



Down to Zero Size of VoIP Packet Payload

Mosleh M. Abualhaj^{*}, Qusai Y. Shambour, Abdelrahman H. Hussein and Qasem M. Kharma

Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan *Corresponding Author: Mosleh M. Abualhaj. Email: m.abualhaj@ammanu.edu.jo Received: 28 October 2020; Accepted: 29 November 2020

Abstract: Voice over Internet Protocol (VoIP) is widely used by companies, schools, universities, and other institutions. However, VoIP faces many issues that slow down its propagation. An important issue is poor utilization of the VoIP service network bandwidth, which results from the large header of the VoIP packet. The objective of this study is to handle this poor utilization of the network bandwidth. Therefore, this study proposes a novel method to address this large header overhead problem. The proposed method is called zero size payload (ZSP), which aims to reemploy and use the header information (fields) of the VoIP packet that is dispensable to the VoIP service, particularly the unicast IP voice calls. In general, these fields are used to carry the VoIP packet payload. Therefore, the size of the proposed ZSP method showed a considerable improvement in the bandwidth utilization of the VoIP service. For example, the saved bandwidth in the tested scenario with the G.723.1, G.729, and LPC codecs reached 32%, 28%, and 26% respectively.

Keywords: Voice over internet protocol; VoIP protocols; bandwidth utilization

1 Introduction

The explosive growth of the Internet has generated a considerable number of new services such as voice over Internet protocol (VoIP) [1,2]. VoIP adoption by companies, schools, universities, and other institutions is increasing exponentially due to various advantages such as affordable cost [3,4]. However, bandwidth utilization and quality of service are two main issues that create difficulties for VoIP service adoption [5,6]. The concern of the present study is the bandwidth utilization of the VoIP service. The main cause of poor bandwidth utilization of the VoIP service is a large overhead cost attached to the small VoIP packet payload [1,7]. On one hand, the typical VoIP packet payload size is 10–30 bytes based on the used codec. A codec produces a fixed-size voice frame (VoIP packet payload). Tab. 1 shows some of the common VoIP codecs [8,9]. On one the other hand, the typical VoIP packet header consists of 12 bytes of real-time transfer protocol (RTP), 8 bytes of user diagram protocol (UDP), and 20 bytes of IP (a total of 40 bytes RTP/UDP/IP) [10,11]. Attaching 40 bytes RTP/UDP/IP header to a small VoIP packet payload (10–30 bytes) induces high packet header overhead, thereby resulting in wasted



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

network bandwidth [10–12]. A packet header overhead is calculated by dividing the packet header size by the total packet size (header size plus payload size). Tab. 2 shows the header overhead ratio of RTP/UDP/IP protocols with different VoIP packet payload sizes.

As mentioned, the main reason for the inefficient bandwidth utilization of VoIP service is header overhead, which results from attaching a 40-byte RTP/UDP/IP header to a small VoIP packet payload [10–12]. The existing UDP and IP are general protocols used to carry all types of data [13]. In addition, the RTP protocol is used with various applications to carry the real-time multimedia (audio and video) data over IP networks, including VoIP, video conferencing, webcasting, and TV distribution [14–16]. Accordingly, a strong argument is that RTP/UDP/IP protocols provide many features and header information (fields) unnecessary for VoIP servers, particularly for unicast IP voice calls (a point-to-point RTP session between two participants). The header information consumes a considerable amount of bandwidth without any use through unicast IP voice calls [15,17–19].

Code name	Frame size (B)	Bit rate (kbps)			
G.723.1	20	5.3			
G.726	30	24			
LPC	14	5.6			
G.729	10	8			
G.728	10	16			

 Table 1: Common VoIP codes

 Table 2: Header overhead ratio

Header	Payload size (byte)	Total number	
40-byte RTP/UDP/IP	10	$=((40)/(40+10)) \times 100\%$	80%
	14	$=((40)/(40+14)) \times 100\%$	74%
	20	$=((40)/(40+20)) \times 100\%$	67%
	30	$=((40)/(40+30)) \times 100\%$	57%

