International Journal of Innovative Computing, Information and Control
Volume **15**, Number **3**, June **2019**

pp. 1067–1077

CA-ITTP: AN EFFICIENT METHOD TO AGGREGATE VOIP PACKETS OVER ITTP PROTOCOL

Mosleh M Abualhaj

Faculty of Information Technology
Al-Ahliyya Amman University
Amman 19328, Jordan
m.abualhaj@ammanu.edu.jo

Received July 2018; revised November 2018

Abstract. *Over the last twenty years, Voice over IP (VoIP) was engendered as a new technology to carry voice calls over IP networks. However, VoIP suffers from some problems that hinder its spreading. One of the key problems is the large header of the VoIP packet in comparison to the packet payload, which dissipates the network bandwidth. The main solution of the VoIP packet overhead is packet aggregation methods, in which several VoIP packets that share the same path are aggregated in one header. In this article, we have proposed a new method, named Compression-Aggregation over ITTP (CA-ITTP). As the name implies, CA-ITTP works with the ITTP protocol. The CA-ITTP consists of Sender-Compression-Aggregation (SCA) and Receiver-Compression-Deaggregation (RCD) components. SCA aggregates the packets at the sender side VoIP gateway. RCD de-aggregates the packets at the receiver side VoIP gateway. The results showed that the CA-ITTP nearly halves the bandwidth consumption, in comparison to the traditional ITTP without aggregation.*
Keywords: ITTP, VoIP protocols, Packet aggregation, Network bandwidth, Codec

1. **Introduction.** Over the last twenty years, Voice over IP (VoIP) was engendered as a new technology that carries voice calls over IP networks. VoIP was widely used and deployed in place of the legacy land line systems (Public Switched Telephone Network (PSTN)) [1,2]. The main driver of adopting VoIP technology is the very cheap call rate or even sometimes free calls. Another important driver is that we can make a call from almost anywhere to the other end of the world using any smart device, such as a mobile, iPad, and laptop [3,4]. On the other hand, VoIP suffers from some problems that hinder its spreading. The main problem is the low quality of the voice call in comparison to the traditional PSTN. Another noticeable problem is the large header of the VoIP packet in comparison to the packet payload [5,6]. Whereas, the typical packet payload is between 10 to 30 bytes, based on the used codec [7,8]. Table 1 shows some of the most well-known codecs and their frame sizes.

VoIP technology uses two different protocol categories to make a voice call. The first category is the signaling protocols. The aim of the signaling protocols is to initiate a call session between the caller and the callee. The two common signaling protocols are H.323 and Session Initiation Protocol (SIP). H.323 is the first signaling protocol that was widely used in VoIP technology. SIP is currently replacing the H.323 and dominating the VoIP technology. The second category of the VoIP protocols is the media transmission protocols. The aim of the media transmission protocols is to carry the voice data over the IP network [9-11]. The 12-bytes Real-time Transport Protocol (RTP) is the first

DOI: 10.24507/ijicic.15.03.1067

Table 1. Voice codec

Codec 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 2. Media transmission protocols header overhead

Protocol name	Protocol size	Header overhead (%)		
		Payload size 10 bytes	Payload size 20 bytes	Payload size 30 bytes
IAX/UDP	12	120%	60%	40%
RTP/UDP	20	200%	100%	66.6%
ITTP	6	60%	30%	20%

Figure 1. ITTP protocol format

standardized media transmission protocol. Typically, RTP works with the 8 bytes User Datagram Protocol (UDP) to carry the voice data [12,13]. Inter-Asterisk eXchange (IAX) and Internet Telephony Transport protocol (ITTP) are other media transmission protocols that are used to transfer voice data. Like RTP, the 4 bytes IAX works with UDP to carry the voice data. On the other hand, unlike RTP and IAX, ITTP does not work with any other protocol and carry the voice data by itself [9,14,15].

As mentioned earlier, the main issue in VoIP technology is the dissipated bandwidth, resulting from the big packet header in comparison to the small payload. The packet header size varies depending on the used media transmission protocol. Table 2 summarizes the dissipated bandwidth ratio, resulting from the packet header overhead of each of the media transfer protocols [13,15-17].

