Delta-Multiplexing: A Novel Technique to Improve VoIP Bandwidth Utilization between VoIP Gateways

Mosleh M. Abu-Alhaj[1], Manjur S. Kolhar[1], Lingeswari V. Chandra[1], O. Abouabdalla[1], Ahmed M. Manasrah[1]
National Advanced IPv6 Center, University Science Malaysia
{mosleh, manjur, lingeswari, omar, ahmad} @nav6.org

Abstract—**Gradually, Voice over Internet Protocol (VoIP) has been dominating the telecommunications world. Unfortunately, its applications are injecting a huge number of small packets in the network, which produces high overhead and therefore wastes network bandwidth. This paper proposed the use of a novel multiplexing technique, Delta-Multiplexing, to save the wasted bandwidth. In the Delta-Multiplexing technique, the VoIP packets destined to the same destination gateway are aggregated in a single UDP/IP header, therefore reducing the header overhead and saving network bandwidth. Moreover, the Delta-Multiplexing technique reduces the size of the packets payload by transmitting the difference between the consecutive packets payloads. Accordingly, the Delta-Multiplexing technique greatly saves bandwidth. We have simulated the Delta-Multiplexing technique using a 14-byte LPC codec. The result showed that Delta-Multiplexing is capable of saving between 68% and 72% as compared to conventional techniques (without multiplexing). Moreover, the Delta-Multiplexing technique reduces the number of VoIP packets running over the network, therefore reducing network traffic, overload, and congestion, thus improving the overall network performance.**

Keywords-component; multiplexing, compression, bandwidth utilization, and VoIP.

I. INTRODUCTION

In the last decade, Voice over Internet Protocol (VoIP) has emerged as a new technology in the telecommunications industry. VoIP was conceived from existing Internet infrastructure, mechanisms, and features to make high-quality phone calls to any place around the world using the Internet. Gradually, VoIP technology will start dominating the telecommunications world and will replace the current public switched telephone network (PSTN) technology in the future [1] [2]. There are many drivers behind the domination of VoIP over PSTN technology. The most important point is that VoIP technology uses the existing Internet environment, which reduces the call cost tremendously. In addition, VoIP technology provides a higher reliability factor since it automatically handles or bypasses network problems, such as the physical problems of over congestion. Another vital advantage is its capability to make calls from anywhere using a personal computer or any network device like PDAs [1] [3] [4].

In spite of the numerous advantages of VoIP, however, there are some problems that hinder it from making headway. A typical VoIP packet consists of a 40-byte RTP/UDP/IP header (12 bytes RTP, 8 bytes UDP, and 20 bytes IP). Meanwhile, the typical payload size (codec frame) is between 10 bytes and 30 bytes [5]. Obviously, this small payload size leads to two main problems. First, the network performance is degraded, thus affecting the QoS [1] [5]. This problem is caused by small VoIP packets flooding the network, which overloads it. Thus, increasing the packets congestion results in rising packets loss as well [1] [5] [6]. Second, bandwidth utilization is inefficient, which is caused by header overhead due to attaching a big header (40 bytes) to small payloads (10 bytes to 30 bytes) [1] [7]. Thus, network bandwidth is wasted by transmitting non-useful data. Accordingly, if we consider the previously mentioned payloads with different frame sizes, the wasted bandwidth, which can be calculated as the relative ratio between useful and non-useful data, will be 57% to 80%. This ratio variation is due to the different codec mechanisms that produce different frame sizes. Table I shows some of the codecs used in VoIP [8]. The objective of this paper is to improve the bandwidth utilization for the evaluation and improvement of network performance, thus enhancing VoIP QoS.

There are several techniques used to improve VoIP bandwidth utilization. These techniques can be divided into two groups. The first group reduces the VoIP packet header overhead. Packet multiplexing is the main technique used in header overhead reduction. Combining VoIP with a frame size of 10 bytes to 30 bytes to a header with a size of 40 bytes produces enormous packet overhead. Meanwhile, the use of multiplex multiple VoIP frames in a single header reduces the header overhead, thus improving bandwidth utilization [5] [6] [9]. Another important technique is header compression. Most VoIP packet header fields are constant through the call period, and some other fields are increased by a constant value for the consecutive packets. Using these properties, the VoIP packets header was compressed from 40 bytes to 2 bytes in the best case [10].

TABLE I. SOME OF THE VOIP CODECS

Codec	Frame size	Frame size	Bit rate
LPC	20 ms	14 B	5.6 kbps
G.729	10 ms	10 B	8 kbps
G.723.1	30 ms	24 B	6.3 kbps
G.723.1	30 ms	20 B	5.3 kbps
G.726	5 ms	15 B	24 kbps
G.728	5 ms	10 B	16 kbps

978-0-7695-4108-2/10 $26.00 © 2010 IEEE
DOI 10.1109/CIT.2010.86

The second group extremely improves bandwidth utilization. This group affects the voice data itself (packet payload). Voice compression is a vital technique in this group. The codec (compression/decompression) is a device or computer program that converts the voice analogue signal to digital data. After this, the codec uses compression algorithms to compress the digital data. Finally, the compressed data are converted to frames (packet payload). The sizes of frames vary depending on the algorithm used by the codec. The codec reduces the data size greatly, therefore consuming less bandwidth during transmission [9] [11]. Thus, this results in improved bandwidth utilization. Table 1 shows some of the voice codecs. Another vital technique is the silence suppression, which is also known as voice activity detection (VAD). Typically, one of the call parties is speaking, and the other one is listening. Accordingly, around 60% of the phone call conversation is silence. Transmitting the silence wastes the bandwidth usage in garbage data. For this reason, VAD applications use a suitable mechanism to suppress the silence, thus saving bandwidth [12] [13] [14]. This paper proposed a novel technique to improve bandwidth utilization through the reduction of both header overhead and payload size.

