

Received April 17, 2020, accepted April 23, 2020, date of publication April 28, 2020, date of current version May 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991064

# Unidirectional and Bidirectional Optimistic Modes IP Header Compression for Real-Time Video Streaming

DOUMA BOUTHIBA FAROUQ<sup>1</sup>, ALA ABDUSALAM ALAROOD<sup>2</sup>, NAHLA ALJOJO<sup>3</sup>, AND ADAMU ABUBAKAR<sup>4</sup>

<sup>1</sup>Security Engineering Department, Diyar United Company, Doha 92693, Qatar

<sup>2</sup>College of Computer Science and Engineering, University of Jeddah, Jeddah 21959, Saudi Arabia

<sup>3</sup>Information System and Information Technology Department, College of Computer science and Engineering, University of Jeddah, Jeddah 21959, Saudi Arabia

<sup>4</sup>Department of Computer Science, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia

Corresponding author: Adamu Abubakar (adamu@iiu.edu.my)

**ABSTRACT** Communication over Internet Protocol (IP) networks, has become crucial component of day everyday activities. They are utilized over the Internet to support a wide range of services. The flexibility of this kind of transmission relies on the IP User Datagram Protocol (UDP), IP/UDP/Real-time Transport Protocol (RTP) and IP/Transmission Control Protocol (TCP). Unfortunately, the weight of encapsulated protocol headers affects the transmission efficiency. This research aims at improving a technique that reduce the packets header size by compression. Performance analysis of the enhanced efficient techniques in both unidirectional and bidirectional optimistic modes applied to real-time video streaming traffic for UDP/IP and HTTP/TCP flows over free error channel has been conducted. The finding shows that the header compression ratio in each case is good and better than the previous studies. The technique achieved a reduction up to 90% for RTP/UDP/IP, 89% for UDP /IP and 77.5 to 86.5 % for TCP/IP profile. This research contribution is restricted to compression gain and saving for  $0 \times 0000$ ,  $0 \times 0001$ ,  $0 \times 0002$  and  $0 \times 0006$  profiles in the unidirectional and bidirectional optimistic modes.

**INDEX TERMS** Datagram protocol, real-time transport protocol, transmission control protocol, protocol headers.

## I. INTRODUCTION

Bandwidth is the most expensive resource in wireless and cellular links comparing with the processing power [1]. The current evolution of multimedia services tends to respond to the need for mobility among users. This has resulted an exchange of information between different types of networks (Internet and mobile Networks) [2]. The information to be exchanged may have different characteristics and each network treats them differently. The Internet and mobile networks are becoming more and more blended to provide consistent, end-to-end service [3]. IP become a common way to carry data and all future cellular links are heading toward the full IP networks [4]. The adoption of IP stacks among

mobile devices have become more general to support all types of flows, not only audio and video, but also web browsing, emails, online gaming and so on.

Nowadays Internet based communication is quite prominent in the industry and also in the market. With the wide distribution of smartphones, the users are connected to the service providers constantly, regardless whether they are staying at a fixed point or moving [5]. The global IP traffic is expected to be increasing. In the meantime, this is forcing the hardware/software producers to find a way for cheaper solution [2]. Wireless communication is characterized by limited bandwidth, and higher error rate. For each transmission over a session, there are a lot of header data associated with any information over a transmission session. This requires a special measure for correction. While, there are a lot of data associated with any information over a transmission session.

The associate editor coordinating the review of this manuscript and approving it for publication was Noor Zaman<sup>id</sup>.

Every transmission medium specifies its Maximum transmission unit (MTU) for any packet carried over it. In order to guarantee a good transmission reliability, small packet size is transferred faster than a larger one [6]. A packet with a size larger than MTU might causes a large memory allocation on Network Interface Card (NIC) and Operation system buffers. That is why IP protocol fragment larger packet size into smaller sizes in order to transmit them faster, but the encapsulated header size remain fixed for any smaller fragment of IP packet [7].

Header compression usually compresses the protocol stack above the link layer. This is possible, because most of the fields in an IP traffic contain values that are constant or rarely change during a session. Previous research studies have adopted this technique and save nearly 80% of the IP/UDP/RTP headers just by utilizing Robust Header Compression [8]. Let's take as an example a typical RTP packet with IPv4 Internet layer protocol. The overall average size of such a packet is 87 bytes, including the IP/UDP/RTP headers' 40 bytes. With a typical stream, it is possible to achieve a constant compression ratio of almost 60% (given a free error channel) [8], [9].

The simplicity of implementation or calculation of a header compression scheme is thus of lesser importance than its compression ratio and its robustness [9]. ROHC header compression defined in the IETF RFC3095 allows a very large increase in network capacity in the case of interactive multimedia streams [10]. RoHC is adopted by different mobile technologies. Universal Mobile Telecommunication System (UMTS) used it in order to ensure better performance in low bandwidth networks [11].

The problem with IP over mobile and wireless networks is the high header redundancy in video streaming and voice data (VoIP) [12] regarding that 70 % from intent traffic belong to multimedia services [13], these applications will most likely be carried by RTP protocol defined in [RFC1889] in addition to the link layer headers [14]. A packet will have an IP header (20 bytes) a UDP header (8 bytes) and an RTP header (12 bytes) and this will be a total of 40 bytes [15]. Local broadcast mechanisms, media streaming and all online games data are encapsulated into UDP/IP stack with 28 bytes' headers. The size of the payload depends on the video and audio encoding (codecs) and the frame sizes used and can be as low as 15 to 20 bytes. That is why this study examined ROHC status with its application in the unidirectional and bidirectional optimistic modes IP Header compression for real-time video streaming.

The remaining parts of this paper are organized as follows: Apart from the present section that provides the overview of this study, section 2 present related works. Section 3 presents the research methodology. Section 4 present algorithm development. Section 5 present experimental analysis, and section 6 is the presentation of the results. Section 7 presents the discussion of the major findings, while section 8 is the conclusion of the paper.

## II. RELATED WORK

The overall objective of IP header compression schemes is to reduce bandwidth consumption, by reducing header redundancy in order to enhance transmission efficiency. Unfortunately, many improvements are still needed to implement and evaluate on existing techniques. One of the major promising tool for achieving this, is, robust header compression (ROHC). The technique serves as an important component of modern packet-switched [16]. It has been adopted by the Electro Electric Systems Research Institute of Huawei for Ship Area Sensor Network (SASN). Nam *et al.* [17].

Following the implementation of header compression in many areas, this current study reviewed empirical studies towards improving header compression. Jian *et al* [18] investigate and developed an optimization scheme of RoHC parameters in LTE MBMS networks in order to maximize the bandwidth saving of multimedia broadcast and multicast service. The work was focused on the revenue and cost of radio resources. In a different approach, an evaluation of the three implementations of header compression was used by IP-based cellular network traffic [10]. It was observed that the compressed header savings typically range from 50 % to about 60 % for both RoHC versions and it is at 85 % for the IPHC with occasional uncompressed refreshes. The result was that both RoHC version perform at about the same level, however, IPHC outperforms them easily by at least 25 %. In order to simulate the TCP compression scenario, in place of large TCP payload. Experiment shows that IPHC is capable to achieve 10 % to 20 % per cent better than RoHC which is more reliable and can ensure robustness by having a stable and persistent compressed header size.

Wu *et al.* [16] proposed a markov compressor technique for RoHC U-Mode which it closely imitate a general timer-based compressor and can be effectively optimize under various channel settings. The unidirectional mode of ROHC header compression scheme in U-mode is a stable hardware accelerator based on ROHC principles. But the compression scheme shows that an RTP packet header with 40 bytes' length could be compressed into only 2 bytes in the best compression state of the accelerator [19].

In an attempt to improve the ROHC compression algorithm, Kaur and Gupta [20] propose to reduce the size of second order packets, then that of standard ROHC mechanism by omitting sending of context ID in SO packet in modified ROHC algorithm, and supplementing this absence of information by inbuilt mechanism of algorithm to generate the identification of SO packet at decompressor end. The motivation behind modifying the RoHC algorithm is further improve upon average header length. A small reduction in the size of SO packet improve the average header length.