There are several methods that are used to reduce the dissipated bandwidth resulting from the VoIP packet header. One of the most well-known methods is VoIP packet aggregation. The core idea of the packet aggregation is to aggregate the VoIP packets that share the same path in one large packet, which reduces the header overhead and improves bandwidth utilization [12,18]. In this paper, we will propose and implement a new aggregation method that works with ITTP protocol. The proposed method is named Compression-Aggregation over ITTP (CA-ITTP). The CA-ITTP reduces the bandwidth consumption in two dimensions. The first dimension is reducing the header overhead resulting from adding an IP header to each packet. This is achieved by aggregating many VoIP packets, from several calls that share the same path, into one header. This will reduce the header overhead and improve bandwidth utilization. The second dimension is reducing the VoIP packet payload size. This is achieved by finding and sending the difference between the successive packets. This will reduce the VoIP packet payload size and improve bandwidth utilization as well. The following section will discuss the CA-ITTP method in detail. Figure 1 shows the typical ITTP packet format.

The rest of this article is organized as follows. Section 2 discusses some of the existing VoIP multiplexing methods. Section 3 discusses the proposed CA-ITTP method in detail. It will first discuss the network architecture in which the proposed method (CA-ITTP) will be implemented. Then, the Compression-Algorithm (C-Alg) steps will be demonstrated with some examples. After that, the mini-header will be discussed. Finally, the call set-up process will be illustrated. Section 4 discusses the performance of the proposed CA-ITTP method in comparison to the traditional ITTP protocol. Last section, Section 5, will conclude the paper.

2. **Related Works.**

There is a concerted effort by researchers to improve VoIP bandwidth utilization. Packets aggregation is widely used to improve VoIP bandwidth utilization [7,19]. This section discusses some of the packets aggregation methods.

One of the first multiplexing methods was proposed by Sze et al. in 2002. The proposed method improves bandwidth utilization by compressing the packet header, and then aggregating the packets from the different sources into one aggregated packet. Header compression is achieved based on some properties in the RTP protocol. Sze et al. successfully reduce the 12 bytes RTP to 2 bytes mini-header using these properties. The mini-header is, then, combined with the codec frame to make mini-packet. After that, several mini-packets are aggregated in one UDP/IP header. The sender and receiver side gateways keep a mapping table in order to be able to reinstate the original packets. As we can see, the proposed method uses two techniques to improve bandwidth utilization, packet multiplexing and header compression. Therefore, it achieved a considerable bandwidth saving, whereas, the result showed the bandwidth saving improved by 72% [20].

A more recent multiplexing method was proposed by Pereira and Tarouco in 2009. They proposed an adaptive multiplexing method, whereas, the packets retention time varies based on certain parameters. The proposed method investigated the multiplexing of VoIP packets over IPsec protocol. It consists of two main components. The first one is adaptive multiplexing that is used for aggregating the VoIP packets based on the E-model. The second component is Quality of Service (QoS) monitor that is used to monitor the quality of calls and send the quality parameter to the adaptive multiplexing component, in order to specify the packets retention time. The results showed that the proposed method achieves better bandwidth utilization while keeping the call quality within the acceptable range [21].

In 2013, Roay invented and patented another multiplexing method. The invented multiplexing method was made general so it can work with IPv4 or IPv6 protocols. When serval VoIP packets from several sources are traveling through the same path, they are aggregated in a single IPv4 or IPv6 header. The UDP/RTP header remains unchanged for each packet. The receiver gateway inspects the UDP/RTP header to separate the mini-packet in the aggregated packet. The original packets are then reconstructed and transmitted to their destinations [22].

The aforementioned multiplexing methods work with UDP/RTP protocols. Unlike the previous methods, in 2015, Abualhaj has proposed a new multiplexing method, named ITTP-MUX, which works with the ITTP protocol. ITTP-MUX aggregates the packets that belong to different VoIP calls and share the same path into one packet. At the sender side gateway, the ITTP-MUX removes the IP header of each packet, adds a mini-header to constitute a mini-packet, aggregates the mini-packets that share the same path in on packet, and adds IP header to the aggregated packets. At the receiver side, the reverse process is performed. Whereas the incoming multiplexed packet is split to mini-packets by searching the mini-header, the mini-header of each mini-packet is removed, and an IP header is attached to each mini-packet in place of the mini-header to reconstitute the original packet that was arrived at the receiver gateway. The evaluation of the ITTP-MUX showed that the bandwidth saving is enhanced by up to 29.1% in certain scenarios [23].

As we can see, the ITTP-MUX achieved a noticeable improvement of bandwidth saving. However, the ITTP-MUX concentrated only in the header overhead reduction. In this paper, we proposed a new method (CA-ITTP) that works with ITTP protocol and combines between lessening the header overhead and lessening the payload size, which achieves better bandwidth saving.

3. **Proposed CA-ITTP.** The main idea of the CA-ITTP method is to aggregate many VoIP packets, from several calls that share the same path, into one header. Therefore, CA-ITTP method reduces the header overhead resulting from adding an IP header to each packet, and, thus, it enhances bandwidth utilization. In addition to packet aggregation, CA-ITTP reduces the VoIP packet payload size, which reduces the needed bandwidth to send the packets. That is, it improves bandwidth utilization as well. This section discusses the proposed CA-ITTP method in detail.

