International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 4, No.2 (Apr-2015)

http://dx.doi.org/10.12785/ijcds/040206

Survey of Header Compression Techniques over Multiprotocol Label Switching (MPLS)

Mohammad Ahsan Chishti[1] and Ajaz Hussain Mir[2]

[1] *Computer Science & Engineering Department, National Institute of Technology Srinagar, Kashmir, India*
[2] *Electronics & Communication Engineering Department, National Institute of Technology Srinagar, Kashmir, India*

Received 20 Dec.2014, Revised 20 Feb. 2015, Accepted 15 Mar. 2015, Published 1 Apr. 2015

Abstract: Due to the rapid evolution of Internet as well as services over the Internet, including high bandwidth consuming applications like audio and video streaming, it has become need of the day to enhance the Internet infrastructure for bandwidth efficiency. Developed by the Internet Engineering Task Force, MPLS allows networks to offer several services on the single network architecture with improved forwarding speed of routers by solving problem of longest prefix match in IP networks. In addition to this, it offers Traffic Engineering, Virtual Private Networks and Quality of Service guarantees. MPLS in combination with Internet Protocol version 6 (IPv6) has been seen as a technology for next generation Internet, which can revolutionize the Internet with speed, services and functionality. IPv6 provides huge number of addresses and guaranteed support for the ever increasing demand. However, there is a disadvantage of bigger packet header size compared to the payload size which leads to excessive overhead in case of real-time multimedia applications. Bandwidth can be conserved by reducing the amount of redundant IP header transmitted with every packet for the same packet stream through header compression/suppression techniques. The objective of this paper is to survey of various header compression technologies which can be implemented over MPLS with IPv6 as the addressing protocol, which can be used as a bandwidth conserving technology. The current efforts in the area, both standardised as well as ongoing research has been discussed in detail and also the problems that are yet to be addressed are examined.

Keywords: Header Compression, Suppression, MPLS, IPv6, CRTP, VJHC, IPHC, ROHC, SCPS, RTPHC, ECRTP, QoS.

1. INTRODUCTION

This With the rapid growth of Internet and with evolving new technologies like next generation Internet Protocol (IP) addressing and increasing demands on real-time multimedia services, the Quality of Service (QoS) requirements over existing infrastructure have accordingly increased. QoS is the overall performance of network. It is a set of technologies that enables network administrators to manage the effects of congestion on traffic flows by using network resource optimally rather than by conditionally adding extra capacity [1]. It represents the set of techniques necessary to manage network bandwidth, delay, jitter, and packet loss which are commonly used parameters [2] [3] and QoS is a major concern for the ISPs in supporting multimedia applications [4]. The two generally used QOS approaches are: Integrated Service (IntServ) and Differentiated Service (DiffServ). IntServ framework aimed at providing per-flow QoS guarantees to individual application sessions assigning end applications to request the QoS they require from routers along their data path using Resource Reservation protocol (RSVP) [5]. However, IntServ suffers from scalability problems because of excessive overhead where as DiffServ is more scalable, manageable and easy deployable for service differentiation in IP networks [6] [7].

Traditional IP forwarding is based on Layer 3 destination address with lookups at every hop. There is drawback associated with this routing that is of the destination address based lookup is needed at every hop. Moreover, IPv6 [8] is the next generation protocol for networks which has a bigger size of header, increased to 128 bits from 32 bits of IPv4. In addition to this, introduction of flow label field was further major change in IPv4 header for QoS purpose [9]. In addition to huge address space, IPv6 puts forward an important enhancement with respect to built-in security, mobility, auto-configuration and enhanced multicast support [10].

MPLS (Multi-Protocol Label Switching) became popular due to its advantage of fast forwarding in its early time, which is no more an advantage due to the capacity of fast forwarding by IP Layer 3 routers. However, now the main advantages of MPLS is providing unified

E-mail: ahsan@nitsri.net, ahmir@nitsri.net

network architecture, BGP free core, QoS, Traffic Engineering, optimal traffic flow etc [11]. MPLS is called Layer 2.5 as it is performed between the L2 and L3 network. IPv6 over MPLS is considered as an able mixture of protocols on layer 2 and layer 3 for routing of packets.

Various header compression and suppression technologies have been proposed to compress the UDP, IP, RTP and TCP headers. The main purpose of these compression technologies like Compressed Real-Time Protocol (CRTP), Internet Protocol Header Compression (IPHC), Van Jacobson Header Compression (VJHC), Robust Header Compression (ROHC) etc. is to improve link efficiency for network which is in terms of reduced bandwidth consumption. Many approaches have been proposed and implemented with respect to Header Compression. However, a lot more needs to be done for header compression over MPLS.

This paper gives a detailed survey of the Header Compression mechanisms that can be implemented over MPLS to improve the Quality of Service and thereby improving the overall performance of the network, in addition to improving link efficiency for network. Also, due to the excessive overhead of the Next Generation IP addressing protocol (IPv6), the need to header compression/Suppression becomes necessary. The aim is to provide the researchers of QOS over MPLS an easy way to understand the essence of header Compression and implement the various header compression technologies which are in place. The various related works have been researched and discussed so that the researchers can easily get the idea of the state of art and the future possible work in this field.

The outline of this survey paper is organized as follows; this section is the introduction to the paper with the background, contribution of this research and scope of this paper. Introduction of the fundamentals and architectures of QoS, IPv6 and MPLS are explained. The next section is the details of Multi Protocol label Switching (MPLS), its architecture and working. The third section with the next generation IP addressing i.e. IPv6 over MPLS explained. Section 4 is dedicated for Header Compression techniques which have been explained in detail, with the types, their advantages and disadvantages. Finally, the last section concludes the survey work with possible future work.

2. MULTI PROTOCOL LABEL SWITCHING (MPLS)

Multiprotocol Label Switching (MPLS) [12] [13] [14] has been introduced as the essential technology for the next-generation packet networks. The main factor for evolution of MPLS is high speed packet switching, forwarding and large scalability which helps Internet Service Providers (ISPs) to offer several services on the single network architecture [15]. It was intended to improve the forwarding speed of routers; however it is offering several important technologies like Traffic Engineering, Virtual Private Networks (VPN), routing performance etc. available at low cost and with minimum configuration overhead [16] [17] [18]. In addition, MPLS can provide QoS guarantees [19] [20] with ability of one-to-many connection, solving the problem of performance bottleneck due to longest prefix match in IP networks. It solves the excessive overhead of network management in IP and problems with overlay models like IP over ATM. MPLS is viewed by some as one of the most important network developments of the 1990's [21]. MPLS allows routing with QoS restrictions, using signalling protocols like Constraint Based Routing over Label Distribution Protocol (CR-LDP) or Reservation Protocol (RSVP) to establish the path adapted to QoS's restrictions [22] [23] [24].

The idea that MPLS is faster than IP is a no more a valid reason because of the fact that nowadays Application-Specific Integrated Circuits (ASIC) are used in routers making the packets switched as fast as that of a label. However, by using MPLS, it will enable carrying protocols other than IP, known as Any Transport over MPLS (AToM) [25] [26] and better IP over ATM integration, in addition to optimal traffic flow and traffic engineering [11].

All routers in the network must be MPLS-enabled, in order to apply MPLS to an existing IP network [27]. MPLS technology is called Layer 2.5 technology as it functions between the Layer 2 (Data Link Layer) and Layer 3 (Network Layer), since it is a packet forwarding technology that's capable of layer 3 to layer 2 route mapping [28]. The idea is to use MPLS labels of 32 bit length instead of longer IP addresses (32 bits in Internet Protocol version 4 and 128 bits in Internet Protocol version 6) in switching of packets. Fig.1 depicts the syntax of MPLS Label [13] [29].

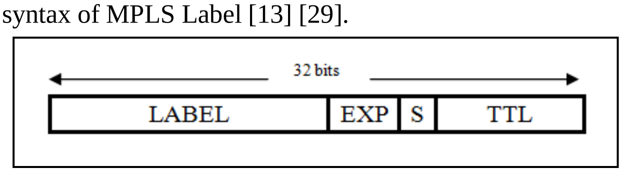

Figure 1. MPLS Label

MPLS header, also known as Shim Header, is inserted between the layer-2 header and layer-3 header as shown in Fig. 2. MPLS header is fragmented into 4 fields: Label (20 bit), EXP (3 bits), S or BoS (1 bit) and TTL (8 bits). Label is used for lookup and gives the next hop to which the packet is to be forwarded and the operation to be performed on the label stack. EXP is experimental bits, reserved for Quality of Service (QOS). S or BoS (Bottom of Stack) is 0, unless this is the bottom label in the stack. TTL is Time-to-Live, used to avoid routing loop. It is decremented by 1 at each hop, shows how far the header could travel along the route [11] [29] [30] [31].

Layer 2 Header	Shim Header	Layer 3 Header

Figure 2. MPLS Shim header position

A label is used for making forwarding decisions in the MPLS network, rather than the IP destination address. Label Switch Router (LSR) is capable of understanding these labels and helps in forwarding of labelled packets. LSR has three types: Ingress LSR, responsible to add a label; Egress LSR, responsible to remove a label and Intermediate LSR responsible for correct switching of the packet. Ingress and Egress LSRs are Label Edge Routers (LER). A sequence of LSRs in MPLS network forms a Labelled Switched Path (LSP).

MPLS flows are connection-oriented and packets are routed along pre-configured LSPs, incorporating label swapping forwarding paradigm with network layer routing. When a packet enters MPLS domain, it is assigned a label by the ingress LER specifying the path that the labeled packet has to take within the MPLS domain. A different label is used for each hop, and it is chosen by the LSR performing the forwarding operation. At the egress, LSR receives the labeled packet, removes the label and forward them based on layer 3 addresses for normal IP routing. Fig. 3 shows an example of forwarding IP packets using MPLS [11] [32] [33] [34].

