abbreviated as 6LoWPAN [3]. The standards and specifications of 6LoWPAN were first released by Internet Engineering Task Force (IETF) in year 2007 [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 device memory of 16 kb RAM and 128kb ROM [5]. IPv6 packet size is of 1280 bytes whereas in wireless network the allowable packet size is 127 bytes. Out of 127 bytes, 23 bytes is for link layer header, 21 bytes for security header, 5 bytes for fragment header and 2 bytes for footer [6]. This leaves only 76 bytes for upper layer headers and payload, as illustrated in Fig.1.



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			

Figure 1. 802.15.4 Frame Format

Headers of IPv6, TCP and UDP are of 40, 21 and 8 bytes in length respectively. If transmission of IPv6 packet takes place using TCP then only 16 bytes are left for payload and in case of UDP, only 28 bytes for actual payload as shown in Fig. 1. Hence compression of IPv6 and UDP/TCP header is needed so as to increase the bytes available for payload. In 6LoWPAN, UDP is preferred over TCP because of its less complex and small size header. 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 desirable.

This paper shows the various compression techniques for compressing IPv6 header under different scenarios. The rest of the paper is organized as follows: 6LoWPAN protocol stack is shown in section II. IPv6 header format is depicted in section III. Section IV covers various header compression techniques along with their limitations. Comparison of all the techniques is presented section V. Finally conclusion is given in section VI.

II. 6LoWPAN protocol stack

As explained in section I 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 as shown in Fig.2. It handles fragmentation, reassembly of packets, header compression, decompression and mesh under routing over multiple hops [6] [7].

Ruchi Garg, Sanjay Sharma
Department of Mathematics & Computer Applications
Maulana Azad National Institute of Technology, Bhopal
India

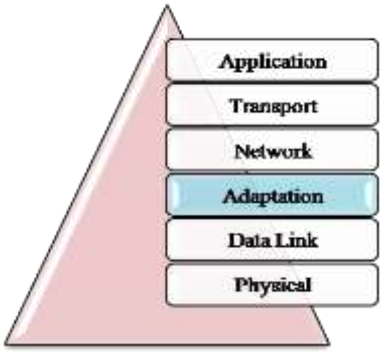
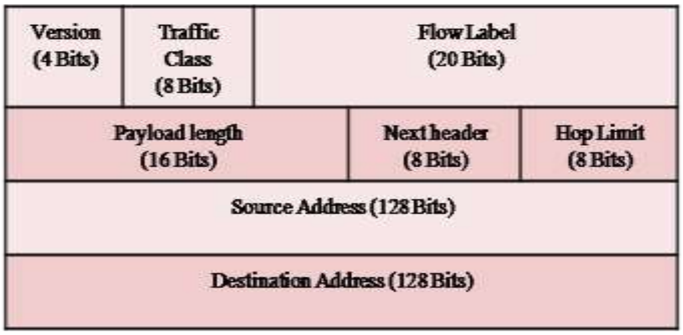


Figure2. 6LoWPAN Architecture

Similar to IPv6, 6LoWPAN also have stacked headers. It contains three sub headers i.e. Mesh Addressing, Fragmentation & IPv6 compressed header. Mesh addressing header [5] is required to forward 6LoWPAN packets through several hops. Fragmentation header [5] is attached with every fragment of the packet. Headers Mesh Addressing & Fragmentation are not always required. Explanation of these headers is not covered in this paper.

III. Format of IPv6 header

IPv6 header [8] has eight fields as shown in Fig. 3.



