Pygmy: Innovative Technique to Increase the Utilization of IEEE 802.11n Networks while Running IPv6 Telephony

Mosleh M. Abualhaj[1] and Yousef H. Alraba'nah[2]
[1] Department of Networks and Information Security, Al-Ahliyya Amman University, Amman, Jordan
[2] Department of Software Engineering, Al-Ahliyya Amman University, Amman, Jordan
Email: {m.abualhaj; y.alrabanah}@ammanu.edu.jo

***Abstract*—The large packet header that IPv6 Telephony service uses has resulted in squandering a significant portion of bandwidth available on the IEEE 802.11n network. For example, the portion of the channel duration and capacity consumed by the packet header reaches 86% in IPv6 Telephony over IEEE 802.11n networks. IPv6 Telephony inventors are exerting extensive effort to handle this problem. This study contributes to the current effort by inventing a novel method called Pygmy. The heart of the Pygmy approach is built on using extra fields in the IPv6 Telephony packet's header to maintain packet payload (i.e., voice data), thus decreasing or zeroing the payload of the IPv6 Telephony packet. Pygmy is a method that drastically cuts down on the enhanced bandwidth of IPv6 Telephony IEEE 802.11n networks. Also, unlike similar methods, the Pygmy method works in any network situation and doesn't add any extra work to the network devices. Findings indicate that the Pygmy approach is capable of decreasing the amount of wasted bandwidth caused by the G.723.1 codec by 23%. Therefore, the bandwidth and channel time of IPv6 Telephony over IEEE 802.11n networks can be successfully reduced by utilizing the Pygmy approach.**

***Index Terms*—IP Telephony, IPv6, IEEE 802.11n, bandwidth utilization**

I. Introduction

Technology is constantly evolving and making new advances to improve upon previously developed technologies with innovative new services and features that make people's lives significantly simple, convenient, and satisfying. The IEEE 802.11n standard is an upgrade of the IEEE 802.11 wireless standard. Owing to improved bandwidth and download rates, the IEEE 802.11n standard offers the possibility of improving a wide variety of services and technologies. In particular, the Medium Access Control (MAC) layers that were part of the older 802.11 standards have been subjected to several improvements. Block acknowledgment and frame aggregation are primary advancements that have been made in the MAC layer [1]-[3]. Many types of technology, such as IPv6 Telephony, utilize the characteristics of IEEE 802.11n to improve the provided service.

IPv6 Telephony is an exciting new technology that offers subscribers a wide range of advantageous new features. With IPv6 Telephony, subscribers are able to freely communicate with anyone, anywhere in the world using their smartphones or any other smart devices of their choice. Accordingly, the number of people using mobile IP telephony is anticipated to exceed three billion. In 2015, the Internet was responsible for transporting over 156 petabytes of IP telephony traffic each and every month [4], [5]. The IPv6 Telephony technology requires data transmission with a minimum of latency. Therefore, IPv6 Telephony is one of the services that will experience a significant boost in quality as a result of working with the IEEE 802.11n standard. IPv6 Telephony is projected to completely integrate the IEEE 802.11n standard. Moreover, IPv6 Telephony will be able to maximize the fast speed of the IEEE 802.11n standard, which will improve call quality to the point where users are satisfied. IPv6 Telephony will also be able to take advantage of the high bandwidth offered by the IEEE 802.11n standard, which will significantly increase the call capacity [2], [6].

In spite of the potential advantages, IPv6 Telephony is currently squandering a sizeable portion of bandwidth made available by IEEE 802.11n for no good benefit. This issue arises because the header of the IPv6 Telephony packet is relatively lengthy compared with the payload [7], [8]. One example is the G.729 codec, which generates digital speech data (packet payload) consisting of 10 bytes. In certain cases, the VoIP packet payload consists of more than one voice frame. Assuming the typical case, in which each packet payload consists of only one voice frame (10 bytes of G.729 codec), up to 86% of the bandwidth designated for an IPv6 Telephony call is lost owing to the header of the IPv6 Telephony packet. Fig. 1 shows the IPv6 Telephony packet format and allocated capacity for various voice codecs.

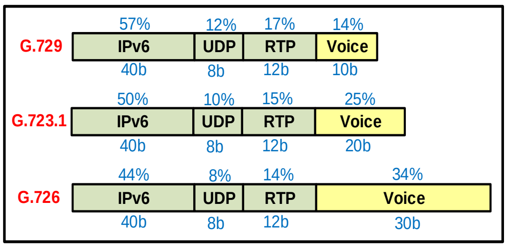


Fig. 1. IPv6 telephony packet with different codecs.

Manuscript received Sept. 26, 2022; revised October 30, 2022; accepted November 3, 2022.

Corresponding author: Mosleh M. Abualhaj (email: m.abualhaj@ ammanu.edu.jo).

TABLE I: CONSUMED BANDWIDTH WITH DIFFERENT VOICE CODES

Codec	Packet Length	Voice Data Length	Consumed bandwidth
G.729	70	10	86%
G.723.1	80	20	75%
G.726	90	30	66.7

Table I shows the consumed bandwidth by packet header with different voice codes. Evidently, a significant quantity of unused bandwidth is the result of all the codecs [9]-[11]. In theory, this method also uses the same proportion of the total transmission time used by IEEE 802.11n networks.