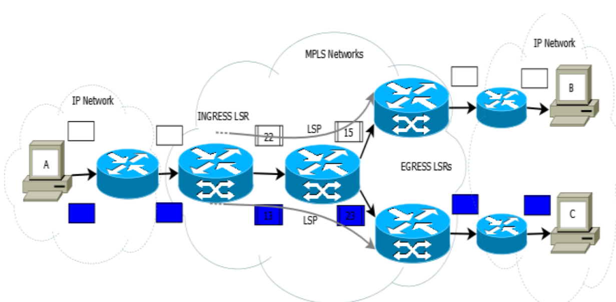

Figure 3. IP and MPLS Network

The network in Fig-3 has three subnets, with two IP based networks and one MPLS network having four core routers called Label Switch Routers (LSR) and two paths called Label Switch Paths (LSP). LSPs are unidirectional for each pair of LSRs. LSR that transmits with respect to the direction of data flow, given as a line with arrow pointing towards Egress router in the fig, is called upstream. LSR that receives the MPLS packet is called downstream. The MPLS edge routers are called E-LSR (Edge-LSR) with the first LSR denoted as Ingress Router and the last as Egress router.

Sender A intends to send traffic to destination B and C. A utilizes IP routing till it reaches MPLS network, after that LER classifies a packet into Forward Equivalence Class (FEC) and attaches a label. FEC is a subset of packets that are all treated in the same way by the router and are mapped to a label [34]. After assigning FEC, there is no need to further analyze the header by the successive routers thus improving the performance. To forward an unlabeled packet, MPLS first relates the FEC with an entry in its next hop forwarding equivalence class table. This table contains the operations like pop, push etc, the next hop and if needed, a new label. The ensuing table is called the label forwarding information base (LFIB) [12] [35].

LSR utilizes IP routing till it reaches the Ingress router, using the destination IP address to determine the next hop and initial label for each packet, given as 22 and 15 in the figure above. The next LSR after receiving the packet utilizes these labels to identify the LSPs to determine the next hops and labels. Once the packet reaches the egress routers, the label is popped off and IP routing takes the packet to the destination [32].

3. INTERNET PROTOCOL VERSION 6 (IPV6) OVER MPLS

Internet Protocol version 4 (IPv4) has been around since 1980s, which was intended to interconnect research universities and government facilities [36] but nowadays it is having tremendous scalability problems due to the phenomenal increase of Internet users, devices and applications growing at a rapid rate. The solution to this problem is IPv6, as it provides a larger address space, along with several other features like inbuilt security, Jumbograms, better traffic routing etc. IPv4 uses 32-bit addresses which can support about 4.3 billion devices and due to this reason the IPv4 address have depleted whereas IPv6 uses 128-bit addresses which can support 2 to the power of 128 that is approximately 3.4×10^{38} addresses. The IPv6 header is of 40 octets (bytes) divided into eight fields as shown in Fig. 4.

<table>
<tr><td>Version</td><td>Traffic Class</td><td colspan="2">Flow Label</td></tr>
<tr><td colspan="2">Payload Length</td><td>Next Header</td><td>Hop Limit</td></tr>
<tr><td colspan="4">Source Header</td></tr>
<tr><td colspan="4">Destination Header</td></tr>
</table>

Figure 4. IPv6 packet header format [37]

Three tuples of IPv6 header, which are the IP source address, IP destination address, and flow label, represent the IPv6 flow signature [38]. Traffic Class field in the IPv6 header is used to identify different classes or different priorities of IPv6 packets. Based on this class, the network forwards the packet. It is 8-bit field where in first 6 bits are used for differentiated service, which classifies the packet and last 2 bits for Explicit Congestion Notification (ECN), providing congestion control. It provides similar functionality to the IPv4 Type of Service (TOS) field [39] [40]. The first 6-bits can be used to create traffic classes for 64 distinct classes and for QoS as well as MPLS label identification. IPv6 has a same 20-bit label field known as Flow Label [41]. Table I shows systematic classification of how all fields in IPv6 header are expected to change.

TABLE I. EXPECTED CHANGES IN IPV6 HEADER [42]

Field	Change/No Change
Version	No Change
Traffic Class	No Change
Flow Label	No Change
Payload Length	Inferred
Next Header	No Change
Hop Limit	No Change
Source Address	No Change
Destination Address	No Change

MPLS does not define any new QoS architecture, but follows DiffServ architecture applied in the MPLS environment [43] [44] [45]. A flexible solution for support of DiffServ over MPLS network has been given by mapping between IP packets and FECs by the ingress router [46] [47] [48]. MPLS is independent from both network layer protocols and data link layer media [49]. MPLS infrastructure has minimal core impact to provide IPv6 services [50]. IPv6 over MPLS is considered to be the best available and most efficient combination of protocols on layer 2 and layer 3 for routing of packets with protocol transparency that can have minimal core impact to provide IPv6 services [51] [52]. MPLS labels and IPv6 labels serve different network functions, and they are not interchangeable because of the fact that MPLS labels are used to create connection-oriented Label Switched Paths (LSP) whereas IPv6 is a connectionless protocol.

MPLS labels are distributed by label distribution protocols and change at every hop whereas Flow Labels are used to identify end-user traffic and do not change. Also, various MPLS services use the shim header and if it is stacked on 40 Byte IPv6 header, it would be mammoth overhead [53] [54]. However as per [55], Flow Label of IPv6 can be used to hold the MPLS label without increasing the complexity of the model and all other shim header fields can be completely mapped into IPv6 headers by introducing IP Next Generation Label Switching (IPngLS). In addition to proving integration of MPLS and IPv6, IPngLS also decreases complexity by eliminating extra headers, no extra QoS mappings, as MPLS reserves only 3 bits to classify packets into QoS classes while IPv6 is fully compatible with the Differentiated Services. It is only suitable for IPv6 networks, still needing MPLS to interoperate with IPv4 networks. The mapping of the Label field from the MPLS shim header can be done on a 1-to-1 basis since both are 20-bit long [43] [55] [56] [57].

There are several approaches for implementing IPv6 over MPLS [58] [59] [60]:

i. IPv6 Provider Edge over MPLS: Extend edge LSRs only, with IPv4 routing and signaling on core LSRs. IPv6 packets transit MPLS network through LSPs which is originally established for IPv4 traffic by routing and signaling mechanism in IPv4 in 6PE model [49] [61]. This method relies on the distribution of IPv6 prefixes among the edge LSRs using standard BGPv4 over IPv4, where the Next Hop is identified by an IPv4 address [62].

ii. IPv4 CE-to-CE Tunnels: This has no impact on existing IPv4 over MPLS core and only Customer Edge routers are IPv6 aware.

iii. Native IPv6 MPLS: Core infrastructure requires full control plane of IPv6. This network is in uniformity to the IPv4 as it requires that all routers in the MPLS network become dual-stack and use IPv6 routing protocols with IPv6-enabled LDP. This is simple to implement however, practically most expensive [62].

iv. Extend routing and signaling protocols to support IPv6: This has no impact on existing IPv4 over MPLS core and only MPLS Edge routers need to support circuit over MPLS.

4. HEADER COMPRESSION/SUPPRESSION

For good bandwidth utilization, it is necessary to reduce the unnecessary packet overhead for each packet. An IP packet is a combination of header and payload, and header compression is taking out the redundant header and then transmitting payload thereby helping in reduction of header information between consecutive packets. Suppression of parts of the header leads to a compressed header. The receiver has to restore the header at the receiving end. In many applications, the data is almost equal to that of the header.

Header compression relies on many fields being constant or changing rarely in consecutive packets belonging to the similar packet flow. If the packets have the same flow that is moving to same destination, some fields such as next header, version, flow label, source address, and destination address fields are same and thus are unnecessary overhead in a packet. That part of the information that does not modify is sent at the start or updated after certain interval of time or after some change has occurred. Even though header is an important part of the packet for communication, still at times these can be excessive or redundant overhead, taking up bandwidth un-necessarily. Header compression or suppression makes it possible to save this bandwidth in addition to reduction of packet loss and improved response time [63].

Some of the header compression gains are given below in table II:

TABLE II. HEADER COMPRESSION GAINS [63]

Protocol headers	Total header (bytes)	Min. compressed header (bytes)	Compression gain (%)
IPv4/TCP	40	4	90.00
IPv4/UDP	28	1	96.40
IPv4/UDP/RTP	40	1	97.50
IPv6/TCP	60	4	93.30
IPv6/UDP	48	3	93.75
IPv6/UDP/RTP	60	3	95.00

When moving from IPv4 to IPv6, the header size will increase from 40 bytes in IPv4 to 60 bytes in IPv6, and to 80 bytes in IPv6 with encryption encapsulation. This increase in header size requires significant bandwidth as the IPv6 header size is almost twice as large as an IPv4 header [64] [65]. The header size in an IPv6/UDP/RTP packet is between 60 to 120 bytes and the payload is between 15 and 20 bytes using compression voice algorithms and real time constraints [66] [67]. Therefore, there is a need for Header Compression in IPv6. The IPv6 header consists of a base portion header and extension headers. The base portion header (40 bytes) has seven fields of version field, priority field, flow label field, payload length field, next header field, hop limit field, and address field. Extension headers provide extra functionality [68] as shown in Fig. 5.

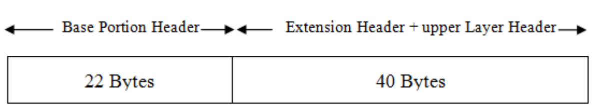

Figure 5. Header Format of IPv6 [68]

Fig. 6 depicts the general concept of header compression, in which the packet which consists of the payload plus data is compressed at the source before sending it and the compressed header is sent instead of the complete header. The compression at the source is done by the compressor and at the destination; the de-compressor helps in decompressing the header.

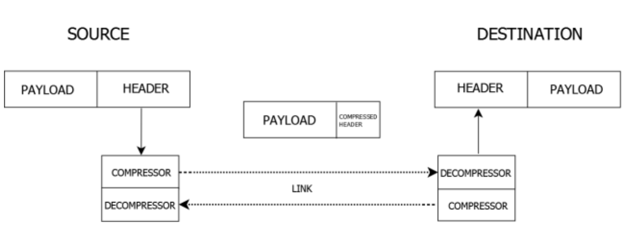

Figure 6. Header Compression in general [94]

4.1 HEADER COMPRESSION/SUPPRESSION

Packets used for header compression are uncompressed packets, compressed packets, and feedback packets. Uncompressed packet is a packet with complete header plus payload whereas compressed packet has compressed header plus payload. Information necessary to compress or decompress packets is stored in a context state database. Compressor and de-compressor operate according to a well defined protocol. The compressor compresses the headers with respect to a reference state that it shares in common with the de-compressor. Both have common reference state, therefore both need to operate according to a protocol. Compressor has the job of conveying context state updates to the de-compressor whenever there is an update in the network and the context state database changes. There is context synchronization until the de-compressor can successfully process these context states updates [65] [69] [70] [71]. The various header compression techniques are explained below.