Version (4 Bits)	Traffic Class (8 Bits)	Flow Label (20 Bits)
Payload length (16 Bits)	Next header (8 Bits)	Hop Limit (8 Bits)
Source Address (128 Bits)		
Destination Address (128 Bits)		

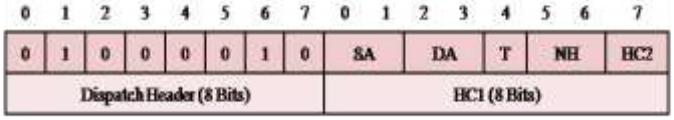
Figure 3. IPv6 Header

The first field version indicates the version number of the IP and its value is 6. Traffic class field is used to give priority of the packet and used during traffic congestion. Flow label field provides handling for a particular flow of data. Payload length field of 16 bits indicates the length of the IP packet. The next header is an 8-bit field used to indicate the header that will be followed by IPv6 header. 8-bit hop limit field indicates the maximum number of hops for a packet while being transmitted from source to destination node. The source address field is of 128-bit which indicates the Internet address of source node from where the packet is transmitted. The destination address field is 128-bit long Internet address which indicates the final destination.

IV. Various techniques of IPv6 Header Compression

A. *6LoWPAN_HC1*

HC1 was first IPv6 header compression technique for 6LoWPAN suggested in RFC 4944 [5] in year 2007. 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 during decompression. The 2 bytes in HC1 encoding are dispatch header and HC1 as illustrated in Fig.4.



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)							

Figure 4. 6LoWPAN_HC1 Encoding

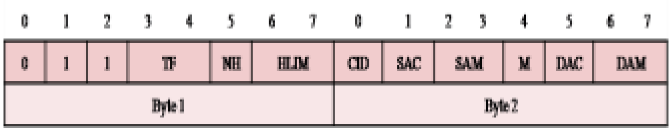
Dispatch header indicates which header will be coming next. Bit numbers 0-1 in dispatch header indicates IPv6 header will be followed after it. Bit numbers 6-7 has value 10, are indicative of the presence of compressed IPv6 (using HC1) header. Source address (SA) and destination address (DA) fields in HC1 header can be compressed to 0 bits in case of link local address. In such a case the prefix is inferred from link local prefix and interface identifier from link layer address at the time of decompression. Field T represents traffic class and flow label which could be 0/1 in HC1. If the value of T is 0 then values of traffic class and flow label are carried inline otherwise fully elided. Field next header (NH) indicates which header will be followed by compressed IPv6 header. Next header 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 using HC2 encoding technique [5]. Hop limit is uncompressible field in this technique hence it has to be carried inline.

HC1 technique works well for link local addresses but doesn't support compression when global unicast & multicast addresses are there [9]. Hence, global address is carried inline (128 bits of address without compression).

B. *6LoWPAN_IPHC*

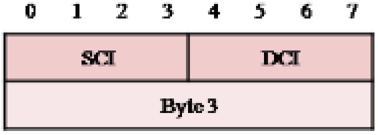
Header compression technique which can compress link local, global unicast and multicast IPv6 addresses is Internet Protocol Header Compression (IPHC) [12]. This encoding could be of 2 bytes (in link local communication), 3 bytes (when additional context encoding is present). IPHC encoding [13] is shown in Fig.6 and Fig.7.

Proc. of the Intl. Conference on Advances in Information Processing and Communication Technology - IPCT 2016.
Copyright © Institute of Research Engineers and Doctors. All rights reserved.
ISBN: 978-1-63248-099-6 doi: 10.15224/ 978-1-63248-099-6-33



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							

Figure 6. IPHC encoding



0	1	2	3	4	5	6	7
SCI				DCI			
Byte 3							

Figure 7. Context identifier extension of IPHC

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 [12]. 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) [12] will follow the DAM bits. 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.

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 [9].

C. *S&SFHC*

S&SFHC is the acronym for Second and Subsequent Fragment Header Compression. This header compression technique is suggested when packets need to be fragmented. Ideally IPv6 header is to be carried with all the fragments belonging to same packet. In S&SFHC the header from first fragment is stored in the header dictionary [15]. Header Dictionary keeps a copy of a header that is being received in the first fragment. Receiver sends a Link Unique Identifier (LUI) of 8 bits to sender from its Free Unique Identifier List (FUIList). Sender replaces the IPv6 header by that identifier in its subsequent fragments. S&SFHC uses two approaches i.e. Standalone and Integrated as shown in fig. 8. In Standalone approach, the header in first fragment is send without any compression. In subsequent fragments header is replaced by S&SFHC compressed header i.e. LUI send by receiver.