The IPv6 protocol is becoming widely used every year, and this trend is expected to continue. According to [12], the rate of worldwide IPv6 usage rose from 2.8% in 2014 to 23.48% as of 2018, thereby representing a significant growth. The implementation of IPv6 in networks globally has been increasing at a rapid pace, reaching an annualized average of over 51% [13]. Eventually, IP Telephony and IPv6 will be integrated across IEEE 802.11n networks. Consequently, the previously mentioned issue of squandering the resources of IEEE 802.11n networks must be considered and thoroughly researched. Developers of IPv6 Telephony have come up with a variety of methods to address the large header of IPv6 Telephony. These methods include coalescing the IPv6 Telephony packet into a single header, compressing the header of the packet, replacing the protocols in the IPv6 Telephony header with new protocols specific to IPv6 Telephony, and using the superfluous fields in the header to carry the voice data of the packet [14]-[16].

The utilization of surplus header fields is the final proposed option, and one of its primary benefits is that numerous of the suggested solutions falling under it are compatible with current standards and devices. Consequently, they are suitable for use in the existing environment and do not require significant adjustments. However, none of these techniques have been optimized for use with IPv6 telephony over IEEE 802.11n networks. This paper focuses on increasing the usage of bandwidth in IEEE 802.11n networks while simultaneously operating IPv6 Telephony, and proposes a new approach to achieve this goal. The digital voice data contained within the packet can be transported owing to the proposed approach, thereby maximizing the unused fields of the IPv6 protocol. Accordingly, payload size can be reduced using the proposed method, or it can be eliminated entirely. Given this scenario, bandwidth utilization of networks being used, such as IEEE 802.11n networks, is improved. A decrease in the extent of airtime is required for the transmission of an IPv6 Telephony packet over IEEE 802.11n networks. This state has an indirect impact on the quality of the voice link because it results in a reduction in the amount of packet loss and delay.

The remainder of this article is structured as follows. Section II presents the primary methods to the IP telephony bandwidth issue. Section III introduces the suggested approach, along with its main concept of resolving the bandwidth problem by exploiting the UDP/IPv6 protocols fields to convey the packet content. Section IV covers the evaluation of the suggested approach compared with other similar approaches. Lastly, Section V outlines the results of this investigation.

II. RELATED RESEARCH

The limitation imposed by bandwidth has, for some time now, slowed down the propagation of IPv6 Telephony. Different ideas have been proposed to solve this problem and accelerate the transition to IPv6 telephony. The concept of gathering all packets that go along the same route and multiplexing them in a single header, which is known as the packet/frame multiplexing method, is one of the first proposed solutions. A large variety of other approaches have been developed within the confines of this packet/frame multiplexing method. Some of the multiplexing approaches have been developed to multiplex the packets at the network layer in a single header, while others have been developed to multiplex the frames at Datalink layer in a single header. Nevertheless, these methods are included in the “frame aggregation” category. Better usage of available bandwidth can be achieved by doing the aggregate at layer 2. Saving as much bandwidth as possible requires optimizing the amount of aggregated packets, which is determined by the characteristics of the environment [3], [17]. Nomura Y. et al. devised a technique for extremely efficient packet aggregation that considers multi-rate transmissions and IP Telephony service. In case the number of IP Telephony packets in the buffer likely exceeds the permitted delay constraint, the suggested solution includes adding IP Telephony priority transmission, which carries out the preferable selection of IP Telephony mobile terminals. This strategy is done in case the buffer contains more IP Telephony packets than what is permitted. In configuring the length of the wireless frame, the access point uniformizes transmission times across all spatial streams by adjusting the number of aggregated packets while considering the MCS level of each packet. The goal is to reduce the amount of time spent in the space channel during the MU–MIMO transmissions. Findings of computer simulations indicate that the proposed method raises system throughput, space channel time ratio during the MU–MIMO transmissions, and maximum delay time for IP Telephony packets in WLANs supporting multi-rate transmissions and IP Telephony service. These improvements are the result of a reduction in the amount of time spent waiting for packets to be delivered. As a direct result, the recommended technique for packet aggregation performs admirably for the downlink MU–MIMO channels in contemporary WLAN configurations [18].

A technique called header compression can be utilized to reduce the size of a packet’s header. This technique involves removing individual fields from the header to reduce its overall size. Given that they are following specific patterns; the other end of the call is able to derive their values easily. When IPv4 is used, the header is reduced to as few as 2 bytes, thereby helping save a significant amount of bandwidth. Ruchi Garg and Sanjay Sharma proposed a compression method called modified and improved IPv6 header compression (MIHC), which is intended for low power wireless personal area network over IPv6 (6LoWPAN). The MIHC compression algorithm relies on the correlation of headers that are already present in the packets being delivered from the source to destination nodes. MIHC has been implemented in Contiki 3.0 by the authors, and simulations have been run using the Cooja simulator. The findings indicate that MIHC performs better than the other techniques with throughputs that are 20% and 76% higher, delays that are 13% and 38% lower, round trip times that are 12% and 37% lower, and packet sizes that are 13% and 39% smaller, respectively. When applied to the measured values, T-test reveals a statistically significant difference ($p < 0.5$) between the no header compression case (i.e., NO COMP) and the suggested approach (i.e., MIHC) [19].