The header overhead problem has been addressed through two main approaches: packet multiplexing and header compression. The packet multiplexing approach aims to unite several VoIP packet payloads from different sources into a single VoIP packet header to replace one packet header for each payload. Thus, the header overhead is reduced and bandwidth utilization is improved. The more payloads are multiplexed together, the greater is the improvement of bandwidth utilization. The header compression approach has successfully compressed the 40-byte UDP/RTP/IP VoIP packet header to only 2 bytes. Therefore, the header compression approach greatly reduces header overhead, thereby improving bandwidth utilization. The core of the header compression is based on redundancy of the fields in the RTP/UDP/IP VoIP packet header [6,10,20]. Apart from these two approaches, a new protocol called inter-asterisk exchange (IAX) has been proposed to carry IP voice calls [18,21]. The IAX protocol along with packet multiplexing and header compression approaches are discussed in detail in the following section. This article aims to propose a new approach (other than header compression and packet

multiplexing) to improve VoIP service bandwidth utilization, particularly the unicast IP voice calls. The core idea of the new approach is to reemploy and utilize the header information (fields) of the RTP/UDP/IP protocol that is dispensable through unicast IP voice calls. In general, these fields are used to carry the VoIP packet payload. Therefore, reducing the size of the payload saves bandwidth.

The rest of this paper is organized as follows. Section 2 presents the main approaches in handling the header overhead problem of the VoIP packet. Section 3 describes the proposed method including its core operations. Section 4 evaluates the performance of the proposed method. Finally, Section 5 concludes the paper.

2 Related Works

The researchers have exerted considerable effort to solve the header overhead problem resulting from attaching the 40-byte RTP/UDP/IP header to a small VoIP packet payload. This section presents an overview of the main approaches to these efforts.

The first approach to handle the header overhead problem in VoIP application is packet multiplexing. Several methods have been proposed under the multiplexing approach. Vulkan et al. suggested and patented one of the best multiplexing methods. This method contains a multiplexing entity located at the sender-side VoIP gateway and a de-multiplexing entity located at the receiver-side VoIP gateway. The multiplexing entity aggregates the VoIP packets heading toward the same VoIP gateway into one UDP/IP header. The packets are aggregated together until the new multiplexed packets reach a predetermined size or period of time. The de-multiplexing entity segregates the multiplexed packet to restore the original VoIP packets and transmit them to their destinations [12]. Another important packet multiplexing method to enhance the VoIP service bandwidth utilization is called VoIP piggyback (VoIPiggy) and was proposed by Salvador et al. [22]. This method aggregates the VoIP frames heading toward the same receiver into one 802.11 MAC layer header. Similar to the previous method, the VoIPiggy method contains a multiplexing entity that aggregates the VoIP frames into one large frame and a de-multiplexing entity that segregates the large frame to restore the original VoIP frames. The aggregation process continues until the large frame reaches a specific predefined size. This frame aggregation method leads to a very high header overhead reduction. Furthermore, only one acknowledgment is sent to the large frame instead of a separate acknowledgment of each frame. Thus, the suggested VoIPiggy method greatly improves the bandwidth utilization of the VoIP service. Results of testing the suggested VoIPiggy method in different scenarios under various conditions (data rate, number of devices, load, and others) showed that a double amount of VoIP data can be transmitted under the same network conditions. However, many difficulties are encountered in the VoIP packet multiplexing approach. One difficulty is that the packet multiplexing approach imposes additional delay on the VoIP packets, thereby degrading the quality of the VoIP service. Another considerable issue is that all the multiplexed stream packets have the same QoS while traveling throughout the network. In reality, some of the multiplexed streams may be more important than others and thus require better OoS [3,23,24].

The second approach to enhance VoIP-service bandwidth utilization is header compression. As discussed, the 40-byte RTP/UDP/IP header causes a considerable waste of VoIP service bandwidth. Sandlund et al. [19] proposed a technique called robust header compression (ROHC) to lessen the VoIP packet header size from 40 bytes to generally 2 bytes. The ROHC technique utilizes two key properties of the 40 RTP/UDP/IP VoIP packet header to achieve this extraordinary size reduction. The first property is that many elements of the RTP/UDP/IP header are

unchanged during the entire call time. Accordingly, these elements are transmitted at the beginning of the call and eliminated from the rest of the packets. The other property is that many other elements can be derived from the previous packets based on certain mechanisms. Therefore, again, these elements are transmitted at the beginning of the call and eliminated from the rest of the packets. The ROHC technique achieves 97% reduction of the typical RTP/UDP/IP header size. Nevertheless, similar to the packet multiplexing approach, the header compression approach faces many difficulties. For example, the header compression approach is complicated and requires extensive processing and substantial resources. Thus, this approach burdens the network devices and imposes additional delay on the VoIP packets. In addition, the header compression approach wastes the bandwidth when packet loss occurs because (i) some packets that have been transmitted are simply ignored and (ii) a complete header packet must be transmitted to refresh the context table at the receiver side [15,25].

