

Performance Evaluation and Compression of IP Packets in a Wireless Local Area Network (WLAN)

Bernard Sentala*, Charles S. Lubobya, Ackim Zulu

Department of Electrical Engineering, University of Zambia, School of Engineering, Lusaka, Zambia

Abstract The study focuses on the Performance Evaluation and Compression of IP Packets in a Wireless Local Area Network (WLAN). WLAN has many network challenges such as network delay, buffering requirement, bandwidth, and network congestion. The aim of the study is to design and develop a WLAN model that would assist test the Performance Evaluation and Compression of IP Packets in a WLAN. The key specific objectives are to assess the effectiveness and efficiency of per-interface compression in a WLAN during IP packet transmission; to analyze the effects of TCP/IP header compression and per-virtual circuit compression on performance and/or quality of service (QoS) in a WLAN, and to design and develop the best compression scheme in transmitting IP Packets in a WLAN. OPNET simulation software tool has been used to simulate the four (4) IP Packet Compression Schemes that include; No compression, Per-interface Compression, TCP/IP Header Compression, and Per-virtual Circuit Compression. The results indicate that Per-virtual Circuit Compression Scheme performed below Per-Interface and TCP/IP Header Compression Schemes. No Compression Scheme is utilized solely for comparative reasons and the scheme recorded the lowest score of all the compression schemes. In terms of overall performance, the Per-Interface Compression algorithm excelled over all other compression schemes tested in the study.

Keywords Compression, OPNET, WLAN, Efficiency, and performance

1. Introduction

In general, IP packet compression is the process that reduces the packet size or file size of digital signals without changing the amount of digital file signals during data transmission without affecting the signal amount of digital files in a Wireless Local Area Network (WLAN) [1] [2]. To simulate and come up with a better solution, the study employed Optimized Network Engineering Tool (OPNET) software for computer network simulation, modeling, and analysis. This was done in order to create optimized and/or mitigated IP Packet delivery challenges, the study aim was to design and develop a model that would help in testing the Performance Evaluation and Compression of IP Packets in a Wireless Local Area Network (WLAN) using four (4) compression schemes while taking performance and Quality of Service into consideration. The key specific objectives were: to assess the effectiveness and efficiency of the per-interface compression in a WLAN during IP packet transmission; to analyze the effects of TCP/IP header and per-virtual circuit compression on performance and/or quality of service (QoS) in a WLAN, and to design and develop the best compression scheme in transmitting IP

Packets in a WLAN. The network problems such as Queuing, transmission, propagation, and processing delays, including buffering needs and network congestion of IP packets, could be reduced and/ or mitigated when the available resources are employed as efficiently and effectively as possible [1] [2] [3] [4].

The study focused on the Performance Evaluation and Compression of IP Packets in a Wireless Local Area Network (WLAN) as a strategy to optimize and reduce packet network delay problems in connection with IP Packet delivery [2]. Data compression is a coding system that allows characters to be eliminated from data frames on the sending side of a transmission device and appropriately substituted on the receiving side [3]. Compression is classified into two (2) major categories: lossy and lossless. Lossy compression occurs when data is compressed to reduce the file size by permanently deleting unnecessary information, such as a picture, and image quality is lowered to a minimum. A lossless compression system is one in which X and Y are identical, whereas a lossy compression technique gives substantially better compression than lossless compression but enables Y to diverge from X [3] [4] [5].

The study found that the Per-Virtual Circuit Compression technique continuously trailed behind the Per-Interface Compression and TCP/IP header compression schemes. None, no compression scheme is used merely for comparison purposes, and this scheme obtained the lowest

* Corresponding author:

bsentala@gmail.com (Bernard Sentala)

Received: Oct. 13, 2022; Accepted: Oct. 24, 2022; Published: Nov. 14, 2022

Published online at <http://journal.sapub.org/jwnc>

score of all the compression schemes tested. The per-interface compression Scheme has provided the best performance and good quality of service during the study.

The paper is therefore, organized through sections consisting of the following headings: introduction, related work, data compression, research methodology, results and discussion, and conclusion.

2. Related Work

Compressing and decompressing of data files for storage is basically the same as sending and receiving compressed data. IP packet compression in Wireless Local Area Network (WLAN) and other networks such as mobile ad hoc networks (MANETs) and satellites have received a lot of attention in the recent past. This section gives a review of related works on data compression schemes.

According to Cha, Hyejin, et al. [6], audio and video packets are sent through IP, UDP, and RTP across a streaming media network. Inefficient data transmission might result from redundant header data on continually sent packets. The Robust Header Compression (RoHC) method was employed to prevent redundancy header data that could be detected. It employed Window Least Significant Bit (WLSB), an encoding technique that makes advantage of the window offset value. The research developed an adaptive approach for adjusting the window shift value when sending RTP packets. According to the findings, the window shift value is changed based on the amount of packet losses. The suggested method's experimental findings show that as the window shift value is changed, the header compression ratios rise and the average header size decreases. The redundancy detected in individual packet headers and subsequent packets belonging to the same IP flow determines the reduction in protocol overhead [6].

Mate Tomoskozi et al. [7], states that the 5th Generation (5G) mesh networks may have protocol encapsulation overhead that surpasses the amount of data to be provided. One way, according to the study, would be to use header compression technologies to reduce the size of individual protocol headers. Robust Header Compression (RoHC) is a well-established standard that properly handles IP-based compression. Because compression functions on a peer-to-peer, single-hop basis, the existing approach at the time of the study did not allow for the installation of wireless mesh networks. The study indicates that payload delivery overhead for RTP transmissions could be cut in half even when a standard like RoHCv2 has difficulties and may fail to compress the IP packet. The study only examined RTP transport; hence the results may differ in TCP and UDP transport scenarios done under WLAN platform [7].