The creation of additional IP Telephony-specific protocols, such as the Inter-Asterisk Exchange (IAX) protocol, has been proposed by several engineers working on IP Telephony software. IAX is an underlying communications protocol included in the Asterisk private branch exchange (PBX) software. In addition, several additional softswitches, PBX systems, and softphones offer support for IAX. It is used in transferring IP Telephony sessions between servers and terminal devices. IAX is an IP Telephony protocol that, although being capable of handling any type of streaming media, including video, primarily focuses on IP voice conversations. IAX protocol utilizes a single protocol for simultaneously controlling and transmitting media. Its open design enables the inclusion of new payload types, which are required to offer additional services. IAX replaces the 12-byte RTP protocol with its own 4-byte mini-header to convey digital speech data. This outcome results in a significant reduction in bandwidth use. In addition, it uses compact encoding, which reduces the amount of bandwidth used and is considered an excellent choice for Internet telephony services [15].

One of the potential solutions that has been considered recently is to use the redundant fields within the header to transport the packet's speech data. The premise that different types of application data are transferred using the IP Telephony packet header protocol is the foundation for this proposed approach. Consequently, certain fields contained within these protocols are required by specific programs, but other applications do not require them. The latter adds unnecessary strain to the bandwidth while serving no useful purpose [16], [20]. These fields have been called unnecessary fields. This final approach to solving the bandwidth issue uses the unnecessary fields to transport speech data included within the packet. Hence, payload can be reduced or eliminated entirely, thereby saving bandwidth. Short voice frame (SVF), which was proposed by Abualhaj et al., is one of the most effective methods to implement the preceding solution. On the basis of a few key presumptions, the SVF technique is able to effectively use the seven empty fields contained within the packet header. Each IPv4 Telephony packet, which is usually between 50 and 70 bytes in length, can be reduced by 17 bytes owing to the aforementioned parameters. Findings demonstrate that the amount of saved bandwidth can approach around 29% in some circumstances [20]. To use excess fields, another approach called zero size payload (ZSP) has been devised. Compared with the SVF approach, the ZSP method operates under a distinct set of presumptions. Accordingly, it is able to use the 10 unnecessary fields while only utilizing 19 bytes. Findings indicate that the amount of bandwidth that can be conserved may approach 32% in some circumstances [16]. Similar to SVF and ZSP, M. Kolhar has proposed a method called Zeroize that utilizes the redundant header fields in the packet header [5]. However, the Zeroize method is intended for IPv6 telephony, not IPv4. The Zeroize utilizes only one field in the packet header, and that is the source IPv6 address. Though it is one field, the source IPv6 address is 16 bytes and succeeded in saving up to 20% of the 5G network bandwidth.

In running IP Telephony, developers have made significant attempts to reduce bandwidth use. Owing to the huge length of the IPv6 header, the issue of bandwidth consumption becomes markedly difficult to solve when merging IP telephony and the IPv6 protocol. However, bandwidth issues affecting IP telephony and IPv6 protocol have not been addressed adequately, particularly in IEEE 802.11n wireless networks. This research develops a novel approach to solve this issue by using the unnecessary fields contained inside the IPv6 Telephony packet header. Digital voice data of the IPv6 Telephony packet is carried by the source IPv6 address, as well as the source port number, according to the recommended approach. Pygmy is the strategy designed to reduce the size of the IPv6 Telephony packet payload as much as possible or even bring it down to nil. The Pygmy approach increases the quality of communications while also increasing the amount of bandwidth that may be utilized.

III. Pygmy Technique Design

This section breaks down and explains the functioning algorithm of the Pygmy method. When IP Telephony over the IPv6 protocol is used, the primary objective of the Pygmy approach is to boost the capacity of the IEEE 802.11n network links as much as possible. Pygmy is a mechanism used on a calling client (e.g., MicroSIP) and on wireless equipment that acts as an intermediate (e.g., wireless router). Either one of them can have their channel capacity increased by employing the Pygmy approach with the same degree of effectiveness. For the purposes of this study, the assumption is that the Pygmy approach is applicable to the caller client. Moreover, the Pygmy approach can be utilized to implement the topology depicted in Fig. 2.

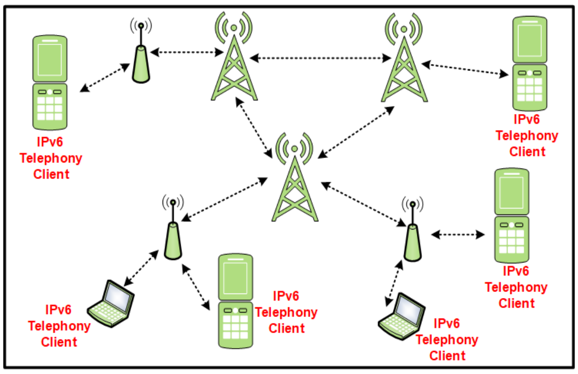


Fig. 2. Pygmy method topology.