Apart from the aforementioned approaches, the IAX protocol was proposed by Spencer et al. to replace the RTP protocol and handle the header overhead issue. The header size protocol of IAX is only 4 bytes with only the necessary information to carry unicast IP voice calls. IAX has greatly reduced the header overhead, where the IAX protocol is 4 bytes whereas the RTP is 12 bytes. Nevertheless, the combination of IAX, UDP, and IP (IAX/UDP/IP) protocols still causes the same header overhead problem of the RTP/UDP/IP protocols, whereas the header overhead resulting from the IAX/UDP/IP protocols is between 51% and 76%, which is still a considerable overhead [18,21,26].

Accordingly, we propose a new approach to handle the header overhead problem resulting from the 40-byte VoIP packet RTP/UDP/IP header. The new approach aims to reemploy and utilize these features and header information (fields) unnecessary in the RTP/UDP/IP protocols for the benefit of VoIP applications, particularly for unicast IP voice calls. In general, these fields are used to carry the VoIP packet payload. Therefore, bandwidth can be saved by reducing the payload size or keeping it to zero. The proposed approach is called zero-size payload (ZSP), which is discussed in detail in the following section.

3 Proposed ZSP Method

The main goal of the ZSP method is to contract the size of the VoIP packet payload and make it as small as zero bytes, thereby enhancing the bandwidth utilization of VoIP servers, particularly the unicast IP voice calls. The ZSP method can be implemented at the client side or at the VoIP gateway connected to the wide area network (WAN) link. The ZSP method is recommended for implementation at the VoIP gateway for several reasons. The first reason is the ability to use any VoIP client from any vendor with any device without the concern as to whether or not the client supports the ZSP method. The second reason is that the local area network usually has a great amount of free bandwidth, whereas the WAN link bandwidth is limited and expensive. The third reason is that the ZSP method can be implemented with other methods, such as VoIP aggregation methods, which are usually implemented at the VoIP gateway [12,25,27].

The ZSP method consists of two main modules. The first is called payload contracting (PC) module, which resides at the sender side gateway. The second is called payload restoration (PR) module, which resides at the receiver side gateway. The PC module contracts the voice frame (VoIP packet payload) and produces a VoIP packet with a smaller or even a zero payload. On the other hand, the PR module reverts the voice frame to its normal size and produces the original VoIP packet. Sections 3.3 and 3.5 discuss the PC and PR modules, respectively. Fig. 1 shows a network topology, including the location at which the ZSP method can be implemented.



Figure 1: Location of ZSP method modules

3.1 Core Concept of Proposed ZSP Method

As mentioned, the main goal of the ZSP method is to contract the size of the VoIP packet payload, thereby enhancing the bandwidth utilization of the VoIP service. The ZSP method intends to do so by reemploying some of the RTP/UDP/IP header fields. These fields are used to keep the voice data of the VoIP packet payload. The ZSP method reemploys and utilizes the RTP/UDP/IP header fields under certain conditions. First, the field must be dispensable by the VoIP service, particularly unicast IP voice calls. In other words, the VoIP service need not perform its function of transporting the voice data between the call parties. Second, keeping a value other than the original typical value in a field of the RTP/UDP/IP header does not cause any misinterpretation of the VoIP packets by the end clients. Third, no context/state table should be kept at the network devices through the path between the call parties, including the sender and receiver VoIP gateways, to restore the original values of the fields. This third condition aims to avoid some of the problems resulting from the previous approaches (i.e., header compression approach), including avoiding the consumption of network resources (CPU, memory, and bandwidth) [15,25]. Fourth, in some cases, the original value of a field can be replaced at the PC module and restored at the PR module without breaking the third condition before reaching the end client. This field is also utilized by the ZSP method to keep part of the VoIP packet payload. Accordingly, these fields are (i) Identification (16 bits), Flags (3 bits), Fragment Offset (3 bits), Protocol (8 bits), and Source IP Address (32 bits) in the IP protocol header; (ii) Source Port (16 bits), Length (16 bits), and Checksum (16 bits) in the UDP protocol header; and (iii) Synchronization Source (SSRC) (32 bits) in the RTP protocol header. The overall size of the caching fields is 152 bits (19 bytes). Henceforth, these fields are referred to as caching fields. The following section discusses the caching fields and why they can be utilized by the ZSP method without affecting the unicast IP voice telephony.