Matias and Refua [8], argues that data compression for IP Packet networks, is transmitted by partitioning it into IP Packets. IP Packet Compression allows better bandwidth utilization of a communication line resulting in much smaller amounts of the packet drops, more simultaneous sessions,

and smooth behaviour of applications. IP Packet Compression can be obtained by a combination of header compression and payload compression, which are complementary methods. However, the study does not suggest or recommend the use of other compression schemes such as Per-Interface compression as the approach for optimisation that can be undertaken in a WLAN platform [8].

Carlos Feres [9], argues that Robust Header Compression (RoHC) is used in packet-switched wireless networks such as 4G and 5G cellular systems to reduce Packet Data Convergence Protocol (PDCP) header length and improve payload efficiency. Recent works on RoHC control based on a Partially Observable Markov Decision Process (POMDP) formulation has emphasized the need for employing a trans-layer strategy that makes use of lower layer information. The POMDP approach has the advantage of reducing the computational complexity of traditional RoHC. The study emphasizes simplicity by building RoHC compressors using lower layer information, such as channel adaptive transport block size due to link adaptation, which is typical in many wireless networks. The model was directly aimed at practical implementation and can achieve transmission efficiency equivalent to optimal POMDP compression [9].

Lubobya et al. [10], investigated the computation time of Single Input Multiple Data (SIMD) when compressing static video packets (or datagrams) on a stand-alone personal computer. In their study, they achieved SIMD computation time increases of 14,000ms and 17,000ms for pairs of Integer DCT and Hadamard transforms, which were limited to static video that was compressed without sending the video over a Wireless Local Area Network (WLAN) [10].

In this study, we evaluated the comparative compression of video conferencing by simulating four (4) compression algorithms using the Optimized Network Engineering Tool (OPNET) software to assess the performance evaluation and compression of IP packets in a full-function WLAN platform.

3. Data Compression

The increasing expansion of data in the digital age has necessitated the development of effective data compression solutions. Data compression, in general, originally occurred in the early nineteenth century. Data compression techniques result in more efficient use of available storage space and bandwidth, which improves performance and throughput during packet delivery. Data compression is the process of decreasing the amount of data into a smaller yet more compact form. Compression is the process of transforming a set of data into a code in order to reduce the demand for data storage and transmission speed [5] [11].

Data compression entails identifying models for the many different sorts of structures found in various forms of data. These might take the shape of patterns that can be identified simply by changing the data, or statistical structures that

need a more sophisticated approach [12]. Data compression is essential because uncompressed data takes up a lot of space, reducing the effectiveness and efficiency of CPU power owing to the restricted hard drive space available. Compression provides benefits such as reduced resource utilization and cost [11] [12].

Lossless and lossy compression techniques are the most common types of compression algorithms. Text data, for example, is represented in a compact form through lossless data compression. Lossy data compression allows for data loss during the compression process. Due to human ears senses that are imprecise, certain tones in compressed audio data are not perceptible. Huffman, Lempel Ziv Welch, and Run Length Encoding are examples of compression algorithm approaches [11] [12] [13]. Below is Figure 1. Lossless and Lossy Compression [14].

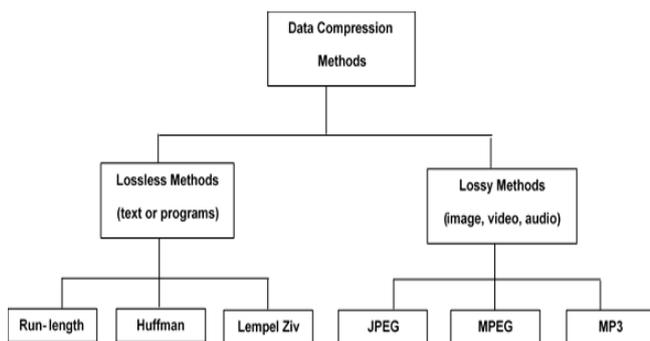


Figure 1. Lossless and Lossy Compression [14]

3.1. Packet Compression Techniques

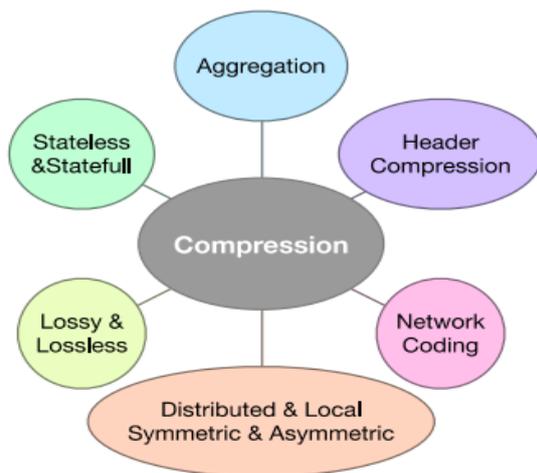


Figure 2. Compression techniques [15]

The recent appearance and popularity of the wireless internet have created a desire for increased transmission efficiency, particularly if the link has a high cost of communication bytes. IP Packet compression at the connection layer may speed up delivery and make better use of available bandwidth capacity. IP Packet compression can be configured to compress simply the header, only the payload, or both sections of the packet. There are numerous approaches for compressing IP packets, that may include

stateless and stateful compression, lossless and lossy compression, and header compression. There are several techniques for compressing IP Packets that are indicated below in Figure 2. compression techniques [15].

The study's main focus is on three (3) compression techniques namely; Stateless and stateful, Lossless and Lossy, and Header compression.

3.1.1. Stateless and Stateful

When using stateless compression, packets are compressed and decompressed separately, with no reference to any other packet. When packets come out of order at the receiving end, the compression and decompression strategy is advantageous. Stateless meaning that it compresses any pieces of data it encounters without relying on past inputs [16].