A. Core Idea

Call capacity can be increased with the help of the Pygmy approach by utilizing the unused fields in the IPv6 Telephony packet header. The unnecessary fields are utilized to preserve digital speech data within the packet. Findings of the research [16] and [20] indicate that many fields of the IPv4 Telephony packet are considered unnecessary for IPv4 Telephony applications. This conclusion was reached on the basis of specific rules.

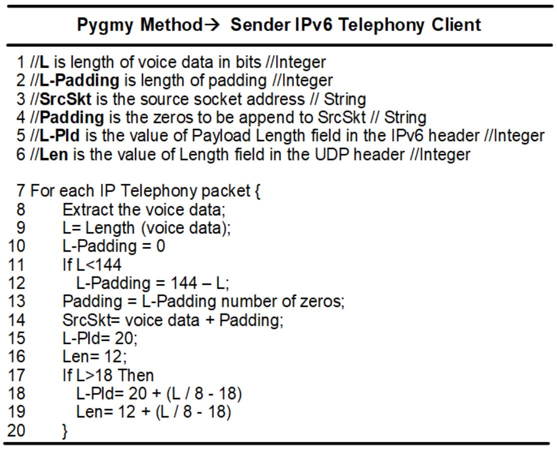
The source IPv4 address and source port number are two of the redundant fields found across the RTP, UDP, and IPv4 protocols. By contrast, the Pygmy technique is designed for use with IPv6 Telephony. Consequently, only the source IPv6 address and source port number (source socket) are considered redundant fields and used to maintain the packet's digital voice data. Details are presented in the following sections.

B. Source Socket

In networks operating based on packets, the source socket's job is to determine the original sender of data. This source socket is utilized by the receiver so that a response can be transmitted back to the sender [20], [21]. Signaling and media transmission are the two processes involved at the beginning of a call initiated using IPv6 Telephony applications. During the signaling step, a particular protocol (e.g., H323 protocol) is applied to initiate the call and decide the call settings, such as the sockets of the callee and caller. This step begins the process of placing the call. At this point, each of the calls is aware of the socket located at the other end. During the media transfer step, another protocol (e.g., RTP protocol) is utilized to transmit speech data [22], [23]. Call settings from the first step, including the source socket, are used by RTP, UDP, and IPv6 to transport voice data to the location intended to receive it. The important takeaway is that each speech data packet contains the source socket (media packets). Meanwhile, the IPv6 Telephony media transmission session is not a request-and-response exchange, and the receiver client is not responsible for independently sending a response to the caller. In the event that the callee responds to the caller, the former will use the source socket, which has been established from the beginning of the process [16], [20], [21]. Other devices comprising the network, such as switches, routers, and firewalls, are unaffected by this issue. The switch typically works at the Datalink layer, while the IPv6 protocol works at the Network layer protocol. The router uses the destination IPv6 address to achieve its primary purpose of routing. As for the firewall systems, they are only denying an IP Telephony call in the call initiation process using SIP or H323 IP Telephony signaling protocols. It will not deny the call during transmitting speech media data using RTP/UDP/IP protocols. In addition, not a single one of the other protocols will be modified as a result of this suggested Pygmy method. Consequently, the source socket included in the header of the packets is redundant and might be reassigned to serve another goal, such as transporting the actual speech data. Given this scenario, the amount of bandwidth utilized by each call is decreased, resulting in an increase in the channel's call capacity.

C. Conveying the Voice Data

Voice data are transferred across the packet using the source socket by employing the Pygmy technique. Data pertaining to the sender's voice are extracted from the packet by the sender's client. Thereafter, data pertaining to the voice are entered into the source socket area. If voice data are relatively modest in size, the remaining bits of the socket will be padded. This operation modifies the length of the packet payload, thereby altering the payload length field in the IPv6 header and length field in the UDP header [21]. Hence, values of these fields should be adjusted so they accurately reflect the new size. The technique used by the IPv6 telephony client on the sender side is explained clearly in the pseudocode shown in Fig. 3. Variables comprising the pseudocode can be identified in Lines 1 to 6. Speech data are placed in the source socket fields using Lines 8 to 14. Lines 8 to 14 are used to put zeros in the residual bits of the socket fields. Lines 15 to 19 are used to update the value of the payload length field in the IPv6 header. Voice data are extracted from the source socket by the IPv6 telephony client and placed thereafter in the playout buffer at the callee. Only the actual speech data (without the padding) should be retrieved.