4.1.1 VAN JACOBSON HEADER COMPRESSION (VJHC)

Van Jacobson proposed the original transport header compression scheme in RFC 1144 for the Transmission Control Protocol/Internet Protocol (TCP/IP) naming it Van Jacobson Header Compression (VJHC) [70]. In VJHC, the 40 byte TCP/IP packet header is reduced to less than 5 bytes for the average case. Van Jacobson (VJ) TCP header compression significantly reduces TCP protocol overhead in a noiseless environment with smaller packets exhibiting better VJ compression ratios and can get about 50% compression ratio [72]. TCP/IP VJHC is implemented with the Point-to-Point (PPP) link protocol achieving compression of TCP/IP header from 40 bytes down to 3-5 bytes. It was especially designed to improve TCP/IP performance over low- speed serial links [73]. It treats the physical link as consisting of two simplex links, one in each direction going from compressor to de-compressor, implying that there is no direct backwards flow of information from the de-compressor to the compressor [74]. It was proposed to improve TCP based interactive performance of applications over low-speed links with improvement in link utilization [72].

VJHC is performed on per hop basis at the link layer, maintaining connection state tables which contain states for each connection consisting of the last uncompressed TCP and IP headers sent or received on that connection. Unique Compression Identifier (CID) is allocated by the compressor for the connection and saving the first TCP/IP headers sent and all successive headers are built by sending only the changes from the previous headers. The de-compressor at the destination un-compresses the header by applying the changes contained in the newly received compressed header to the saved header. Fig. 7 depicts the VJHC mechanism.

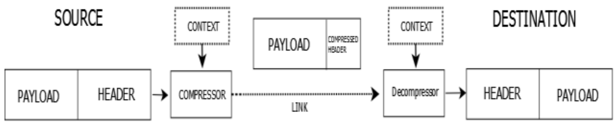

Figure 7. Van Jacobson Header Compression [94]

VJHC is very commonly used header compression method despite several other Header Compression mechanisms being implemented [73] [75]. Compressor from the source is located between Network Layer and Data Link Layer and relies on framer for in-order packet delivery and error detection, without any feedback between compressor and de-compressor [76]. In case of a lost or corrupted packet, invalid uncompressed header will be created. All packets delivered after the lost or corrupted packet will be decompressed improperly, thus will be discarded by the destination and requires TCP/IP re-transmission. Hardly any experimental results are there to support impact of TCP/IP's VJHC in lossy communication channels, particularly for low bit-rate wireless and satellite links [77]. The state is synchronized once again after the sender retransmits the original lost or corrupted packet. Resynchronization is done by uncompressed retransmissions [78].

4.1.2 SPACE COMMUNICATION PROTOCOL SPECIFICATION (SCPS)

Space communications require protocols which are reliable and efficient, as simple TCP performs poorly over space communications links.TCP was developed for earthly wired networks whereas for satellite communication, there are long delays and high Bit Error Rate (BER) producing unsatisfactory results. As the congestion control mechanism in TCP has unnecessary overhead of rate control, it leads to low bandwidth utilization. As an example, the achieved throughput is only about 200 Kbps, even though the satellite link capacity reaches 1.5 Mbps with a BER at 10^{-6} [79] [80]. Many TCP enhanced protocols such as Scalable TCP (STCP) [81], FAST AQM Scalable TCP (FAST TCP) [82], eXplicit Control Protocol (XCP) [83], Variable-structure congestion Control Protocol (VCP) [84] and Westwood protocol [85] etc. have been developed to improve its performance and the most successful among these is SCPS [79]. It contains four protocols named SCPS-FP, SCPS-TP, SCPS-NP, and SCPS-SP. As per ISO network model, SCPS-FP is an application layer protocol, SCPS-TP is a transport layer protocol, SCPS-NP is a network layer protocol, and SCPS-SP protocol is between transport layer and network layer [86].

Compression techniques are available in Space Communication Protocol Specification– Network Protocol (SCPS-NP) and Space Communication Protocol Specification–Transport Protocol (SCPS-TP). The SCPS-NP header construction approach is based on the header compression concepts elaborated in RFC 1144 [70] and uses a technique called 'capability driven header construction' as a means to control bit overhead which means that the packet has only those header fields that are essential for that packet only [87]. Header compression in SCPS deals independently with transport and network layer headers and compression of the SCPS-TP header is agreed in the initial uncompressed three-way handshake. SCPS-TP is an extension of TCP, thus 4 byte TCP option is built-in the header [76].

4.1.3 INTERNET PROTOCOL HEADER COMPRESSION (IPHC)

IPHC was introduced in RFC 2507 [88], compressing both IPv4 and IPv6 packets base and extension headers. It also compresses both TCP and UDP transport protocol header, aiming at improvement of RFC 1144 [70] with extra features and protocols using delta coding technique [89]. This compression technique intends to provide better response time, line efficiency, loss rate, and bit overhead and is intended for point-to-point links [76]. For the UDP (Fig. 8) and TCP (Fig. 9) headers, many header fields remain unchanged for an ongoing flow of packets. In both headers, sequential packets in the same stream will have the same source and destination port fields and length field can be inferred from lower layers of the protocol stack [90].

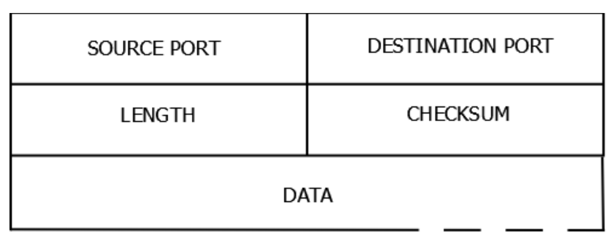

Figure 8. UDP Header Format [91]

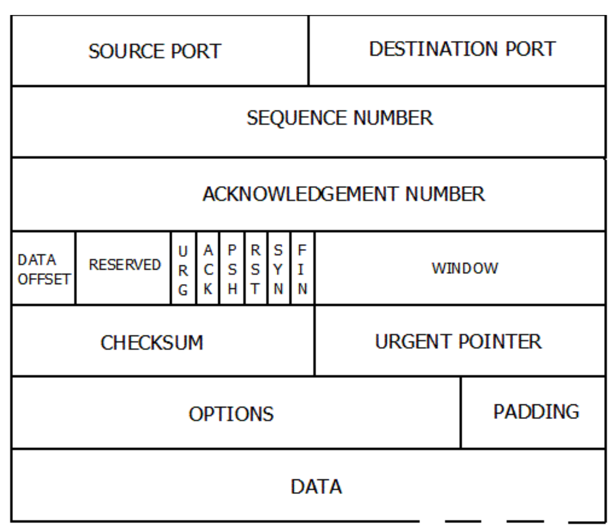

Figure 9. TCP Header Format [92]

In IPHC Operation, compressor sets up static header fields as constants and full headers carrying the context identifier (CID) are transmitted over the link. De-compressor, at the receiving end stores this context to use it for decompressing succeeding packets with the same CID, thus all the succeeding packets pass on the compressed header only. The compressor and de-compressor store more or less all header information and CID mapping in a context table. In case any of the static header changes, the complete header has to be sent as resynchronization has to be done to update the context.

In addition to compression of static header fields, IPHC can do compression of dynamic headers as well with a delta-based differential encoding scheme. IPHC protocol operation is shown in Fig. 10. IPHC utilizes 'twice' algorithm for management of error recovery which is better than VJHC for wireless networks. Some TCP fields change from packet to packet in expected way, which is represented as 'deltas' from previous value. These deltas are used to re-computer and repair context [90] [93].

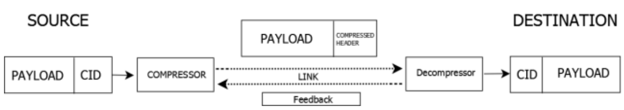

Figure 10. IP Header Compression Mechanism [94]

Compression efficiency is high in IPHC but is susceptible to packet errors that are common for all kinds of delta coding. Thus, when a packet with error arrives at the destination, the de-compressor cannot decompress it correctly and the consecutive packets are not reliable [95]. It has been observed that IPHC has long round-trip times on wireless links and studies have revealed that IPHC-based CRTP scheme do not work well over lossy links with long RTTs. Also, IPHC is not optimized for all upper-layer protocols [90] [96].

4.1.4 RTP HEADER COMPRESSION

The Real-time Transport Protocol (RTP) [97] delivers real-time data streams over UDP/IP. In streaming audio, major part of the packets is mostly the same, thus RTP packet considerably introduces header overhead. As an example, the RTP/UDP/IPv4 header takes 40 bytes, which requires up to 67% bandwidth utilization in transporting audio stream [98] [99]. Apart from RTP header of 12 octets, the packet will have an IP header of 20 octets, UDP header of 8 octets making a total of 40 octets and if IPv6 is used instead of IPv4, it will introduce another 40 octets of header. For audio transmission utilizing codec, the payload may be ranging from 15 to 20 octets only, thus a considerable size of header is present [100]. This is depicted in Fig. 11.

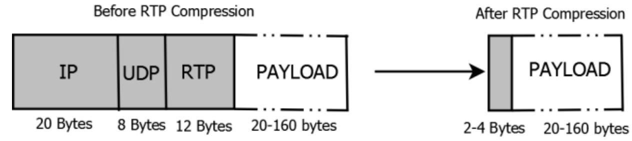

Figure 11. RTP Header Compression [94]

As a replacement of compressing the RTP header alone, better efficiency is obtained by compressing the combined RTP/UDP/IP headers together and this compression technique is not intended to work in conjunction with RTCP (Real-time Transport Control Protocol) as it adds to its complexity which is undesirable [76]. Compressing IP/UDP/RTP header was introduced in RFC 2508 to reduce the redundant header in real time traffic flows like audio and video. With CRTP, the minimal compressed headers are 2 octets if the UDP checksum is disabled and using this header compression technique, simulations have shown to have the lowest packet loss for audio traffic with codec [100] [101]. It is suitable for the voice and video packet compression, reducing bandwidth overhead in single hop links and works well in links with a small RTT. This is due to the reason that there will not be excellent synchronization in case the RTT is long and will reduce voice quality. Thus, due to the error-proneness and long RTT, CRTP is not appropriate for wireless links [72].