3.1. **CA-ITTP network architecture.** The main idea of the CA-ITTP method is to aggregate several ITTP flows, which share the same path to a certain destination, in one flow. There are many scenarios in which several ITTP flows could travel through the same path. A common example would be a company (e.g., a bank) with several branches, in which the employees in the branches call each other to perform the daily tasks. The calls' flow from one branch to another will, typically, take the same path. Any of the branches could be the sender or the receiver of the calls at any time. Figure 2 shows a network architecture scenario in which several flows take the same path.

The proposed CA-ITTP is a method made up off two main components. The first component is called Sender-Compression-Aggregation (SCA) and located at the sender side. SCA performs packets aggregation and payload size reduction to the flows traveling through the same path. The second component is called Receiver-Compression-Deaggregation (RCD) and is located at the receiver side. The RCD segregates the incoming aggregated flows and restores the payload to its normal size. Figure 3 demonstrates

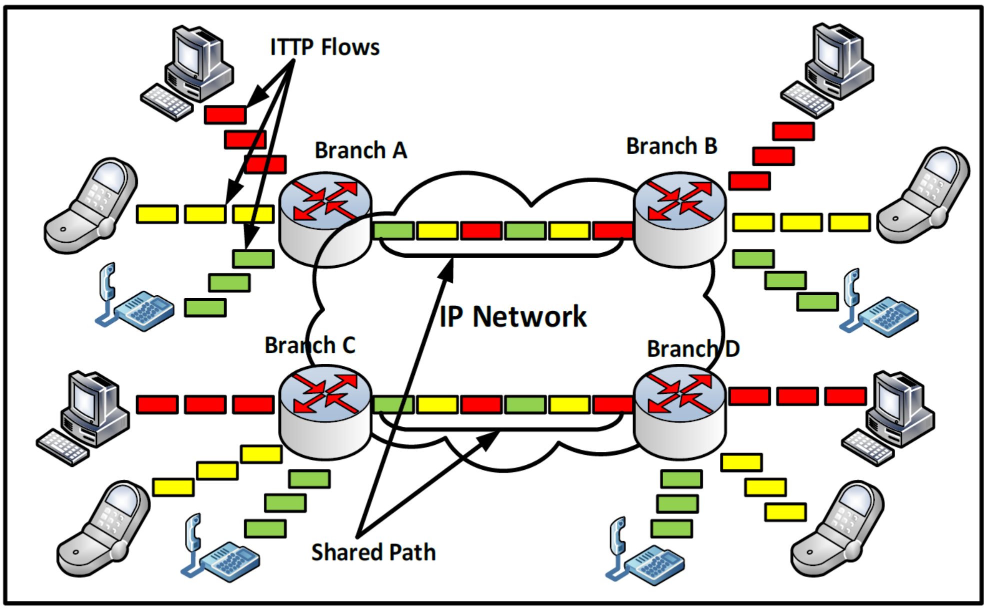

FIGURE 2. ITTP flows sharing the same path

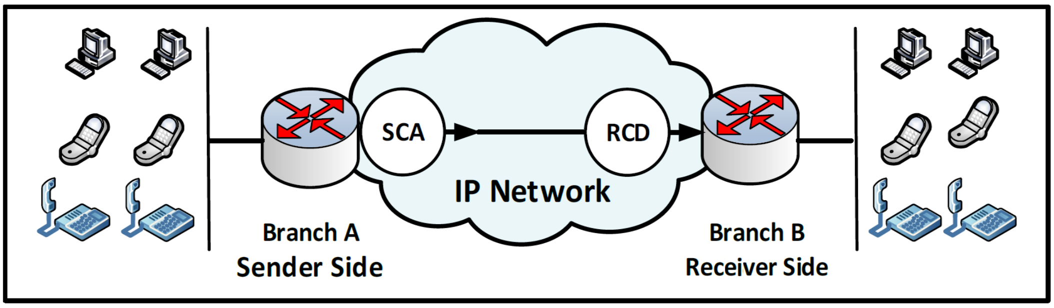

FIGURE 3. CA-ITTP components (SCA and RCD)

FIGURE 4. Mini-packet format

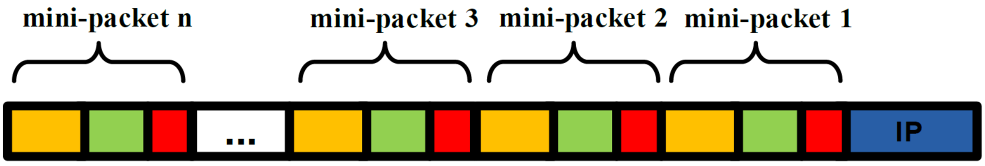

FIGURE 5. m-pkt format

the proposed CA-ITTP method components, namely, SCA and RCD. The following two sub-sections will discuss the SCA and RCD components.

