

Contracting VoIP Packet Payload Down to Zero

Mosleh M. Abualhaj¹, Mayy M. Al-Tahrawi², Mahran Al-Zyoud¹

¹Department of Networks and Information Security, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan

²Department of Computer Science, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan

E-mails: m.abualhaj@ammanu.edu.jo mtahrawi@ammanu.edu.jo m.zyoud@ammanu.edu.jo

Abstract: *The inefficient use of the IP network bandwidth is a fundamental issue that restricts the exponential spreading of Voice over IP (VoIP). The primary reason for this is the big header size of the VoIP packet. In this paper, we propose a method, called Short Voice Frame (SVF), that addresses the big header size of the VoIP packet. The main idea of the SVF method is to make effective use of the VoIP packet header fields that are unneeded to the VoIP technology. In particular, these fields will be used for temporarily buffering the voice frame (VoIP packet payload) data. This will make the VoIP packet payload short or even zero in some cases. The performance evaluation of the proposed SVF method showed that the use of the IP network bandwidth has improved by up to 28.3% when using the G.723.1 codec.*

Keywords: *VoIP, VoIP protocols, bandwidth utilization.*

1. Introduction

Over the years, Internet Protocol (IP) networks have embraced a big number of new technologies such as Voice over IP (VoIP) [1, 2]. VoIP is consistently evolving and substituting the old telecommunication technology, in both business and domestic sectors. Businesses are using VoIP for several reasons such as lessening their communications costs and enhancing the interaction and service level with employees and clients [3, 4]. Nevertheless, VoIP is encountering two main issues that slow down its domination of the telecommunication world. Namely, the low quality of the VoIP calls in comparison to the old telecommunication technology and the inefficient use of the IP network bandwidth [5, 6]. This paper focuses on the IP network bandwidth issue.

In general, two main phases are performed to make a VoIP call [1]. The first phase requires a caller and callee to set up a session to verify each other and agree upon some parameters that are needed to make a successful VoIP call, such as the used VoIP codec. To achieve this, VoIP technology uses so-called signaling protocols

[1, 7]. Session Initiation Protocol (SIP) is the dominant signaling protocol due to its simplicity, scalability, extensibility, and some other features that provide better VoIP service [8, 9]. In the second phase, after the call session is established, the media voice data begin transferring between the two ends of the call. To transfer the voice media, VoIP technology uses so-called media transfer protocols [1, 10]. Real-time Transport Protocol (RTP) is the dominant and most used media, both voice and video, transfer protocol [11, 12]. The RTP protocol works in collaboration with the UDP protocol to be able to convey the voice data over the IP network. The size of the RTP protocol is 12 bytes, the UDP protocol is 8 bytes, and the IP protocol is 20 bytes. Therefore, the VoIP packet header size comprises of 40 bytes RTP/UDP/IP [12, 13]. On the other hand, the VoIP codecs produce a very small voice frame (VoIP packet payload) between 10 and 30 bytes. For example, the G.726 produces 30 bytes voice frame, G.723.1 – 20 bytes, the LPC – 14 bytes, and G.729 – 10 bytes [14-16]. 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, the ratio of the header size (40 bytes RTP/UDP/IP) to the payload size (10 and 30 bytes) is considerable and consumes much bandwidth. For example, the portion of the bandwidth consumed by the 40 bytes RTP/UDP/IP with each of the above codec is 57.1%, 66.6%, 74%, and 80%, respectively. Thus, the big header size, resulting from the RTP/UDP/IP protocols, is the main cause to the inefficient use of the IP network bandwidth [12, 13, 17].

The RTP/UDP/IP are generic protocols that contain a lot of information (fields) in their header to transfer various types of data for different applications [18]. Most of this information is unneeded to transfer the VoIP applications' data, particularly the point-to-point voice calls (voice calls between two participants) [19-21]. As discussed, these unneeded RTP/UDP/IP protocols header information are the main cause of the inefficient use of the IP network bandwidth [12, 13, 17]. The packet aggregation and header compression are the two main approaches that have been designed to handle the inefficient use of the IP network bandwidth issue [12, 22, 23]. In addition, the Inter-Asterisk eXchange (IAX) have been designed as a specified protocol to carry the point-to-point voice calls [24]. The packet aggregation and the header compression approaches and the IAX protocol will be discussed in the following section. This article intends to design a new approach to conserve the network bandwidth when using VoIP applications, particularly the point-to-point voice calls. The new approach will reemploy and utilize the RTP/UDP/IP protocols header's information (fields) that are unneeded to transfer the point-to-point voice calls. The proposed method is designed to make effective use of these fields. In particular, these fields will be used for temporarily buffering the voice frame data of the VoIP packet.

The remaining of this article is organized as follows: Section 2 discusses the main approaches to tackle the inefficient use of the IP network bandwidth by VoIP applications; Section 3 provides a detailed explanation of the proposed method including its components, internal operations, and the core principle of the method; Section 4 analyzes the proposed method in comparison to the traditional method; Section 5 presents the outcome of this research.

2. Related works

There are various approaches used to handle the header overhead issue of the VoIP packets. This section presents these approaches.