CRTP utilizes delta encoding to compress header. The compressor only transmits packets with the differences of headers from the previous header. The compressed header consists of a 4-bit sequence number to detect the packet loss. The de-compressor maintains context which represents the current header and how it is expected to change etc. and the differences carried in each compressed packet update the context. If there is a packet loss, the compressor and de-compressor will be out of synchronization [96] [98]. Context is invalidated due to a packet loss or corrupt packet and this invalidated context is fixed by explicit signalling messages. With this signalling, the de-compressor asks for a context update from the compressor and during this time interval, all the packets will be discarded. Thus, CRTP does not perform well on error prone links such as wireless links [102] [103].

4.1.5 EXTENDED COMPRESSED REAL TIME PROTOCOL (ECRTP)

Since RTP introduces a significant overhead, therefore CRTP [89] was introduced. It was designed for reliable point to point links, with short delays but it does not perform well over links with high rate of packet loss, packet reordering and long delays [104]. Since for wireless links, the high BER inflicts CRTP to reject a huge amount of successfully received packets with compressed headers, therefore the playback quality is drastically degraded [105] [106] [107]. Therefore, ECRTP [104] was introduced as RTP/UDP/IP header compression scheme to enhance CRTP by sending all full header packets and also those packets which contain update information N+1 times so that loss of fewer than N consecutive packets will not lead to context invalidation. ECRTP compressed UDP header is shown in Fig. 12.

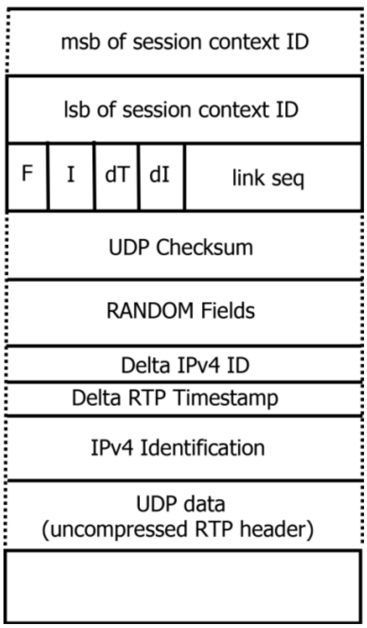

Figure 12. ECRTP Compressed UDP header [109]

Therefore, ECRTP [104] was introduced as RTP/UDP/IP header compression scheme to enhance CRTP by sending all full header packets and also those packets which contain update information N+1 times so that loss of fewer than N consecutive packets will not lead to context invalidation. ECRTP compressed UDP header is shown in Fig-12.

Compressed headers include the changes which occur in the dynamic header fields and for changes in static field from the de-compressor always trigger transmission of full header packets [108]. ECRTP also uses a delta based differential encoding mechanism. There are three major changes that have been introduced in CRTP for ECRTP; allowing updating the differential RTP values and to selectively update the absolute values for some header fields, Network efficiency is improved by inserting header checksum and extensions for updating context state which also minimize context invalidation and all updates in the context are replicated in several compressed headers to secure the establishment of updates [69] [109].

ECRTP performs retransmission by predicting whether the retransmitted packets can arrive in header decompressor earlier than their planned playback times and header compression determines the proper retransmitted packets with full or compressed headers [98]. The performance of ECRTP is effected by its shortcomings like transmission of full header during updates, since it cannot selectively update dynamic fields in the IP header, thus not effectively predicting changes in header fields [69] [104].

4.1.6 ROBUST HEADER COMPRESSION (ROHC)

Due to significant error rates, long round-trip times and bandwidth limited capacity in links, it became necessity to design highly robust and efficient header compression scheme based upon a highly extensible framework [76]. Thus, Robust Header Compression scheme [110] [111] was developed for compression of headers in IP packets i.e. UDP/IP, RTP/UDP/IP, ESP/IP, TCP/IP in order to resolve the problem caused by high error rate and long RTT appeared in wireless links. ROHC is considered stateful protocol, as it exchanges information over the air between compressor and decompressor to enable compression. ROHC builds state between the compressor and decompressor by sending full and incremental headers periodically. Fig. 13 shows the application ROHC defines state machines in the compressor and de-compressor in order to evaluate the consistency of the context information.

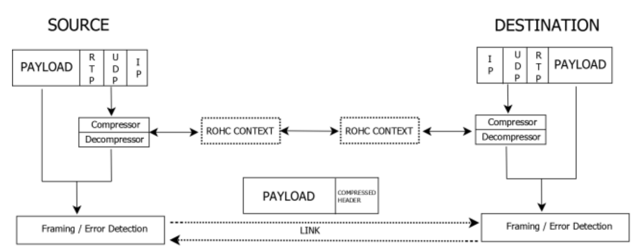

Figure 13. Application of ROHC in a protocol stack [112]

The compressor and de-compressor have three different operational states and start in the lowest compression state and attempt to work toward the higher state. The transitions between states do not need to be synchronized between the compressor and de-compressor. ROHC compressor has three compression levels: Initialization and Refresh (IR), First Order (FO) and Second Order (SO).

To start with, IR state in compressor establishes the static information and sends the full header (context) information. In FO compression state, it establishes the change pattern of dynamic fields and signifies partial context established between de-compressor and compressor. Finally, when the compressor is certain of the delivered context information, the compressor switches to the SO state, and sends compressed headers with minimal information as shown in Fig. 14. If there is an update or error has occurred, the compressor shifts back to upper compression levels and returns back to the SO compression state only after re-transmitting the updated information and establishing again the change pattern in the de-compressor. Context is used to store the information with regards to the header fields and size of the compressed header depends on the compression level and the header information required by the de-compressor [114].

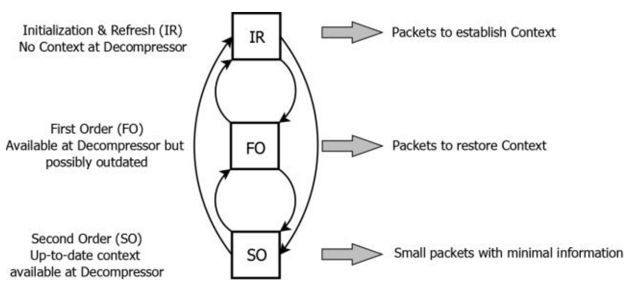

Figure 14. ROHC compressor states [113]

The de-compressor works at the destination and decompresses the headers based on the information of the context. The ROHC de-compressor operates in 3 finite states: No Context (NO), Static Context (SC) and Full Context (FC). To start with, the de-compressor starts in the No Context state, as it has no context information available in the beginning of the packet flow and shifts to Full Context and Static Context states as per the success of the decompression as it has received both static and dynamic information. This is shown in Fig. 15.

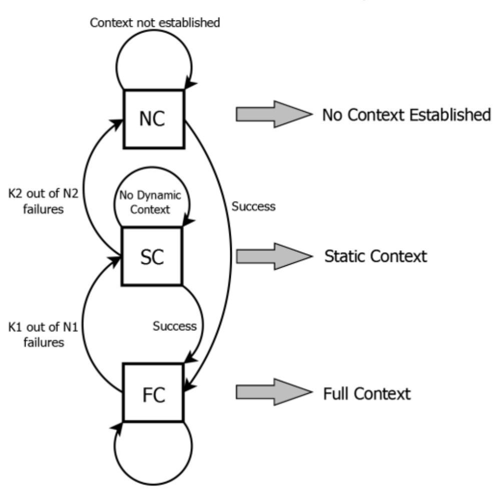

Figure 15. ROHC De-compressor states [113]

The successful decompression of an Initialization and Refresh from the compressor will generate the context information at the de-compressor side. However, in case of an update or occurrence of an error, the de-compressor shifts from FC to lower state. When the compressor sends an IR packet, the context can be restored at the de-compressor. It is necessary that synchronization occurs so that decompression happens correctly [113] [115] [116].

ROHC header compression has three operation or run modes: Unidirectional (U), bi-directional Optimistic (0) and bi-directional Reliable (R). In unidirectional mode, packets are sent from the compressor to the de-compressor in one direction only and this compression has to start with U-mode only. U-mode is inefficient as there is cyclical return to lower states and no feedback for errors recovery and thus has lower transmission rates as well. The difference between Unidirectional mode and Bi-directional mode is that in Bi-directional ROHC, operations start from U-mode and then may transit to O-mode or R-mode depending on the feedback information. Transition to the Bidirectional mode can be done once a packet has reached the de-compressor and the de-compressor has replied with an ACK feedback packet representing a mode transition is required [72] [117].

Bidirectional optimistic run mode (O-mode) represents compression over a bidirectional link and has a feedback channel from the de-compressor to the compressor. This feedback mechanism is used by de-compressor for signifying wrong decompression and acknowledgment of significant context updates [94]. This mode tries to improve the compression efficiency and increase the usage rate of sloppy feedback channels. Bidirectional reliable mode (R-mode) compared to O-mode, has dependable ROHC scheme which tries to make the most of it by having a better robust scheme in lost and damaged packet scenarios, thus helping in a minimization of context invalidation done by frequent usage of the feedback channel [72]. ROHC is comparably better than VJHC and IPHC due to high robustness and improved efficiency. However, quality enhancement of the small size data communication is constrained by processing overhead due to complicated communication and functional complexity from ROHC [118] [119]. Fig. 16 depicts RTP/UDP/IP compressed header packet in which the header size is compressed up to 1 byte.