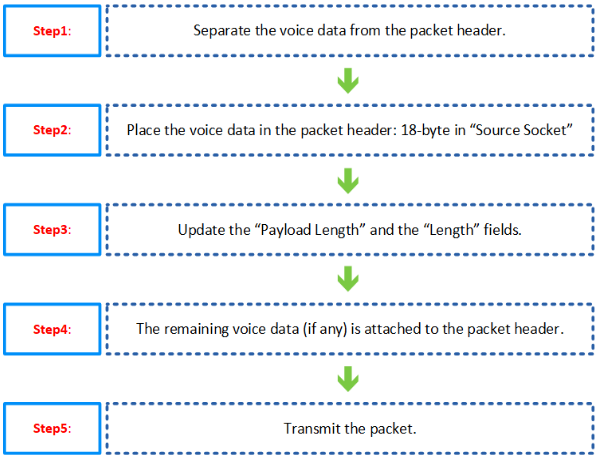
Pygmy Method→ Sender IPv6 Telephony Client

```
 1 //L is length of voice data in bits //Integer
 2 //L-Padding is length of padding //Integer
 3 //SrcSkt is the source socket address // String
 4 //Padding is the zeros to be append to SrcSkt // String
 5 //L-Pld is the value of Payload Length field in the IPv6 header //Integer
 6 //Len is the value of Length field in the UDP header //Integer

 7 For each IP Telephony packet {
 8       Extract the voice data;
 9       L= Length (voice data);
10       L-Padding = 0
11       If L<144
12          L-Padding = 144 – L;
13       Padding = L-Padding number of zeros;
14       SrcSkt= voice data + Padding;
15       L-Pld= 20;
16       Len= 12;
17       If L>18 Then
18          L-Pld= 20 + (L / 8 - 18)
19          Len= 12 + (L / 8 - 18)
20       }
```

Fig. 3. Pygmy technique algorithm: Caller side.

To select the speech data length, the IPv6 telephony client should check the codec currently utilized. During the signaling stage of an IPv6 Telephony call, the codec that will be utilized will be supplied. Fig. 4 and Fig. 5 substantially explain the process implemented by the IPv6 telephony client on the sender and recipient sides of the conversation.



Step1: Separate the voice data from the packet header.

Step2: Place the voice data in the packet header: 18-byte in "Source Socket"

Step3: Update the "Payload Length" and the "Length" fields.

Step4: The remaining voice data (if any) is attached to the packet header.

Step5: Transmit the packet.

Fig. 4. Pygmy technique process: Caller side.

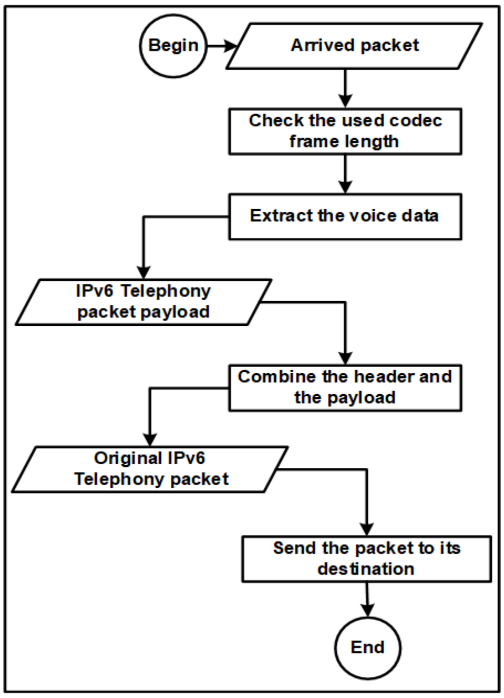

Begin
Arrived packet
Check the used codec frame length
Extract the voice data
IPv6 Telephony packet payload
Combine the header and the payload
Original IPv6 Telephony packet
Send the packet to its destination
End

Fig. 5. Pygmy technique process: Callee side.

IV. Pygmy Method Performance Analysis

This section provides an analysis of the Pygmy approach. The Pygmy approach is compared with the Zeroize method [5], the ZSP method [16], the SVF method [20], and the conventional RTP (C-RTP) method. The Zeroize, SVF and ZSP methods are chosen because of their similarities to the proposed Pygmy approach. In IP Telephony applications, the Zeroize, SVF and ZSP techniques aim to achieve the similar fundamental goal as the Pygmy technique, which is to reduce the amount of bandwidth consumed by the network. Conveying the speech data in the unnecessary fields of the packet header is the core notion behind the Zeroize, SVF and ZSP approaches, in which the three methods aim to reduce bandwidth use. However, the SVF and ZSP methods are only compatible with IPv4 networks and are designed to be used for point-to-point connections. On the other hand, while, Pygmy and Zeroize are intended for IPv6 networks. Comparisons are made between the proposed Pygmy approach and other techniques with regard to link capacity and saved bandwidth. For comparison, G.726, G.729, and G.723.1 codecs will be assumed.

The Network simulation 3 (NS3) was utilized to act out the suggested Pygmy technique. After simulating the Pygmy technique, a network topology utilizing NS3 was created to assess the Pygmy technique whilst comparing it with the SVF technique, ZSP technique, and Zeroize technique. The simulation topology includes two elements of the Pygmy technique, which are positioned at the sender and receiver sides, as discussed in Sections III. Each element uses a buffer with a maximum size of 5. The link between the two elements is simulated as a first-in-first-out buffer. A CBR traffic producer is attached to each ends.

