BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **21**, No 4
Sofia • 2021 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2021-0048

An Efficient Method to Enhance IP Telephony Performance in IPV6 Networks

Mosleh M. Abualhaj[1], *Sumaya N. Al-Khatib*[2], *Qusai Y. Shambour*[3], *Ahmad Adel Abu-Shareha*[2]

[1]*Department of Networks and Information Security, Al-Ahliyya Amman University, Amman, Jordan*
[2]*Department of Computer Science, Al-Ahliyya Amman University, Amman, Jordan*
[3]*Department of Software Engineering, Al-Ahliyya Amman University, Amman, Jordan*
E-mails: m.abualhaj@ammanu.edu.jo sumayakh@ammanu.edu.jo q.shambour@ammanu.edu.jo a.abushareha@ammanu.edu.jo

Abstract**:** *IP telephony have played an essential role during the COVID 19 pandemic lockdown. One of the issues that lower the service level of the IP telephony solutions is the inefficient bandwidth exploitation. This paper proposes a Smallerize/Zeroize* (*SmlZr*) *method to enhance bandwidth exploitation. The SmlZr method is explicitly designed for the P2P IP telephony calls over IPv6 networks. The essence concept of the proposed method is to use the unnecessary fields in the header to keep the voice media of the packet. Doing so leads to smallerize or zeroize the packet payload and, thus, enhance the bandwidth exploitation. The SmlZr method has outperformed the RTP method for all the comparison parameters. For instance, the SmlZr method shrinks the bandwidth by 25% compared to the RTP protocol. Bandwidth saving is helpful for P2P IP telephony calls because it alleviates the traffic load. Thus, improve the call capacity boosts the call clarity.*

Keywords**:** *IP telephony, bandwidth exploitation, voice codec, IPv6.*

1. Introduction

In the first half of 2020, the COVID 19 pandemic has forced a lockdown in most countries worldwide. Many sectors have changed to virtual communications, including on-site businesses and educations, to keep functioning [1, 2]. Video conferencing, real-time online education, and IP telephony (i.e., VoIP) have become essential solutions in organizations and personal lives. The forecasting for IP telephony solutions is to have 3 billion users in 2021 [3]. One of the vital reasons behind spreading IP telephony is the significantly lesser price or free calls. Accordingly, the performance of the IP telephony solutions should be risen to reach the user's satisfaction level. Bandwidth exploitation of IP telephony networks is one of the issues that should be discussed to raise the level of IP telephony performance.

Improving bandwidth exploitation also impacts on the clarity of the IP telephony calls [4].

Typically, two principal stages are carried out to establish a call when using IP telephony solutions [5]. The principal stage requires that one side of the call starts a call with another side of the call to confirm one another and concur upon specific parameters needed to settle on an effective IP telephony call, including the utilized IP telephony codec. To accomplish this, IP telephony solutions utilize signalling protocols [5, 6]. SIP is the prevailing signalling protocol because of its effortlessness, adaptability, extensibility, and particular highlights that enhance IP telephony solutions [7, 8]. In the subsequent stage, after establishing the call, the media speech information starts moving between the two parties of the call. IP telephony systems utilize media transport protocols such as the RTP protocol [5, 9].

IP telephony systems encapsulate the speech media by RTP, UDP, and IPv4/IPv6 protocols. The size of the RTP, UDP, and IPv4/IPv6 protocols is 12-byte, 8-byte, 20/40-byte, respectively. While the size of the speech media produced by the IP telephony codec is between 10-byte and 30-byte. Clearly, the size of the three protocols (RTP, UDP, and IPv4/IPv6) is considerably larger compared to the speech media samples. IP telephony systems are sometimes encapsulated and several speech samples may be in one packet's payload. However, the encapsulated samples do not exceed a certain size to avoid increasing the packetization delay and, thus, reduce the call clarity [10, 11]. Table 1 shows the default speech sample size along with the typical packet payload for the well-known codecs. Still, in the three protocols the packet headers are significant compared to the packet payload (multiple samples). Table 2 shows the bandwidth consumed by the packet header compared to the packet payload in IPv6 networks. The bandwidth consumed by the packet header is calculated by dividing the header size on the total packet size. [10, 12, 13]. As we can see, the wasted bandwidth when using the IPv6 protocol is considerably large.