Stateful compression processes are those that do not yet have access to memory but utilize it to make choices. There would always be a discussion between the sender and the receiver in the case of stateful compression. During the transmission, both sides agree on the semantics of the compression [12].

When using stateless compression, packets are compressed and decompressed separately, with no reference to one other. Since there is no need to presume anything before the negotiation, the stateful technique would allow for significantly higher compression rates than the stateless approach. Due to the nature of wireless communication, some packet loss and bit mistakes are unavoidable; nonetheless, if the losses and errors are extremely minimal, the stateless technique outperforms the stateful approach [12]. Data Compression methods can be divided into lossy and lossless techniques.

3.1.2. Lossless Techniques

The goal of the lossless approach is to return the original data. This approach may be used to compress data such as text or programmes, but it can also compress picture and video data. There are several lossless approaches or techniques, which include:

Run Length Encoding: RLE is a fundamental data compression algorithm that is supported by bitmap file formats such as bitmap picture files (BMP). RLE compresses data by reducing the physical size of a repeated string of letters. It can also be combined with other compression methods [17].

Lempel-Ziv-Welch (LZW): The LZW is a general-purpose compression method that can handle practically any sort of data. It substitutes the real data with references to a database of strings that are often found in the data being compressed. The table is created during compression when the data is encoded and decoded [17].

Huffman Coding: Huffman coding is used to compress data containing American Standard Code for Information Interchange (ASCII) characters. It is used to compress several sorts of data, including text, audio, video, and images.

This approach is based on creating a complete binary tree for all of the symbols in the original file [18].

3.2. Lossy Techniques

Lossy compression differs from lossless compression in that some data may be lost throughout the process, as the term suggests. Lossy compression methods are commonly employed to compress data that originated as analog signals, such as voice, video, images, and music files that may be cut at the margins. Pictures, unlike text and processing files, do not require reconstruction to be identical to the original, especially if the data lost is minor or undetected. Lossy compression might be used in the video, picture, and audio/voice applications. JPEG, MP3, and MPEG are the most common lossy compression-based picture formats, and the many approaches used include Chroma subsampling, colour reduction, fractal compression, transform coding, vector quantization, and many more [17].

3.3. Header Compression

A network packet is a packet that comprises data and header information from several protocols. The header section is noteworthy because it contains duplicate information about distinct protocol headers, particularly across consecutive packets belonging to the same flow. This type of redundant data can be omitted, combined, or eliminated. Not all of that information is required over a single link, and some of it can be deleted temporarily [18].

4. Research Methodology

This section dealt with research procedures for gathering data, variables involved, an explanation of how data was analysed and any other information that may be required to efficaciously achieve the aim and objectives of the study.

The data gathering instrument tool employed during the study was the Optimized Network Engineering (OPNET) Tool Modeler 14.5 Release, which were designed to answer the key questions of the research. The data obtained were exported to Microsoft Excel for analysis and presentation of the results that was collected from the simulations. During simulation, the metrics measured were throughput, Data dropped, Delay, Load, Network Load, Traffic sent and Media Access Delay that are supported by OPNET simulation software tool. The OPNET Modeler simulator was chosen because it supports the four (4) IP Packet Compression algorithms via its IP Attribute Config object, which accepts just one of the four (4) values: i) None (no compression), ii) TCP/IP Header Compression (just the TCP/IP header is compressed), iii) Per-Interface Compression (the complete packet is compressed), and iv) Per-Virtual Circuit Compression (only the packet's payload, not the header information is compressed). Simulations were undertaken for all the four (4) scenarios that the study had planned to undertake.

4.1. Research Methodology Process

In this sub-section, the strategy was to describe a general plan about how the researcher intended to answer the key questions relating to the study. The study undertook four (4) scenarios as indicated below in Figure 3., to determine its performance in terms of Effectiveness and Efficiency. The results of the research were then analysed, validated, and documented. The model was developed using the following process shown in Figure 3. Research Methodology Process.

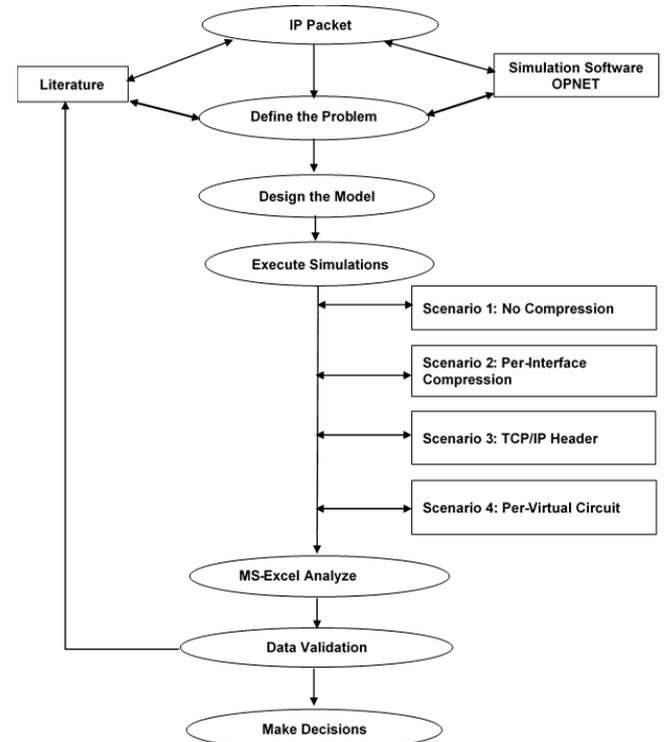


Figure 3. Research Methodology Process

4.2. Network System Design

The Network Topology implementation of the Wireless Local Area Networks (WLAN) was done by designing the network that included the Server, Access Point (AP), IP-Cloud, Gateway Router (Ethernet4_slip_gtwy), Seven (7) Wireless Workstations, Applications, Profiles, Task and IP Attribute Config object as shown in Figure 2. WLAN Topology. The 100base-T cable was used to connect the devices from Access Point (AP) to the Gateway router and from Gateway router to the IP cloud PPP_DS3 was used. Similarly, from IP Cloud to PPP Server, PPP_DS3 cable was used. It is important to note that the 100Base-T link represents an Ethernet connection operating at 100Mbps (i.e. 10 times faster than standard Ethernet). The Basic Service Set Identifier (BSSID) was set to 1 for all the nodes and Access Point (AP). The BSSID is the Media Access Control (MAC) physical address of the Access Point or wireless router which was used to connect to the Wi-Fi and that the term is used in wireless network. The Basic Service Set (BSS) is the cornerstone topology of any 802.11 network.