3.2 Caching Fields

The first caching fields are the **Fragmentation** fields (Identification, Flags, and Fragment Offset) in the IP protocol header. These fields are used to reassemble the fragmented packets at the receiver side in case the packet is fragmented. In general, the VoIP packet sizes are extremely small at 50–80 bytes. These sizes are smaller than any maximum transmission unit (MTU) of any common technology (e.g., ethernet MTU is 1500 bytes and the internet minimum MTU is 576 bytes). Reference [28] shows the MTU for most of the common technologies. Accordingly, the VoIP packets are not fragmented. Therefore, the **Fragmentation** fields are dispensable in the case of the VoIP service. Furthermore, the value of the **Fragmentation** fields is considered only if the packet is fragmented. Therefore, changing its value neither affects the delivery of the packet to its destination nor misinterprets the packets through the intermediary devices or end clients [12,29,30].

The second caching field is Protocol in the IP protocol header. This field is used to identify the layer 4 protocol. The layer 4 protocol used with VoIP is fixed and its UDP protocol has a value of 17 in the Protocol field. Changing this value may lead to misinterpretation of the VoIP packet. However, the PR module at the receiver VoIP gateway can always set the value of this field to 17 for any incoming VoIP packet from the PC module at the sender VoIP gateway. Thus, the Protocol field can be used by the ZSP method [13,31].

The third caching field is Source IP Address in the IP protocol header. This field is used to identify the sender of the data. Typically, the receiver needs the source IP address to be able to respond to the received data. However, VoIP sessions are not a request/response session and there is no response to the messages by the receiver client. Therefore, the source IP address is dispensable for the VoIP service and can be used to keep part of the VoIP packet payload [13,31].

The fourth caching field is Source Port in the UDP protocol header. However, the UDP source port is optional and is used only with certain applications in certain scenarios. In the case of VoIP, similar to the source IP, there is no response to the messages by the receiver client; thus, the source port is not needed to identify the source of the connection. Therefore, the source port is dispensable for the VoIP service and can be used by the ZSP method to keep part of the VoIP packet payload [13,31,32].

The fifth caching field is Length in the UDP protocol header. This field is used to keep the total length of the UDP datagram (UDP header and data). Similar to the discussion of the Protocol field, changing this value may lead to misinterpretation of the VoIP packet by the end client. However, the PR module at the receiver VoIP gateway can reset the value of the length field to its original value (as it leaves the client at the sender side) based on the Total Length field in the IP protocol header. However, the Total Length field in the IP protocol header. In the case of VoIP, the IP protocol header size is fixed and equal to 20 bytes. Accordingly, the PR module can find the original value of the Length field of the UDP protocol beader [13,19,32].

The sixth caching field is Checksum in the UDP protocol header. The UDP Checksum is an optional field that can be disabled by setting it all to zero. Many advanced algorithms can be used to construct the corrupted packets, particularly the VoIP packets. Therefore, disabling the optional checksum field allows the application layer to use these advanced algorithms to salvage the corrupted VoIP packets, thereby improving the VoIP calls quality. Accordingly, the VoIP service can benefit from disabling the optional Checksum field more than enabling it. Therefore, the Checksum field is dispensable for the VoIP service and can be used to keep part of the VoIP packet payload [15,33,34].

The seventh caching field is SSRC Identifier in the RTP protocol header. The SSRC Identifier is used to uniquely identify the source of the call in case of multicast VoIP sessions with a group of participants or even in a unicast session that has a translator or mixer. However, in the unicast IP voice calls between two participants (point-to-point call), the SSRC Identifier of the RTP protocol is dispensable. Thus, the proposed ZSP method can utilize the SSRC field to keep some of the voice frame data [10,15].