A. Channel Capacity

This subsection presents for consideration the channel capacity of the proposed Pygmy technique. The capacity of each link with bandwidth between 100 kbps and 1000 kbps has been examined. The fact that the packets are being lost indicates that the link is full. Consequently, the channel capacity of a link is equal to the number of calls that can occur prior to the loss of packets, and it is measured in accordance with this equation. Channel capacity of the Pygmy approach is examined in Fig. 6, Fig. 7, and Fig. 8 in contrast to the C-RTP, Zeroize, SVF, and ZSP techniques with the G.723.1, G.726, and G.729 codecs, respectively. Compared with the C-RTP and Zeroize approachs, channel capacity achieved with the Pygmy method results in a greater number of successful calls. Difference in channel capacity increases with the amount of bandwidth currently available. This issue has emerged as a result of payload elimination by using an unnecessary field (source socket) of the header to convey the payload. Nevertheless, channel capacity achieved with the Pygmy technique is inferior (resulting in considerably few calls) compared with that achieved with the SVF and ZSP methods. The reason is that the IPv6 protocol header requires more bandwidth than that of the IPv4 protocol, even though the former is twice as large as the latter. Sizes of the unnecessary fields produced by the SVF and ZSP methods are significantly larger than those produced by the Pygmy approach.

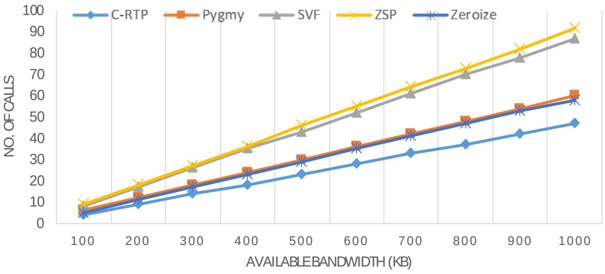


Fig. 6. Channel capacity (G.723.1 codec).

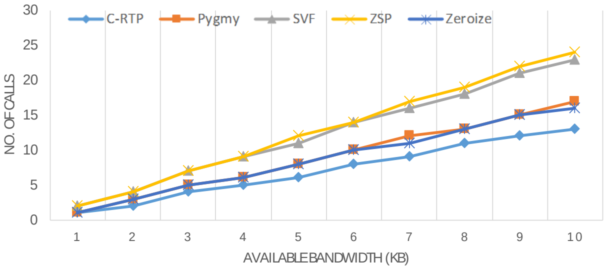


Fig. 7. Channel capacity (G.726 codec).

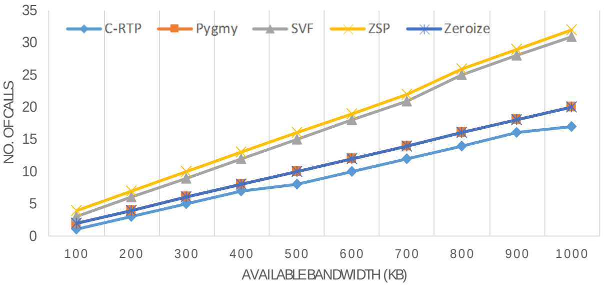


Fig. 8. Channel capacity (G.729 codec).

B. Saved Bandwidth

This subsection discusses the allocated bandwidth ratio that the Pygmy approach uses. Between 10 and 100 calls are used in the analysis of the allotted bandwidth ratio. This study evaluates and compares the Pygmy, Zeroize, SVF, and ZSP methods with the C-RTP methods. The Pygmy, Zeroize, SVF, and ZSP approaches result in lower overall bandwidth use compared with C-RTP. As shown in Fig. 9, compared with the C-RTP approach, when uses the G.723.1 codec, the Pygmy method results in approximately 23% reduction in the amount of bandwidth consumed by the allocated data transmission. However, bandwidth savings achieved with the Pygmy method are significantly lower than those achieved with the SVF and ZSP approaches. The reason is that the IPv6 protocol header requires more bandwidth than that of the IPv4 protocol, even though the former is twice as large as the latter. Sizes of the unnecessary fields produced by the SVF and ZSP methods are significantly larger than those produced by the Pygmy approach.

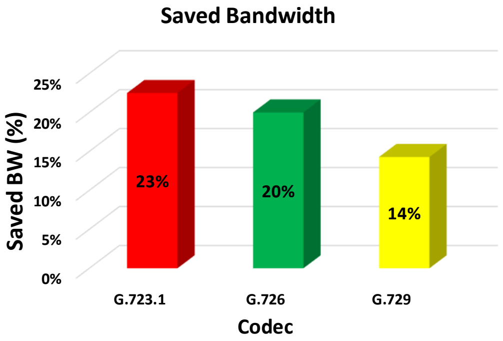


Fig. 9. Bandwidth reduction.