Below is Figure 4. WLAN Topology.

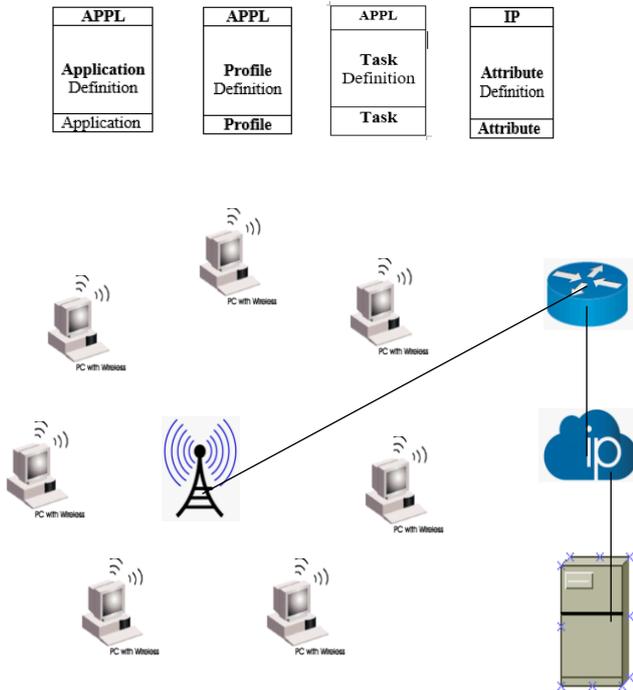


Figure 4. WLAN Topology

4.3. OPNET

For the purpose of this study, OPNET simulator tool is used as it supports IP Packet Compression through the IP Compression Information Attribute in the IP Attribute Config object. By default, the IP Compression Information attribute in OPNET contains a set of pre-configured compression schemes that include; None no compression, TCP/IP header compression, per-interface compression, per virtual circuit compression, image compression, and Telnet application compression [3].

4.3.1. Simulation Setup

In this study, the access point and mobile node settings were configured as given in table 1.

Table 1. Access Point and Mobile Node parameter

Wi-Fi 1802.11g		
	AP 1(Access 1Point)	Mobile 1Node
Tx 1Power	0.1W	0.1W
Data 1Rate	11Mbps	11Mbps
Received 1Power 1Threshold	-95dBm	-95dBm
Buffer 1Size	1024000 1bits	1024000 1bits
Short 1retry 1Limit	7	7
Long 1Retry 1Limit	4	9
Large 1Packet 1Processing	Fragment	Fragment
Access 1Point 1Functionality	Enabled	Disabled

The IP Attribute Config object was set as shown below in table 2. IP Attribute Config parameters.

Table 2. IP Attribute Config parameters

Scenario(s)	IP Attribute Config object	Traffic Sent	Compression Ratio	Simulation Time (Sec)
One (1)	No compression	Video	0.5	1200
Two (2)	Per-Interface Compression	Video	0.5	1200
Three (3)	TCP/IP Header Compression	Video	0.5	1200
Four (4)	Per-Virtual Circuit compression	Video	0.5	1200

5. Results and Discussion

In this subsection, the study shows and analyzes the results of OPNET simulations performed using the approach provided earlier under the subtitle Research Methodology. The researcher ran OPNET simulations over a simulated WLAN using four (4) compression schemes: i) none, no compression, ii) compression at the interface, the entire packet compressed, iii) TCP/IP header compression, only the TCP/IP header is compressed, and iv) Per-Virtual Circuit Compression, only the packet payload is compressed. Each compression strategy was performed as a scenario, with seven (7) clients/users, a time of 1200 seconds, and a data rate of 11 Mbps was considered.

The following stage was to evaluate the performance and Quality of Service of all compression techniques using measurable metrics such as throughput, data lost, traffic sent, and media access latency. These measures are decisive in terms of overall performance and quality of service. The obtained data were exported to Microsoft Excel for additional analysis and comparative discussion using graphs.

5.1. Throughput

The number of information bits successfully delivered or received per second is known as throughput [19]. It indicates how much data can be moved from one point to another in a given length of time across a network. Network throughput monitors wireless media and wireless bandwidth usage, which is a finite resource in a network, hence making the best use of these resources is vital. The study examined the throughput graphs for the four (4) compression schemes. Figure 5. depicts the results of the IP packet compression approaches obtained from the OPNET simulation. The x-axis (horizontal) represents time, whereas the labels on the y-bar denote throughput (vertical) bits per second.

In Figure 5., we observed that the time needed to establish a WLAN connection was from 0 to 96 seconds, after which the graphs for all compression algorithms began to rise.

According to the results, the time for the Per-Interface compression graph increased from 96 to 108 seconds, and the throughput increased from 0 to 432925 bits per second. The time for None no compression graph increased from 96 to 108 seconds and throughput from 0 to 1873536 bits per second. The time for the Per-Virtual Circuit compression graph increased from 96 to 108 seconds, and throughput from 0 to 1873536 bits per second. While the time for TCP/IP header compression graph increased from 96 to 120 seconds and throughput from 0 to 4031784 bits per second.