3.1.1. *SCA component.* The purpose of the SCA is to aggregate the ITTP packets from different calls into one packet and to reduce their payload sizes. The resulting packet is named multiplexed packet (m-pkt). There are several steps occurring at the SCA component, at the sender side, before producing the m-pkt.

Step 1. Collect and group the packets according to their destination.
Step 2. Separate the ITTP header and the payload from the packet and keep them in the multiplexing buffer.
Step 3. Separate the ITTP header from the packet's payload, apply the SCA C-Alg on the successive packets' payload, which produces a smaller payload (s-pld). SCA C-Alg will be explained in Section 3.3.1 of this paper.
Step 4. Re-combine the ITTP header to the s-pld and attach a mini-header to them, which produces a small packet, named mini-packet. Figure 4 depicts the mini-packet format. The mini-header will be discussed in Section 3.2 of this paper.
Step 5. Aggregate the mini-packets together in one IP header to produce the m-pkt and, then, transmit the m-pkt to its destination VoIP gateway. Figure 5 shows the m-pkt format.

Figure 6 shows the SCA component at the sender side.

3.1.2. *RCD component.* The purpose of the RCD is to segregate the incoming m-pkt to mini-packets and restore the packet's payload to its normal size. There are several steps that occur at the RCD component, at the receiver side, in order to reconstruct the original ITTP packets.

Step 1. Inspect the mini-header in the received m-pkt, in order to retrieve the mini-packets.

Step 2. Remove both mini-header and ITTP header and, then, use the RCD C-Alg to reinstate the s-pld to normal size payload. RCD C-Alg will be explained in Section 3.3.2 of this paper.
Step 3. Attach the ITTP header along with the IP header to the payload from the previous step, which reconstructs the original packet that has been arrived at the sender side.
Step 4. Transmit the packets generated in Step 3 to their destinations.

Figure 7 shows the RCD component at the receiver side gateway.