Table 1. Speech codecs

Codec	Speech Sample Size	Typical Payload Size	Mean Opinion Score
G.729	10	20	4.1
G.728	10	60	3.61
G.723.1	20	20	3.8
G.726	20	80	3.85

Table 2. Consumed bandwidth by packet header

Codec	Common Payload Size	Packet Size	Header Consumed Bandwidth
G.729	20	80	(80 – 20) / 80 = 75.0%
G.728	60	120	(120 – 60) / 120 =50.0%
G.723.1	20	80	(80 – 20) / 80 =75.0%
G.726	80	140	(140 – 80) / 140 =42.9%

The main reason for the large header is that the RTP protocol contains the necessary information to convey data for all types of real-time applications, including video conferencing, webcasting, and IP telephony applications [14, 15]. Besides, the UDP and IPv6 protocols contain the necessary information to convey data for all applications that run over IP networks [16]. Therefore, a considerable part of the RTP/UDP/IPv6 information is extra for the IP telephony applications [17-19]. This paper will analyze the information in the RTP/UDP/IPv6 protocols to find extra information for IP telephony applications. This paper mainly concerns the point-to-point (P2P) IP telephony calls (the calls between only two clients) over IPv6 networks. All information in the RTP/UDP/IPv6 protocols that are extra for P2P calls will enhance bandwidth exploitation. In fact, the fields of the extra information in the RTP/UDP/IPv6 protocols will be exploited to carry the speech media. This will smallerize or zeroize the packet payload and, thus, enhance the bandwidth exploitation of IP telephony solutions.

This paper is organized in five sections. Section 2 analyses the key approaches related to bandwidth exploitation of IP telephony. Section 3 discusses the main idea of the proposed method and its main modules. Section 4 evaluates the proposed method current protocols that are used to carry the real-time voice media. Section 5 summarizes the finding of this study.

2. Related works

Methods for enhancing the bandwidth exploitation of IP telephony can be divided into two groups according to their primary working mechanism. Namely, the packet multiplexing group and header compression group. As stated, each IP telephony packet contains a sizable header that drains the bandwidth. Therefore, the packet multiplexing methods pack several IP telephony packets in one header. The common property among the packed packets is that they share one traveling route to the destination. Accordingly, the packet multiplexing methods will highly alleviate draining the bandwidth by the IP telephony packet header. Packing more packets in one header gives better bandwidth exploitation. However, the number of the packed packets is constrained by several parameters such as the maximum transmission unit, allowable IP telephony delay, and other parameters that might impact the clarity of the call. A considerable number of methods have been designed under the packet multiplexing group. Each of which has been designed to work in certain environments and use different parameters to control the packed packets' number [20-25]. For instance, S. Seytnazarov and K. Young-Tak [20] have designed a multiplexing method for IEEE 802.11n wireless networks. The method works at layer 2 of the OSI model, in which the IP telephony frames are multiplexed in one Aggregation MAC Protocol Data Unit (A-MPDU). If one of the frames within the (A-MPDU) has been corrupted, only the corrupted frame is sent again. In addition, the method controls the size of the multiplexed frames based on the channel load, the reported delay from the RTP/RTCP protocols, buffering delay, 150 ms delay, and the average delay to access the medium. Besides, a specified access category is used to cluster the IP telephony frames. The simulation result has shown that the method is providing better performance than the comparable methods. For instance, the performance has been improved by 160% than the comparable methods in the tested scenarios. Nevertheless, the multiplexing group suffers from several handicaps. First, the packets from different sessions are multiplexed in a single chunk. The multiplexed packets will have the same service while going on the network. As a matter of fact, some sessions should have better service than other sessions. Second, IP telephony applications use specific mechanisms to conceal the lost packet and improve call clarity. The concealment mechanisms are effective with the typical small IP telephony packets. However, the concealment mechanisms will not be effective with a large chunk of several packets. Thus, the lost packet will not be concealed, and the call clarity will be degraded. Third, multiplexing more packets in one chunk will give a better bandwidth exploitation efficiency. In case of a small number of sessions, the packets should wait in the buffer until enough packets arrived to be multiplexed. In addition, the multiplexing process itself takes time depending on the utilized method. Therefore, the waiting time in the buffer along with the multiplexing process time will force some delay and, thus, impact on the call clarity. Finally, the multiplexing process consumes the resources of the multiplexing device [4, 26, 27, 28].

The second group to handle the sizable header of the IP telephony packets is header compression. In general, the header compression methods utilize two main features in the IP telephony packets to alleviate the header size. The first feature is based on the "no change" fields in the header, and the other one is based on the steadily increasing fields. These two features have compressed the header into 4-byte when activating the Checksum field or 2 bytes when disabling the Checksum field [25, 29, 30]. The Robust Header Compression (RoHC) is a variant of the traditional header compression approach, which is more suitable for specific IP telephony applications [19]. Pedro Fortuna and Manuel Ricardo [31] have designed a model to investigate the performance of RoHC when running IP telephony over 802.11 channels. The RoHC U-mode has been chosen in the design model. Besides, a new element called RoHCGain has been suggested to measure the amount of additional bandwidth other streams can utilize due to the use of RoHC with IP telephony traffic. The investigation results showed that RoHC is applicable only when 802.11 links are congested or transferring greedy flows [19, 31]. Regardless of the utilized method, the header compression group suffers from several handicaps as well. First, the header compression methods are not working well in high packet loss or long round trip time. Second, header compression contains many complex operations at the compression and decompression devices. These operations overwhelm the compression/decompression devices and waste their resources. In addition, performing the compression/decompression operations on the packet will impose a new source of delay on IP telephony applications [30-33].