Many other new generation technologies that are expected to be designed in this area of compression will be for mobile phones and IoT. The adoption of ROHC technique in modern android mobile devices, is becoming widespread [10]. ROHC technology will not increase power consumption and it has

a potential of decreasing mobile battery drain by 0.05 Watt, when packet payloads are small [10].

It is worthwhile to demonstrated the benefit of compression in IP network communication by the previous studies. But majority of these studies where not considering real time audio and video streams compression test with more advanced profiles (e.g. RTP/UDP/IP, TCP/IP, UDP/IP). This is expected to achieve a high bandwidth gain. Some studies successfully developed RoHC compression algorithm and tests its efficiencies [9], [10] Quite a number of RoHC implementations were also revealed in [16] but do not use a real time RoHC measurements. For example, only U-mode was Implemented in those studies. As a result, this study intended to further and examined unidirectional and bidirectional optimistic modes IP header compression for real-time video streaming.

### III. METHODOLOGY

The research methodology adopted for this study is in the context of the procedural steps that describes an approach to on how to solved IP packet header problems. This research build on three different steps namely” 1) Highlighting theoretical framework system architecture, 2) Algorithm development 3) Experimental Evaluation.

#### A. CONCEPTUAL BACKGROUND OF THE STUDY

The conceptual background of the study dwells on ROHC compression and decompression of an IP header. An IP interface between two nodes in the network can have N compressors and N decompressors. An IP interface can indeed use several channels of the lower layer to transport the packets.

The ROHC Compressor has three operational states: Initialization and Refresh state, First Order state, And Second Order state (SO). It is preferable and efficient for compressor to work in the higher states [18].

In the “Initialization and Refresh state” the technique is set to initialize the non-changing (statics) parts of the decompressor context. It is also responsible for recovering after a compression or decompression failure [21]. The compressor maintains the IR state and keep sending packets with full headers with all the dynamic and non-changing fields in the uncompressed format and plus some additional information until it is assured that the decompressor has received all the static information related to the stream.

In the “First order (FO) state” the compressor efficiently communicates the changing and irregularities of the packet stream. When the compressor operates in this level it rarely sends all the header static fields [18]. Only few static fields can be updated, and the information transferred to the decompressor is usually compressed partially. In the “Second order (SO)” SO state, the compression efficiency is optimal. The compressor enters to this state, when the sequence number (RTP SN) of next packet header to be compressed is completely predictable and the compressor is completely sure that the decompressor has acquired all sequence number function parameters of the other fields [18].

The decompressor has three states NO\_CONTEXT is the first state, The STATIC\_CONTEXT state and The FULL\_CONTEXT state.

In the NO\_CONTEXT, while working in No Context state the decompressor uses has not received an initiation packet or when the context is lost. In this state, all packets are lost except initiation packets (IR). The STATIC\_CONTEXT state is reached after detecting some errors in the compression, and the decompressor has only the static part of the header. It expects a periodic retransmission in the one-way case where it sends a NACK to have the dynamic part. The FULL\_CONTEXT state is reached after the first correct decompression of a packet (reception an IR packet with static and dynamic information). The decompressor remains in the FULL\_CONTEXT state until the dynamic part of the header is lost [21].

There are three modes of operation of ROHC in order to increase the performance of the compression. 1) Unidirectional mode (U), 2) The Bidirectional Optimistic mode (O), and 3) The Bidirectional Reliable mode (R). The ROHC mechanism can switch from one mode to another to adapt its compression. Each compression mode has its own behavior. The optimistic one-way (U) and bidirectional (O) modes are more complex than the bidirectional reliable (R) mode because the compressor and decompressor do not know all the parameters and the information for compression [22].

The unidirectional mode compression always starts in the unidirectional mode (U). In this mode, the receiver does not send acknowledgments. To avoid excessive drift in the contexts between the compressor and the decompressor, complete headers sent periodically. The period depends on the parameters such as the estimation of the quality of the connection that the compressor can have. The unidirectional operation mode has different algorithms of regeneration control [18]. It uses two timers (IR\_TIMEOUT and FO\_TIMEOUT) to perform regeneration of the complete header. It also uses a trusted system (L) called the confidence variable. This approach used to increase the compression level. In this operation mode, the compressor controls the size of the used header. If any change detected in the header, it may return to a lower compression level to get all the information needed for the decompressor. Since the decompressor cannot communicate with the compressor, the acknowledgments have not sent and everything based on the control mechanisms. This is the least efficient mode of operation but the protocol guarantees the correct arrival of most headers. The main application of this mode is on military or satellite networks where it is impossible or even dangerous to transmit data [21].

The Bidirectional Optimistic mode is the optimistic bidirectional operation mode that is similar to the unidirectional operation mode, but in this mode, the decompressor can send negative acknowledgments to indicate errors. The optimistic mode does not use both timers but keeps the confidence system (L) [21]. Transitions in compression levels achieved by two negative acknowledgments (NACK and STATIC-NACK) and (L) approach. The negative acknowledgments (NACK)

make a backward transition of a compression level (FO or IR). Static negative acknowledgments (STATIC-NACK) make a negative transition to the lower compression level (IR). If the compressor receives a packet updating the context, the compressor will change its compression level by sending a larger header. The approach (L) is used by the decompressor, to move to a higher level of compression [22].

The Bidirectional Reliable mode is the reliable bidirectional operation mode that works only with the acknowledgments received from the decompressor, whenever it receives a positive or negative acknowledgment; the compressor changes its compression level. It returns to the initial state when it receives a static acknowledgment. In this operation mode, only packets with a CRC of 7 or 8 bits can update the context and used as a reference for subsequent decompression [21]. Therefore, the integrity of the context will have better protection.

The profiles define the protocol headers that must be compressed. They allow the decompressor to know the IP version, if the stream uses RTP or ESP (Encapsulating Security Payload) or if it is a UDP stream. Currently there are five defined profiles; this number may evolve in the future: Profile 0: without compression [18]. If this profile is chosen, only the ROHC identifier is added so that the decompressor can recognize the packets but there is no compression; Profile 1: compression of the IPv4/v6/UDP/RTP headers. This profile is the most generic; it compresses the entire header from IP to the RTP header. This is also the most useful in the case of Voice over IP and the simplest for compression because the time stamp and the RTP sequence number will facilitate the updating of the context; Profile 2: compression of IPv4/v6/UDP headers. This profile is a variation of profile 1 except that compression here stops at UDP. The formats of the header of the ROHC packets are different from those of the profile 1; Profile 3: compression of IPv4/v6/ESP headers. This profile compresses the ESP protocol; it can also be taken into account as a variation of the profile 2; Profile 4: compression of IPv4/v6 headers. This profile only compresses the IP protocol, and its packets are created on the formats of packets in profile 2. Note that the case of the TCP protocol will be treated differently. Currently, it is expected that future versions of mobile networks will have to use CTCP. Work in the ROHC group plans to use EPIC, compression by the Huffman algorithm, to reduce the size of the header [22].

RoHC uses cyclic redundancy checks as the primary means for detecting erroneous decompression.

#### IV. ALGORITHM DEVELOPMENT

Deployments of ROHC protocol are already made in mobile phones. A performance evaluation application was developed using a free Opensource Robust Header Compression (ROHC) library. In order to develop an algorithm for IP packet header compression, two modules should be involved. The compression module and the decompression module.

The compression module is created by passing the type of CID to use and a call-back to a function that generates random numbers. The compression profiles that defines the packets types will be activated. Then enable a call back function to trace the compression process. Then the packet will be pass for compression. Algorithm 1 is formulated in such a way that after detecting a suitable profile and the existing CID, the compressor will check for an existing feedback to piggybacked. Then, the compressed header will be concatenated to the packet payload to form the ROHC compressed packet. When the compression process is done, the context is destroyed.