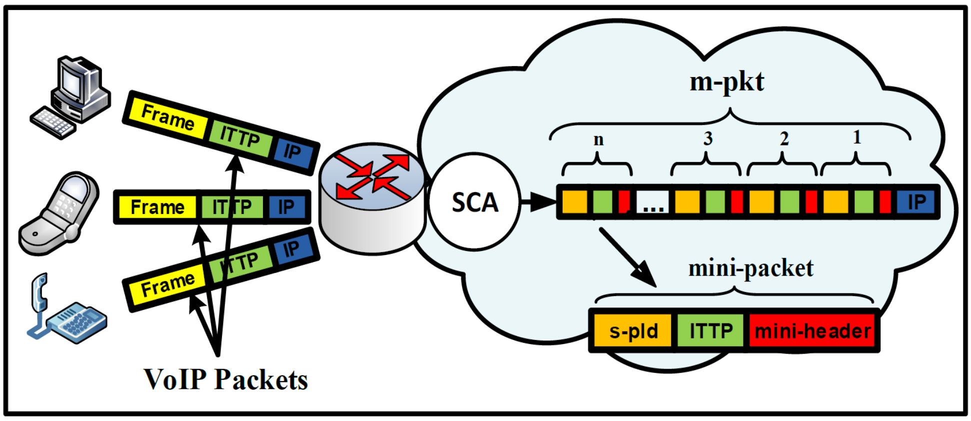

FIGURE 6. SCA component at sender side gateway

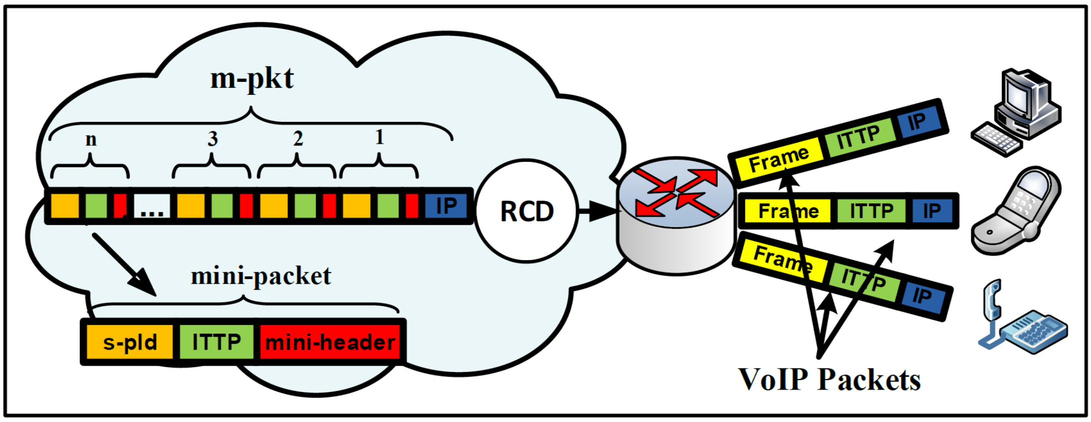

FIGURE 7. RCD component at the receiver side gateway

FIGURE 8. mini-header format

3.2. **Mini-header.** As mentioned earlier, a mini-header is added for each mini-packet at the sender side. The RCD component at the receiver side uses this mini-header to separate the mini-packets within the m-pkt, reinstate the normal payload size, and reconstruct the original packet arrived at the receiver gateway. The mini-header contains three fields, namely Call ID (CID), Greater Flag (F-Grt), and Normal-Payload Flag (F-NP). Figure 8 shows the mini-header format.

CID (6 bits) – A unique CID between 0 and 63 is selected, by the SCA at the sender side during call set-up, for each call. The RCD, at the receiver side, uses the CID to find the address of the destination from the state table. The state table, located at the RCD (receiver side gateway) and SCA (sender side gateway) components, maps the destination address to its corresponding CID. The CID of each mini-packet must be unique within the same m-pkt. Therefore, single m-pkt can have up to 64 mini-packets.

F-Grt (1 bit) – The RCD C-Alg, at the receiver side, uses this flag to reinstate the payload of the mini-packet to its normal size. The F-Grt determines which packet between the successive packets is greater. Setting the F-Grt to 1 indicates that the n packet is greater than $n+1$ packet. Setting the F-Grt to 0 indicates that the $n+1$ packet is greater than n packet.

F-NP (1 bit) – The RCD C-Alg, at the receiver side, uses this flag to reinstate the payload of the mini-packet to its normal size. The F-NP flag determines whether the mini-packets contains a normal size payload (original payload) or a reduced size payload. Setting the F-NP to 1 indicates a normal size payload (original payload). Setting the F-NP to 0 indicates a reduced size payload.

3.3. **Compression-Algorithm (C-Alg).** The aim of the C-Alg is to lessen the packet payload (frame) size. In order to achieve this, C-Alg treats the frames as decimal numbers. Accordingly, C-Alg subtracts the successive frames from each other, which produces a smaller payload (s-pld) than the normal payload size. The s-pld is encapsulated and sent, instead of the normal full payload size, which saves the bandwidth. For example, assume both the payload of packet1 (pp1) and the payload of packet2 (pp2) are fifteen bits, 111001100110101 and 110011101001001 respectively. Then the result of subtracting pp2 from pp1 is thirteen bits 101111101100, which is smaller than the normal payload size (110011101001001), and thus lessens the consumed bandwidth. In the following two sub-sections, we will explain the SCA C-Alg at the sender side, which reduces the packet payload size, and the RCD C-Alg at the receiver side, which restores the normal payload size.

3.3.1. *SCA C-Alg.* The SCA C-Alg (C-Alg at the sender side) lessens the size of the payload by subtracting the successive packets from each other. The first packet's payload remains unchanged since no packets precede it. Starting from the second packet, the size of the payloads of each packet will be reduced, if possible, to produce the s-pld based on a certain algorithm (SCA C-Alg). For example, assume that the first packet is A and the second packet is B. If the payload of packet A (A-pld) is greater than the payload of packet B (B-pld), then subtract B-pld from A-pld. Otherwise, subtract A-pld from B-pld. If the subtraction result is less than B-pld, then change the S-pld value to the subtraction result value. Otherwise, change the S-pld value to B-pld value. Figure 9 shows the pseudo code of SCA C-Alg algorithm.

3.3.2. *RCD C-Alg.* The RCD C-Alg (C-Alg at the receiver side) reinstates the size of the payload by subtracting or adding the successive packets from each other. For example, assume the first packet is A and the second packet is B. If SCA C-Alg subtracts A-pld from B-pld, then the RCD C-Alg will sum A-pld and B-pld. Otherwise, if SCA C-Alg subtracts B-pld from A-pld, then the RCD C-Alg will perform the same operation (subtract). The first packet's payload remains unchanged since no packets precede it. Starting from the second packet, the size of the payload of each packet will be reinstated to produce the normal payload size. Figure 10 shows the pseudo code of RCD C-Alg algorithm.