In explaining the results in Figure 5., we note that throughput is one of the parameter metrics used to measure performance and Quality of Service of the network. When

the graphs in figure 5., heads upwards as the time progresses and the number of clients/users increases, this indicates that the server is processing the requests without any problems as well as the network bandwidth is sufficient.

Once the graphs become relatively flat as the time progresses and the number of clients/users increases, then there is low bandwidth or requests have piled up at the server but the server is still processing the requests without any queue errors. If the issue is with the network bandwidth, then we can see such type of flat graphs because low bandwidth limits the amount of data which is transferred between the client/user and the server.

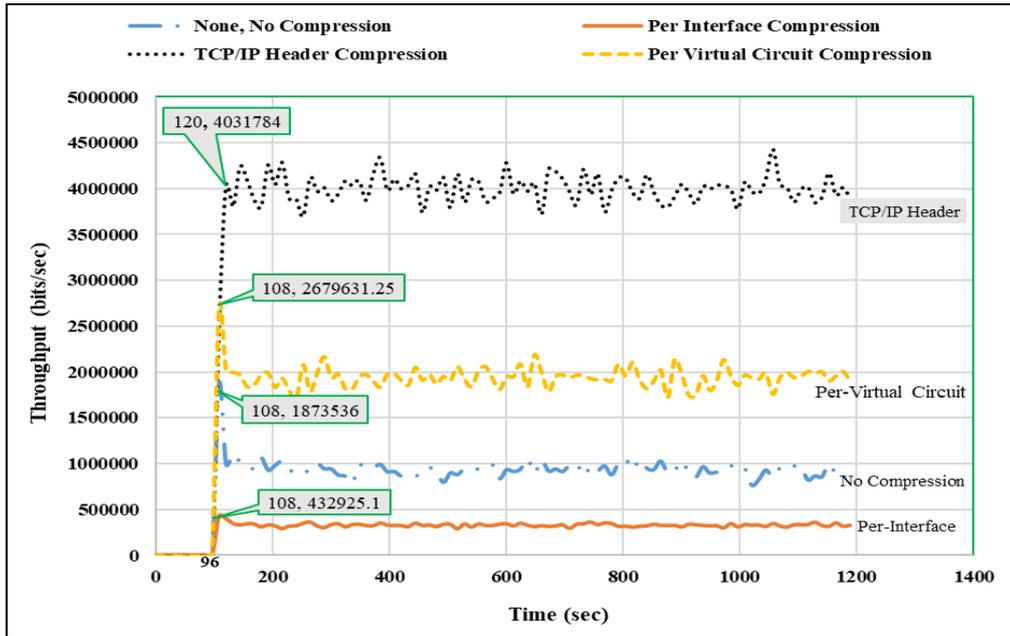


Figure 5. Throughput (bits/sec)

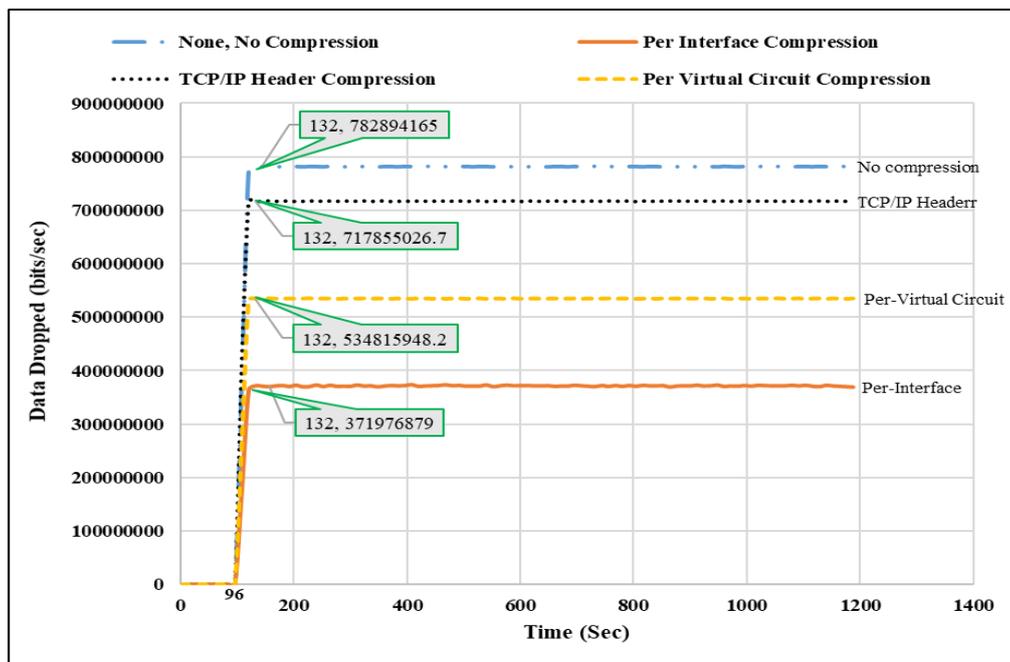


Figure 6. Data dropped (bits/sec)

If a certain drop appears in the graphs during the steady state, this indicates that network bandwidth is insufficient or the problem could be with the server.

We can therefore, state that TCP/IP Header compression had the highest throughput and better network connectivity to achieve good performance leading to good Quality of Service for the WLAN. The Per Interface Compression scheme had the lowest throughput and probably poor network connectivity which could not achieve good performance and Quality of Service for the WLAN.