---

#### Algorithm 1 RoHC\_Compressor

---

**Input:** IP Packet header for communication from  $A_i \rightarrow B_j$

**Output:** IP packet header compressed

initialize Packets  $X = \{x_1, \dots, x_n\}$

```

1: While packets are received do
2:   if suitable profile is found for the packet then
3:     if context is not synchronized then
4:       create a new context
5:     end if
6:     code CID using the context
7:     if ROHC feedback is found Then
8:       Piggyback ROHC feedback
9:     end if
10:    compress packet header using the context and profile
11:    concatenate compressed header ad payload
12:  end if
13:  send packet to the decompressor
14: End while

```

---

After the compression and the compilation of the compressed packets, the next is the decompression. The ROHC decompression API of the ROHC library allows the evaluation program to decompress the incoming ROHC packet from the decompressor into their original uncompressed format. The ROHC decompressor algorithm is more complex than compressor algorithm.

Any padding octets will be stripped if found on add CID field, then the decompressor need to decode the CID, if an Add-CID octet is found when ROHC channel operate with LARGE-CID, this packet is erroneous and it will be discarded. If an ROHC feedback is detected in the compressed packet it can decoded only if the compressor and decompressor are synchronized. Otherwise the packet will be discarded due to undetermined ROHC feedback length. Then the decompressor will decode the packet CID based on the encountered LARGE-CID, else the CID will be assumed zero if no Add-CID octet was detected and SMALL-CID will be used. After this, the decompressor will decompress the ROHC packet according to the decompression routines presented. Finally, it destroys the context when it finishes the process.

In order to test the performance of ROHC protocol, virtual machine was used. The performance evaluation

**Algorithm 2** RoHC\_Decompressor**Input:** IP Packet Compressed Header  $A_i \rightarrow B_j$ **Output:** IP packet header

```

initialize compressed headers
1: while packets are received do
2:   if padding octets are found then
3:     strips padding octets
4:   end if
5:   if add CID is found then
6:     if decompressor uses LARGE-CID then
7:       discard the packet
8:     end if
9:   Else
10:    decode CID
11:  end if
12:  if decompressor uses LARGE-CID then
13:    decode CID from LARGE-CID octet
14:  Else
15:    CID is Zero
16:  end if
17:  if DecompressorContext = No Context then
18:    if IR packet is received then
19:      Verifies Packet CRC + set DecompressorCon-
text = full context
20:    else
21:      discard packet
22:    end if
23:  else
24:    if compressorState = Static context then
25:      if IR packet is received then
26:        Verifies Packet CRC + set DecompressorCon-
text = full context
27:      else
28:        if normal packet is received then
29:          remove CID + Reconstruct IP packet
30:        end if
31:      else
32:        if compressorState = Full_Context then
33:          if IR packet is received then
34:            Verifies Packet CRC
35:          else
36:            if normal packet is received then
37:              remove CID + Reconstruct IP packet
38:            end if
39:          end if
40:          send packet to the upper layer
41:        end while

```

model consists of IP packet generator, RoHC compressor/decompressor and comparator. This study uses “VLC player” to generate streaming media packets according to the RTP protocol. The objective of this experiment is to show the robustness of the ROHC protocol and the gain of size on real packets resulting from streams. The idea is to see how it would actually compress the packets and see compression

ratio of each type of packet which is supported right now by the library which implemented the protocol. Since the ROHC protocol compresses different types of packets, this research capture those packets and simulate a compression of those packets and the decompression but not in unidirectional mode in order to see the actual results on the O Mode when the decompressor sends feedbacks to the compressor.

For the different packets that ROHC already implements the main ones implemented are the IP/TCP, IP/UDP and the IP/UDP/RTP protocols.

**V. EXPERIMENTAL ANALYSIS**

To perform the RoHC performance tests under Linux platform. We have developed a C++ program that define the two ROHC instances (compressor & decompressor) and take the captured packet flows as an input and checks ROHC Compression and Decompression statistics. The developed user space application run all ROHC library test into the Linux kernel by reading the capturing packets and send them to the kernel module. Finally, after sending the captured flow through compressor/decompressor modules, the program calculate the average bit compressed per second for the given flow.

The obtained results were achieved on an Intel Core I7-4558U machine running Ubuntu 16.04 LTS. We use our program by running the measurement scenarios and executing shell commands. the operation outputs a CSV format file and log files (As shown in Table 4.1) generated by the compression /decompression measures which is interpreted by excel and converted to graphs. Then the experiments are carried out, followed by an analysis of the results in order to know if the RoHC meets the expectations in the studied protocol.

The Compression Gain was calculated by using the following formula: Packet Compression gain (%) = (Original Packet Size - Compressed Packet Size) / (Original Packet Size) x 100. The Header Gain of each packet is calculated by the following formula: Header Grain = 1 - (Payload / (Header + Payload)) = Header / (Header + Payload). The performance test is done by using different scenarios.

The experimental testbed (see Figure 1 and 2) consists of two PCs, the first PC run under Linux (GNU/Linux Ubuntu 16.04 LTS) and the second PC runs Windows operation system. Both PCs are connected to the same local network through wireless 802.11 router. The compressor and decompressor were implemented in the Linux machine. For the first experiment the VLC streaming machine is configured to stream a video under RTP/MPEG Transport stream (RTP/UDP/IP profile), and this stream is played using the local network on the Linux machine. To compress this RTP streaming flow all streaming packets have been captured from the wireless Ethernet network device’s using the main.ccp program which can filter packet from the wireless network interface based on IP addresses and Port numbers of the stream source and destination. The program stores all the sniffed RTP traffic information (number of packets, packet

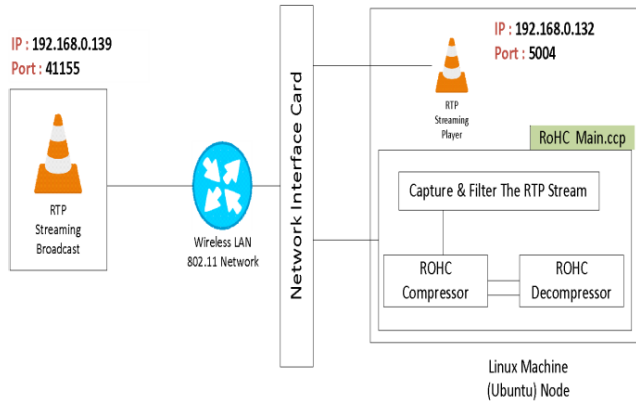


FIGURE 1. RTP and UDP streaming scenario.

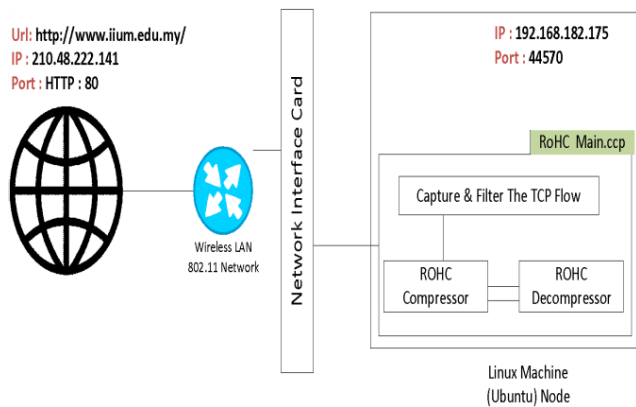


FIGURE 2. TCP flow compression scenario.

size, packet processing time) in the statistic file and export as a CSV format. This was enough to show the performance on different types of packets.

The same simulation is repeated several times by using different ROHC operation modes, unidirectional mode and Bidirectional optimistic mode.

TCP Flow compression experiment was undertaken in this scenario, where a TCP flow was initiated (source address & source port) and its ending (destination address and destination port), to all the follow TCP Stream.

VI. PRESENTATION OF RESULTS

The results of the initiation and refresh compression efficiency analysis in each state is presented in Figure 3 and 4. For all the difference compression state, RTP/UDP/IP profile in U-Mode and O-Mode are described. The result indicated that of the initialization state shows that the IR header compression for RTP /UDP/IP packets are in the optimistic mode. In this mode just four packets with full headers were sent. The compression gain is 0 (keep the 40 bytes' size during the IR state), as shown in Figure 3.