3.3 ZSP Method: PC Module

The PC module performs a set of operations at the WAN gateway on the sender side. Initially, the VoIP packet payload is separated from the VoIP packet header. Thereafter, the separated VoIP

packet payload is stored in the VoIP packet header protocol (RTP/UDP/IP) fields, specifically, the caching fields discussed above. The VoIP packet payload data are stored in the caching fields in the following order: the first four bytes in the Fragmentation fields, the next byte in the Protocol field, the following four bytes in the Source IP Address field, the following two bytes in the Source Port field, the following two bytes in the Length field, the following two bytes in the Checksum field, and the following four bytes in the SSRC field. This order accounts for a total of 19 bytes of VoIP packet payload. The remaining bytes of the VoIP packet payload (if any) is kept as a normal VoIP packet payload. If the size of the VoIP packet payload is less than that of the caching fields (19 bytes), then the remainder of the caching fields is set to zero and no packet payload exists. Tab. 3 demonstrates the distribution of the VoIP packet payload in the caching fields with four different payload sizes. The size of G.726 and G.723.1 is greater than the size of the caching fields; 11 bytes and 1 byte are placed as a packet payload when using each of the codecs respectively. The size of LPC, G.729, and G.728 is smaller than the size of the caching fields; there is no payload when using any of these three codecs. This operation produces a new contracted VoIP packet. Subsequently, the internet header length (IHL) field of the IP protocol in the packet is set to a certain value, as discussed in the following. Finally, the new packet is transmitted to the VoIP gateway of the receiver. Fig. 2 illustrates the PC module operations.

Payload size	Size of stored voice data in caching fields from payload									
	Fragmentation	Protocol	Source IP address	Source port	Checksum	Length	SSRC	Packet payload		
G.726-30 bytes	4 bytes	1 byte	4 bytes	2 bytes	2 bytes	2 bytes	4 bytes	11 bytes		
G.723.1-20 bytes	4 bytes	1 byte	4 bytes	2 bytes	2 bytes	2 bytes	4 bytes	1 byte		
LPC-14 bytes	4 bytes	1 byte	4 bytes	2 bytes	2 bytes	1 byte and zero	Zeros	N/A		
G.729-10 bytes	4 bytes	1 byte	4 bytes	1 byte and zero	Zeros	Zeros	Zeros	N/A		
G.728-10 bytes	4 bytes	1 byte	4 bytes	1 byte and zero	Zeros	Zeros	Zeros	N/A		

Table 3: Distribution of VoIP packet payload in caching fields

3.4 IHL Field and Intermediary Layer Three Devices

In typical situations, a large amount of packets of different types and sources passes through the intermediary layer of three devices (e.g., routers). The routers are unable to find the type of data passing through it using the traditional IP protocol header. For the proposed ZSP method, the intermediate routers (between and including the sender and receiver VoIP gateways) need to distinguish the VoIP packet generated by the PC module from the other types of packet. This approach is intended to avoid misinterpreting the new values of the VoIP packet header fields after being modified by the PC module. In other words, for all the packets generated by the PC module, the routers must not process the value of the Fragmentation, Protocol, and Source IP Address fields of the IP protocol header.

The ZSP method uses the IHL field of the IP protocol to distinguish the VoIP packet generated by the PC module from all the other packet types. Typically, the IHL field of the IP protocol is used to keep the size of the IP protocol header. The VoIP packet uses the traditional IP protocol header without any extra options. Thus, the IHL always contains the minimum value of five when the IP protocol is used with VoIP [13]. The PC module of the ZSP method reemploys the value of the IHL field to indicate that the VoIP packet is not a normal packet but one with modified header values. Typically, the minimum value of the IHL field is five and all values

less than five are unused. Therefore, any value less than five can be used to indicate a packet with modified header values; however, the PC module uses the value of one. Accordingly, when a router receives a packet with the IHL field equal to one, (i) the router considers the IP header size equal to 20 bytes and (ii) the router must not process the values of Fragmentation, Protocol, and Source IP Address fields. Definitely, similar to what happened when the CRTP and ROHC header compression standards are proposed and implemented [19], the routers' internal operations must be modified to be able to interpret the new values of the VoIP packets generated by the PC module.