Apart from packet multiplexing and header compression groups, the researchers have proposed to replace the current IP telephony transport protocol with a new dedicated one [9,18]. One of the prominent transport protocols for IP telephony calls is the Inter-Asterisk Exchange (IAX) protocol. One of the design goals of the IAX is to alleviate the drained bandwidth from the dominating RTP IP telephony transport protocol. The IAX protocol imposes a 4-byte header to the IP telephony packet, while the RTP imposes a 12-byte. Therefore, the bandwidth drained by the header has been reduced by 13.3% when utilizing IAX in place of RTP, assuming the IPv6 protocol is used. Though 13.3 % is a good saving of the bandwidth, the IP telephony packet header is still draining plenty of bandwidth when using the IAX protocol. For instance, assume the G.728 codec is used with a standard packet payload, as shown in Table 1. Then, the IP telephony packet header consumes 46.4% when using the IAX/UDP/IPv6 header. More importantly, the IAX protocol is designed to be a signalling and media transfer protocol by itself. In other words, the IAX protocol is not compatible with the dominating signalling protocol, namely SIP and H.323. Therefore, the chances of deploying the IAX protocol in IP telephony solutions are limited [18, 30, 34].

In summary, the packet multiplexing and header compression approaches are achieving good bandwidth exploitation for IP telephony solutions. However, as stated, they impose many problems and are not suitable for many scenarios and environments. Though IAX solves some packet multiplexing and header compression problems: i) the wasted bandwidth when using IAX is unacceptable, and ii) the spreading of the IAX protocol is minimal. Accordingly, a new approach will emerge in this paper to handle the bandwidth exploitation problem of IP telephony solutions. The new approach should be workable with current standards protocols. In addition, it should exploit the bandwidth of IP telephony systems while handling or at least alleviating the problems of the packet multiplexing approach and header compression approach. The new approach will be designed specifically for the P2P IP telephony calls over IPv6 networks. The fields of the extra information in the header will be exploited to carry the speech media. Thus, the packet payload will be smallerized or zeroized and, thus, the bandwidth exploitation will be enhanced. The new approach, called Smallerize/Zeroize (SmlZr), will be discussed in the next section in detail.

3. Smallerize/Zeroize (SmlZr) method

The purpose of building the SmlZr method is to enhance the bandwidth exploitation of P2P IP telephony calls over IPv6 networks. The SmlZr method is assumed to be deployed at the IP telephony user agent (UA) such as KPhone. The SmlZr method includes a module that works at the sender UA (called Sender SmlZr [SmlZr-S] module), and another one works at the receiver UA (called Receiver SmlZr [SmlZr-R] module). After creating the IP telephony packet, the SmlZr-S module will interfere with moving the packet payload (speech media) to the extra fields in the packet header (RTP/UDP/IPv6 protocols). The detail of the SmlZr-S module is discussed in Section 3.2. The extra fields are addressed in Section 3.1. Upon arriving at the SmlZr-R, the speech media will be extracted from the header of the IP telephony packet and placed in the packet payload. The detail of the SmlZr-R module is discussed in Section 3.3. The proposed SmlZr method is independent from the type of the network because it is deployed at the UA. Fig. 1 shows a scenario at which the SmlZr method could be implemented.

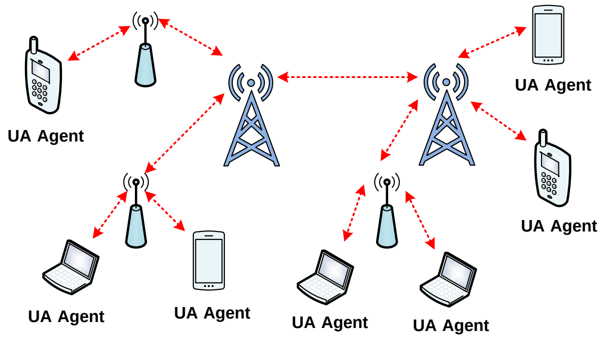

Fig. 1. SmlZr method topology

3.1. Analyzing the RTP/UDP/IPv6 fields

The primary source of the drained bandwidth in IP telephony is the large RTP/UDP/IPv6 header of the packets. As stated, most of these fields are extra for the P2P IP telephony calls. This section will analyze the RTP/UDP/IPv6 header to find the extra fields. The SmlZr method will use the extra fields to carry the speech media of the IP telephony packet.