SCA C-Alg Algorithm

```
// A is packet n
// B is packet n+1
// A-pld is the payload of packet n
// B-pld is the payload of packet n+1
// F-Grt is a flag in the mini-header
// F-NP is a flag in the mini-header

if (A-pld > B-pld) {
  result= A-pld – B-pld
  set F-Grt to 1
else
  result= B-pld – A-pld
  set F-Grt to 0
}
if (result < B-pld) {
  s-pld= result
  set F-NP to 0
else
  s-pld= B-pld
  set F-NP to 1
}
```

FIGURE 9. SCA C-Alg algorithm

RCD C-Alg Algorithm

```
// A is packet n
// B is packet n+1
// A-pld is the payload of packet n
// s-pld is the payload of packet n+1
// F-Grt is a flag in the mini-header
// F-NP is a flag in the mini-header

if (F-NP is equal to 1)
  Payload-Size= s-pld
else if (F-NP is equal to 0) {
  if (F-Grt is equal to 0)
    Payload-Size= s-pld + A-pld
  else if (F-Grt is equal to 1)
    Payload-Size= A-pld - s-pld
  }
```

FIGURE 10. RCD C-Alg algorithm

3.4. **Call set-up and CID selection.** In VoIP, the two ends of the call are required to establish a session before starting the call. Call establishment process must be modified to suit the CA-ITTP method. The modification of the call establishment process identifies the CID for each call. The CID identification procedure is summarized in the following few steps.

Step 1. After establishing a connection between the two call sides, the RCD component at the sender side examines for the presence of a connection with the receiver side.

Step 1.a. If there is a connection available with free CID, then a CID is chosen and preserved for that call.

Step 1.b. If no session exists between the sender and receiver side gateways or if there is no CID available in the present session, then a new session is initiated between the sender and receiver side gateways.

Step 2. When choosing the CID of the call, the SCA retains the CID and its corresponding callee socket (IP address: port number) in the state table.

Step 3. The SCA transmits the CID and its corresponding callee socket to the RCD at the receiver side gateway.

Step 4. The RCD also retains the received CID and callee socket in a state table.

The steps of CID identification must be performed in the two sides of the call (both directions). Table 3 demonstrates the socket information and the corresponding CID within the state table.

TABLE 3. State table

SCA component (sender side gateway)			RCD component (receiver side gateway)	
CID	IP address: Port		CID	IP address: Port
3	11.0.0.1:13		3	11.0.0.1:13
5	11.0.0.2:15		5	11.0.0.2:15
6	11.0.0.3:16		6	11.0.0.3:16
7	11.0.0.4:10		7	11.0.0.4:10
11	11.0.0.5:11		11	11.0.0.5:11

4. CA-ITTP Performance Evaluation.

This section discusses the performance of the proposed CA-ITTP method, in comparison to the traditional ITTP protocol. The main goal of the comparison is to show the bandwidth efficiency of the CA-ITTP method versus ITTP. We have measured the bandwidth efficiency based on two factors, namely, header overhead and bandwidth consumption. The 14-bytes LPC was used as voice codec. Figure 11 shows the header overhead of CA-ITTP method against the ITTP. As