The first approach to enhance VoIP application bandwidth exploitation is packet aggregation. In this approach, the VoIP packets that are headed toward the same receiver are aggregated into one header to replace one packet header for each payload. This approach reduces the header overhead of each packet and greatly improves bandwidth utilization. One of the best coalescence methods, to improve the VoIP technology bandwidth exploitation, was suggested by Salvador P. et al. [25] in 2013. The suggested methodology is named VoIP Piggy back (VoIPiggy). The VoIPiggy technique coalesces the VoIP frames, which are going towards the same recipient, in one 802.11 MAC layer preamble. Like the previous technique, the VoIPiggy technique contains a coalescence module that coalesces the VoIP frames in a single large frame and a de-coalescence module that de-coalesces the large frame to recreate the original VoIP frames. The coalescence cycle proceeds until the large frame gets to a particular predetermined size. This frame coalescence technique prompted an extremely high preamble overhead decrease. Besides that, just an acknowledgment is transmitted to the large frame rather than a separate acknowledgment of every frame. Therefore, the suggested VoIPiggy technique has profoundly enhanced the bandwidth exploitation of VoIP technology. The result, of testing the proposed VoIPiggy technique in various situations with different criteria (rate, number of apparatuses, load, and ..., etc.), showed twice the amount of VoIP information can be communicated under the same network criteria [25]. However, there are many problems that hinder the spread of the packet aggregation techniques. Firstly, the packet aggregation necessitates multiple streams with a regular arrival time. Where, if the frames from different steams arrive at different times to the aggregation device, then some packets will be waiting until the rest of the frames arrive. Therefore, impose an extra delay. Moreover, the packet aggregation/de-aggregation process itself imposes some delay. Therefore, impose extra delay as well. In addition, some streams are more important than the other. Therefore, those streams require better QoS. However, aggregation of multiple streams together imposes giving all the streams the same QoS [3, 25-27].

The second approach to handle the inefficiency of using the bandwidth is the VoIP packet header optimization. The RObust Header Compression (ROHC) is a header optimization technique that manages to compact the VoIP packet header to 2 bytes. The ROHC relies on two primary observations to achieve that. The first observation is that a considerable number of fields in the RTP/UDP/IP header are constant throughout all the packets in the VoIP call. In this manner, these fields can be transmitted when the call starts and discarded from the rest of the packets of the call. The second observation is that another considerable number of fields in the RTP/UDP/IP header is changing in a specific calculable manner in the consecutive packets. Hence, these fields can be easily determined based on a simple mathematical formula. In this manner, once more, these fields can be transmitted when the call starts and discarded from the rest of the packets of the call. Following these two observations, the ROHC technique has accomplished 97% bandwidth utilization

improvement [28]. Despite that, numerous issues have resulted when deploying the header optimization approach. First, the header optimization approach is consisting of many different processes and parts. Thus, it loads the equipment by a considerable amount of processing. Furthermore, the header optimization approach is not effective over the erroneous connections (connections with high packet loss and delay). In addition, the loss of numerous packets wastes bandwidth because some packets that have been transmitted are simply ignored and because a full header packet has to be transmitted to refresh the context stored on the equipment [20, 29, 30].

Aside from the previously mentioned methods, the Inter-Asterisk eXchange (IAX) protocol has been created by Spencer et al. to supplant the RTP protocol and deal with the header overhead problem. IAX has several features including the built-in signaling function, channels aggregation, and supports trunking. IAX protocol has two main headers: The Full-Frames header and the Mini-Frames header. The Mini-Frames header is the one that is used to carry IP voice calls. IAX Mini-Frames is just four bytes header size protocol with just the fundamental data to convey IP voice calls. IAX has exceptionally diminished the header overhead, where the IAX protocol is 4bytes while the RTP is 12 bytes [24]. In spite of that, the composition of IAX, UDP, and IP (IAX/UDP/IP) protocols causes a similar header overhead issue of RTP/UDP/IP protocols. Whereas, the header overhead caused by IAX/UDP/IP protocols is between 51% and 76%, which is still a significant overhead. Another important issue is that the IAX protocol has been designed to perform the signaling function by itself, without the need for SIP protocol or even compatibility with it. However, since SIP is the dominant signaling protocol, the IAX has not spread and its usage has been limited to only the initial application [24, 29, 30].

In this article, we intend to come out with a new approach, other than all the above exciting approaches, to conserve the network bandwidth when using VoIP applications, particularly the point-to-point voice calls. The new approach will reemploy and utilize the RTP/UDP/IP protocols header's data (fields) that are unneeded to transfer the point-to-point voice calls. These fields burden the network resources with no useful usage by point-to-point voice calls. The proposed method is designed to make effective use of these fields. In particular, these fields will be used for temporarily buffering the voice frame data of the VoIP packet. The proposed approach is called a Short Voice Frame (SVF). The SVF approach is discussed in detail in the following section.

3. The proposed SVF method

The objective of designing the proposed SVF method is to conserve the network bandwidth when using VoIP applications, particularly the point-to-point voice calls. The key idea of conserving network bandwidth, by the proposed SVF method, will be discussed in the following section. The proposed SVF method will be deployed on the VoIP appliance that is connected to the WAN for two main reasons. An important reason is the LAN bandwidth is sufficient and free as opposed to the WAN bandwidth which is limited and costly. Another important reason is that the SVP method can be deployed and integrated with the packet aggregation methods, which

are mostly deployed at the VoIP appliance that is connected to the WAN [17, 29, 32]. The design structure of the proposed SVF method comprises two components. The first component must be activated in the VoIP appliance located at the source of the call. The main function of this component is to decrease the size of the VoIP packet payload as small as zero, as explained in Section 3.2. The name of this component is Payload Squeezing (PS). The second component must be activated in the VoIP appliance located at the destination of the call. The main function of this component is to bring back the former original size of the VoIP packet payload, as explained in Section 3.3. The name of this component is Payload De-Squeezing (PDS). Fig. 1 demonstrates the location of the PS and PDS components.