The first header of the IP telephony packet is the IPv6, which is used with all applications in the IP-based networks. The Source IPv6 (SIPv6) address information is not needed by IP telephony applications because IP telephony applications are not "request/response" applications [16, 35]. In fact, the call ends knew the IPv6 address of each other during the call setup. While making the conversation, the IP telephony packet is transmitted to the already known IPv6 of the destination, not as a reply to the received packets [30, 36]. Therefore, the field that carries the SrcIPv6 address information is extra for the P2P IP telephony calls. The information in the Next Header (NH) field is necessary to identify the upper-layer protocol with all applications, including IP telephony. However, the upper layer protocol in IP telephony is always UDP with the value 17 in the NH field [30, 36]. Therefore, SmlZr-S module can use the NH field to keep the voice media, and the SmlZr-R module can set back the NH field to 17 upon receiving the packet.

The second header of the IP telephony packet is the UDP, which is used with many applications in IP-based networks. The Source Port (SP) information is not needed by IP telephony applications for the same reasons as the SIPv6 address. In fact, the SP field is optional, and it can be disabled by many applications, including IP telephony. The Checksum (Ch) field is also optional, and it can be disabled by many applications, including IP telephony. The Length (Ln) field information is necessary to determine length of the layer 4 segment, including the UDP header. However, the Ln field is equal to the Payload Length field in the IPv6 header in case of IP telephony. Therefore, SmlZr-S module can use the Ln field to keep the voice media, and the SmlZr-R module set back the Ln field based on the value of the Payload Length [16, 30, 35, 37].

The third header of the IP telephony packet is the RT. The RTP header contains the necessary information for various types of real-time multimedia applications in IP-based networks. In P2P IP telephony calls, not all the information in the RTP header is essential. The information in the Synchronization Source (SSRC) field is used to identify the source. The SSRC assists in fixing the conflict when the initial value of the sequence number is the same for two sources. However, there is one source in the case of P2P calls [15, 17, 30]. Thus, the SSRC field is extra and can be used to carry the voice media of the IP telephony packets.

Only the fields that can be used to carry the voice media are discussed above. All the other information in the RTP/UDP/IPv6 headers is needed to convey the P2P call packets in different network types and with various UA applications. Accordingly, the SIPv6, NH, SP, Ch, Ln, and SSRC fields can be utilized by the SmlZr to carry the voice payload of the P2P calls packets. The total size of these fields is 27-byte. Therefore, the SmlZr method will smallerize or zeroize the payload of the P2P calls packets and, thus, improve the bandwidth exploitation.

3.2. The SmlZr-S module

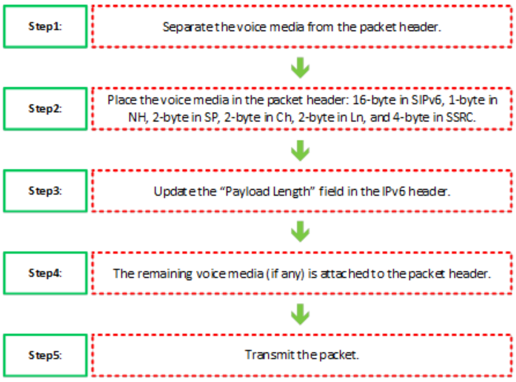

Fig. 2. SmlZr-S module operations

The primary purpose of the SmlZr-S module is to move the voice media (up to 27-byte) of a packet into the packet header. The function performed by SmlZr-S module goes through several steps, as shown in Fig. 2. In Step2, if the voice media is greater than 27-byte, then the residual voice media is kept as a packet payload. For instance, the typical packet payload when using the G.728 codec is 60-byte. Therefore, the residual voice media is 33-byte (60 – 27 = 33) as a packet payload. In Step3, the "Payload Length" field in the IPv6 header is updated because the size of the voice media encapsulated in the packet has been shortened. Therefore, the "Payload Length" must have the new size of the payload in order to process the packet correctly by the network devices. Fig. 3 shows the process of updating the "Payload Length" field.

SmlZr-S Module → Updates the "Payload Length"

```
//Lvm: is the length of the voice media //Integer
//20: is 12-byte RTP + 8-byte UDP
//27: is the size of the extra fields in the header

If Lvm > 27 Then
   Payload Length = 12 (RTP) + 8 (UDP) + (Lvm - 27)
else
   Payload Length = 12 (RTP) + 8 (UDP)
```

Fig. 3. Updating the "Payload Length" field

3.3. The SmlZr-R module

The primary purpose of the SmlZr-R module is to move the voice media from the packet header into the packet payload. The function performed by SmlZr-R module goes through several steps, as shown in Fig. 4. In Step1, the SmlZr-R should checks the size of the voice media to extract the exact voice media from the packet header, as shown in Fig. 5. The size of the voice media is negotiated during the call setup based on the used codec (discussed in Section1). In Step3, the “Payload Length”, NH, Ch, Ln fields are updated. The “Payload Length” = is equal to the voice media length plus 20-byte (RTP/UDP), NH is equal to 17, Ch is equal to zero, and Ln is equal to “Payload Length”. The Ch field is set to zero to avoid misinterpretation by the UA. The SIPv6 and SP fields are simply ignored because they are not needed by the UA to process the packet correctly.