Figure 2: PC module operations

3.5 ZSP Method: PR Module

The PR module performs a set of operations at the WAN gateway of the receiver side to restore the original VoIP packet format and values. Initially, the IHL field value of the IP protocol of the incoming packets is inspected to distinguish the VoIP packets generated by the PC module of the ZSP method from all other packets. Then, the payload (if any) of the VoIP packets generated by the PC module is separated from the VoIP packet header. Thereafter, the voice data are extracted from the caching fields of the VoIP packet header. The VoIP packet payload and voice data from the previous two steps are combined to construct the original VoIP packet payload. The combination process is performed in the following order: the value of the Fragmentation fields is placed first, followed by the value of the Protocol field, value of the Source IP Address field, value of the Source Port field, value of the Length field, value of the Checksum field, and value of the SSRC field. The VoIP packet payload (if any) is placed last. Thereafter, the value of the Protocol field of the IP protocol and the value of the Length field of the UDP protocol is reset to their original value, as explained in Section 3.2. Subsequently, all the remaining fields used by the PC module at the sender side gateway are set to zero to avoid misinterpretation at the destination VoIP client. Before the final step, the resulting original payload is attached to the VoIP packet header (RTP/UDP/IP), which constitutes the original VoIP packet. Finally, the VoIP packet is transmitted to its destination client. Fig. 3 illustrates the PR module operations.



Figure 3: PR module operations

4 ZSP Method Performance Analysis

The performance of the proposed ZSP method is estimated in this section. The method was tested and estimated against the traditional method (the standard 40-byte RTP/UDP/IP header)

of transmitting the VoIP packets. For simplicity, the traditional RTP/UDP/IP method is called the RUI method. The bandwidth utilization efficiency was used to compare the proposed ZSP method against the traditional RUI method. The bandwidth utilization efficiency was measured based on the network capacity and saved bandwidth. To perform a realistic estimation, we measured the network capacity and saved bandwidth with three different codecs, namely, G.723.1, G.726, and LPC. The simulation model scheme contains the two components of the ZSP method: (i) PC Component, which resides at the sender side gateway, and (ii) PR Component, which resides at the receiver gateway. Each component uses a queue with a maximum size of 5. The two components are assumed to be connected using a WAN link, which is simulated as a first-in-first-out queue. The processes of PC and PR Components are explained in Sections 3.3 and 3.5, respectively.

4.1 Capacity

The capacity of the proposed ZSP method refers to the number of concurrent connections that can run through the network without packet loss. The capacity of the proposed ZSP method against the traditional RUI method was estimated at bandwidths of 100,200... to 1,000 kbps. The number of concurrent connections was increased for each link bandwidth until the packet loss began. The beginning of the packet loss denoted that the link capacity was exceeded. Thus, the number of concurrent connections for each link was equal to the number of connections before packet loss begins, based on which the capacity was measured. Figs. 4–6 show the capacity of the ZSP method compared with that of the RUI method with G.726, G.723.1, and LPC, respectively. As we can observe, the capacity when using the ZSP method is better (more calls are enabled) than that of the RUI method with the three codecs. This condition is due to placing all or part of the VoIP packet payload in the caching filed of the RTP/UDP/IP VoIP packet header. Furthermore, the difference in the capacity between the ZSP and RUI methods vary from codec to another because the ratio of the packet payload in the caching fields to the total packet size varies when different codecs are used.



4.2 Saved Bandwidth

This section discusses the saved bandwidth ratio of the proposed ZSP method. The saved bandwidth ratio was calculated based on the link capacity. The saved bandwidth ratio of the ZSP method in comparison with the RUI method was estimated using the G.726, G.723.1, and LPC codecs. As shown in Fig. 7, the ZSP method has sizable bandwidth saving compared with the RUI method with the three codecs. The bandwidth saving with G.723.1, G.726, and LPC is 32%, 28%, and 26%, respectively. These results are due to placing all or part of the VoIP packet payload in the caching field of the RTP/UDP/IP VoIP packet header. Furthermore, the difference between

the ZSP method and RUI method capacity varies from one codec to another. The reason is that the ratio of the packet payload in the caching fields to the total packet size varies when different codecs are used.