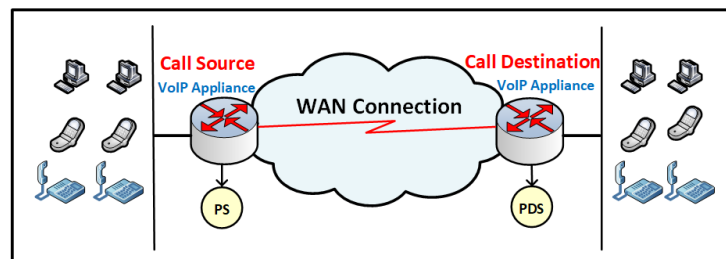


Fig. 1. Location of PS and PDS components

3.1. The abstract idea of proposed SVF method

The RTP/UDP/IP protocols contain several fields that are unneeded to transfer the point-to-point voice calls. These fields burden the network resources with no useful use by point-to-point voice calls [18-21]. The proposed SVF method is designed to make effective use of these fields. In particular, these fields will be used for buffering the voice data of the VoIP packet, thus, they are called the buffering fields. An RTP/UDP/IP protocols field is considered a buffering field based on certain criteria. Firstly, the field must be unneeded to deliver the packet to the intended destination. Secondly, the field must be unneeded to interrupt the packet in the correct way. Thirdly, some previous methods (e.g., header compression) are based on saving certain information on the intermediary devices between the call clients in order to retrieve the original value of the fields. The proposed SVF method must not do this so as to avoid burdening the intermediary devices and deteriorate the call quality [20, 29]. Fourthly, the value of some fields is pre-known and unchanged whatsoever [18]. In addition, the value of some other fields can be calculated from the value of the other fields, without keeping any information at the intermediary devices between the call clients. In both cases, these fields are considered buffering fields. Finally, the processes performed by the intermediate layer three devices must not be altered or updated to be able to interrupt the new values of the header. This is to avoid the compatibility issues of the proposed method with the networks' devices. Accordingly, the RTP/UDP/IP protocols fields that meet the above criteria and will be considered as buffering fields are: i) Identification (16-bit), Protocol (8-bit), and Source IP address (32-bit) in the IP protocol header; ii) Source Port Number (16-bit), Length (16-bit) and Checksum (16-bit) in the UDP protocol header; iii) Synchronization Source (SSRC) Identifier (32-bits) in the RTP protocol header. The size of the

buffering fields together is 136 bits (17-bytes). Table 1 summarizes and discusses the buffering fields.

Table 1. Buffering fields

Buffering field	Discussion
Identification	The IP protocol uses the Identification field in the fragmentation process (if applied). The typical VoIP packet size is less than 70 bytes, which is less than all known technologies for layer two. Therefore, the VoIP packet is never fragmented and the Identification field is unneeded by the VoIP applications [17, 33-35]
Protocol	The IP protocol uses the Protocol field to identify the layer four protocol. The VoIP data is carried over the UDP layer four protocol with a fixed value of 17 in the Protocol field. Therefore, the Protocol field may carry voice data and then its value reset to 17 at the receiver VoIP appliance to avoid misinterpretation of the packet by the client application [18, 36]
Source IP address	Typically, the receiver uses the Source IP Address field to respond to the arrived packets. However, the receiver VoIP client does not respond to the arrived packets. Therefore, the Source IP Address is unneeded by the VoIP applications [18, 36]
Source port number	Typically, the receiver uses the Source Port Number field to respond to the arrived segments. However, the receiver VoIP client does not respond to the arrived segments. Therefore, the Source Port Number is unneeded by the VoIP applications [18, 36, 37]
Length	The receiver client uses the Length field to determine the size of the layer four segments. This field value can be calculated by subtracting 20 (20 is the size of the IP protocol when used with VoIP) from the Total Length (TL) field of the IP protocol. Therefore, the Length field may carry voice data and then its value reset at the receiver VoIP appliance to avoid misinterpretation of the packet by the client application [18, 28, 37]
Checksum	The receiver client uses the optional Checksum field to check the integrity of the arrived packets. The packets that fail to pass the check are dropped. VoIP applications can use a certain algorithm to rescue these packets and improve the overall voice quality. Therefore, the optional Checksum field is unneeded by the VoIP applications [20, 38, 39]
SSRC	The SSRC field is needed to identify the sender of the packets with a multicast VoIP session and when the translator or mixer is used in VoIP sessions. However, it is not needed with the point-to-point voice calls. Therefore, the SSRC field may carry voice data [12, 20]

3.2. SVF method: PS component function

The function of the PS component is to decrease the size of the VoIP packet payload as small as zero. The operation at the PS component goes through several steps to achieve that. These steps are four.

Step 1. The VoIP packet payload is separated from the VoIP packet header.

Step 2. The VoIP packet payload is buffered in the buffering fields. The first 2-B (Bytes) are buffered in the identification field, the next 1-B in the protocol field, the next 4-B in the Source IP Address field, the next 2-B in the Source Port Number field, the next 2-B in the Length field, the next 2-B in the Checksum field, and the last 4-B in the SSRC field. The total size of the buffering fields is 17 bytes. This

produces a VoIP packet header with new values. The name of this header is the Buffering Header (BH).