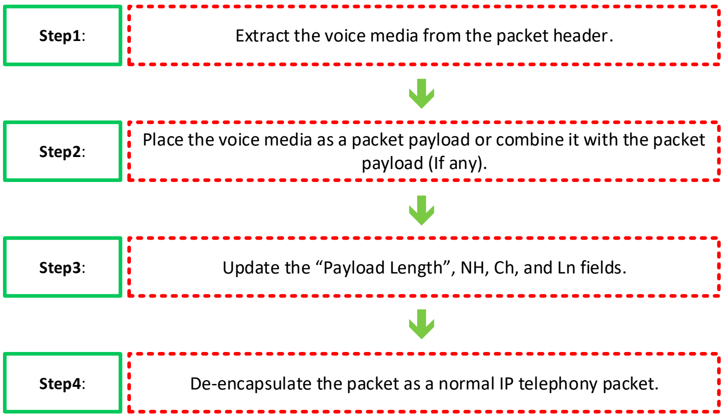

Fig. 4. SmlZr-R module operations

SmlZr-R Module → Extract the Voice Media

```
//Lvm: is the length of the voice media // Integer
//27: is the size of the extra fields in the header // Integer
//Vm: is voice media stored in the header // String
//Lvmh: is the length of the voice media stored in the header // Integer

If Lvm > 27 Then {
   Lvmh = Lvm – 27
else
   Lvmh = Lvm
}
Vm = extract Lvmh byte from the extra fields of the header
```

Fig. 5. Extract the voice media

4. The SmlZr performance evaluation

This section evaluates the success of the proposed SmlZr method. The SmlZr method has been evaluated against to the IAX/UDP/IPv6 method (IAX protocol) and the common RTP/UDP/IPv6 method (RTP protocol) of conveying the IP telephony traffic. The SmlZr method, IAX protocol, and the RTP protocol have been compared in three criteria, namely, the call capacity, saved bandwidth ratio, and buffer utilization enhancement. To make a reasonable comparison, each of these three criteria has been tested with three different codecs, G.726 G.728, and G.723.1.

4.1. Call capacity

This section evaluates the call capacity of the proposed SmlZr method against to the IAX protocol and RTP protocol. The call capacity has been evaluated at various link bandwidths from 100 kbps to 1000 kbps. Figs 6, 7, 8 present the call capacity of the SmlZr method against that of the IAX protocol and RTP protocol using G.726 G.728, and G.723.1, respectively. Clearly, with the three codecs, the proposed SmlZr method outperformed the IAX and RTP protocols. For instance, the call capacity is 31, 33, and 40 when running the IAX protocol, RTP protocol, and SmlZr method, respectively (Assuming G.728 codec at 1000kbps bandwidth). Clearly, this improvement in call capacity is keeping up to 27-byte of the voice media in the packet header. Additionally, the change in the call capacity between the SmlZr method, IAX protocol, and RTP protocol differs from one codec to another. This is because the ratio of the voice media in the packet header to the full packet size varies when diverse codecs are utilized.

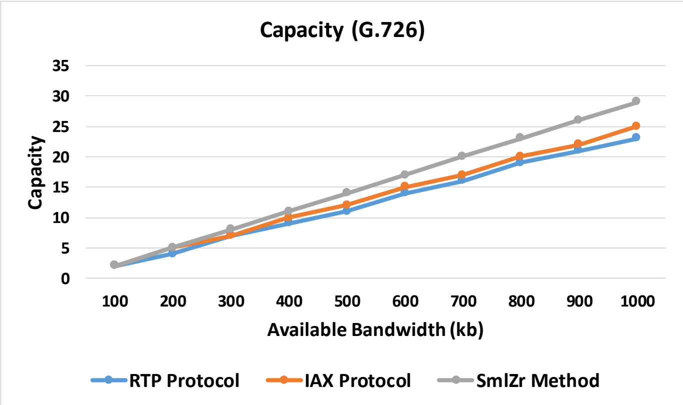

Fig. 6. Call capacity (G.726)

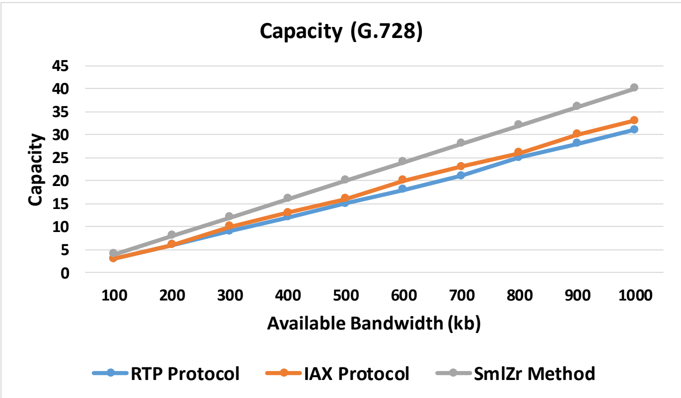

Fig. 7. Call capacity (G.728)