5.2. Data dropped (Packet Loss)

The Data dropped is the total amount of bits sent by a wireless node but never received by the other node due to reasons such as network congestion and delay, router problems, network congestion queues, and the longer the packet travels, the greater the probability of data loss [20]. It might also be due to a lack of media access, which has a significant impact on WLAN dependability. The study compared the Data dropped graphs for the four (4) compression techniques evaluated. The results of the IP packet compression techniques acquired from the OPNET simulation are shown in Figure 6. below. The x-axis (horizontal) represents time in seconds, while the labels on the y-bar (vertical) reflect data decrease in bits per second.

Figure 6. shows that the time needed to establish a WLAN connection was from 0 to 96 seconds, after which the data dropped for all compression algorithms began to rise. The findings show that the time for compression using Per-Interface increased from 96 seconds to 132 seconds, and data dropped increased from 0 to 371976979 bits per second. The time for the Per Virtual Circuit compression increased from 96 to 132 seconds, and data dropped increased from 0 to 534815948.2 bits per second. The time for the TCP/IP Header Compression increased from 96 seconds to 132 seconds, and data dropped increased from 0 to 717855026.7 bits per second. While the time for None no Compression increased from 96 to 132 seconds, and data dropped increased from 0 to 782894165 bits per second.

Figure 6. demonstrates that the Operating System had a lower capacity than it was intended for at the start-up of the network. This may have made the network less robust, making it difficult to handle the requests or packets, leading to lost packets. A WLAN may potentially lose packets as a result of weak signals, distance, and actual obstructions like walls.

Once the graph becomes relatively flat as the time progresses from 96 to 132 seconds, then there is sufficient bandwidth or all requests are being processed by the server without any queue errors. If the issue initially of Data dropped was with bandwidth as clients/users were trying to access the sever at the same time, then we can see a flat graph like the one in figure 6. If the graph shows some drop during steady state, it means that the network bandwidth is more than sufficient and the server can handle all the requests without any queue errors.

Per-Interface compression scheme had a better network connection to be able to achieve high performance, resulting in good quality of service in the WLAN. This is attributed to the compression that had taken place for both the header and payload as the frames become smaller and the overall transmission delay reduced despite the scheme losing time due to compression and decompression processes that consumed a certain amount of time, though response time was faster.

The compression scheme with the highest data loss was None no Compression which had poor network connectivity which could not achieve good results in a WLAN compared to other compression schemes. The Per Interface compression had the lowest data dropped in this case which had good network connectivity that achieved good results in a WLAN. In video transmission a single packet drop can result in the video impairment and could degrade the video quality. When there are many packet drops in a short period of time, that may lead to poor video quality that may become unacceptable.

5.3. Traffic Sent

Traffic sent, also known as network traffic or data traffic, refers to the volume of data moving across a computer network at any given time [21]. The study compared Traffic Sent for the four (4) compression schemes evaluated. The results of the IP packet compression schemes acquired from the OPNET simulation are shown below in Figure 7. Time is shown on the x-axis (horizontal), while the labels on the y-bar (vertical) signify Traffic Sent in bytes per second.

Figure 7, shows Traffic Sent for each of the four (4) compression techniques. It was observed that the time needed for the WLAN to establish the connection ranged from 0 to 96 seconds, after which Traffic Sent began to increase for all the compression algorithms. The results provided in Figure 7. demonstrate that the time for Per-virtual circuit compression increased from 96 seconds to 120 seconds, and traffic sent increased from 0 to 76137600 bytes per second. The time for Per-Interface Compression increased from 96 seconds to 120 seconds, and the traffic sent increased from 0 to 98926080 bytes per second. The time for the None no Compression increased from 96 seconds to 120 seconds, and traffic sent increased from 0 to 106508160 bytes per second. In the case for TCP/IP header compression, the time increased from 96 seconds to 120 seconds, and the traffic sent increased from 0 to 121,777,920 bytes per second as the highest traffic sent on the network.

Figure 7. shows that as the graph progresses from 96 seconds to 120 seconds and the number of clients/users attempting to access the network at the same time increases, so does the Traffic Sent. When the graph is rising upwards over time and the number of clients/users increases, it indicates that the server is handling requests without any problem and that the network capacity is enough. If the graph becomes relatively flat over time as the number of clients/users rises, this indicates that bandwidth is getting

low or that Traffic Sent has piled up on the network server. If the graph displays a drop during steady state, it suggests that the network bandwidth is insufficient or that there is a problem with the network or may be the problem may be with the server and other devices on the network.

As a result of the above results, we can state that TCP/IP header compression had the highest transmitted traffic and better network connection to achieve high performance, leading to improved network quality of service. Furthermore, we observed that compression and decompression occur at each individual node in the TCP/IP header scheme. TCP/IP

header compression lowers packet transmission overhead and is advantageous when the application payload is smaller than the header. The scheme also minimises serialisation delay and uses less data, resulting in increased throughput and more available bandwidth. The fixed time required to clock a video/speech or any data frame onto the network interface is what is referred to as serialization delay. The Per-Virtual Circuit Compression technique had the least amount of transmitted traffic and the worst network connection, making it unable to achieve high network performance and quality of service.