The next round of running the experimental scenario is with the same streaming media but under unidirectional mode of compression and decompression. The result of the experiment has yielded some important results (see Figure 4).

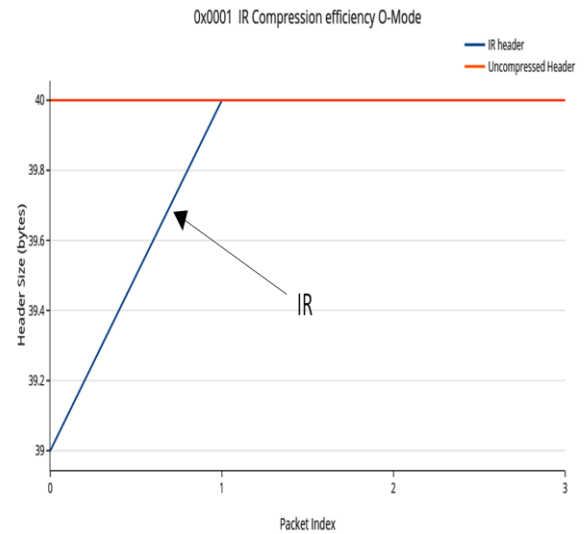


FIGURE 3. RTP/UDP/IP IR state compression efficiency in O-mode.

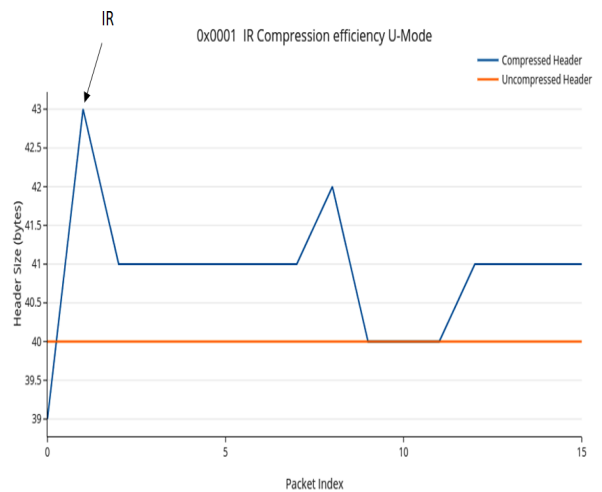


FIGURE 4. RTP/UDP/IP IR state compression efficiency U-mode.

It indicated that in the Initialization and fresh state, the compressor sends 16 IR packets with full headers size to the decompressor. Some packets are larger than the original header size due to association of CID (1 bytes), profile identifier (1 bytes) and CRC (1 bytes). This make up to 43 bytes IR header during the initialization stage. It has been observed that in the unidirectional mode 16 IR packet have been sent while in the optimistic mode the compressor sends just 4 packets, this is caused due to lack of feedback in U-Mode so the compressor cannot know if all packets have been received successfully by the decompressor .to avoid the context damage, the compressor sends IR packets periodically to update the context if needed.

After the initialization, in the first compression level, the compressor has send 2345 packets in the First order state. due to the constancy of compression ratio we have chosen the first 200 packets to plot the graphs in Figure 5 and Figure 6. The average compressed packet size in O-mode is 10.03 bytes

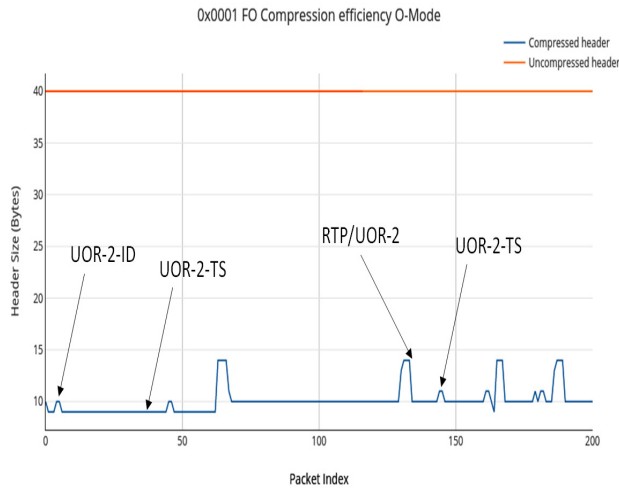


FIGURE 5. RTP/UDP/IP FO state compression efficiency in O-mode.

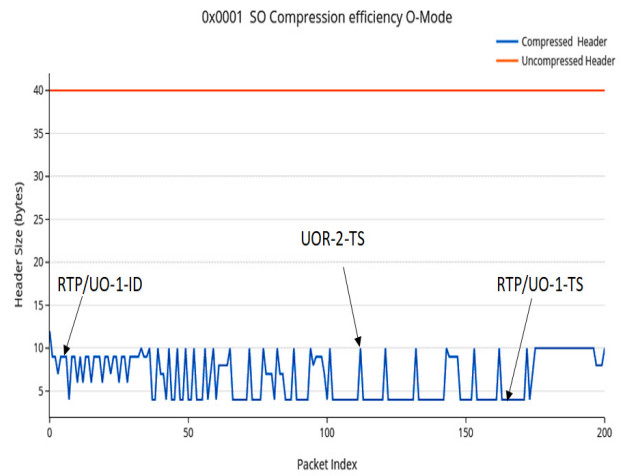


FIGURE 7. RTP/UDP/IP SO state compression efficiency.

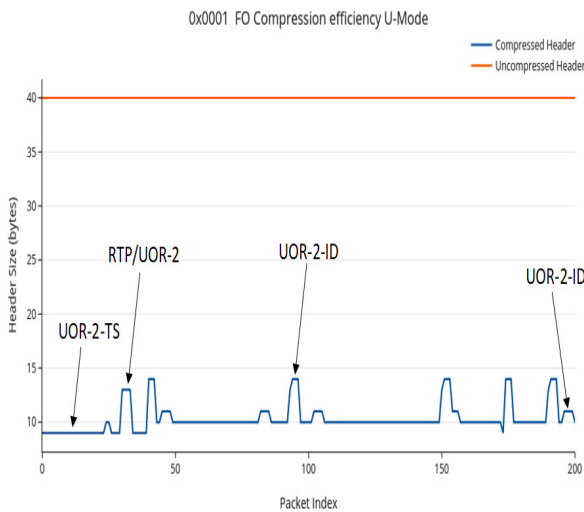


FIGURE 6. RTP/UDP/IP FO state compression efficiency U-mode.

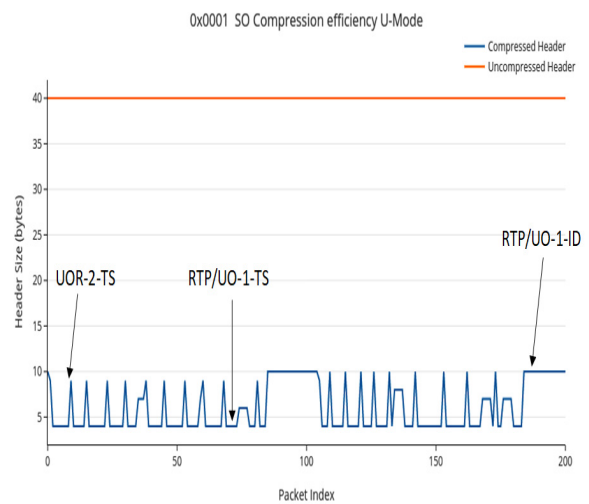


FIGURE 8. RTP/UDP/IP SO state compression efficiency U-mode.

(73 % Size gain). We have obtained a header gain as low as 9 bytes, 77.5 % (for UOR-2-TS packets).

The transmission uses 2464 packets in the unidirectional first order state with an average compressed packet size of 10.32 bytes with 9 bytes minimum 14 bytes' maximum.