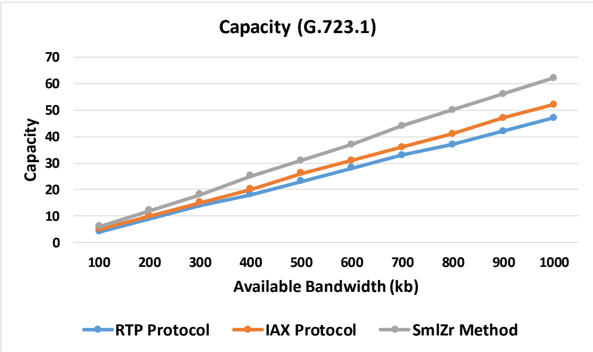

Fig. 8. Call capacity (G.723.1)

4.2. Saved bandwidth ratio

This section evaluates the saved bandwidth ratio of the proposed SmlZr method against to the IAX protocol RTP protocol. Fig. 9 presents the saved bandwidth ratio when utilizing the SmlZr method against the IAX protocol and RTP protocol, based on the call capacity, with the three codecs. Using the SmlZr method, the saved bandwidth ratio surpasses that of the IAX protocol and RTP protocol with the three codecs. For instance, the proposed SmlZr method saves the bandwidth by 25% compared to the RTP protocol, while the IAX protocol saves the bandwidth by 10% only (Assuming G.723.1 codec). Clearly, the cause of this improvement in saved bandwidth ratio is keeping up to 27-byte of the voice media in the packet header. Additionally, the change in the call capacity between the SmlZr method, IAX protocol, and RTP protocol differs from one codec to another. This is because the ratio of the voice media in the packet header to the full packet size varies when diverse codecs are utilized.

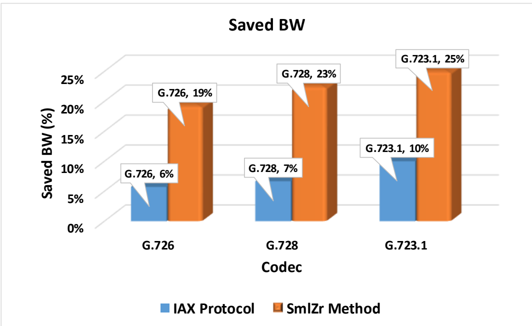

Fig. 9. Saved bandwidth ratio

4.3. Buffer utilization enhancement

This section examines the buffer utilization enhancement of the proposed SmlZr method contrary to the IAX protocol RTP protocol, with the G.726 G.728, and G.723.1 codecs. When the quantity of received traffic is more than the buffer size, then the received traffic is lost. Thus, the loss ratio and the delay are rises. Equation (1) can be utilized to find the quantity of received traffic the buffer can keep

(1) $$B_S = \frac{B_S}{\mathrm{Pkt_l}},$$

where B_s is the buffer capacity in a packet, is the buffer size in bytes and Pkt_l is the packet length in byte. Equations (2) and (3) can be utilized to find the buffer utilization enhancement ratio when running the proposed SmlZr method against RTP and IAX protocols, respectively

(2) $$\text{RTP Improvement } \% = \frac{\text{SmlZr}_{\text{bc}} - \text{RTP}_{\text{bc}}}{\text{RTP}_{\text{bc}}} * 100\%,$$

(3) $$\text{IAX Improvement } \% = \frac{\text{SmlZr}_{\text{bc}} - \text{IAX}_{\text{bc}}}{\text{IAX}_{\text{bc}}} * 100\%,$$

where SmlZr_{bc} is the SmlZr method buffer capacity in a packet, IAX_{bc} is the IAX protocol buffer capacity in a packet, RTP_{bc} is the RTP protocol buffer capacity in a packet. For instance, assuming that the buffer capacity is 1000 bytes, and utilizing Equations (1), (2), and (3). Fig. 10 presents that the SmlZr method improves the buffer utilization over IAX protocol and RTP protocol with G.726, G.728, and G.723.1 codecs. For instance, the SmlZr method enhances the buffer utilization by 33.3% compared to the RTP protocol, while the IAX protocol enhances the buffer utilization by 11.1% only (Assuming G.723.1 codec). This leads to less packet loss and delay and boosts call clarity. The reason behind this enhancement is the same as the saved bandwidth ratio in Section 4. Additionally, the change in the enhancement of buffer utilization ratio between the SmlZr, IAX protocol, and RTP protocol differs from one codec to another, for the same reasons of the saved bandwidth ratio, in Section 4.