In Integrated approach [16] the header in first fragment is compressed using IPHC technique and by S&SFHC compressed header in subsequent fragments. The simulation results shown in the paper [16] shows improvement in integrated approach as compared to Standalone approach.

Uncompressed Header	Frag 1 Payload	S&SFHC header	Frag 2 payload	Further fragments

IPHC/HC1 Header	Frag 1 Payload	S&SFHC header	Frag 2 payload	Further fragments

Figure 8. S&SFHC standalone and Integrated approach [15]

V. Comparison of Header Compression Techniques

Various header compression techniques are studied and analyzed in section IV. Comparison of all techniques based on various points is shown in this section.

A. *Comparison on the basis of source and destination address bits*

As shown in Table I, HC1 cannot compress global unicast and multicast addresses hence 128 bits each of source & destination addresses are carried inline. In IPHC encoding SA & DA bits send inline could be 64/16/0. This is the case when SAC = 1 and SAM has 01/10/11 bit values respectively. At the time of decompression the address is formed using the context information and the bits which are carried inline. Same applies for destination address when M=0, DAC = 1 & DAM bits values are 01/10/11. When M=1 and DAC = 1, then this is the case of multicast context based destination address compression. The compressed multicast address may be of any one form as given below for different values of DAM where X are the nibbles which are carried inline.

FFXX :: 00XX:XXXX:XXXX	48 bits inline
FFXX :: 00XX:XXXX	32 bits inline
FF02::00XX	8 bits inline

In case of global unicast and multicast addresses, IPHC outperforms HC1.

TABLE I. COMPARISON ON THE BASIS OF SOURCE & DESTINATION ADDRESS BITS CARRIED INLINE

IPv6 header field	Uncompressed IPv6	HC1	IPHC
Link-Local (Source/Destination)	128	0	0
Global Unicast (Source/Destination)	128	128	0/16/64
Global Multicast (Destination only)	128	128	8/32/48

B. *Comparison on the basis of IPv6 header field bits carried inline*

Version field bits are completely elided in HC1 & IPHC because in 6LoWPAN the version is always IPv6. In HC1 traffic class and flow label fields are either totally elided or fully inline whereas in IPHC these fields are compressed to various degrees (0/8/24) according to application need. Hop limit is an uncompressible field in HC1 whereas this field is compressible in IPHC. Payload length field which is of 2 bytes in IPv6 header is elided in HC1and IPHC as at the time of decompression this value can be derived from data link layer header. From table II it is clear that IPHC carries less number of bits than HC1 in traffic class, flow level and hop limit fields.

TABLE II. COMPARISON ON THE BASIS OF IPV6 HEADER FIELD BITS CARRIED INLINE

IPv6 header field	Uncompressed IPv6	HC1	IPHC
IP version	4	0	0
TC&FL	28	0/32	0/8/24
Payload length	16	0	0
Next header	8	0	0
Hop limit	8	8	0

C. *Comparison on the basis of various other parameters*

HC1 performs stateless header compression in which no flow state of the link is maintained where as in IPHC statefull header compression is done. Statefull header compression is context based compression which maintains the flow of the data transfer between source and destination. HC1 does not support the presence of extension headers because HC1 encoding bits are immediately followed by HC2 encoding bits. IPHC supports extension headers as in both encodings uncompressed IPv6 header fields comes after IPHC encoding. Header compression 2 (HC2) [5] and Next header Compression (NHC) [12] are UDP header compression techniques used in HC1 and IPHC respectively.

TABLE III. COMPARISON ON THE BASIS OF VARIOUS OTHER PARAMETERS

Parameter	HC1	IPHC
Type of compression	Stateless	Stateless /Statefull
Hop Limit	Carried inline	Elided
Extension headers	No support	Full support
Next header encoding	HC2	NHC
Checksum of UDP header	Carried inline	Elided

D. *Comparison on the basis of performance analysis measures*

The results in table IV are on the basis of implementation results shown in paper [13] [16]. Simulations results shows that S&SFHC outperforms in throughput and packet delivery ratio. IPHC consumes more memory than HC1.