The Pygmy method comes out on the losing end with regard to channel capacity and saved bandwidth compared with the SVF and ZSP approaches. However, the Pygmy approach is compatible with IPv6 networks compared with the SVF and ZSP methods, which are only compatible with IPv4 networks. The SVF and ZSP techniques convey speech data using numerous fields, some of which are considered redundant information. Given this limitation, they are only usable in certain circumstances, which are precisely the types of circumstance they are intended for. However, the Pygmy technique uses nothing more than source socket fields as redundant fields to transport voice data. Given this criterion, the Pygmy approach becomes a considerably general method that may be applied to any situation. Compared with the Pygmy approach, the SVF and ZSP methods stresses on the network owing to numerous reasons. These approaches were developed with the goal of functioning at gateway routers. Consequently, resources of these routers are utilized to their full potential, particularly owing to the large volume of calls. Pygmy is a method intended to work on the client side rather than burdening routers with its processing. Moreover, Pygmy simply uses the source socket to transport voice data and does not require the values of socket fields to be reset to their initial states. By contrast, the SVF and ZSP techniques use multiple fields to convey speech data. Accordingly, there are now more operations and calculations that must be performed, such as finding the new value of the length field for the UDP protocol and changing the value of the Internet header length field for the IP protocol. These processes and calculations also require that any unnecessary fields at the receiver gateway must have their values reset. The reason is for speech data to possibly be carried via various fields used by the SVF and ZSP approaches. Accordingly, performing these actions on each and every call packet uses up the resources of routers, which is particularly evident when there are too many calls. The SVF and ZSP techniques place additional strain on the network compared with the Pygmy approach.

V. Conclusion

IPv6 Telephony services offer a multitude of capabilities, which contribute to their considerably broad applicability in the telecommunications field. Accordingly, escaping the inevitable merger of IPv6 Telephony and IEEE 802.11n networks is impossible. Given this integration, up to 86% of bandwidth and airtime that an IEEE 802.11n network has available are wasted. The Pygmy technique is this study's contribution to the current effort to find a solution to the preceding problem. The proposed solution is successful in maximizing the source socket to convey the packet's digital speech data, enabling the IPv6 Telephony packet payload to be minimized or eliminated entirely. Voice data are extracted from the packet by the sender client and placed thereafter in the source socket fields. Moreover, voice data are retrieved from the source socket fields by the IPv6 Telephony client at the receiver and appended thereafter to the end of the packet in the same manner as any other normal payload.

The C-RTP approach, Zeroize method, SVF method, and ZSP method are compared with the Pygmy technique by utilizing three distinct codecs (i.e., G.729, G.726, and G.723.1). The Pygmy technique results in a 23% reduction in the amount of bandwidth lost in contrast with the C-RTP approach using the G.723.1 codec. In reducing the amount of unused bandwidth and airtime that IEEE 802.11n networks experience when running IPv6 Telephony, the method that has been described is an intriguing potential solution. However, bandwidth savings achieved with the Pygmy method are significantly lower than those achieved with the SVF and ZSP approaches, because the IPv6 protocol header is twice as large as the IPv4 protocol header. In addition, the SVF and ZSP techniques are only usable in certain situations, while the Pygmy technique may be applied to any situation. Finally, the SVF and ZSP techniques impose an additional burden on the network compared with the Pygmy technique.

In the future, the suggested Pygmy technique will be integrated with other approaches, which handle the bandwidth utilization problem, such as the packet multiplexing approach. Also, the Pygmy technique will be tested with different network settings and parameters to make sure that the results are correct.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Both Mosleh M. Abualhaj and Yousef H. Alraba'nah contributed to the design and implementation of the research, to the analysis of the results, and to the writing of the manuscript.

References

[1] M. H. D. Z. Mahfouz, A. B. J. Kokkeler, A. Meijerink, and A. A. Glazunov, "Impact of ultra-narrowband interference on Wi-Fi links: An experimental study," *IEEE Trans. on Wireless Communications*, vol. 20, no. 5, pp. 3016-3030, 2021.

[2] Z. Farej and M. Jasim, "Investigation on the performance of the ieee802. 11n based wireless networks for multimedia services," in *Proc. of 2nd Int. Conf. for Engineering, Technology and Sciences of Al-Kitab*, 2018, pp. 48-53.

[3] S. Seytnazarov, J. G. Choi, and Y. T. Kim, "Enhanced mathematical modeling of aggregation-enabled WLANs with compressed blockACK," *Journal of IEEE Trans. on Mobile Computing*, vol. 18, no. 6, pp. 1260-1273, 2018.

[4] M. M. Abualhaj, M. Kolhar, K. Qaddoum, and A. A. Abu-Shareha, "Multiplexing VoIP packets over wireless mesh networks: A survey," *KSII Trans. on Internet and Information System*s, vol. 10, no. 8, pp. 3728-3752, 2016.

[5] M. Kolhar, "Zeroize: A new method to improve the utilization of 5G networks when running VoIP over IPv6," *Applied System Innovation*, vol. 4, no. 4, 2021.

[6] K. Hassine and M. Frikha, "A VoIP focused frame aggregation in wireless local area networks: Features and performance characteristics," in *Proc. of 13th Int. Wireless Communications and Mobile Computing Conf.*, 2017, pp. 1375-1382.