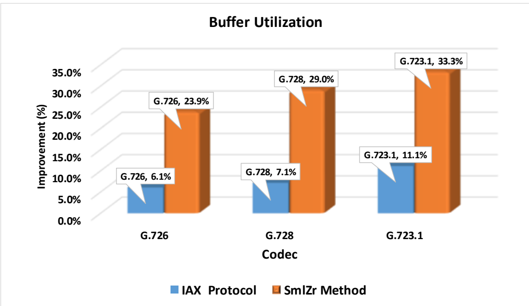

Fig. 10. Buffer utilization enhancement

5. Conclusion

This paper proposes the SmlZr method to enhance the bandwidth exploitation of IPv6 networks when running P2P calls. The SmlZr method saves the bandwidth by exploiting the unnecessary fields of the large RTP/UDP/IPv6 header of the packet. These fields are used to shorten the packet payload by carrying the voice media of the packet. The SmlZr method has been evaluated against the IAX protocol and the RTP protocol regarding the call capacity, saved bandwidth ratio, and buffer utilization enhancement criteria. The SmlZr method has shown a considerable improvement in the three criteria against the comparable protocols. For instance, the proposed SmlZr method saves the bandwidth by 25% compared to the RTP protocol. Therefore, the network load is reduced, theoretically, by 25%, which reduces the delay and packet loss and reflects the call quality. In addition, the SmlZr method modules (SmlZr-S and SmlZr-R modules) are implemented at the UA, with no change required to the network devices. Accordingly, The SmlZr method is entirely compatible with existing IP telephony solutions and network devices and can be readily deployed.

References

1. Raycheva, L., N. Velinova, N. Miteva, M. Tomov. Impacts of Virtual Communication During Social Isolation of Covid'19. – In: Proc. of International Conference of Human Systems Engineering and Design: Future Trends and Applications, Springer, 2020, pp. 63-68.
2. Garro-Abarca, V., P. Palos, M. Aguayo-Camacho. Virtual Teams in Times of Pandemic: Factors That Influence Performance. – Frontiers in Psychology, Vol. **12**, 2021, p. 232.
3. Hoffman, J. VOIP Adoption Statistics for 2019 & Beyond. **https://wisdomplexus.com/blogs/voip-adoption-statistic-2019-beyond/**
4. Abualhaj, M. M., A. H. Hussein, M. Kolhar, M. A. Al Hija. Survey and Analysis of VoIP Frame Aggregation Methods over A-MSDU IEEE 802.11n Wireless Networks. – Comput. Mater. Contin, Vol. **66**, 2020, pp. 1283-1300.
5. 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.
6. 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.
7. Thejashwini, S., M. S. Kumar, S. A. Alex. IMS Based Session Initiation Protocol in Robot Framework for Telephony Services. – In: Proc. of 2018 International Conference on Inventive Research in Computing Applications (ICIRCA'18), 2018, IEEE, pp. 1218-1223.
8. Rozhon, J., E. Gresak, J. Jalowiczor. Using LSTM Cells for SIP Dialogs Mapping and Security Analysis. – In: Proc. of 26th Telecommunications Forum (TELFOR'18), 2018, IEEE, pp. 1-4.
9. 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.
10. Abualhaj, M. M., M. M. Al-Tahrawi, M. Al-Zyoud. Contracting VoIP Packet Payload Down to Zero. – Cybernetics and Information Technologies, Vol. **21**, 2021, No 1.
11. Narayan, S., M. Gordon, C. Branks, Li Fan. VoIP Network Performance Evaluation of Operating Systems with IPv4 and IPv6 Network Implementation. – In: Proc. of 3rd International Conference on Computer Science and Information Technology, Vol. **5**, IEEE, 2010, pp. 669-673.
12. Tomoskozi, M., P. Seeling, P. Ekler, F. H. Fitzek. Regression Model Building and Efficiency Prediction of Rohcv2 Compressor Implementations for Voip. – In: Proc. of 2016 IEEE Global Communications Conference (GLOBECOM'16), IEEE, 2016, pp. 1-6.
13. Gupta, N., N. Kumar, H. Kumar. Comparative Analysis of Voice Codecs over Different Environment Scenarios in VoIP. – In: 2nd International Conference on Intelligent Computing and Control Systems (ICICCS'18), IEEE, 2018, pp. 540-544.
14. Hartpence, B. Packet Guide to Voice over IP: A System Administrator's Guide to VoIP Technologies. O'Reilly Media, Inc., 2013.
15. Gao, J., Y. Li, H. Jiang, L. Liu, X. Zhang. An RTP Extension for Reliable User-Data Transmission over VoIP Traffic. – In: International Symposium on Security and Privacy in Social Networks and Big Data, Singapore, Springer, 2019, pp. 74-86.
16. Tomsho, G. Guide to Networking Essentials. Cengage Learning, 2012.
17. Perkins, C. RTP: Audio and Video for the Internet. Addison-Wesley Professional, 2003.
18. Spencer, M., B. Capouch, E. Guy, F. Miller, K. Shumard. IAX: Inter-Asterisk Exchange Version 2. IETF RFC 5456, 2010.