TABLE IV. COMPARISON ON THE BASIS OF PERFORMANCE ANALYSIS MEASURES

Performance criteria	Results
Throughput	S&SFHC>IPHC>HC1>Uncompressed
Energy	Uncompressed>HC1>IPHC>S&SFHC
Packet delivery ratio	S&SFHC>IPHC>HC1>Uncompressed
Memory usage	Uncompressed>IPHC>HC1>S&SFHC

VI. Conclusion and Future work

By using header compression techniques, the size of IPv6 header can be greatly reduced resulting in more number of bytes to carry payload in 6LoWPAN. When address is link local then it can be completely elided in all techniques as it can be derived from link layer header. IPHC is suited for global unicast and multicast address. According to comparison table IV, IPHC consumes more memory than HC1. Hence in Integrated S&SFHC we aim towards designing a “Type of address” based algorithm. If address is link local then Integrated S&SFHC approach will use HC1 technique otherwise IPHC. Also S&SFHC approach uses Automatic Repeat Request (ARR) algorithm to keep fragments in order. The advantage of ARR is that sequence of fragments is being maintained but it causes delay as it follows stop & wait mechanism. In real time transmission, delay should be as minimum as possible. Hence in our future work we will propose to use fragment header in place of ARR algorithm to minimize delay. We aim towards implementation of these compression encodings by incorporating our proposed work and its simulation in NS3.

Proc. of the Intl. Conference on Advances in Information Processing and Communication Technology - IPCT 2016.
Copyright © Institute of Research Engineers and Doctors. All rights reserved.
ISBN: 978-1-63248-099-6 doi: 10.15224/ 978-1-63248-099-6-33

References

[1] G. Feller, "The Internet of Things: In a Connected World of Smart Objects," Accenture & Bankinter Foundation of Innovation, (2011).

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

[3] Z. Shelby, "6LoWPAN: The Wireless Embedded Internet," Wiley; 1 edition December 21, (2009).

[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", RFC4944, IETF network working group, Sep.2007.

[6] A Study on protocol stack in 6LoWPAN model by Nurul Halimatul Asmak Ismail, Rosilah Hassan, Khadijah W. M. Ghazali, 2012

[7] 6LoWPAN Demystified by Jonas Olsson, Texas Instruments, October 2014.

[8] B. A. Forouzan, *Data Communication & Networking*, 4th ed.

[9] Wang Huiqin, Dong Yongqiang, "An Improved Header Compression Scheme for 6LoWPAN Networks", Ninth International Conference on Grid and Cloud Computing, 2010.

[10] Jonathan Hui, David Culler, Samita Chakrabarti, "6LoWPAN: Incorporating IEEE 802.15.4 into the IP architecture", Internet Protocol for Smart Objects (IPSO) Alliance, White paper # 3, Jan 2009.

[11] J. Hui, D. Culler, "Stateless IPv6 Header Compression for Globally Routable Packets in 6LoWPAN Sub networks", draft-hui-6lowpan-hc1g-00, June 2007.

[12] 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.

[13] Alessandro Ludovici, Anna Calveras, Marisa Catalan, Carles Gómez, and Josep 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.

[14] EFFNET AB, "An Introduction to IPv6 Header Compression." white paper (February 2004), http://www.effnet.com/sites/effnet/pdf/uk/Whitepaper_Header_Compres sion.pdf.

[15] Samer A.B. Awwad, Chee Kyun Ng, K. Noordin, B. M. Ali, F. Hashim, "Second and Subsequent Fragments Headers Compression Scheme for IPv6 Header in 6LoWPAN Network", seventh international conference on sensing technology, IEEE, 2013.

[16] Samer A.B. Awwad, Chee Kyun Ng, K. Noordin, B. M. Ali, F. Hashim,"The Integrated versus Standalone Operation Mode for Second and Subsequent Fragments Headers Compression Scheme in 6LoWPAN", Sensing technology: current status and future trends, Springer, 2015

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.