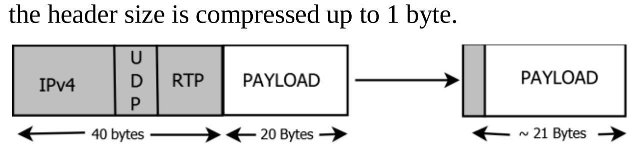

Figure 16. ROHC format for Header Compression [120]

The Robust Header Compression consists of the following profiles [115] [121]:

- *ROHC Uncompressed (Profile 0)*

Those packets are compressed, which cannot be compressed by any of the following profiles.

- *ROHC RTP (Profile 1)*

Packets with IP/UDP/RTP protocol headers are compressed.

- *ROHC UDP (Profile 2)*

Packets with IP/UDP protocol headers are compressed.

- *ROHC ESP (Profile 3)*

Packets with IP/ESP protocol headers are compressed.

The ROHC version 2 (ROHCv2) [121], extends the previously mentioned ROHC by increasing in overall simplicity and robustness, with equivalent or enhanced performance. ROHCv2 is a complete new design for performance enhancement [114]. It has a robustness method used by a compressor to boost the prospect of decompression success when packets can be lost and/or reordered on the ROHC channel [117]. Also, ROHC has improvements against the high packet loss and retransmissions in wireless links [122]. Fig. 17 represents ROHCv2 ladder diagram for flows in a context setup between two nodes in unidirectional mode.

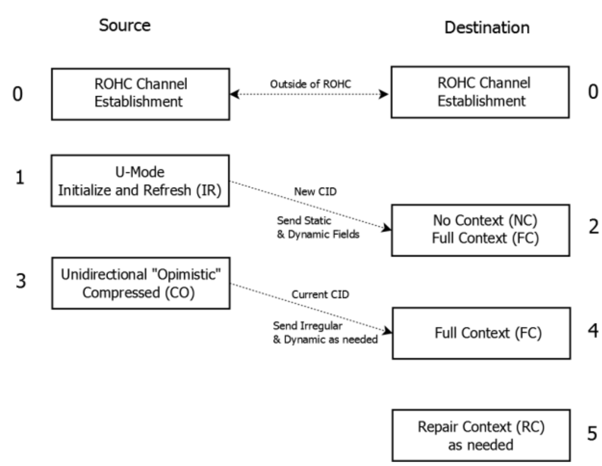

Figure 17. ROHCv2 Context Setup in Unidirectional mode [117]

The step 0 has channel establishment and step 1 has context setup. New CID is created and sent by the compressor in a U-mode IR packet with and the full IP headers as well at the beginning and when the decompressor receives it, the new CID is added to decompressor context. With no errors in decompression, the state for this CID moves from U-mode no context (NC) to U-modefull context (FC). Fig. 18 represents ROHCv2 with feedback channel for acknowledgements in Bi-directional mode in which the de-compressor verifies for context damage and then it may transit to RC or NC State depending on the severity of the context damage.

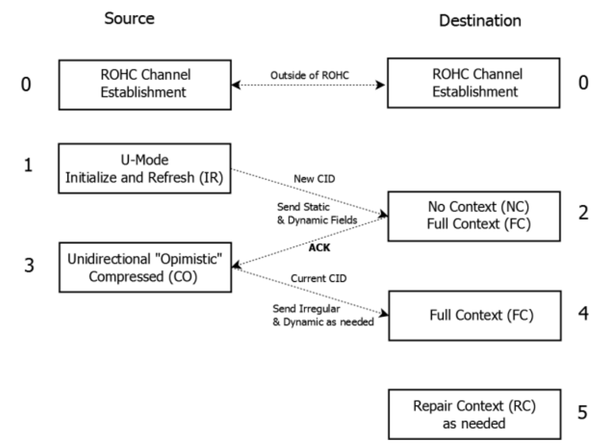

Figure 18. ROHCv2 Context setup in Bidirectional mode [117]

5. HEADER COMPRESSION OVER MPLS

Compressed packets can be routed through MPLS based network for reducing the packet over-head significantly by utilizing various Header Compression/Suppression schemes. MPLS together with Header Compression increases bandwidth efficiency and processing scalability of the maximum number of simultaneous compressed flows. Little research has been done for implementing Header Compression/Suppression over MPLS, with two RFC's proposing the implementation of Header Compression over MPLS. The first RFC 4247 [123] has given the requirements for Header Compression over MPLS mostly for voice/RTP/UDP/IP/MPLS-labels. Since IPv4 packet header is around 48 bytes, while the voice payload is often no more than 30 bytes, thus payload size is smaller than the relative size of the header [124]. With IPv6, the header size is increased, so is the overhead as well. The second RFC [125] has given the Protocol Extensions for Header Compression over MPLS defining how to use MPLS to route Header-Compressed (HC) packets over an MPLS label switched path. The concept of using MPLS pseudowire to transport the HC context and control messages between the ingress and egress MPLS label switching routers is given. This RFC provides guidelines as requirements for any future development of compression techniques in MPLS.

Using RFC4247 [123], Implementation of Payload Header Suppression (PHS) has been done in MPLS based network. For high speed and high throughput, transmission media at the backbone requires low and fast complexity approach for Header Compression (HC) and Header Decompression (HD) without unnecessary delay in feedback and signalling. The IPv6 first-order difference (static fields) is bigger in size and there are frequent state transitions for other header compression mechanisms such as ROHC. The congestion in the core of MPLS domain was reduced by 40% in case of IPv6 packet and throughput was thereby increased [124]. HC over MPLS can be implemented by applying HC algorithm at the ingress router of the Label Switched Path (LSP) and decompressed at the egress router where the HC ends [123]. HC over MPLS is depicted in Fig. 19.

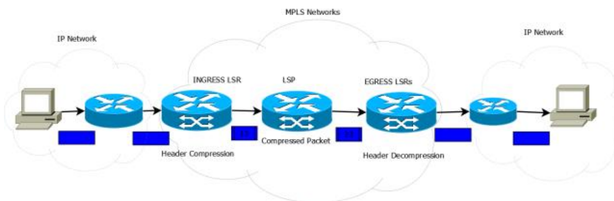

Figure 19. Header Compression over MPLS

Ingress router is the place where header compression is done and egress router is where header decompression is done. After compression, compressed packet is routed through LSP of MPLS Network up till egress router. MPLS LSP carries voice/compressed-header/MPLS-labels in place of voice/RTP/UDP/IP/MPLS-labels thereby saving 30 octets or more per packet [123].

MPLS supports high speed and high traffic networks with low bit error rate (BER) and compression with Payload Header Compression (PHS) was preferred over ROHC to avoid complexity in Header Compression and Decompression, redundant delays in feedback and signalling due to less BER in MPLS and also inefficient frequent state transitions for ROHC for MPLS [124]. Also, LSP-PHS implementation adds less complexity overhead and decrease in resource utilization thereby 37 out of 40 bytes IPv6 header was suppressed between source and destination in MPLS enabled network. The simulation result demonstrates substantial decrease in delay of UDP packets by approximately 50% with reduced packet drop of real-time traffic and better bandwidth utilization [38].

ROHC was developed for header compression over high Bit Error (BER) links and has mechanisms for quick context resynchronization. ROHC has compression of ESP/IP, UDP/IP, RTP/UDP/IP with better encoding scheme for the fields which keep on changing dynamically. The compressor compresses the RTP/UDP/IP packets into the appropriate compressed packets and sends them to the decompressor and works on the feedback information from the decompressor. This is done taking care of the states and modes and CIDs are to be managed practically and efficiently [72]. The advantage of having ROHC is that it has by default the ability to identify the packet type in the compression header and thus there is no need to further extend the identify packet type. There is a need for some changes in the existing MPLS network for handling compression and decompression such as [123]:

i. Expansion in MPLS signaling to discover the LSP from Header Compressor to Header Decompressor

ii. Negotiate the HC algorithm used and protocol parameters

iii. Negotiate the Session Context IDs (SCIDs) space between the ingress and egress routers on the MPLS LSP

iv. Signal HC over MPLS tunnels with the Label Distribution Protocol (LDP).

RFC 4901 [125] defines the MPLS Pseudo wires (PWs) which can be used to transport the HC context and control messages between the ingress and egress MPLS label switching routers with extension method for future header compression protocols. HC mechanisms have been normally intended to be used for single point-to-point links, whereas in MPLS, we can have header compression over LSP involving a number of hops. Pseudo Wire Emulation Edge-to-Edge (PWE3) is a method which imitates the characteristics of service such as ATM, Frame Relay or Ethernet over a Packet Switched Network (PSN) with functions that consist of encapsulating service-specific PDUs arriving at an ingress port, and carrying them across a path or tunnel, managing their timing and order. This can be utilized for MPLS domain as well as PW is supposed to be an unshared link from edge to edge [126].

Fig. 20 shows the reference pseudo wires model. The purpose of PWs also includes encapsulating packets arriving at an ingress router and transmitting them across an IP path or MPLS tunnel and also complete act on managing the timing and order of packets. Pseudo wires does encapsulation of service-specific PDUs or circuit data arriving at the provider edge carries the encapsulated data across a tunnel and does the management of the signalling, timing and order of the PW [127].

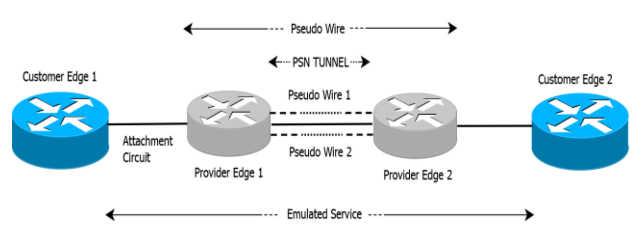

Figure 20. PWE3 Reference Model [14x]

RFC 5602 [130] defines the MIB module for PW operation over MPLS. It is designed to support both manually configured and signalled PWs, point-to-point PW connections, enable the use of any emulated service and supports MPLS-TE outer tunnel, non-TE MPLS outer tunnel (an outer tunnel signalled by LDP or set up manually), and no outer tunnel (where the PW label is the only label in the MPLS stack).