[7] J. Karjee, S. Khatter, D. Sarkar, H. L. C. Tammineedi, and A. K. R. Chavva, "5G-NR cross layer rate adaptation for VoIP and foreground/background applications in UE," in *Proc. of IEEE 3rd 5G World Forum (5GWF)*, 2020, pp. 80-85.

[8] V. Geetha, A. Bhat, and S. Thanmayee, "New bit pattern based IPv6 address compression techniques for 6LoWPAN header compression," *IEEE Access*, vol. 10, pp. 80055-80070, 2022.

[9] S. Narayan, M. Gordon, C. Branks, and L. Fan, "VoIP network performance evaluation of operating systems with IPv4 and IPv6 network implementations," in *Proc. of 3rd Int. Conf. on Computer Science and Information Technology*, 2010, pp. 669-673.

[10] F. Zampognaro, R. Aurigemma, and W. Munarini, "VoIP CODEC assessment and performance evaluation in satellite-based scenarios," in *Proc. of 2021 4th Int. Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, 2021.

[11] S. Dhar and S. Chatterjee, "A study of VoIP codecs performance over IEEE 802.11n," in *Proc. of Devices for Integrated Circuit* (DevIC), 2017, pp. 121-124.

[12] Internet Society, State of IPv6 Deployment. (2018). [Online]. Available: https://www.internetsociety.org/wp-content/uploads/2018/06/2018-ISOC-Report-IPv6-Deployment.pdf

[13] C. Sun. PERSPECTIVE: Future of the Internet Depends on Global Adoption of IPv6. (2018). [Online]. Available: https://www.hstoday.us/subject-matter-areas/cybersecurity/future-of-the-internet-depends-on-global-adoption-of-ipv6/

[14] C. Vulkan, A. Rakos, Z. Vincze, and A. Drozdy, "Reducing overhead on voice traffic," U.S. Patent No. 8,824,304. 2 Sept., 2014

[15] P. Kadam, M. Kulkarni and V. Gaikwad, "Bandwidth management for VoIP calling through asterisk," in *Proc. of 2nd Global Conf. for Advancement in Technology*, 2021.

[16] M. M. Abualhaj, Q. Y. Shambour, A. H. Hussein, and Q. M. Kharma, "Down to zero size of voip packet payload," *CMC-Computers Materials & Con-Tinua*, vol. 68, no. 1 pp. 1271-1283, 2021.

[17] R. F. Aji, S. Yazidb, H. Suhartantoc, and I. Budid, "Throughput fairness using packet aggregation on 802.11g networks," *Journal of Advanced Research in Computing and Applications*, vol. 2, no. 1, pp. 19-24, 2016.

[18] Y. Nomura, K. Mori, K. Naito, and H. Kobayashi, "High efficient packet aggregation scheme for multi-rate and IP telephony packet transmissions in next generation MU-MIMO WLANs," in *Proc. of Int. Conf. on Advanced Technologies for Communications,* 2014, pp. 517-521.

[19] R. Garg and S. Sharma, "Modified and improved IPv6 header compression (MIHC) scheme for 6LoWPAN," *Wireless Personal Communications*, vol. 103, no. 3, pp. 2019-2033, 2018.

[20] M. M. Abualhaj, M. M. Al-Tahrawi, and M. Al-Zyoud, "Contracting VoIP packet payload down to zero," *Cybernetics and Information Technologies*, vol. 21, no. 1, pp. 137-150, 2021.

[21] *The Art of Service, IPv6 A Complete Guide*, 1st ed. The Art of Service, 2021.

[22] A. Khudher, "Sip aspects of IPv6 transitions: Current issues and future directions," *Journal of Engineering Science and Technology*, vol. 14, no. 1, pp. 448-463, 2019.

[23] A. H. Hussein, M. Al-Zyoud, K. Nairoukh, and S. N. Al-Khatib, "Enhancing VoIP BW utilization over ITTP protocol," *Int. Journal of Advanced Computer Science and Applications*, vol. 11, no. 10, pp. 505-510, 2020.

Mosleh M. Abualhaj is a senior lecturer in Cybersecurity and Networks Dept., faculty of Information Technology at Al-Ahliyya Amman University. He received his first degree in computer science from Philadelphia University, Jordan, in July 2004, master degree in computer information system from the Arab Academy for Banking and Financial Sciences, Jordan in July 2007, and doctorate degree in multimedia networks protocols from Universiti Sains Malaysia in 2011. His research area of interest includes IDPS systems, machine learning, VoIP, and congestion control. Apart from research, Prof. Abu-Alhaj also does consultancy services in the above research areas and directs the Cisco academy team at Al-Ahliyya Amman University.

Yousef H. Alraba'nah was born in Jordan 1990. He received his B.Sc. degree in software engineering from Zarqa university (Jordan) in June 2012. In February 2013, Yousef obtained fully-funded scholarship from Zarqa University to earn his M.Sc. degree in computer science. He graduted in June 2015 with excellent degree. He currently works as a lecturer in the faculty of information technology at Al-Ahliyya Amman University, Jordan. His main research interests include: distributed systems, networks security, and machine learning.