Step 3a. If the size of the VoIP packet payload is smaller than the size of the buffering fields (17-B), then the unused portion of the buffering fields are set to zeros. This produces a VoIP packet without a payload, only the BH header.

Step 3b. If the size of the VoIP packet payload is larger than the size of the buffering fields (17-bytes), then the remaining portion of the packet payload is attached to the BH header. This produces a VoIP packet with a payload. In both cases (Step 3a and Step 3b), this new packet is called the BH Packet (BHP).

Step 4. The BHP packet is transmitted to the VoIP appliance that is located at the destination of the call.

Table 2 shows the allocation of the VoIP packet payload in the buffering fields with four different codecs. Fig. 2 demonstrates the steps that have occurred at the PS Component.

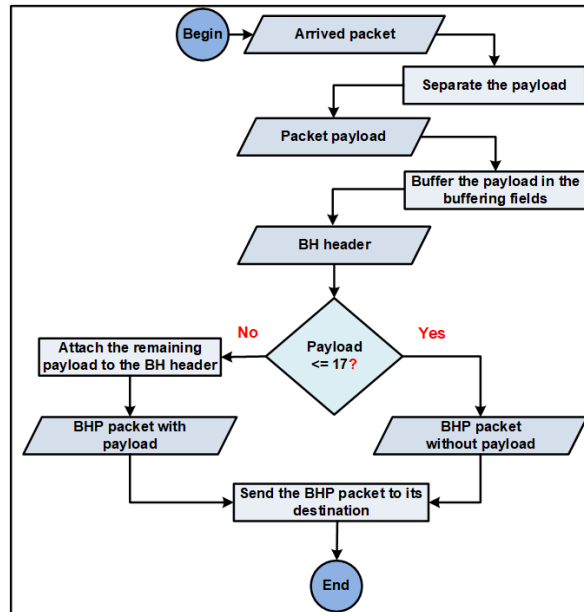


Fig. 2. PS component process

Table 2. Buffering fields' allocation

Payload size	Allocation of the VoIP packet payload in the buffering fields							
	Identification	Protocol	Source IP address	Source port number	Check-sum	Length	SSRC	Packet payload
G.726-30-B	2-B	1-B	4-B	2-B	2-B	2-B	4-B	13-B
G.723.1-20-B	2-B	1-B	4-B	2-B	2-B	2-B	4-B	3-B
LPC-14-B	2-B	1-B	4-B	2-B	2-B	2-B	1-B and Zeros	N/A
G.729-10-B	2-B	1-B	4-B	2-B	1-B and Zeros	Zeros	Zeros	N/A
G.728-10-B	2-B	1-B	4-B	2-B	1-B and Zeros	Zeros	Zeros	N/A

3.3. SVF method: PDS component function

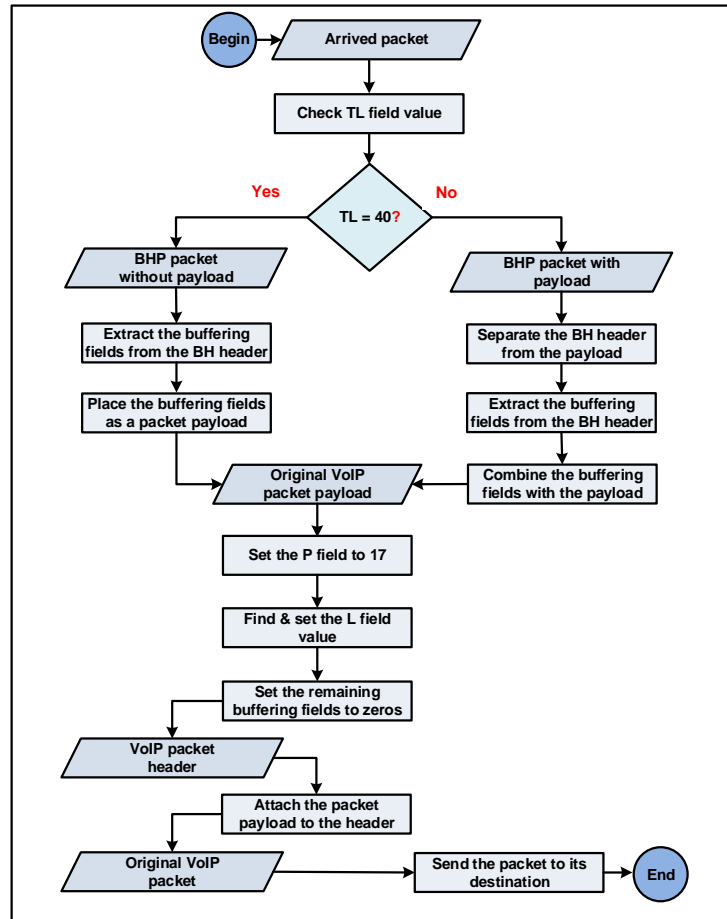


Fig. 3. PDS component process

The function of the PDS component is to bring back the former original size of the VoIP packet payload. The operation at the PDS component goes through several steps to achieve that. These steps are seven.