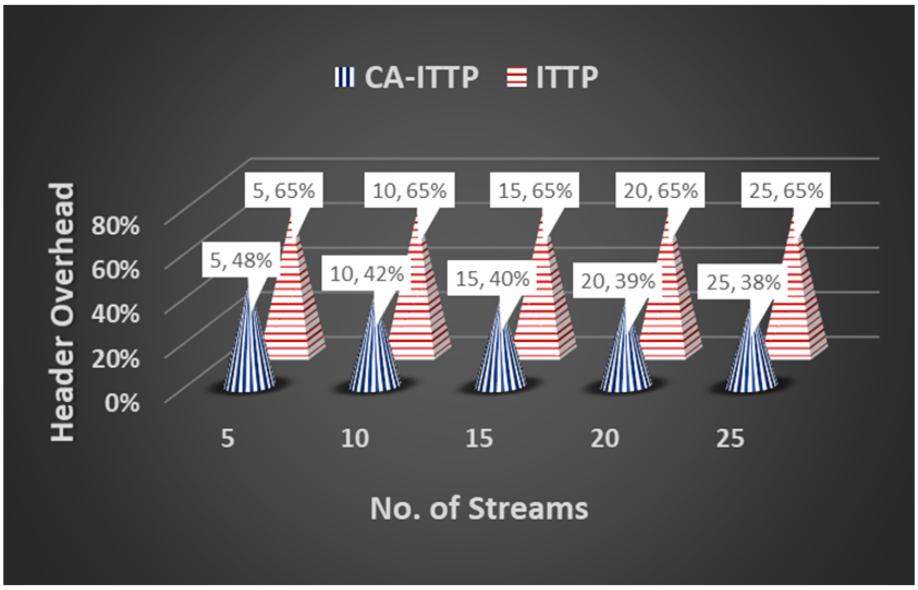

FIGURE 11. Header overhead

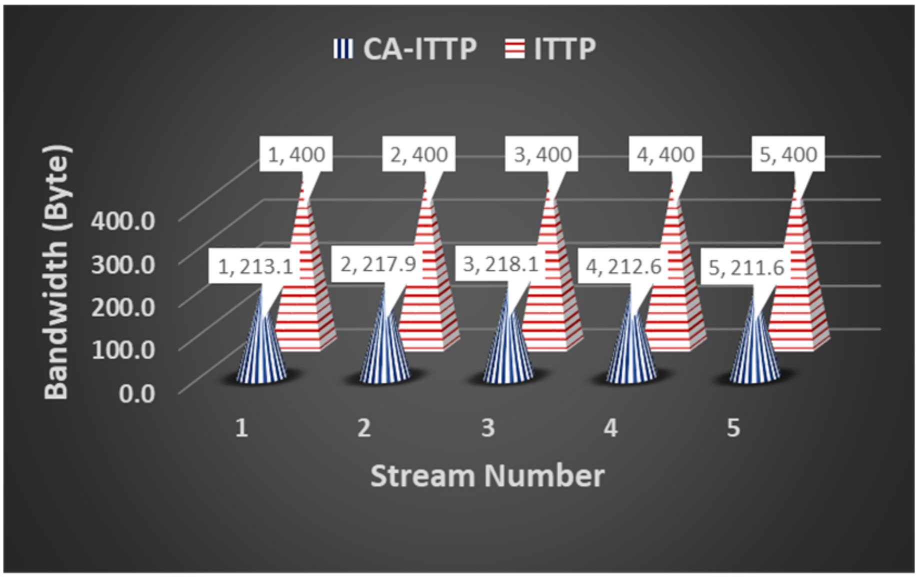

Figure 12. Consumed bandwidth

we can see, CA-ITTP achieved a high reduction in the header overhead in comparison to ITTP. Apparently, this is due to the aggregation of the ITTP VoIP packets, which are sharing the same path, in a single IP header, instead of adding a separate header to each packet. Figure 12 shows the consumed bandwidth when using CA-ITTP method against the ITTP. As we can see, CA-ITTP achieved a high reduction in the bandwidth consumption in comparison to ITTP. This is due to two main reasons. First, as discussed earlier, reduce the consumed bandwidth when using the CA-ITTP method instead of ITTP header overhead. Second, as discussed earlier too, reduce the size of the VoIP packet payload, which reduces the consumed bandwidth. Accordingly, CA-ITTP has succeeded in improving bandwidth utilization, especially in comparison to the traditional ITTP protocol without multiplexing.

5. **Conclusion.** VoIP is one of the most emerging technologies in the current era. There is a concerted effort by researchers to improve VoIP bandwidth utilization. VoIP packets aggregation is one of the main methods that are used to improve VoIP bandwidth utilization. In this paper, we have proposed a new method, called Compression-Aggregation over ITTP (CA-ITTP), which works with the ITTP protocol. The CA-ITTP method achieved high bandwidth utilization improvement by i) aggregating the packet that shares the same path in one header and ii) reducing the payload size of the VoIP packets. The CA-ITTP consists of two components, namely, Sender-Compression-Aggregation (SCA) and Receiver-Compression-Deaggregation (RCD). The SCA component aggregates and reduces the VoIP packet payload at the sender side VoIP gateway, while the RCD deaggregates and reinstates the VoIP packet payload at the receiver side VoIP gateway. The results showed that using the CA-ITTP method, we can almost carry the double of VoIP packets over the same network bandwidth in comparison to the traditional ITTP without aggregation. As future work, the CA-ITTP method will be discussed and evaluated in comparison to other multiplexing methods. In addition, more evaluation parameters will be used to measure the performance of the CA-ITTP method.

REFERENCES

[1] S. K. Manjur, M. M Abualhaj, O. Abouabdalla, T.-C. Wan, R. Budiarto and A. M. Manasrah, Conference gateway for heterogeneous clients: Real time switching clients and interasterisk exchange

clients, *International Journal of Innovative Computing, Information and Control*, vol.7, no.1, pp.395-406, 2011.