Within an MPLS network, the compression of packets is not hop-by-hop rather the compression is per LSP from ingress to egress LSRs. Pseudo Wires help in transmitting of messages between the compressor and the decompressor by providing a logical link between the two. HC channel is supported by PW like single point-to-point link; however these PWs carry only compressed packets. Two levels of label stacks are used in PWs, with PW label at the Bottom of Stack (BoS) for identification for PW. With header compression enabled, the compressor adds label at the BoS and likewise the removal of this label is done by the decompressor. This label is left untouched without any changes being done by the intermediate label switching routers of the LSP. MPLS labels are added after this label higher in the stack and are known as Packet Switch Network (PSN) labels. The packets are forwarded within the MPLS network with the higher labels in the label stack.

When the packet reaches the decompressor, the label at the BoS also known as PW label is used with CID for appropriate decompression. HC over MPLS additionally carries HC control parameter which contains a packet type field and a packet length field. The packet type field defines the packet type used in header compression such as a IR packet, feedback packet or compressed packet. With RoHC scheme, the length of headers is not mentioned as it is determined from the lower layer except for packets transmitted over Ethernet link [125].

RFC 4447 [128] details a LDP signalling to set up, tear down, and manage PWs for establishing and maintaining the pseudowire, using extensions to Label Distribution Protocol (LDP). Certain FEC elements, other parameters and codes have been defined for LDP to enable it to identify PWs. Pseudowire endpoint utilize TLVs in LDP to bind a demultiplexor field value to a pseudowire and transmits information to the other remote endpoint of this binding. The status updates and other changes including the releasing of binding is also specified. Pseudowire demultiplexor field is an MPLS label with transmission of MPLS packets through the pseudowire as MPLS tunnel with the exception of PHP behaviour. MPLS LSP can act as the PSN tunnel.

With extensions to LDP for PWs and now further extensions with respect to header compression specifically RoHC, certain elements like TLVs are reused to negotiate HC over the Point to Point Protocol with specifications for negotiating given in RFC3544 [129]. RoHC Bi-directional mode has feedback mechanism which is used to send error recovery request as well as acknowledgements of context updates if required. PWs are unidirectional, therefore there is a need to set up the PWs in each direction in order to have Bi-directional RoHC mode as given in fig. 21 [125].

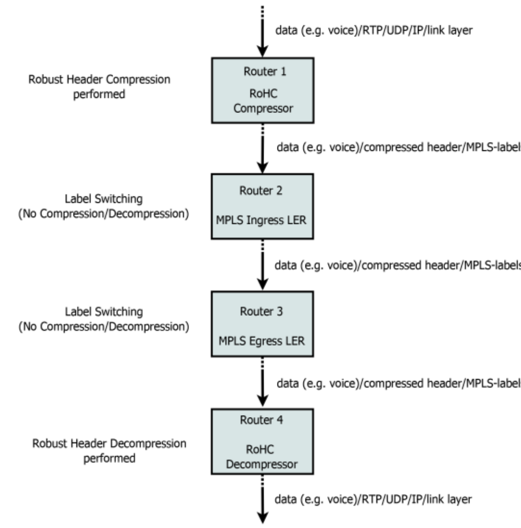

Figure 21. RoHC over MPLS Network [125]

6. CONCLUSION

IPv6 has bigger header which takes a major part of bandwidth thus affecting the bandwidth efficiency. Header compression can help in the reduction of redundant header in IP packets. With the help of MPLS, networks can handle IPv6 traffic efficiently over the IPv4 backbone network. Several header compression schemes have been developed and deployed in order to use the bandwidth efficiently. This paper provides a survey of available header compression schemes that can be implemented on MPLS over IPv6 and enable conservation of resources over the network. The findings from this survey contribute to a better understanding for researchers on how to enhance the bandwidth with header compression by which the excess IP header is removed before transmitting the packet on a link and at the destination decompressing them to their original state. A number of header compression schemes were discussed namely, VJHC, SCPS, IPHC, CRTP, ECRTP and ROHC with their advantages and disadvantages. Very less research has been done in implementation of these header compression schemes in MPLS networks. Future work would include simulation and implementation of various header compressions in MPLS over wired and wireless networks with IPv6 as well as multicasting.

REFERENCES

[1] Bernet Y., “The Complementary Roles of RSVP and Differentiated Services in the Full-Service QoS Network,” IEEE Communication Magazine, Feb. 2000, pp. 154-162.

[2] Sekib F. A., McClellan S., Manpreet S.and Chakravarthy S., “End-to-End Testing of IP QoS Mechanisms”, IEEE Magazine 2002.

[3] Moshin M., Wong W. and Bhatt Y., “Support for real-time traffic in the Internet, and QoS issues,” A survey of QoS protocols for the Internet, March 30, 2002.

[4] Rahimi, M.; Hashim, H.; Rahman, R.A, "Implementation of Quality of Service (QoS) in Multi Protocol Label Switching (MPLS) networks," *Signal Processing & Its Applications, 2009. CSPA 2009. 5th International Colloquium on* , vol., no., pp.98,103, 6-8 March 2009

[5] Braden R., Zhang L., Herzog S.and Jamin S., ”Resource ReSerVation Protocol (RSVP)- Version 1 Functional Specification,” RFC2205, 1997

[6] Fgee, E.-B.; Kenney, J.D.; Phillips, William J.; Robertson, W.; Sivakumar, S., "Comparison of QoS performance between IPv6 QoS management model and IntServ and DiffServ QoS models," Communication Networks and Services Research Conference, 2005. Proceedings of the 3rd Annual , vol., no., pp.287,292, 16-18 May 2005

[7] Bernet Y. et al,” A Frame work for Differentiated Services,” Internet Draft –ietf-diffserv- framework-02.txt, Feb 1999.

[8] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” RFC 2460 (Draft Standard), Internet Engineering Task Force, Dec. 1998, updated by RFCs 5095, 5722, 5871.

[9] Wai Yee Tai, Chong Eng Tan, Sei Ping, Lau, "Towards Utilizing Flow Label IPv6 in Implicit Source Routing for Dynamic Source Routing (DSR) in Wireless Ad Hoc Network", in the proceedings of 2012 IEEE Symposium on Computers and Informatics, Penang, Malaysia, 18-20 March, 2012

[10] A. Liakopoulos, D. Kalogeras, V. Maglaris, D. Primpas, and C. Bouras, “QoS experiences in native IPv6 networks,” International Journal of Network Management, vol. 19, no. 2, pp. 119–137, 2009.

[11] Luc De Ghein. 2006. MPLS Fundamentals. Cisco Press.

[12] E.Rosen, A. Viswanathan, R.Callon. “Multiprotocol Label Switching Architecture,” RFC 3031. January 2001.

[13] E.Rosen, G.Fedorkow, Y.Rekhter, D.Farinacci, T.Li, A.Conta. “MPLS Label Stack Encoding,” RFC 3032. January 2001.

[14] R. Callon, P. Doolan, N. Feldman, A. Fredette, G. Swallow, and A. Viswanathan, “A framework for multiprotocol label switching,” September 1999.

[15] T. Mallik and F. Francois, “Implementation of IPv6 Services over a GMPLS-Based IP/Optical Network,” Cisco Systems, IEEE Communications Magazine, May 2005

[16] D. Awduche, J. Malcolm, “Requirements for Traffic Engineering over MPLS,” RFC 2702. 1999

[17] Solano, F., Fabregat, R., & Marzo, J. L. “On optimal computation of MPLS label binding for multipoint-to-point connections”, IEEE Transactions on Communications, 56(7), 1056–1059. 2008

[18] Sahel A., Sa’ed A., M. Kharbutli, J. Bassam, “MPLS technology in wireless networks,” Wireless Networks, November 2013

[19] Rosen E, Tappan D, et al. “MPLS Label Stack Encoding”. RFC 3032, IETF, January 2001.

[20] Chunying Tian; Shanzhi Chen; Yuhong Li; Yaning Liu, "Experimental Study of MPLS and MIPv6 Integration Technology," Communications and Networking in China, 2006. ChinaCom '06. First International Conference on , vol., no., pp.1,3, 25-27 Oct. 2006

[21] http://www.protocols.com/papers/mpls.htm [online]

[22] Padilla, J.; Huerta, M.; Paradells, J.; Hesselbach, X., "IntServ6: an approach to support QoS over IPv6 networks," Computers and Communications, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium on , vol., no., pp.77,82, 27-30 June 2005

[23] O. Aboul-Magd and B. Jamoussi, "QoS and service interworking using constraint route label distribution protocol (CR-LDP)", Communications Magazine, IEEE, vol. 39, pp. 134-139, 2001.

[24] P. Pan, G. Sawallow, and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", 2004.

[25] L. Martini, E. Rosen, N. El-Aawar, “Encapsulation Methods for Transport of Layer 2 Frames over MPLS Networks”, RFC 4905, Cisco Systems Inc, June 2007

[26] L. Martini, E. Rosen, N. El-Aawar, “Transport of Layer 2 Frames Over MPLS”, RFC 4906, Cisco Systems Inc, June 2007

[27] Imaizumi, H.; Hiroyuki Morikawa; Nakamura, O.; Murai, J.; Aoyama, T., "An alternative approach to multi-path forwarding on IPv6 networks," Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on, vol.2, no., pp.5 pp.,, 18-20 April 2006.

[28] Adibi, S.; Erfani, S., "IP-based IMT network platform formobile-IPv6-MPLS systems," Electrical and Computer Engineering, Canadian Conference on , vol., no., pp.1416,1418, 1-4 May 2005

[29] Jialei Wu; Yuanping Zhang, "A Layered MPLS Network Architecture," *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on* , vol., no., pp.1,4, 23-25 Sept. 2010

[30] T. Onali, “Quality of service technologies for multimedia applications in next generation networks,” Ph.D. dissertation, University of Cagliari, February 2008.

[31] I. Minei and J. Lucek, MPLS-enabled applications: emerging developments and new technologies. John Wiley & Sons Inc, 2005.

[32] http://www.metaswitch.com/wiki/what-mpls-and-gmpls [online]

[33] Nai-Bin Hsu, "The Routing of Healthcare Applications in MPLS Networks," icgec, pp.771-774, 2010 Fourth International Conference on Genetic and Evolutionary Computing, 2010

[34] Artan Halimi, Brikena Statovci-Halimi, “Overview on MPLS Virtual Private Networks”, Photonic Network Communications, May 2002, Volume 4, Issue 2, pp 115-131

[35] Vassiliou, V.; Owen, Henry L.; Barlow, D.; Sokol, J.; Huth, H.-P.; Grimminger, J., "M-MPLS: Micromobility-enabled multiprotocol label switching," Communications, 2003. ICC '03. IEEE International Conference on , vol.1, no., pp.250,255 vol.1, 11-15 May 2003.