Step 1. The incoming BHP packet is inspected whether it has a payload or not. This is done by inspecting the value of the TL field of the IP protocol in the header. If the value of the TL field is equal to 40-B (40-B is the size of the typical RTP/UDP/IP VoIP packet header), then the incoming BHP packet has no payload. Otherwise, the incoming BHP packet has a payload.

Step 2. If the incoming BHP packet has a payload, then separate the BH header from the payload and extract the VoIP packet payload from the buffering fields. Otherwise, if the incoming BHP packet has no payload, then, only, extract the VoIP packet payload from the buffering fields.

Step 3. The separated payload (If any) and the extracted VoIP packet payload from Step 2 are integrated together. The data of the Identification field is positioned first, then the data of the Protocol field, then the data of the Source IP Address field,

then the data of the Source Port Number field, then the data of the Length field, then the data of the Checksum field, then the data of the SSRC field, and the VoIP packet payload (if any) is positioned last. This produces the former original VoIP packet payload.

Step 4a. The Protocol and Length fields are set to their original value, the value before they modified by the PS component, as discussed in Table 1.

Step 4b. The other buffering fields, other than the Protocol and Length fields, are set to zeros. This step must be done to avoid misinterpretation of these fields by the destination client of the call. Step4a and Step4b produce the RTP/UDP/IP VoIP packet header.

Step 5. The VoIP packet payload from step3 and the VoIP packet header from step4 are combined together to produce the VoIP packet.

Step 6. The produced VoIP packet is transmitted to the destination of the call.

Fig. 3 demonstrates the steps that occurred at the PDS Component.

4. SVF method performance analysis

This section measures the achievement of the suggested SVF method. The SVF method was inspected and examined against the common method (the common 40 bytes RTP/UDP/IP header) of carrying the VoIP packets. Henceforth, the common RTP/UDP/IP method is named the RUI method. The bandwidth exploitation proficiency has been exploited to analyze the suggested SVF method against the common RUI method. The bandwidth exploitation proficiency has been estimated dependent on the Maximum number of concurrent Calls (MaxCalls) and the saved bandwidth ratio. Besides the bandwidth exploitation proficiency, the proposed SVF method has been inspected against the common RUI method in terms of the network devices' buffer utilization.

The Network Simulation 2 (NS2) tool has been utilized to simulate the proposed SVF method. After simulating the SVF method, a network topology using NS2 has been designed to evaluate the SVF method whilst comparing it with the RUI method. The simulation model plan comprises of two components of the SVF method: i) PS component, which is situated at the transmitter side appliance, and ii) PDS component, which is situated at the recipient side appliance. The functions of PS and PDS components are discussed in Sections 3.2 and 3.3, respectively. Each component utilizes a buffer with a maximum size of 5. The two components are suggested to be associated with utilizing a WAN connection, which is mimicked as a first-in-first-out buffer. A CBR traffic generator is attached to each end node.

Three different experiments with the three diverse codecs (G.723.1, G.726, and LPC) have been run on the designed topology. In each experiment, the MaxCalls supported by the SVF method and the MaxCalls supported by the RUI method, in various channel bandwidths between 100 kbps to 1000 kbps, have been compared. The starting of packet loss is the indication that the channel is overwhelmed and its MaxCalls is reached. Therefore, the MaxCalls is counted just before the channel is overwhelmed.

4.1. MaxCalls

The section presents MaxCalls of the proposed method in comparison to the common RUI method. Fig. 4, 5, and 6 show the MaxCalls of the SVF method compared with that of the RUI method with G.726, G.723.1, and LPC, respectively. Fig. 7 shows the saved bandwidth ratio when using the SVF method compared with the RUI method, based on the MaxCalls with the three codecs. Observably, the MaxCalls when using the SVF method surpasses the MaxCalls of the RUI method with the three codecs, which reflected to the saved bandwidth ratio as well. Clearly, the reason behind this improvement of both MaxCalls and the saved bandwidth ratio is putting portion or even the whole VoIP packet payload in the buffering filed of the RTP/UDP/IP VoIP packet header. Moreover, the difference in the MaxCalls and saved bandwidth ratio between the SVF and RUI methods vary from a codec to another because the ratio of the packet payload in the buffering fields to the total packet size varies when different codecs are used.

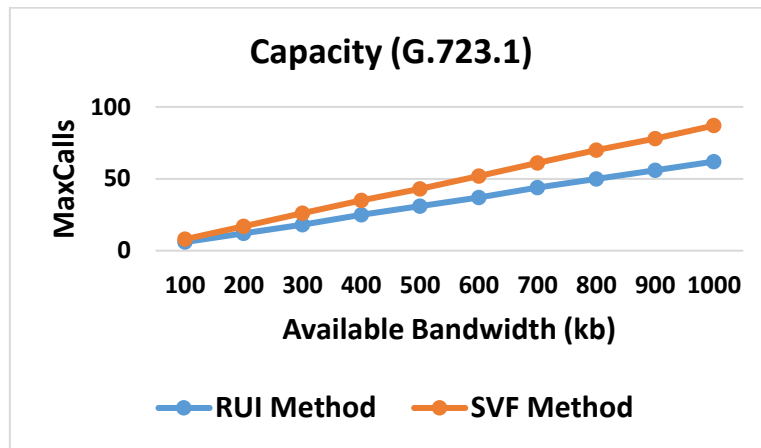


Fig. 4. Capacity (G.723.1)