The second compression level results (Figure 7 and 8) represent the header compression in O and U mode. 2341 packets have been transmitted in the optimistic mode. The higher compression level achieved a better compression rates. The average compressed packet size is 6.50 bytes (83.75 % header gain) the packet size will be as low as 4 bytes for UO-1-TS packets.

The next round indicate uses 2428 packets for transmission in the unidirectional first order state with an average compressed packet size of 5.98 bytes. Overall the compressed packet in SO state U-Mode reach up to 4 bytes' size.

The full compressions result obtained in every profile has been gathered. The RTP/UDP/IP video streaming compression results are presented. The packet generation component

generates packets continuously, and sends the packets to the compressor of our virtual machine and the statistics module. After compressing the packets, the compressor sends the compressed packets to the decompressor and the statistics module. Table 1 shows the result of the compression efficiency of RTP/UDP/IP profile in each state of U-mode and O-mode.

The bidirectional optimistic mode result shows the function of the statistic module, which is counting the number of packets (NOP) received by the compressor, the uncompressed headers sizes (UHS), the uncompressed packets size (UPS), the compressed packet size (CPS) in each compression type, and the compressed Header size (CHS) in each compression type. The results in Table 1 and 2 was provided by counting the component. The compressor compresses the original IPv4/UDP/RTP packet header effectively.

Table 1 present the total compressed header savings for RTP/UDP/IP profile (0 × 0001) in optimistic mode are about 77.26 % for the full RTP flow with IPv4 headers in O-mode.

TABLE 1. RTP/UDP/IP compression efficiency under O mode.

Packet Type	Number of packets	UPS (bytes)	UHS (bytes)	CPS (bytes)	CHS (bytes)
IR	4	5424	160	5423	159
RTP/U O-1-ID	93	126108	3720	123143	577
RTP/U O-1-TS	393	532908	15720	518760	1572
RTP/U OR-2	39	52884	1560	51792	468
UOR-2-ID	278	376968	11120	369249	3401
UOR-2-TS	1538	2085528	61520	2039153	15145
Total	2345	3179820	93800	3107520	21322

TABLE 2. RTP/UDP/IP compression efficiency under U mode.

Packet Type	Number of packets	UPS (bytes)	UHS (bytes)	CPS (bytes)	CHS (bytes)
IR	16	21696	640	21710	654
RTP/UO-1-ID	121	164076	4840	160210	974
RTP/UO-1-TS	510	691560	20400	673200	2040
RTP/UOR-2	40	54240	1600	53120	480
UOR-2-ID	309	419004	12360	410356	3712
UOR-2-TS	1484	2012304	59360	1967502	14558
Total	2480	3362880	99200	3286098	22418

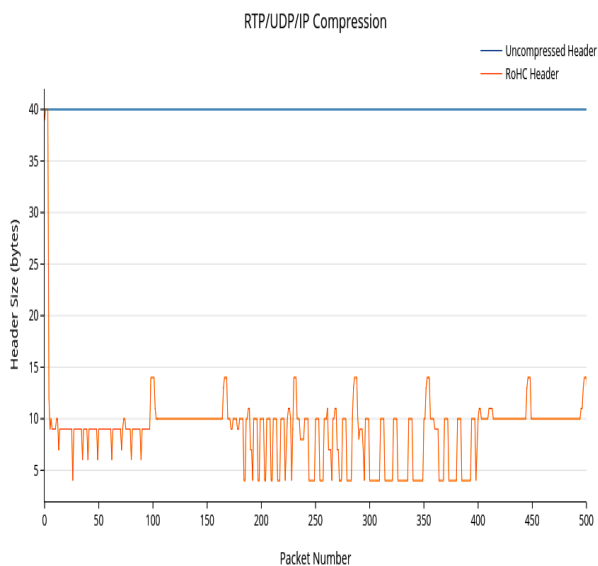


FIGURE 9. RTP/UDP/IP header compression O-mode.

Figure 9 shows the result obtained after compression an RTP/UDP/IP header in O-mode by comparing the size of the original uncompressed packet with the size of ROHC compressed packets. The 500 first packets from the stream have been selected to analyze the compression efficiency for  $0 \times 0001$  profile.

In the experiments, the Header Gain is up to 90 % for UO-1-TS packets, in the average of 500 packets the header size has been reduced from 40 bytes to 8.99 bytes which is considered as 75 % of average header gain. Figure 11 shows that the compressor greatly reduces the header length of the packet, and saves a great bandwidth resources wireless networks.

In Figure 12, the length of IPv4/UDP/RTP packet that used in the video streaming communication is 1356 Bytes, the length of payload is 1316 Bytes. Therefore, in our experiment, when the compressor and the decompressor run stably, the compressor could compress a 1356-byte packet into 1325.16 bytes to transmit when these packets compressed by

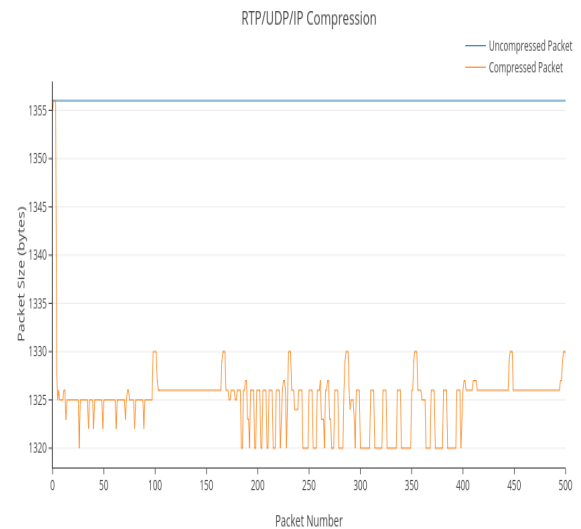


FIGURE 10. RTP/UDP/IP packet compression O-mode.

ROHC protocol. The compression gains for RTP /UDP/IP VLC stream vary between Gains is 2.21% as maximum and 1.91% as a minimum packet compression.

The unidirectional mode (see Table 2), was observes and the total compressed header savings for RTP/UDP/IP profile ( $0 \times 0001$ ) in unidirectional mode are about 77.40 %.

However, for the full RTP flow with IPv4 headers in U-mode. Figure 11 and 12 illustrate the header and packet compression for RTP/UDP/IP of 500 video streaming packets scenario in unidirectional mode. The average packet header compression gain is up to 76.66 %. The full packet stream (header + payload) saving is 2.26 %.

UDP/IP streaming compression results has been gathered. The UDP/IP compression profile result is presented in Table 3. streaming a H264 video with MP3 sound using VLC and encapsulate it in UDP packets. As shown in Table 3 four packet types have been used in UDP header compression (IR, UO-O, UO-1 and UOR-2).

The compressor used three IR packets to initialize and refresh the decompressor context, UO-0 packets have saved 89.28 %, UO-1 57 % header saving, UOR-2 achieved 75 %.



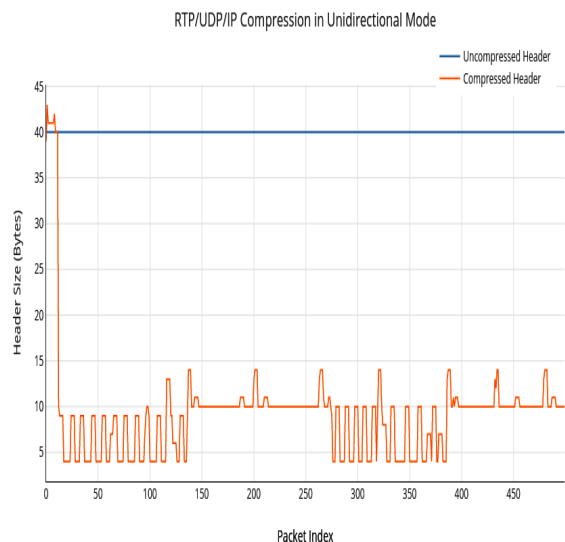


FIGURE 11. RTP/UDP/IP header compression U-mode.