Figure 7: Saved bandwidth ratio

5 Conclusion

VoIP is a widespread IP-based service that is gradually replacing conventional landline telecommunication systems. However, many difficulties are encountered in the VoIP service, thereby hindering its widespread use. Foremost of these difficulties is the poor use of the bandwidth by the VoIP service, which results from the considerable header of the VoIP packet. This study proposed a new novel method called ZSP, which handles the large header overhead problem. The ZSP achieved by reemploying and utilizing the RTP/UDP/IP fields are dispensable by VoIP to maintain the VoIP packet payload. The ZSP method consists of PC and PR modules. On the

one hand, the PC module contracts the VoIP packet payload and generates a VoIP packet with a smaller or even a zero payload. On the other hand, the PR module reverts the VoIP packet payload to its normal size and generates the original VoIP packet. The suggested ZSP method was estimated with G.723.1, G.729, and LPC codecs. The estimation result showed that the bandwidth savings were 32%, 28%, and 26% with the three codecs, respectively. Therefore, the proposed ZSP method is a promising solution to address the header overhead problem and improve the utilization of the VoIP service bandwidth. In the future, the proposed method will be integrated with other approaches, which handle the header overhead problem, such as the packet multiplexing approach.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] M. Abualhaj, "Ca-ittp: An efficient method to aggregate VoIP packets over ittp protocol," *International Journal of Innovative Computing, Information and Control*, vol. 15, no. 3, pp. 1067–1077, 2019.
- [2] Y. Zhang, K. Xie and O. Ruan, "An improved and efficient mutual authentication scheme for session initiation protocol," *PLoS ONE*, vol. 14, no. 3, pp. 1–15, 2019.
- [3] M. M. Abualhaj, M. Kolhar, K. Qaddoum and A. A. Abu-Shareha, "Multiplexing VoIP packets over wireless mesh networks: A survey," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 8, pp. 3728–3752, 2016.
- [4] E. T. Affonso, R. D. Nunes, R. L. Rosa, G. F. Pivaro and D. Z. Rodriguez, "Speech quality assessment in wireless VoIP communication using deep belief network," *IEEE Access*, vol. 6, pp. 77022– 77032, 2018.
- [5] C. Olariu, J. Fitzpatrick, Y. Ghamri-Doudane and L. Murphy, "A delay-aware packet prioritisation mechanism for voice over IP in wireless mesh networks," in *IEEE Wireless Communications and Networking Conf.*, Doha, Qatar, pp. 1–7, 2016.
- [6] Q. Shambour, S. N. Alkhatib, M. M. Abualhaj and Y. Alrabanah, "Effective voice frame shrinking method to enhance VoIP bandwidth exploitation," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 7, pp. 313–319, 2020.
- [7] J. Y. Jung, H. S. Kang and J. R. Lee, "Performance evaluation of packet aggregation scheme for VoIP service in wireless multi-hop network," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1037–1045, 2013.
- [8] N. Gupta, N. Kumar and H. Kumar, "Comparative analysis of voice codecs over different environment scenarios in VoIP," in *Second Int. Conf. on Intelligent Computing and Control Systems*, Madurai, India, pp. 540–544, 2018.
- [9] M. O. Ortega, G. C. Altamirano, C. L. Barros and M. F. Abad, "Comparison between the real and theoretical values of the technical parameters of the VoIP codecs," in *IEEE Colombian Conf. on Communications and Computing*, Barranquilla, Colombia, pp. 1–6, 2019.
- [10] A. H. Hussein, M. Abu-Alhaija and K. Nairoukh, "New RTP packet payload shrinking method to enhance bandwidth exploitation over RTP protocol," *International Journal of Advanced Computer Science* and Applications, vol. 11, no. 8, pp. 139–142, 2020.
- [11] M. M. Abualhaj, M. M. Al-Tahrawi and S. N. Al-Khatib, "A new method to improve voice over IP (VoIP) bandwidth utilization over internet telephony transport protocol (ITTP)," in *Proc. of the 2019* 8th Int. Conf. on Software and Information Engineering, Cairo, Egypt, pp. 192–195, 2019.
- [12] V. Csaba, A. Rakos, Z. Vincze and A. Drozdy, "Reducing overhead on voice traffic, U.S. Patent 8,824,304, 2014.
- [13] G. Tomsho, *Guide to Networking Essentials*, 8th ed. Boston, Massachusetts, United States: Cengage Learning, 2019.