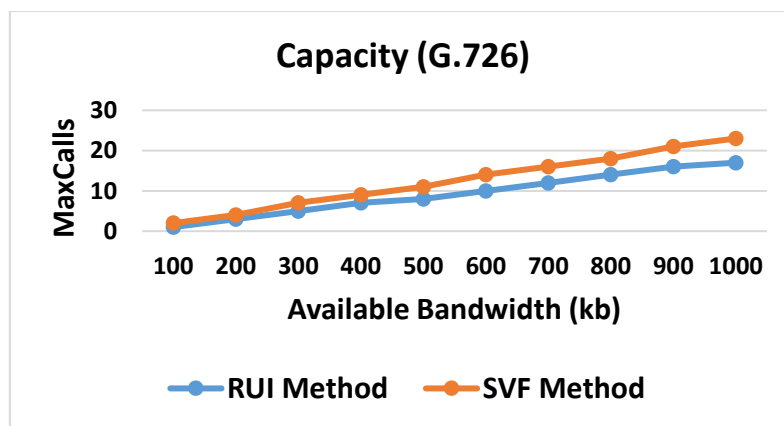


Fig. 5. Capacity (G.726)

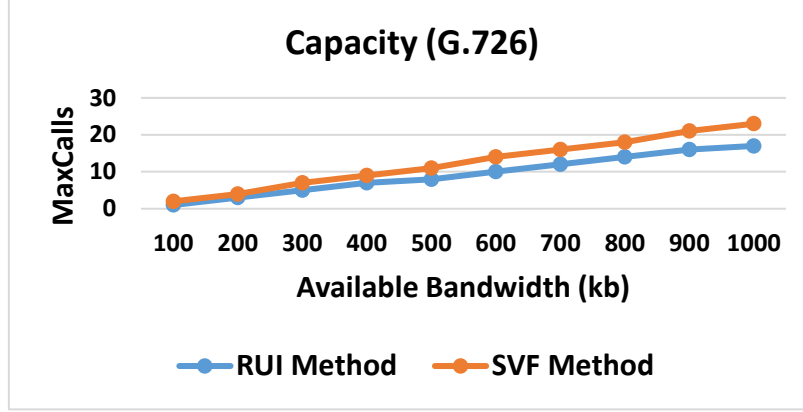


Fig. 6. Capacity (LPC)

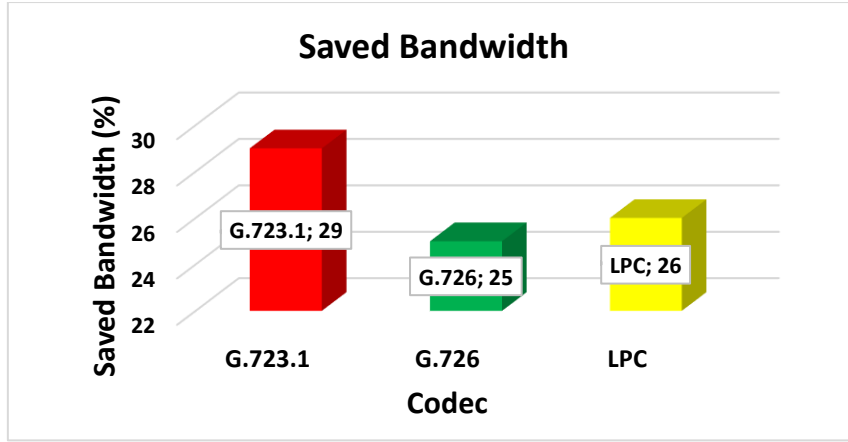


Fig. 7. Saved bandwidth ratio

4.2. Buffer utilization

The section presents the buffer utilization of the proposed method in comparison to the common RUI method. When the device buffer capacity is exceeded, then the incoming packets are lost. In this case, the buffer capacity impacts the packet loss ratio and also the delay. The next equation is used to calculate the buffer capacity:

$$(1) \quad BC = \frac{BS}{P_{kts}}$$

where, BC is the Buffer Capacity in packet, BS is the Buffer Size in bytes, and Pkts is the Packet size in bytes. The next equation is used to calculate the buffer utilization improvement ratio, when using the SVF method rather than the RUI method:

$$(2) \quad \text{Improvement \%} = \frac{SVF_{BC} - RUI_{BC}}{RUI_{BC}} \times 100\%.$$

For example, assuming that the buffer size is 1000 bytes, using Equations (1) and (2), Fig. 8 shows the buffer improvement ratio when using the SVF protocol with G.726, G.723.1, and LPC codecs.

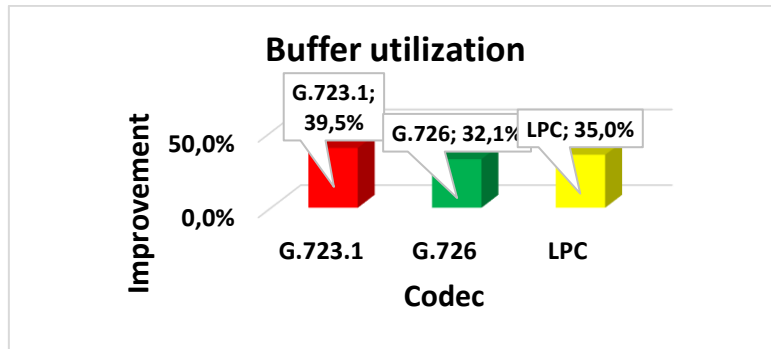


Fig. 8. SVF method buffer utilization improvement ratio

As we can see in Fig. 8, the SVF method improves the buffer utilization over the RUI method, when using G.726, G.723.1, and LPC codecs. With the three codecs, the SVF method substantially improves the buffer utilization in comparison to the RUI method. This results in less packet loss and delay, leading to better voice quality. Clearly, the reason behind this improvement of buffer utilization ratio is putting portion or even the whole VoIP packet payload in the buffering field of the RTP/UDP/IP VoIP packet header. Moreover, the difference in the improvement of buffer utilization ratio between the SVF and RUI methods vary from a codec to another because the ratio of the packet payload in the buffering fields to the total packet size varies when different codecs are used.