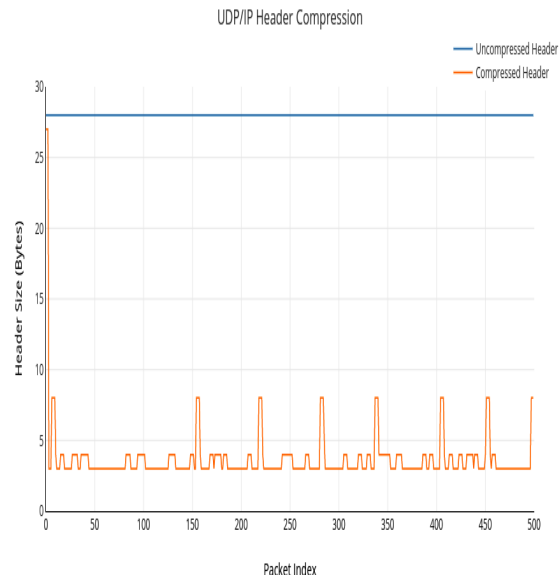


FIGURE 13. UDP/IP header compression savings.

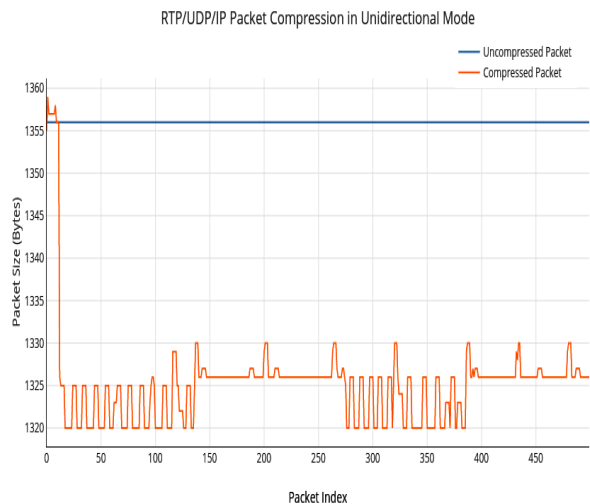


FIGURE 12. RTP/UDP/IP packet compression U-mode.

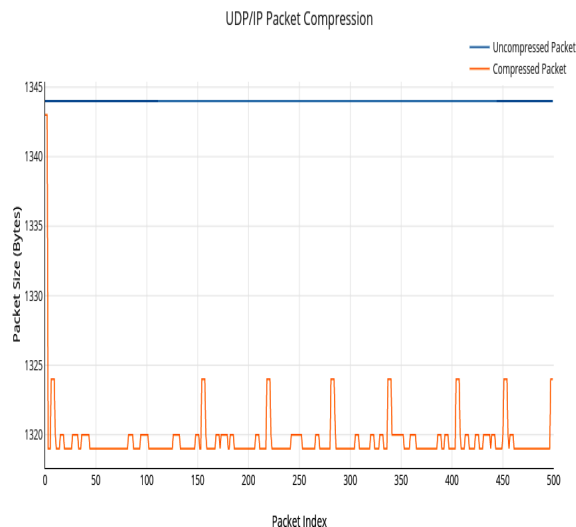


FIGURE 14. UDP/IP packet compression savings.

TABLE 3. UDP/IP flow compression details.

Packet Type	Number of packets	UPS (bytes)	UHS (bytes)	CPS (bytes)	CHS (bytes)
IR	3	4032	84	4029	81
UO-0	1784	2397696	49952	235309	5352
UO-1	618	830592	17304	815760	2472
UOR-2	205	275520	5740	271266	1486
Total	2610	3510450	73080	344415	9391

Figure 13 and 14, present the length of packet that used in the UDP multimedia communication, the value of 1344 bytes were obtained. The length of payload is 1316 bytes. The ROHC UDP/IP profile could compress a 28 bytes’ header to just 3 bytes in its best state with an average of 3.74 bytes for each compressed packet.

The TCP/IP Flow compression results is presented in Table 4. In this experiment 151 IR plus IR-DYN packets were sent to initialize and refresh the decompressor context

In table 4, we observe that the total compressed header savings for TCP/IP profile are about 63.83 % for the full captured TCP flow with IPv4 headers. Figure 15 and 16 illustrates the compression results of 500 HTTP/TCP packet flow. The size of the uncompressed packet range from 52B to 80B. The compressed headers size can go up to 7 bytes. The average header compression gains 63.35 % followed by 55.41 average packet size gain in the 500 packet flow.

The Header compression ratio comparison has been carried out. Figure 17 shows the different compression ratios in each simulated profile. The RTP/UDP/IP U-mode has nearly achieved a similar compression ratio as RTP/UDP/IP O-mode with (0.23 CR, 0.22 CR) for U and O mode. this difference in compression rates is based on the confidence variable, the number of IR packet sent in each mode, as we notice RTP U-mode compressor send 16 IR packets to synchronize

TABLE 4. TCP/IP flow compression details.

Packet Type	Number of packets	UPS (bytes)	UHS (bytes)	CPS (bytes)	CHS (bytes)
IR	90	13796	4976	13314	4494
IR-DYN	61	8831	3172	7851	2192
TCP/ co common	1163	68698	62616	29886	23804
TCP/seq 1	76	5641	5156	2066	1581
TCP/seq 2	0	0	0	0	0
TCP/seq 3	52	2704	2704	538	538
TCP/seq 4	187	9724	9724	1569	1569
TCP/seq 5	1	52	52	13	13
TCP/seq 7	409	24537	21396	8386	5245
TCP/seq 8	43	2944	2944	1342	1342
Sum	2082	136927	112740	64965	40778

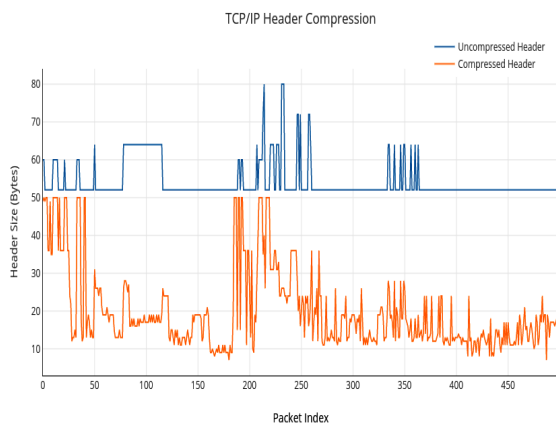


FIGURE 15. TCP/IP header compression gain.

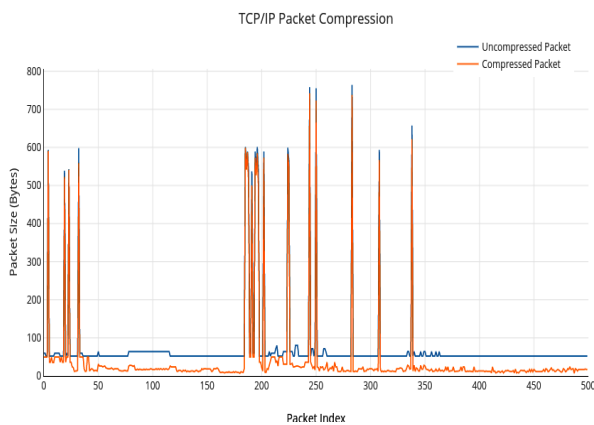


FIGURE 16. TCP/IP packet compression gain.

the decompressor context, while in O-mode just 4 IR packets have been sent which is more efficient in term of bandwidth and header gain. UDP O/U mode have a very similar compression rate with an overall average header compression ratio of 0.13 for U-mode and 0.128 for O-mode with 3 IR packets sent in both modes.