[36] http://www-03.ibm.com/ibm/history/exhibits/mainframe/ mainframe _intro2.html [online]

[37] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification”, RFC 2460, December 1998.

[38] Imad J Mohamad, “End-to-End QoS Improvement using IPv6 Header Reduction over MPLS”, International Journal of Computer Applications 80(11):27-32, October 2013.

[39] S. Deering, R. Hinden, “Internet Protocol version 6 Specification”, RFC 2460, December 1998

[40] Ramakrishnan, K.; Floyd, S.; Black, D. (September 2001) The Addition of Explicit Congestion Notification (ECN) to IP, IETF. RFC 3168.

[41] Aazam, M.; Syed, A.M.; Eui-Nam Huh, "Redefining flow label in IPv6 and MPLS headers for end to end QoS in virtual networking for Thin client," Communications (APCC), 2013 19th Asia-Pacific Conference on , vol., no., pp.585,590, 29-31 Aug. 2013

[42] Degermark, et. al., “IP Header Compression”, RFC 2507, February 1999

[43] K. Nichols et at., “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers,” Cisco Systems, Torrent Networking Technologies, EMC Corporation, RFC 2474, December 1998.

[44] K. Nichols, et. al. , “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, RFC 2474, December 1998

[45] S. Blake, et. al., “An Architecture for Differentiated Services”, RFC 2475, December 1998.

[46] F. Le Faucheur, et. al., “Multi-Protocol Label Switching (MPLS) Support of Differentiated Services, RFC 3270, May 2002.

[47] P. Ciprian, "Deploying IPv6 Networks", Cisco Systems, 2006.

[48] Astudillo, C.A.; Calderón, O.J.; Ortiz, J.H., "PM²PLS: Integrating Proxy Mobile IPv6 and MPLS in Wireless Access Networks," New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on , vol., no., pp.1,5, 7-10 Feb. 2011

[49] Satoshi Uda, Nobuo Ogashiwa, Yojiro Uo, Yoichi Shinoda, "IPv6 support on MPLS networks: Experiences with 6PE approach," saint-w, pp.226, 2003 Symposium on Applications and the Internet Workshops (SAINT 2003 Workshops), 2003

[50] Verrydt, Wim; Popoviciu, Ciprian, "Study of IPv6 Multicast Deployment in MPLS Netw," Computing in the Global Information Technology, 2006. ICCGI '06. International Multi-Conference on , vol., no., pp.4,4, Aug. 2006

[51] Aazam, M.; Syed, A.M.; Eui-Nam Huh, "Redefining flow label in IPv6 and MPLS headers for end to end QoS in virtual networking for Thin client," Communications (APCC), 2013 19th Asia-Pacific Conference on , vol., no., pp.585,590, 29-31 Aug. 2013

[52] Verrydt, Wim; Popoviciu, Ciprian, "Study of IPv6 Multicast Deployment in MPLS Netw," Computing in the Global Information Technology, 2006. ICCGI '06. International Multi-Conference on , vol., no., pp.4,4, Aug. 2006

[53] J. Rajahalme, et. al., "IPv6 Flow Label Specification," RFC 3697, March 2004.

[54] Fineberg, V., "The role of IPV6 and MPLS in the GIG black core," Military Communications Conference, 2005. MILCOM 2005. IEEE , vol., no., pp.1938,1943 Vol. 3, 17-20 Oct. 2005

[55] Roesler, V.; Balbinot, L.; de Andrade, M.; Tarouco, L., "IP next generation label switching," IP Operations and Management, 2002 IEEE Workshop on , vol., no., pp.21,25, 2002

[56] F. Le Faucheur et al., "MPLS Support of Differentiated Services," Cisco Systems, PMC-Sierra Inc., Nokia, Axiowave Networks, Alcatel, Telia Finland, RFC 3270, April 2001.

[57] F. Baker and Y. Rekhter, "Use of Flow Label for Tag Switching," Cisco Systems, Internet-Draft (expired), draft-baker-flow-label-00.txt, October 1996.

[58] Pattrick, G, "IPv6 over MPLS", Cisco Systems, 2001. http://www.ipv6-es.com/02/docs/ Patrick_grossetete_1.pdf [online]

[59] Uda, S.; Ogashiwa, N.; Uo, Y.; Shinoda, Y., "IPv6 support on MPLS networks: experiences with 6PE approach," Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on , vol., no., pp.226,231, 27-31 Jan. 2003

[60] Yunos, Rozita; Noor, N.M.; Ahmad, Siti Arpah, "Performance evaluation between IPv4 and IPv6 on MPLS Linux platform," Information Retrieval & Knowledge Management, (CAMP), 2010 International Conference on, vol., no., pp.204,208, 17-18 March 2010

[61] J. D. Clercq, G. Gastaud, T. Nguyen, D. Ooms, S. Prevost, and F. L. Faucheur., "Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)," RFC 4798, February 2007.

[62] Nawaz, A.; Ashraf, M.; Hong, X.B.; Wu, J.; Lin, J.T., "IPv4 to IPv6 Evolution Strategies of Pakistan Internet Exchange," Networking and Services, 2009. ICNS '09. Fifth International Conference on , vol., no., pp.383,388, 20-25 April 2009.

[63] An introduction to IP header compression, EFFNET AB white paper, February 2004. http://www.effnet.com/sites/effnet/pdf/uk/ Whitepaper_Header_Compression.pdf [online]

[64] Wu, T., Chao, H., Lo, C., "Providing Efficient Secured Mobile IPv6 by SAG and Robust Header Compression", Journal of Information Processing Systems, Vol.5, No.3, September 2009

[65] Westphal, C.; Koodli, R., "Stateless IP header compression," Communications, 2005. ICC 2005. 2005 IEEE International Conference on , vol.5, no., pp.3236,3241 Vol. 5, 16-20 May 2005

[66] Bormann, C., et. Al. "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP and uncompressed", RFC 3095, 2001.

[67] Minaburo, A.; Toutain, L.; Nuaymi, L., "Deployment of IPv6 robust header compression profiles 1 and 2," Computer Science, 2003. ENC 2003. Proceedings of the Fourth Mexican International Conference on , vol., no., pp.278,283, 8-12 Sept. 2003

[68] Lim, J.D.; Stern, H.P., "IPv6 header compression algorithm supporting mobility in wireless networks," Southeastcon 2000. Proceedings of the IEEE , vol., no., pp.535,540, 2000

[69] Brower, E.; Ertekin, E.; Christou, C.A.; O'Keeffe, S., "The application of header compression to IPsec encrypted networks," Military Communications Conference, 2005. MILCOM 2005. IEEE , vol., no., pp.2844,2850 Vol. 5, 17-20 Oct. 2005

[70] V. Jacobson, R. Braden, D. Borman. Compressing TCP/IP headers for low speed serial links IETF, Network working group, RFC 1144, Feb. 1990

[71] Mikael Degermark, Mathias Engan, Bjorn Nordgren, Stephen Pink ¨ LowLoss TCP/IP Header Compression for Wireless Networks Proceedings of the 2nd Annual International Conference on Mobile Computing and Networking (MOBICOM'96), 1996, pp.1-14.

[72] Xiuliang Chen; Fuliang Guo; Ping Dang; Libing Wu, "A survey of ROHC header compression schemes," Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on , vol., no., pp.331,335, 29-31 Dec. 2012

[73] M. Allman, S. Dawkins, D. Glover, J. Griner, T. Henderson, J. Heidemann, H. Kruse, S. Ostermann, K. Scott, et al, "Ongoing TCP/IP Research Related to Activities," Internet Draft, Internet Engineering Task Force, August 1999.

[74] Perkins, S.J.; Mutka, Matt W., "Dependency removal for transport protocol header compression over noisy channels," Communications, 1997. ICC '97 Montreal, Towards the Knowledge Millennium. 1997 IEEE International Conference on , vol.2, no., pp.1025,1029 vol.2, 8-12 Jun 1997

[75] Ruhai Wang, "An experimental study of TCP/IP's Van Jacobson header compression behavior in lossy space environment," Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th , vol.6, no., pp.4046,4050 Vol. 6, 26-29 Sept. 2004

[76] J.A. Ishac, Survey of Header Compression Techniques, NASA/TM-2001-211154, September 2001. http://kom.aau.dk/ group/ 05gr1120/ref/Header_comp.pdf [online]

[77] Ruhai Wang; Horan, S., "Impact of Van Jacobson header compression on TCP/IP throughput performance over lossy space channels," Aerospace and Electronic Systems, IEEE Transactions on , vol.41, no.2, pp.681,692, April 2005

[78] Ruhai Wang; Sreelakshmi Bonasu; Bagasrawala, S., "The impact of TCP-IP header compression on performance over satellite links," Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE, vol.4, no., pp.2021,2026 Vol.4, 21-25 March 2004

[79] Xiaohu Ge; Li, F.Y.; Reichert, F., "An enhanced TCP congestion avoidance scheme and its performance evaluation in high speed satellite networks," Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on , vol., no., pp.1,5, 8-10 Dec. 2009

[80] R. C. Durst, G. J. Miller, and E. J. Travis, "TCP Extensions for Space Communications," Wireless Networks, Vol.3, N. 5, pp.389-403, Oct, 1997

[81] T. Kelly, "Scalable TCP: Improving Performance in High Speed Wide Area Networks," Computer Communication Review, Vol. 33, No. 2, pp. 83-91, Feb. 2003

[82] D. X. Wei, C. Jin and S. H. Low, "FAST TCP: Motivation, Architecture, Algorithms, Performance," IEEE/ACM Tran. on Networking, Vol. 14, No. 6, pp.1246-1259, Dec. 2006.

[83] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High Bandwidth-Delay Product Networks," in Proc. ACM SIGCOMM, June 2002, pp.19-23.