5. Conclusion

In recent years there has been a considerable increase in the demand for VoIP technology both in business and domestic sectors. Nevertheless, VoIP has been encountering inefficient use of the IP network bandwidth issue that slows down its propagation. Several approaches have been proposed to tackle this issue such as packet aggregation and header compression methods and the IAX protocol. This paper has proposed a new approach, called SVF, to tackle the inefficient use of the IP network bandwidth issue. The SVF approach utilizes the VoIP packet header fields that are unneeded for the VoIP applications, especially the point-to-point voice calls. In particular, these unneeded fields will be utilized to carry the VoIP packet payload temporally, till reaching the destination VoIP appliance. The performance evaluation has shown that the maximum number of calls, which can be carried in a specific length bandwidth, is increased by up to 28.3% when using the G.723.1 codec. In addition, the buffer utilization, which could reflect on the call quality, has been improved by up to 39.5% with the same codec. Therefore, the SVF method is an auspicious approach to address the inefficient use of the IP network bandwidth. As future works, despite the fact that the NS2 simulation has been utilized in this study to deploy the proposed method, a real deployment environment in the presence of real network traffic, needs to be considered. In addition, the impact of altering the original value of the buffering fields on the security issues, such as the firewall rules and configuration, needs to be investigated.

References

1. Abualhaj, M. M. CA-ITTP: An Efficient Method to Aggregate VOIP Packets over ITTP Protocol. – International Journal of Innovative Computing, Information and Control, Vol. **15**, 2019, No 3, pp. 1067-1077.
2. Zhang, Y., K. Xie, O. O. Ruan. An Improved and Efficient Mutual Authentication Scheme for Session Initiation Protocol – PloS ONE, Vol. **14**, 2019, No 3, e0213688.
3. Abualhaj, M., M. Kolhar, K. Qaddoum, A. A. Abu-Shareha. Multiplexing VoIP Packets over Wireless Mesh Networks: A Survey. – KSII Transactions on Internet and Information Systems, Vol. **10**, 2016, No 8, pp. 3728-3752.
4. Affonso, E. T., R. D. Nunes, R. L. Rosa, G. F. Pivarro, D. Z. Rodriguez. Speech Quality Assessment in Wireless VoIP Communication Using Deep Belief Network – IEEE Access, Vol. **6**, 2018, pp. 77022-77032.
5. Olariu, C., J. Fitzpatrick, Y. Ghamri-Doudane, L. Murphy. A Delay-Aware Packet Prioritisation Mechanism for Voice over IP in Wireless Mesh Networks. – In: 2016 IEEE Wireless Communications and Networking Conference, April, 2016, pp. 1-7.
6. Shambour, Q., S. N. Alkhatib, M. M. Abualhaj, Y. Alrabanah. Effective Voice Frame Shrinking Method to Enhance VoIP Bandwidth Exploitation – International Journal of Advanced Computer Science and Applications (IJACSA), Vol. **11**, 2020, No 7, pp. 313-319.
7. Atanasov, I. Study on Deployment of Web Services for User Interaction in Multimedia Networks. – Cybernetics and Information Technologies, Vol. **13**, 2013, No 2, pp. 63-74.
8. Thejashwini, S., M. S. Kumar, S. A. Alex. IMS Based Session Initiation Protocol in Robot Framework for Telephony Services. – In: 2018 International Conference on Inventive Research in Computing Applications (ICIRCA'18), IEEE, 2018.
9. Rozhon, J., E. Gresak, J. Jalowiczor. Using LSTM Cells for SIP Dialogs Mapping and Security Analysis. – In: 2018 26th Telecommunications Forum (TELFOR'18), IEEE, 2018.
10. Abualhaj, M. M., S. N. Al-Khatib, Q. Y. Shambour. PS-PC: An Effective Method to Improve VoIP Technology Bandwidth Utilization over ITTP Protocol – Cybernetics and Information Technologies, Vol. **20**, 2020, No 3, pp. 147-158.
11. Gowd, B. R., N. Kumari, S. Thenappan. Real Time Implementation of Multimedia Traffic Unicast and Multicast Groups Using Optical Network. – In: 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT'16), IEEE, 2016.
12. Hussein, A. H., M. Abu-Alhaija, K. Nairoukh. New RTP Packet Payload Shrinking Method to Enhance Bandwidth Exploitation over RTP Protocol – International Journal of Advanced Computer Science and Applications (IJACSA), Vol. **11**, 2020, No 8, pp. 139-142.
13. Abualhaj, M. M., M. M. Al-Tahrari, S. N. Al-Khatib. A New Method to Improve Voice over IP (VoIP) Bandwidth Utilization over Internet Telephony Transport Protocol (ITTP). – In: Proc. of 2019 8th International Conference on Software and Information Engineering, April 2019, pp. 192-195.
14. Gupta, N., N. Kumar, H. Kumar. Comparative Analysis of Voice Codecs over Different Environment Scenarios in VoIP. – In: 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS'18), IEEE, June 2018, pp. 540-544.
15. Ortega, M. O., G. C. Altamirano, C. L. Barros, M. F. Abad. Comparison between the Real and Theoretical Values of the Technical Parameters of the VoIP Codecs. – In: 2019 IEEE Colombian Conference on Communications and Computing (COLCOM'19), IEEE, June 2019, pp. 1-6.
16. Cheng-Yu, Y., Y.-h. Zhong. An Efficient Algebraic Codebook Search for G.729 Speech Codec. – In: 2014 World Congress on Computer Applications and Information Systems (WCCAIS'14). IEEE, 2014, pp. 1-4.
17. Vulkan, C., A. Rakos, Z. Vincze, A. Drozdy. Reducing Overhead on Voice Traffic. U.S. Patent 8,824,304, Issued 2 September, 2014.
18. Tomsho, G. Guide to Networking Essentials. Cengage Learning, 8th Ed. 2019.