19. S a n d l u n d, K., G. P e l l e t i e r, J. L e. The Robust header compression (Rohc) Framework, RFC 5795, March, 2010.
20. S e y t n a z a r o v, S., K. Y o u n g-T a k. Qos-Aware Adaptive a-Mpdu Aggregation Scheduler for Voice Traffic in Aggregation-Enabled High Throughput WLANs. – Journal of IEEE Transactions on Mobile Computing, Vol. **16**, 2017, No10, pp. 2862-2875.
21. A b u a l h a j, M. M., M. K o l h a r, K. Q a d d o u m, A. A. A b u-S h a r e h a. Multiplexing VoIP Packets over Wireless Mesh Networks: A Survey. – KSII Transactions on Internet and Information Systems (TIIS), Vol. **10**, 2016, No 8, pp. 3728-3752.
22. V u l k a n, C., A. R a k o s, Z. V i n c z e, A. D r o z d y. Reducing Overhead on Voice Traffic. United States Patent US 8,824,304, 2014.
23. R o a y, B. Generic UDP Multiplexing for Voice over Internet Protocol (VOIP). United States Patent US 8,553,692, Issued 8 October, 2013.
24. A b u a l h a j, M. M., Q. Y. S h a m b o u r, A. H. H u s s e i n. Effective Packet Multiplexing Method to Improve Bandwidth Utilization. – International Journal of Computer Applications in Technology, Vol. **63**, 2020, No 4, pp. 327-336.
25. S z e, H. P., S. C. L i e w, J. Y. L e e, D. C. Y i p. A Multiplexing Scheme for H. 323 Voice-over-IP Applications. – IEEE Journal on Selected Areas in Communications, Vol. **20**, 2002, No 7, pp. 1360-1368.
26. A b u a l h a j, M. M., M. K o l h a r, K. Q a d d o u m, A. A. A b u-S h a r e h a. Multiplexing VoIP Packets over Wireless Mesh Networks: A Survey. – KSII Transactions on Internet and Information Systems (TIIS), Vol. **10**, 2016, No 8, pp.3728-3752.
27. S a l v a d o r, P., V. M a n c u s o, P. S e r r a n o, F. G r i n g o l i, A. B a n c h s. 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.
28. C h a r f i, E., C. G u e g u e n, L. C h a a r i, B. C o u s i n, L. K a m o u n. Dynamic Frame Aggregation Scheduler for Multimedia Applications in IEEE 802.11n Networks. – Transactions on Emerging Telecommunications Technologies, Vol. **28**, 2017, No 2, p. e2942.
29. C a s n e r, S., V. J a c o b s o n. Compressing IP/UDP/RTP Headers for Low-Speed Serial Links. RFC 2508, 1999.
30. A b u a l h a j, M. M., Q. Y. S h a m b o u r, A. H. H u s s e i n, Q. M. K h a r m a. Down to Zero Size of VoIP Packet Payload. – CMC-Computers Materials & Continua, Vol. **68**, 2021, No 1, pp. 1271-1283.
31. F o r t u n a, P., M. R i c a r d o. Header Compressed VoIP in IEEE 802.11. – IEEE Wireless Communications, Vol. **16**, 2009, No 3, pp. 69-75.
32. P e r k i n s, C. RTP: Audio and Video for the Internet. Addison-Wesley Professional, 2003.
33. S a l d a n a, J., F. P a s c u a l, J. A. C a s t e l l, J. F e r n a n d e z-N a v a j a s, D. D. H o z, J. R u i z-M a s, M. N u ñ e z, D. R. L o p e z, D. F l o r e z. Small-Packet Flows in Software Defined Networks: Traffic Profile Optimization. (No ART-2015-91767), 2015.
34. A b u-A l h a j, M. M., A. M a n a s r a h, M. B a k l i z i, N. A b d u l l a h, L. V. C h a n d r a. Transport Layer Protocols Taxonomy from Voice over IP Perspective. – Advanced Computing: An International Journal, Vol. **2**, 2011, No 4.
35. K o z i e r o k, C. M. The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference. No Starch Press, 2005. ISBN-13:978-1593270476. ISBN-10:15932704X.
36. S i l v i a H a g e n. IPv6 Essentials: Integrating IPv6 into Your IPv4 Network. 3rd Edition. O'Reilly Media, 29 June, 2014.
37. S t e i n, S., E. R i p p e l. Efficient Double Parity Forward Error Correction on a Communication Network. United States Patent US 10,567,102, 2020.

Received: 27.09.2021; Second Version: 03.10.2021; Accepted: 12.10.2021 (fast track)