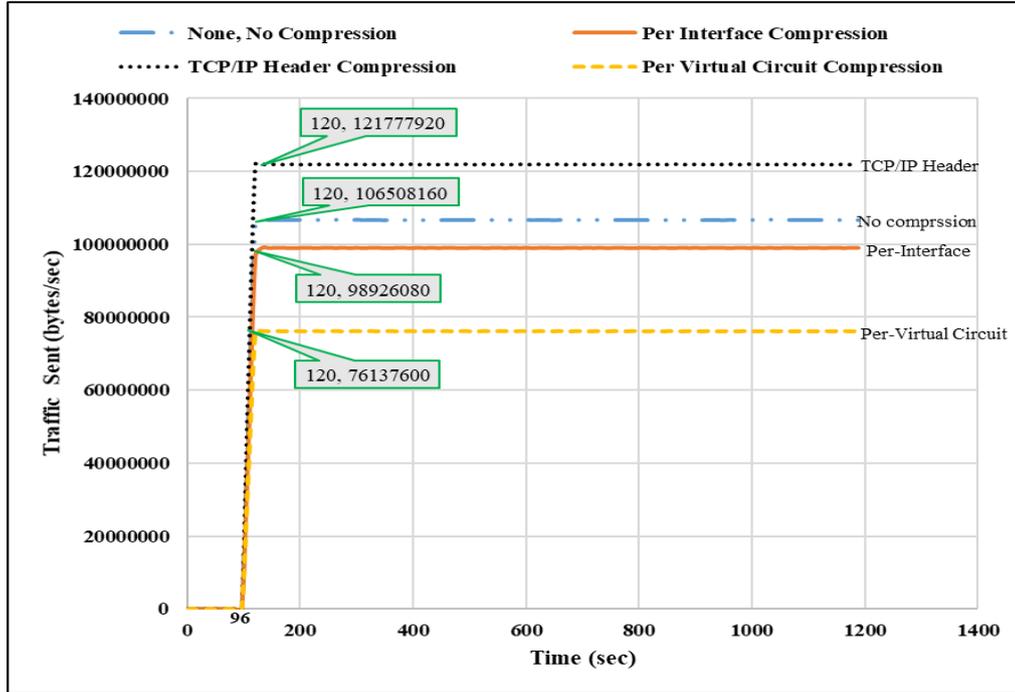


Figure 7. Traffic Sent (bytes/sec)

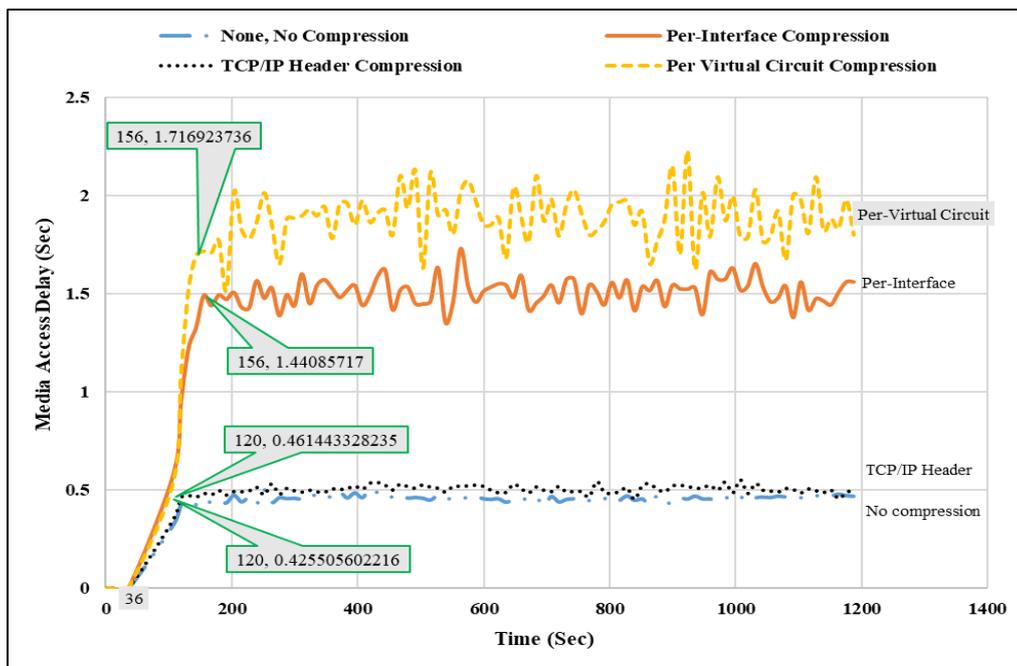


Figure 8. Media access Delay (sec)

5.4. Media Access Delay

Media access delay is the amount of time it takes for data to reach the media access control (MAC) layer and be successfully sent to the network's WLAN interface [21]. The entire amount of queuing time and data packet delay necessary for a signal head to access the medium is referred to as media access delay. The study compared the media access latency of the four (4) compression techniques evaluated. Figure 8. shows the OPNET simulation results for IP packet compression techniques. We have time in seconds on the x-axis (horizontal), and the labels on the y-bar (vertical) shows the media access latency in bits per second.

The results for the four (4) compression schemes are shown in Figure 8. It was observed that the time needed for the WLAN to establish the connection ranged from 0 to 36 seconds; after that, the graphs for all the compression algorithms went up.

According to the results in Figure 8., the time for None-no compression increased from 36 seconds to 120 seconds, and the Media Access Delay increased from 0 to 0.425505602216 second. The time for TCP/IP header compression increased from 36 seconds to 120 seconds, and the Media Access Delay increased from 0 to 0.461443328235 seconds. The time for Per-Interface compression increased from 36 seconds to 156 seconds, and Media Access Delay increased from 0 to 1.44085717 seconds. While the time for Per-virtual circuit compression, increased from 36 to 156 seconds, and Media Access Delay increased from 0 to 1.716923736 seconds.

Figure 8, shows that as the graph progresses from 96 seconds to 120/156 seconds and the number of clients/users attempting to access the network at the same time increases, so does the media access delay. When the media access delay graph rises with time and the number of clients/users increases, it indicates that the clients/users are competing for network media access. Other factors influencing the network include network congestion and capacity concerns caused by poor connections, particularly in WLANs that were not built to handle an increase in client/user numbers or activities.

When the graph becomes generally flat over time, from 36 seconds to 120/156 seconds, the media is sufficiently accessible. If the media access delay was initially caused by a bandwidth problem while several clients/users attempted to access the media at the same time, now there is sufficient bandwidth, we would get a media access delay graph flat similar to Figure 8, above. If there is a drop in the graph during a steady state, it indicates that the media access is adequate and the packets may receive the necessary access without any delays.