19. Hartpence, B. Packet Guide to Voice over IP: A System Administrator's Guide to VoIP Technologies. O'Reilly Media, Inc., 2013.
20. Perkins, C. RTP: Audio and Video for the Internet. Addison-Wesley Professional, 2003.
21. Gao, J., Y. Li, H. Jiang, L. Liu, X. Zhang. An RTP Extension for Reliable User-Data Transmission over VoIP Traffic. – In: W. Meng, S. Furnell, Eds. Security and Privacy in Social Networks and Big Data. SocialSec 2019. – Communications in Computer and Information Science, Vol. **1095**, Springer, 2019.
22. Seytnazarov, S., K. Young-Tak. QoS-Aware Adaptive A-MPDU Aggregation Scheduler for Enhanced VoIP Capacity over Aggregation-Enabled WLANs. – In: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2018.
23. Niu, Y., C. Wu, L. Wei, B. Liu, J. Cai. Backfill: An Efficient Header Compression Scheme for OpenFlow Network with Satellite Links. – In: 2016 International Conference on Networking and Network Applications (NaNA'16), IEEE, 2016, pp. 202-205.
24. Spencer, M., et al. Iax: Inter-Asterisk Exchange Version 2" IETF RFC 5456, 2010.
25. Salvador, P., V. Mancuso, P. Serrano, F. Gringoli, 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**, 2013, No 7, pp. 1640-1652.
26. Abu-Alhaj, M. M. Internet Telephony Transport Protocol (ITTP): An Efficient Transport Protocol for Voip Applications. Dissqution, Universiti Sains Malaysia, 2011.
27. Charfi, E., C. Gueguen, L. Chaari, B. Cousin, L. Kamoun. Dynamic Frame Aggregation Scheduler for Multimedia Applications in IEEE 802.11n Networks – Transactions on Emerging Telecommunications Technologies, Wiley-Blackwell, 2016, pp. 1-17.
28. Sandlund, K., G. Pelletier, J. Le. The Robust Header Compression (RoHC) Framework. 2010.
29. Sze, H. P., S. C. Liew, J. Y. Lee, D. Yip. A Multiplexing Scheme for H. 323 Voice-over-IP Applications – Selected Areas in Communications, IEEE Journal, Vol. **20**, 2002, No 7, pp. 1360-1368.
30. Abu-Alhaj, M. M., A. Mnasrah, M. Balkizi, N. Abdullah, L. V. Chandra. Transport Layer Protocols Taxonomy from Voice over IP Perspective. – Advanced Computing: An International Journal, Vol **2**, 2011, No 4.
31. Boucadair, M. Inter-Asterisk Exchange (IAX): Deployment Scenarios in SIP-Enabled Networks. 1st Ed. John Wiley & Sons, Ltd., 2009.
32. Abualhaj, M. M., S. N. Al-Khatib, M. Kolhar, A. Munther, Y. Alraba'nah. Effective Voice Frame Pruning Method to Increase VoIP Call Capacity – TEM Journal, Vol. **9**, 2020, No 1, pp. 48-54.
33. https://www.juniper.net/documentation/en_US/junos/topics/reference/general/interfaces-media-mtu-size-by-interface-type.html
34. Sado, E., I. E. Zilbershtein. Packet Fragmentation Prevention. U.S. Patent No 9,088,494. 21 July 2015.
35. <https://www.geeksforgeeks.org/fragmentation-network-layer/?ref=lbp>
36. Kozierok, C. M. The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference. No Starch Press, 2005.
37. Lars, E., G. Fairhurst, G. Shepherd. UDP Usage Guidelines. IETF RFC 8085, 10 March 2017.
38. Khan, S. A., M. Moosa, F. Naeem, M. H. Alizai, J. M. Kim. Protocols and Mechanisms to Recover Failed Packets in Wireless Networks: History and Evolution – IEEE Access, Vol. **4**, 2016, pp. 4207-4224.
39. Stein, S., E. Rippel. Efficient Double Parity Forward Error Correction on a Communication Network. U.S. Patent No 10,567,102. 18 February 2020.

Received: 21.10.2020; Second Version: 30.12.2020; Accepted: 11.01.2021 (fast track)