[2] P. H. A. Filho, M. F. Caetano and J. L. Bordim, A packet aggregation mechanism for real time applications over wireless networks, *International Journal of Networking and Computing*, vol.2, no.1, pp.18-40, 2012.

[3] M. Abu-Alhaj, M. S. Kolhar, M. Halaiyqah, O. Abouabdalla and R. Sureswaran, Multiplexing SIP applications voice packets between SWVG gateways, *Proc. of International Conference on Computer Engineering and Applications (ICCEA 2009)*, 2009.

[4] M. Abu-Alhaj, M. S. Kolhar, M. Halaiyqah, O. Abouabdalla and R. Sureswaran, MuxComp – A new architecture to improve VoIP bandwidth utilization, *International Conference on Future Networks*, pp.212-215, 2009.

[5] Y. Nomura, K. Mori, K. Naito and H. Kobayashi, High efficient packet aggregation scheme for multi-rate and VoIP packet transmissions in next generation MU-MIMO WLANs, *International Conference on Advanced Technologies for Communications (ATC)*, pp.517-521, 2014.

[6] 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.

[7] M. M. Abualhaj, M. Kolhar, K. Qaddoum and A. A. Abu-Shareha, Multiplexing VoIP packets over wireless mesh networks: A survey, *KSII Trans. Internet & Information Systems*, vol.10, no.8, 2016.

[8] J. Saldana, J. Fernández-Navajas, J. Ruiz-Mas, J. Murillo, E. V. Navarro and J. I. Aznar, Evaluating the influence of multiplexing schemes and buffer implementation on perceived VoIP conversation quality, *Computer Networks*, vol.56, no.7, pp.1893-1919, 2012.

[9] M. M Abualhaj, S. K. Manjur, R. Sureswaran, T.-C. Wan, I. J. Mohamad 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.1879-1895, 2012.

[10] S. Karapantazis and F. N. Pavlidou, VoIP: A comprehensive survey on a promising technology, *Computer Networks*, vol.53, no.12, pp.2050-2090, 2009.

[11] M. Kolhar, M. M. Abualhaj and F. Rizwan, QoS design consideration for enterprise and provider's network at ingress and egress router for VoIP protocols, *International Journal of Electrical and Computer Engineering (IJECE)*, vol.6, no.1, pp.235-241, 2016.

[12] C. Vulkan, A. Rakos, Z. Vincze and A. Drozdy, *Reducing Overhead on Voice Traffic*, United States Patent US 8,824,304, 2014.

[13] A. Froehlich, *CCNA Voice Study Guide: Exam 640-460*, John Wiley & Sons, 2010.

[14] M. S. Kolhar, M. M. Abu-Alhaj, O. Abouabdalla, T. C. Wan and A. M. Manasrah, Comparative evaluation and analysis of IAX and RSW, *arXiv preprint arXiv:1001.2280*, 2010.

[15] M. Spencer, B. Capouch, E. Guy, F. Miller and K. Shumard, *IAX: Inter-Asterisk eXchange Version 2*, 2010.

[16] M. M. Abu-Alhaj, A. Manasrah, M. Baklizi, N. Abdullah and L. V. Chandra, Transport layer protocols taxonomy from voice over IP perspective, *Advanced Computing: An International Journal (ACIJ)*, vol.2, no.4, 2011.

[17] M. Spencer and F. W. Miller, *IAX Protocol Description*, IXIA Publication, 2004.

[18] S. Boland, *Multiplexing VoIP Streams for Conferencing and Selective Playback of Audio Streams*, United States Patent US 7,936,705, 2011.

[19] S. Yun, H. Kim, H. Lee and I. Kang, 100+ VoIP calls on 802.11b: The power of combining voice frame aggregation and uplink-downlink bandwidth control in wireless LANs, *IEEE Journal on Selected Areas in Communications*, vol.25, no.4, 2007.

[20] H. P. Sze, S. C. Liew, J. Y. Lee and D. C. Yip, A multiplexing scheme for H.323 voice-over-IP applications, *IEEE Journal on Selected Areas in Communications*, vol.20, no.7, pp.1360-1368, 2002.

[21] R. M. Pereira and L. M. Tarouco, Adaptive multiplexing based on E-model for reducing network overhead in voice over IP security ensuring conversation quality, *The 4th International Conference on Digital Telecommunications*, pp.53-58, 2009.

[22] B. Roay, *Generic UDP Multiplexing for Voice over Internet Protocol (VoIP)*, United States Patent US 8,553,692, 2013.

[23] M. M Abualhaj, ITTP-MUX: An efficient multiplexing mechanism to improve voip applications bandwidth utilization, *International Journal of Innovative Computing, Information and Control*, vol.11, no.6, pp.2063-2073, 2015.