[84] Y. Xia, L. Subramanian, I. Stoica, and S. Kalyanaraman, "One More Bit is Enough," IEEE/ACM Trans. on Networking, Vol. 16, No. 6, pp.1281 – 1294, Dec. 2008

[85] M. Gerla, M. Y. Sanadidi, R. Wang, and A. Zanella, "TCP Westwood: Congestion Window Control Using Bandwidth Estimation," in Proc. IEEE GLOBECOM, Nov. 2001, pp.1698-1702.

[86] Hong Jiang; Kai Hu, "Research on SCPS-NP Routing Protocol," *Information Engineering and Computer Science (ICIECS), 2010 2nd International Conference on* , vol., no., pp.1,4, 25-26 Dec. 2010

[87] Space Communications Protocol Specification (SCPS)—Network Protocol (SCPS-NP). Recommendation for Space Data System Standards, CCSDS 713.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 1999.

[88] M.Degermark, B.Nordgren, S.Pink, "IP Header Compression", RFC 2507, Febrary 1999

[89] S.Casner, V.Jacobson, "Compressing RTP/UDP/IP Headers for Low-Speed Serial Links", RFC 2508, February 1999.

[90] Ertekin, E.; Christou, C.A., "Internet protocol header compression, robust header compression, and their applicability in the global information grid," *Communications Magazine, IEEE* , vol.42, no.11, pp.106,116, Nov. 2004

[91] J. Postel, "User Datagram Protocol," RFC 768, August 1980

[92] J. Postel, "Transmission Control Protocol," RFC 793, September 1981

[93] Bow-Nan Cheng; Wheeler, J.; Hung, B., "Internet protocol header compression technology and its applicability on the tactical edge," *Communications Magazine, IEEE* , vol.51, no.10, pp.58,65, October 2013

[94] Manish Raj Shivare ,Yogendra P. S.Maravi, Sanjeev Sharma, "Analysis of Header Compression Techniques for Networks," International Journal of Computer Applications (0975 – 8887) Volume 80 – No5, October 2013

[95] Yongkoo Yoon et. al, Header Compression Method and Its Performance for IP over Tactical Data Link, International Journal of Energy, Information and Communications, pg 61-77, Vol. 2, Issue 2, May 2011

[96] M. Degermark et al., "Evaluation of CRTP Performance over Cellular Radio Networks," IEEE Pers. Commun., vol. 7, no. 4, Aug. 2000, pp. 20–25

[97] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," In IETF RFC 1889, Jan. 1996.

[98] Wen-Tsuen Chen; Din-Wuan Chuang; Hung-Chang Hsiao, "Enhancing CRTP by retransmission for wireless networks," Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on , vol., no., pp.426,431, 2001

[99] T. Hoshi, K. Tanigawa, and K. Tsukada, "Proposal of a Method of Voice Stream Multiplexing for IP Telephony Systems," In Proc. of Internet Workshop, pp. 182-188, 1999.

[100] Degermark, M.; Hannu, H.; Jonsson, L.; Svanbro, K., "Evaluation of CRTP performance over cellular radio links," *Personal Communications, IEEE* , vol.7, no.4, pp.20,25, Aug 2000

[101] S. Casner and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-speed Serial Link," RFC 2508, Feb. 1999

[102] Rahmat, F.; Idris, F.; Azri, M.; Kanathasan, V.; Sarimin, N.; Mohamad, M.H., "Performance analysis of VoIP in Multi-hop Wireless Network," Applied Electromagnetics (APACE), 2012 IEEE Asia-Pacific Conference on , vol., no., pp.51,55, 11-13 Dec. 2012

[103] Larzon, L.-A.; Hannu, H.; Jonsson, L.-E.; Svanbro, K., "Efficient transport of voice over IP over cellular links," Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE , vol.3, no., pp.1669,1676 vol.3, 2000

[104] T. Koren and et al, "Enhanced Compressed RTP (ECRTP) for links with high delay, packet loss and reordering", RFC 3545, July 2003.

[105] M. Degermark and H. Hannu, "Evaluation of CRTP Performance over Cellular Radio Links," In IEEE Personal Communications, pp. 20-25, Aug. 2000.

[106] Wen-Tsuen Chen; Din-Wuan Chuang; Hung-Chang Hsiao, "Enhancing CRTP by retransmission for wireless networks," Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on , vol., no., pp.426,431, 2001

[107] G. Mamais, M. Markaki, M. H. Sherif, and G. Stassinopoulos, "Evaluation of the Casner-Jacobson Algorithm for Compressing the RTP/UDP/IP Headers," In Proc. of IEEE Symposium on Computers and Communications, pp. 543-548, 1998.

[108] Haipeng Jin; Hsu, R.; Jun Wang, "Performance comparison of header compression schemes for RTP/UDP/IP packets," *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE* , vol.3, no., pp.1691,1696 Vol.3, 21-25 March 2004

[109] K. Carl, "Evaluation and Implementation of Header Compression Algorithm ECRTP," Masters Thesis, MSc programmes in Engineering, Department of Computer Science & Electrical Engineering, Lulea University of Technology.

[110] L-E. Jonsson, G. Pelletier,K. Sandlund "The RObust Header Compression (ROHC) Framework", RFC 4995, July 2007

[111] C. Burmeister, M. Degermark, H. Fukushima, Matsushita, H. Hannu, Ericsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3059, July, 2001.

[112] Jesus Arango, Syed Al, Daniel Hampel, "Header Compression for Ad hoc Networks," in Military Communication Conference, MILCOM 2005, 2005.

[113] Maeder, A; Felber, A, "Performance Evaluation of ROHC Reliable and Optimistic Mode for Voice over LTE," Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th , vol., no., pp.1,5, 2-5 June 2013.

[114] Tomoskozi, M.; Seeling, P.; Fitzek, F.H.P., "Performance evaluation and comparison of RObust Header Compression (ROHC) ROHCv1 and ROHCv2 for multimedia delivery," *Globecom Workshops (GC Wkshps), 2013 IEEE* , vol., no., pp.544,549, 9-13 Dec. 2013.

[115] The concept of robust header compression, ROHC, EFFNET AB White Paper, February 2004. http://www.effnet.com/sites/effnet/pdf/uk/Whitepaper_Robust_Header_Compression.pdf [online]

[116] Martinez, E.; Minaburo, A; Toutain, L., "ROHC for multicast distribution services," Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on , vol.3, no., pp.1540,1544 Vol. 3, 11-14 Sept. 2005

[117] Bow-Nan Cheng; Moore, S., "Securing Robust Header Compression (ROHC)," Military Communications Conference, MILCOM 2013 - 2013 IEEE , vol., no., pp.1383,1390, 18-20 Nov. 2013

[118] Sangkil Jung; Sangjin Hong; Peom Park, "Effect of RObust Header Compression (ROHC) and Packet Aggregation on Multi-hop Wireless Mesh Networks," Computer and Information Technology, 2006. CIT '06. The Sixth IEEE International Conference on , vol., no., pp.91,91, Sept. 2006

[119] Iqbal F, “Performance Evaluation of Robust Header Compression Protocol for Low Data Rate Networks,” thesis for the degree of Master of Science at the University of Agder, Faculty of Engineering and Science.

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

[121] G. Pelletier, K. Sandlund, "RObust Header Compression Version 2 (ROHCv2): Profiles for RTP, UDP, IP, ESP and UDP-Lite,” Internet Engineering Task Force (IETF), Network Working Group, RFC 5225, 2008

[122] Szczesny, D.; Traboulsi, S.; Bruns, F.; Hessel, S.; Bilgic, A, "Exploration of energy efficient acceleration concepts for the ROHCv2 in LTE handsets," *Industrial Embedded Systems (SIES), 2011 6th IEEE International Symposium on* , vol., no., pp.232,237, 15-17 June 2011

[123] J. Ash, e.a., “Requirements for Header Compression over MPLS”, RFC 4247, November 2005.

[124] Mohamad, IJ.; Tat-Chee Wan; Alzyoud, F.Y.; Sumari, P., "Optimizing the MPLS support for real time IPv6-Flows using MPLS-PHS approach," *TENCON 2009 - 2009 IEEE Region 10 Conference* , vol., no., pp.1,6, 23-26 Jan. 2009.

[125] J. Ash, et al., “Protocol Extensions for Header Compression over MPLS”, RFC4901, June 2007.

[126] X. Xiao, et al., “Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)”, RFC 3916, September 2004

[127] S. Bryant, P. Pate, “Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture”, RFC 3985, March 2005

[128] L. Martini, et. Al., “Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)”, RFC 4447, April 2006

[129] T. Koren, et. al., “IP Header Compression over PPP”, RFC 3544, July 2003

[130] D. Zelig, T. Nadeau, “Pseudowire (PW) over MPLS PSN Management Information Base (MIB)”, RFC 5602, July 2009.

Mohammad Ahsan Chishti
Mohammad Ahsan Chishti has done his Bachelor of Engineering (B.E.) and M.S. in Computer and Information Engineering from International Islamic University Malaysia with specialization in Computer Networking. Presently he is Assistant Professor and Head of the Department of Computer Science & Engineering, National Institute of Technology Srinagar, Kashmir. He has around 50 publications to his credit and 5 patents. He is a Young Scientist awardee by JK Govt. and also Principal Investigator of Department of Science & Technology, Government of Jammu and Kashmir sponsored research project on IPv6. He has been guiding M.Tech. and B. Tech. research projects and his research area includes Computer Networks, Next Generation Networks, MPLS, IPv6 and Technology Roadmapping.

Prof. Ajaz Hussain Mir
Prof. Ajaz Hussain Mir has done his Bachelor of Engineering (B.E.) in Electrical Engineering with specialization in Electronics and Communication Engineering from REC Srinagar (J&K) India in 1982. He did his M.Tech in Computer Technology and PhD both from IIT Delhi in the year 1989 and 1996 respectively. He was Chief Investigator of Ministry of Communication and Information Technology, Govt. of India project: Information Security Education and Awareness (ISEA). He has been guiding PhD and M.Tech. thesis related to the area of Security and other related areas. He has a number of International publications to his credit. Presently he is working as Professor in the Department of Electronics and Communication Engineering and Dean, Research & Consultancy at NIT Srinagar, Kashmir, India. His areas of interest are Biometrics, Image Processing, Security, Wireless Communication and Networks.