The None No Compression algorithm had the shortest Media Access Delay and the best medium access to the network, resulting in a best performance and a high Quality of Service. The Per Virtual Circuit compression scheme had the longest Media Access Delay and inadequate medium access to the network, making it difficult to achieve good network performance and Quality of Service.

6. Conclusions

The scope and objective of the study were to design and develop a model that would help in examining the Performance Evaluation and Compression of IP Packets in a Wireless Local Area Network (WLAN) using four (4) methods in the transmission of a high bandwidth application, such as video conferencing. The study undertook several performance tests for each of the four (4) compression schemes, and measured performance metrics that include; throughput, data dropped, traffic sent, and media access delay. The simulation results show that Per-Interface compression and TCP/IP header compression outperformed all of the other compression schemes depending on the circumstances and conditions set. In contrast, per-virtual circuit compression had lagged behind Per-Interface and TCP/IP header compression schemes. In terms of overall performance under the used simulation conditions in this particular study of performance of WLAN, Per-Interface Compression Scheme had the best performance and good Quality of Service. It is therefore, important to state that the work given in this study lays the groundwork, yet there is still potential for improvement. As a result, it is evident that similar or more study could be conducted to explore more simulation situations comparable to a larger network real-life scenario to determine if the same findings could be established.

REFERENCES

- [1] I. M. Omar and S. Prasad, "A review of data compression techniques," *International Journal of Advance Research in Science and Engineering*, vol. Volume Number 07, no. Special issue n. 07, pp. 1-8, 2018.
- [2] G. Goetz and L. D. Shapiro, "Data Compression and Database Performance," *ACM/IEEE-CS Symp. on Applied Computing*, p. 10, 1991.
- [3] A. S. Sethi and V. Y. Hnatyshin, "The Practical OPNET® User Guide for Computer Network Simulation," CRC Press Taylor & Francis Group, London New York, 2013.
- [4] K. Anubhuti, M. Saxena and B. Patil, "Voice Data Compression and Decompression," in *Cost Electronics & Communication Conference Proceedings*, 2011.
- [5] K. Sayood, *Introduction to Data Compression*, San Francisco, San Francisco: Elsevier Inc., 2006, pp. 1-5.
- [6] H. Cha, T. Shon, K. Kim and M. Hong, "Improving packet header compression with adaptive sliding window size," Cha, Hyejin; Shon, Taeshik; Kim, Kangseok; Hong, Manpyo (2015). [IEEE 2015 International Conference on Information Networking (ICOIN) - CambInternational Conference on Information Networking (ICOIN), 12-14 Jan. 2015.
- [7] T. Mate, D. Lucani, F. H. Fitzek and P. Ekler, "Unidirectional Robust Header Compression for Reliable Low Latency Mesh Networks," *Department of Engineering, Communications Systems, Aarhus University, Denmark*, pp. 1-6, 2019.

- [8] M. Yossi and R. Refua, "Delayed-Dictionary Compression for Packet Networks," Tel-Aviv University, Israel, pp. 3-5, 2013.
- [9] C. Feres, "Low Complexity Header Compression with Lower-Layer Awareness for Wireless Networks," International Conference on Communications, pp. 1-7, 2019.
- [10] C. S. Lubobya, M. E. Dlodlo and G. J. a. & K. L. Fergu-son, "SIMD implementation of integer DCT and hadamard transforms in H. 264/AVC encoder," Institute of Electrical and Electronics Engineers (IEEE), pp. 1-5, 2011.
- [11] L. A. Fitriya, "A Review of Data Compression Techniques," International Journal of Applied Engineering Research ISSN 0973-4562, Vols. Volume 12, Number 19, p. 8, 2017.
- [12] R. K. Kurmi and S. Gupta, "A Review of Lossless and Lossy Based Image Compression Techniques," International Journal of Engineering and Advanced Technology, Vols. Volume-6 Issue-6, August 2017, no. ISSN: 2249 – 895817, p. 5, August 2017.
- [13] Azeez N A and A. A. Lasisi, "Empirical and Statistical Evaluation of the Effectiveness of Four Lossless Data Compression Algorithms," Nigerian Journal of Technoloical Development, Vols. VOL. 13, NO. 2., pp. 1-2, 2 December 2016.
- [14] B. Souley, P. Das and S. and Tanko, "A comparative analysis of data compression techniques," International Journal of Applied Science and Engineering, 2014.
- [15] P. Koskela, M. Majanen and M. Valta, "Packet Header Compression for the Internet of Things," IFSA Publishing, Vols. Vol. 196., no. Issue 1., pp. 2-3, 2016.
- [16] I. Pu, "Data compression. Undergraduate Study in Computing and Related Programmes," University of London., 2004.
- [17] P. Kavitha, "A Survey on Lossless and Lossy Data," International Journal of Computer Science & Engineering Technology (IJCSSET), p. 5, 2016.
- [18] Z. Mehdi and J. S. Thompson., "Impact of Compression and Aggregation in Wireless Networks on Smart Meter Data," Digital Communications Dept., The University of Edinburgh, p. 2, 2016.
- [19] S. Sharma, "Analysis of Header Compression Techniques for Networks: A Review," Researchgate.net, pp. 1-9, 2013.
- [20] G. A. Mohamed, M. Zaghoul and M. F. a. Z. Mohammed, "Performance evaluation of multimedia streams over Wireless Computer Networks (WLANs)," International Journal of Advanced Science and Technology, pp. 1-14, 2009.
- [21] Madan, R. and Mangipudi, P.S., 2018, August. Predicting computer network traffic: a time series forecasting approach using DWT, ARIMA and RNN. In 2018 Eleventh International Conference on Contemporary Computing (IC3) (pp. 1-5). IEEE.