- [14] B. Hartpence, Packet Guide to Voice Over IP: A System Administrator's Guide to VOIP Technologies, 1st ed. Sebastopol, California: O'Reilly Media, Inc., 2013.
- [15] C. Perkins, RTP: Audio and Video for the Internet, Boston, Massachusetts, United States: Addison-Wesley Professional, 2003.
- [16] J. Gao, Y. Li, H. Jiang, L. Liu and X. Zhang, "An RTP extension for reliable user-data transmission over VoIP traffic," in 5th Int. Symp., SocialSec, Copenhagen, Denmark, 2019.
- [17] M. M. Abualhaj, M. SK., R. Sureswaran, T. Wan and A. M. Manasrah, "ITTP: A new transport protocol for VoIP applications," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 3, pp. 1–10, 2012.
- [18] M. Spencer, B. Capouch, E. Guy, F. Miller and K. Shumard, "IAX: Inter-asterisk exchange version 2," IETF RFC 5456, 2010.
- [19] K. Sandlund, G. Pelletier and J. Le, "The robust header compression (ROHC) framework," *IETF RFC* 5795, 2010.
- [20] S. Seytnazarov and Y. Kim, "QoS-aware adaptive A-MPDU aggregation scheduler for enhanced VoIP capacity over aggregation-enabled WLANs," in 2018 IEEE/IFIP Network Operations and Management Symp., Taipei, pp. 1–7, 2018.
- [21] M. Boucadair, Inter-Asterisk Exchange (IAX): Deployment Scenarios in Sip-Enabled Networks, 1st ed. Hoboken, New Jersey, United States: John Wiley & Sons Ltd., 2009.
- [22] P. Salvador, V. Mancuso, P. Serrano, F. Gringoli and A. Banchs, "VoIPiggy: Analysis and implementation of a mechanism to boost capacity in IEEE 802.11 WLANs carrying VoIP traffic," *IEEE Transactions on Mobile Computing*, vol. 13, no. 7, pp. 1640–1652, 2013.
- [23] M. M. Abu-Alhaj, Internet Telephony Transport Protocol (ITTP): An Efficient Transport Protocol for VOIP Applications, Malaysia: Diss. Universiti Sains, 2011.
- [24] E. Charfi, C. Gueguen, F. Lamia, C. Bernard and K. Lotfi, "Dynamic frame aggregation scheduler for multimedia applications in IEEE 802.11n networks," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 2, pp. 1–15, 2015.
- [25] H. P. Sze, C. L. Soung, J. Y. B. Lee and D. Yip, "A multiplexing scheme for h.323 voice-over-IP applications," *Selected Areas in Communications*, vol. 20, no. 7, pp. 1360–1368, 2002.
- [26] A. Mosleh, M. Ahmed, B. Mahmoud, A. Nibras and C. Lingeswari, "Transport layer protocols taxonomy from voice over IP perspective," *Advanced Computing: An International Journal*, vol. 2, no. 4, pp. 9–21, 2011.
- [27] M. M. Abualhaj, N. Sumaya, M. Kolhar, A. Munther and Y. Alraba'nah, "Effective voice frame pruning method to increase VOIP call capacity," *TEM Journal*, vol. 9, no. 1, pp. 48–54, 2020.
- [28] 2020. [Online]. Available: https://www.juniper.net/documentation/en_US/junos/topics/reference/general/ interfaces-media-mtu-size-by-interface-type.html.
- [29] E. Sadot and I. E. Zilbershtein, "Packet fragmentation prevention. U.S. Patent No. 9,088,494, 2015.
- [30] 2020. [Online]. Available: https://www.geeksforgeeks.org/fragmentation-network-layer/?ref=lbp.
- [31] C. M. Kozierok, *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference*, San Francisco, California, United States: No Starch Press, 2005.
- [32] L. Eggert, G. Fairhurst and G. Shepherd, "UDP usage guidelines," IETF RFC 8085, 2017.
- [33] A. Sheraz, M. Muhammad, N. Farhan, H. Muhammad and K. Jong-Myon, "Protocols and mechanisms to recover failed packets in wireless networks: History and evolution," *IEEE Access*, vol. 4, pp. 4207–4224, 2016.
- [34] S. Stein and E. Rippel, "Efficient double parity forward error correction on a communication network. U.S. Patent No. 10,567,102, 2020.