In the TCP/IP we observed an average compression rate of 0.36 which is less efficient than other profiles, this compressed TCP flow contain a different packet size with a variable payload length which lead, as we compressed

Header Compression Ratio in each Profile

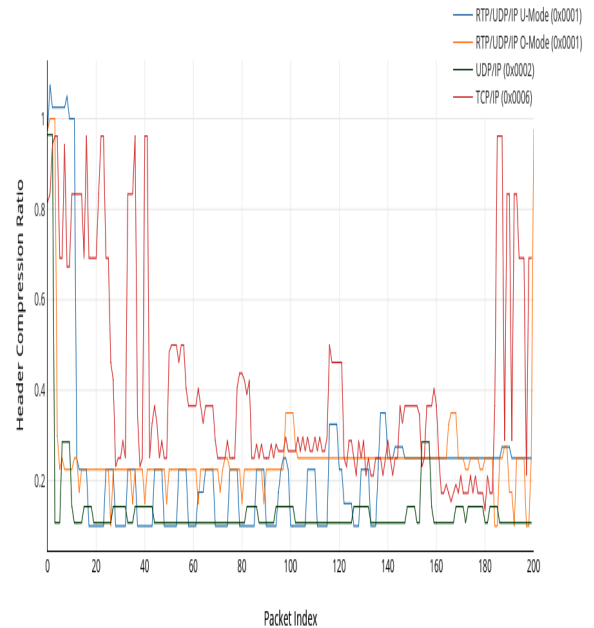


FIGURE 17. Comparison between Different profile header compression ratios.

TABLE 5. Header compression gain (best state).

Protocol	Header size (bytes)	Minimum header size (bytes)	Compression Gain ratios (%)
RTP / UDP / IPv4	40	4	90
UDP /IPv4	28	3	89
TCP/IPv4	52 to 80	7-18	77.5-86.5

and HTTP over TCP flow the destination port have been fixed to 80 but the source port kept changing during the compression, this factors lead the compressor to send periodic refreshes of static and dynamic fields (IR and IR-DYN packets) each time to keep the decompressor context synchronized. Moreover, this factors affect the compression efficiency of TCP/IP profile.

Further details of the result presented in Table 5 indicated that ROHC protocol can achieve compression ratios over 90% for RTP/UDP/IP profile, 89 % for UDP/IP profile and 77.5 to 86.5 % for TCP/IP profile.

The ROHC throughput capacity of compression and decompression is calculated by the number of received bits over the total time in second. That is the amount of data transferred over a given period of time.

Figure 18 shows the average compression/decompression bit-rate for RTP/UDP/IP, UDP/IP and TCP/IP profiles. Processing an RTP/UDP/IP stream in the optimistic mode is 18 kb/s slower than the unidirectional mode. In UDP/IP compression test the scheme can compress 46 kb/s in the unidirectional mode and 16 kb/s in O-Mode while the TCP/IP flow compression is slower that other profiles by 6 kb/s in U-mode and 2.5 kb/s in O-Mode. The interesting thing here is the asymmetry regarding to the throughput capacity of

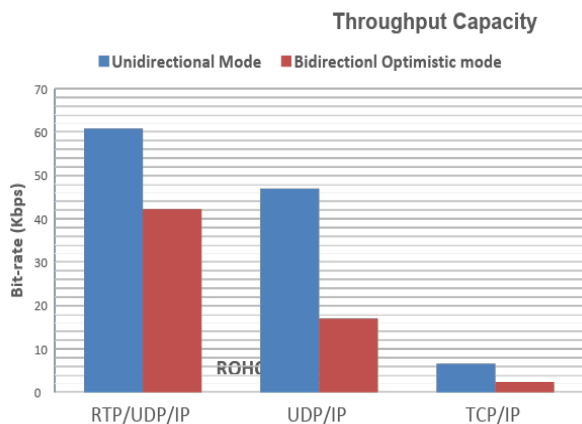


FIGURE 18. Throughput capacity.

U and O modes. We have noticed that it takes significantly longer to process packet in the optimistic mode due to the complexity of the optimistic mode operations. The ROHC TCP scheme efficiency is very low comparing to the RTP and UDP compression.

## VII. DISCUSSION

This research examined robust header compression scheme. The performance evaluation of the robust header compression scheme is crucial nowadays that it is used in mobile networks. The paper has presented the ROHC compression scheme by describing its framework and comparing it with related research of RoHC framework. Firstly, the research introduced the ROHC compression scheme by describing the protocol terminology, profiles and framework. The use of mobile phone has grown up widely, and is still increasing. It is expected that by the year 2022, the global expected IP traffic is going to 333 Exabyte per month. Furthermore, 5G network is expected to start its full implementation and that will increase the volume of traffic around the world. That is why, it is important to perform this kind of study because of the many different Internet protocols like RTP, TCP, UDP, and IP. In, UDP, RTP, TCP and IPv4, the minimum size of these uncompressed headers are respectively 8,12, 20 and 20 bytes, the compression of these headers lead to a very significant increase of the network capacity especially in the case of interactive multimedia streams packet headers. In order to compress RTP/UDP/IP, UDP/IP and TCP/IP packet headers.

The development of the robust header compression protocol U and O mode implementation began by defining the protocol logical architecture and choosing the right libraries and development tools. This study RoHC implementation was basically focused on the RTP/UDP/IP ( $0 \times 0001$ ) and UDP/IP ( $0 \times 0001$ ) profiles in addition to TCP/IP ( $0 \times 0006$ ) flow compression test.

This implementation has been set up in a platform to measure to the different the protocols of compression techniques in order to see the impact of ROHC. We have integrated ROHC compression and decompression modules in Linux based virtual machine (VMware Workstation) by using the ROHC and Pcap libraries. The compression and

decompression were defined in our testbed scenarios and configuration for the performance evaluation. Another experiment on different flows without link proprieties consideration was also carried out. The performance of RoHC is assessed in term of compression efficiency in each compression profile using the experimental results were presented and analyzed.

The principal finding of this study lies with the compression ratio. Based on the original IP header and other research studies compression techniques. This current study has been able to achieved a reduction of up to 90% for RTP/UDP/IP, 89% for UDP /IP and 77.5 to 86.5 % for TCP/IP profile. The media used for this study focused on UDP and RTP. These are the protocols that are mainly used for audio and video media flow. The transmission load of this protocols is much higher than TCP transmission. Similarly, TCP stream compression has been one of the area which work is still underway to improve its efficiency.

In terms of compression gain, in the unidirectional mode, the compressor periodically returns to the levels where it has to send larger packets. In addition, it has no feedback, so the compression level is constant. However, in bidirectional mode, it only returns to lower levels or it receives NACKs. Based on the comparison between the unidirectional modes and a bidirectional optimistic mode implementation. For lossless channels, this study recommends the unidirectional mode for transmission without a feedback channel.

This research contribution is limited to compression gain and saving for  $0 \times 0000$ ,  $0 \times 0001$ ,  $0 \times 0002$  and  $0 \times 0006$  profiles in the unidirectional and bidirectional optimistic modes by using Linux user space measurements including compression and decompression processes. It would be useful for future experiment to repeat this measurement on a real RoHC protocol implementation with channel characteristic considerations and study the different parameters that can affect compression and decompression process like W-LSB window width, Confidence variables, Compression and decompression Timeouts, delay. Besides that, it would be interesting to measure the bidirectional reliable approach performance.

## VIII. CONCLUSION

This study recognizes the issues of improving efficiency of data communication by reducing the header redundancy. IP packets are designated to have specific header for TCP and UDP while transmitting data. Under these protocols it is expected that the data should be within MTU. The size of the payload they would carry will depend on the media type. This study examine transmission on video and audio. As a result, a RoHC technique were utilized. Specifically, the performance analysis of the efficiency of the techniques in both unidirectional and bidirectional optimistic modes applied to the real-time video streaming traffic for UDP/IP and HTTP/TCP has been evaluated. Several analyses were performed. The finding indicated very good percentage of the header compressed on the transmission. The proposed approached for this study has achieved a compression rate

of up to 90% for RTP/UDP/IP, 89% for UDP /IP and 77.5 to 86.5 % for TCP/IP profile. The different compression ratios have been determined in each case by simulated profile. The RTP/UDP/IP U-mode compression ratio to RTP/UDP/IP O-mode with (0.23 CR, 0.22 CR) for U and O mode were almost the same.

## ACKNOWLEDGMENT

The authors, therefore, gratefully acknowledge DSR technical support.

## REFERENCES

- [1] L. Chen and G. Gong, *Communication System Security* (Cryptography and Network Security). Boca Raton, FL, USA: CRC Press, 2012.
- [2] B. Williamson, and S. Wood, S, "Mobile value, spectrum and data demand—A bootstrap approach to estimation," *Digit. Policy, Regulation Governance*, vol. 19, no. 1, pp. 101–112, 2017.
- [3] Y. Al Mtawa, A. Haque, and B. Bitar, "The mammoth Internet: Are we ready?" *IEEE Access*, vol. 7, pp. 132894–132908, 2019.
- [4] S. Sun, K. Adachi, P. H. Tan, Y. Zhou, J. Joung, and C. K. Ho, "Heterogeneous network: An evolutionary path to 5G," in *Proc. APCC*, Oct. 2015, pp. 174–178.
- [5] R. N. Mitra and D. P. Agrawal, "5G mobile technology: A survey," *ICT Express*, vol. 1, no. 3, pp. 132–137, Dec. 2015.
- [6] M. A. Chishty, S. Quershi, and A. H. Mir, *Robust Header Compression (RoHC) Over Multiprotocol Label Switching (MPLS) Networks*. Munich, Germany: Grin Publishing, 2017.
- [7] S. Debbarma, "Performance measurement of TCP/IP header compression," *Int. J. Electron. Commun. Eng.*, vol. 4, pp. 399–404, Jan. 2011.
- [8] A. B. Effnet, "The concept of robust header compression," ROHC, White Paper, 2017. Accessed: Jan. 21, 2019. [Online]. Available: <http://www.effnet.com/products/rohc/>
- [9] M. Tömösközi, P. Seeling, and F. H. P. Fitzek, "Performance evaluation and comparison of ROHCv1 and ROHCv2 for multimedia delivery," in *Proc. IEEE GC Wkshps*, Dec. 2013, pp. 544–549.
- [10] M. Tömösközi, P. Seeling, P. Ekler, and F. Fitzek, "Performance evaluation and implementation of IP and robust header compression schemes for TCP and UDP traffic in static and dynamic wireless contexts," *Comput. Sci. Inf. Syst.*, vol. 14, no. 2, pp. 283–308, 2017.
- [11] A. Couvreur, L.-M.-L. Ny, A. Minaburo, G. Rubino, B. Sericola, and L. Toutain, "Performance analysis of a header compression protocol: The ROHC unidirectional mode," *Telecommun. Syst.*, vol. 31, no. 1, pp. 85–98, Jan. 2006.
- [12] Y. Zhao, B. Zhang, L. Wang, F. Xie, and C. Li, "Selective redundant transmissions for real-time video streaming over multi-interface wireless terminals," in *Proc IEEE GLOBECOM*, Jun. 2016, pp. 1–6.
- [13] K. Kumar, A. Prakash, and R. Tripathi, "Spectrum handoff scheme with multiple attributes decision making for optimal network selection in cognitive radio networks," *Digit. Commun. Netw.*, vol. 3, no. 3, pp. 164–175, Aug. 2017.
- [14] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications*, document RFC 1889, Jan. 2019. [Online]. Available: [https://www.researchgate.net/publication/246511581\\_RTP\\_A\\_Transport\\_Protocol\\_for\\_Real-time\\_Applications](https://www.researchgate.net/publication/246511581_RTP_A_Transport_Protocol_for_Real-time_Applications)
- [15] X. Chen, F. Guo, P. Dang, and L. Wu, "A survey of ROHC header compression schemes," in *Proc. IEEE ICCSNT*, Dec. 2012, pp. 331–335.
- [16] W. Wu and Z. Ding, "On efficient packet-switched wireless networking: A Markovian approach to trans-layer design and optimization of ROHC," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4232–4245, Jul. 2017.
- [17] S.-J. Nam, J. Kang, and D. Moon, "Wireless packet header compression techniques for ship area sensor Network(SASN)," in *Proc. IEEE ICCAS*, Oct. 2015, pp. 2048–2052.
- [18] C. Jiang, W. Wu, and Z. Ding, "IP packet header compression and user grouping for LTE multimedia broadcast multicast services," in *Proc. IEEE ICNC*, Jan. 2017, pp. 52–57.
- [19] M. Yan and S. Zhang, "The design of the ROHC header compression accelerator," in *Proc CCF-NCCET*. Berlin, Germany: Springer, 2012, pp. 75–83.
- [20] J. Kaur, D. Gupta, and M. Chaturvedi, "Performance enhancement of ROHC U-mode in wireless links," in *Proc. IEEE ICACCS*, Jan. 2015, pp. 1–4.
- [21] S. Traboulsi, W. Zhang, D. Szczesny, A. Showk, and A. Bilgic, "An energy-efficient hardware accelerator for robust header compression in LTE-advanced terminals," in *Proc. 22nd Int. Conf. Field Program. Log. Appl. (FPL)*, Aug. 2012, pp. 691–694.
- [22] M. N. Napiyah, M. Y. I. Bin Idris, R. Ramli, and I. Ahmady, "Compression header analyzer intrusion detection system (CHA-IDS) for 6LoWPAN communication protocol," *IEEE Access*, vol. 6, pp. 16623–16638, 2018.
- [23] R. Hermenier, F. Rossetto, and M. Berioli, "On the behavior of ROHC header compression U-mode in channels with memory," *IEEE Trans. Wireless Commun.*, vol. 12, no. 8, pp. 3722–3732, Aug. 2013.
- [24] D. E. Taylor, A. Herkersdorf, A. Doring, and G. Dittmann, "Robust header compression (ROHC) in next-generation network processors," *IEEE/ACM Trans. Netw.*, vol. 13, no. 4, pp. 755–768, Aug. 2005.



**DOUMA BOUTHIBA FAROUQ** received the master's degree in information and communication technology from International Islamic University Malaysia. He is currently a Security Engineer with Diyar United Company, Qatar. His measure research area is on enterprise network and security domain handling design, specializing in IP network management products and services implementation and troubleshooting of wide range of solutions as Firewalls, Web gateways, IPS/IDS, WAF, LoadBalancers, SIEM, and compliance based security solutions. His current main research interests include softwarized and agile network operations, cloud and hybrid based architectures and their impact on network security as well as the IoT, legacy infrastructures, and the performance assessment of these technologies and architectures.



**ALA ABDUSALAM ALAROOD** received the B.Sc. and M.Sc. degrees in computer science, and the Ph.D. degree in computer science from the University of Technology Malaysia. He is currently an Assistant Professor with the Faculty of Computer Science and Engineering, University of Jeddah. His current research interests are in artificial neural networks, machine learning, and cyber security.



**NAHLA ALJOJO** received the Ph.D. degree in computing from Portsmouth University. She is working currently as an Associate Professor with the Information system and information Technology Department, College of Computer science and Engineering, University of Jeddah, Jeddah, Saudi Arabia. Her research interests include adaptivity in web-based educational systems, E-business, leadership's studies, information security and data integrity, E-Learning, education, machine learning, health informatics, environment and ecology, and logistics and supply chain management.



**ADAMU ABUBAKAR** the B.Sc. degree (Hons.) in geography and the P.G.D., M.Sc., and Ph.D. degrees in computer science from Bayero University, Kano, Nigeria and IIUM, respectively. Prior to his completion of M.Sc. degree programme from 2005 to 2007, he has worked in various area of information technology. He is currently an Assistant Professor with the Department of Computer Science, International Islamic University Malaysia (IIUM). He is working in the areas of computer networks, 3D mobile map, information retrieval technologies, digital watermarking, steganography, artificial neural networks, and wireless sensor networks. He had published many articles in conferences, journals, and book chapters during his Ph.D., Postdoctoral and at the current position of an Assistant Professor. During his Ph.D., he worked on many research projects. He received many awards research. Dr Adamu has been a Professional Member of Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE), since 2014.