This paper is organized as follows. Section 2 discusses some of existing studies that aim to reduce the header overhead problem. Section 3 discusses in more details the proposed techniques, followed by the implementation details, and finally the highlights of the effectiveness of the proposed techniques on network performance. Finally, Section 4 concludes this paper.

II. RELATED WORKS

VoIP is one of the main technologies that use the Internet. A huge number of VoIP packets are running over the Internet, which consume a big part of network bandwidth. Thus, there is great effort from network developers to reduce the bandwidth consumed by VoIP packets. This section discusses some of these efforts.

In 1999, a great compression technique was developed by Casner and Jacobson. The proposed technique compressed the RTP/UDP/IP VoIP packet header from 40 bytes to 2 bytes. This considerable compression in RTP/UDP/IP VoIP packets header has greatly reduced the header overhead problem and subsequently improved bandwidth utilization. The RTP/UDP/IP VoIP packet header compression technique is divided into two parts. In the first part, Casner and Jacobson noticed that most of the fields in the RTP/UDP/IP VoIP packet header remained unchanged during the call time period. By taking advantage of this feature, Casner and Jacobson proposed the transmission of these unchanged fields during the call initiation and eliminating them from the other packets. In the second part, Casner and Jacobson noticed that in most of the RTP/UDP/IP VoIP packet headers, the fields are increasing by a constant value. By taking advantage of this feature, Casner and Jacobson applied differential coding to optimize these fields [10].

Apart from the header compression technique, multiplexing the VoIP packets can also be done to reduce packet overhead. The purpose of multiplexing the VoIP packets is to increase the payload size. Thus, this technique reduces the header overhead and improves the bandwidth utilization as a result. A novel packet multiplexing method has been proposed by Hoshi et al. in 1999. Hoshi et al. suggested the aggregation of VoIP streams destined to the same gateway in a single UDP/IP header. After implementing the proposed method on an H.323 standard, the result showed that the consumed bandwidth is decreased by 40% [15].

More recently, Sze et al. proposed a technique that combined the aforementioned techniques: packet multiplexing and header compression. In the proposed technique, the VoIP packets of the same route are multiplexed in a single UDP/IP header. Moreover, the proposed technique employed the header compression technique and applied it on the RTP header. This will eventually save more bandwidth. After implementing the proposed technique on H.323 standard as well, the result showed extreme improvement in bandwidth efficiency [7].

In this paper, we have proposed the multiplexing technique called Delta-Multiplexing. Unlike the aforementioned multiplexing techniques, Delta-Multiplexing provides novelty by combining header overhead reductions and decreasing payload size. Thus, this will save huge bandwidth of between 68% and 72%. Moreover, the Delta-Multiplexing technique improved the network performance in terms of network traffic, overload, and packets congestion. In addition, the Delta-Multiplexing technique can be applied to several environments such as SIP, H.323, and other.

III. DELTA-MULTIPLEXING

Inefficient bandwidth utilizations and network overloads are the two vital issues and problems in the computer network circle. The small VoIP packets are producing a large overhead, which wastes network bandwidth. Moreover, the small VoIP packets overwhelm the network, thus overloading it [1] [6] [7]. Packet multiplexing is one of the best techniques used to improve bandwidth utilization. Furthermore it reduces the overloading because it decreases the number of generated packets. This paper proposed a novel multiplexing technique called Delta-Multiplexing. The Delta-Multiplexing technique is achieved with an extremely competent bandwidth utilization and extreme network performance. Fig. 1 shows the scenario in which the Delta-Multiplexing technique achieves high performance.

A. Delta-Multiplexing Architecture

The Delta-Multiplexing architecture consists of two entities. The first entity is the Multiplexer (Mux), which is located in the sender gateway. The Mux performs payloads size reduction and packets multiplexing. The second entity is the D-Multiplexer (DMux), which is located in the receiver gateway. The DMux performs packets de-multiplexing and returns the payload to its original size.

1) Packets Multiplexing: The Mux in the sender gateway performs a set of procedures aiming to reduce the packet size. First, the Mux checks the destination of the received packets, before extracting the packets payload, to multiplex the packets destined to the same destination gateway together. Afterwards, the size of the extracted payload will be reduced by applying the Delta-Algorithm (D-Alg). This will result in the payload called Compressed-Payload. Subsequently, the RTP header will be attached back together with the small header called Mini-Header to the Compressed-Payload, which constitutes the small packets called mini-packets. Before the final step is done, these mini-packets will multiplex together in one UDP/IP header, with the resulting packets called the Multiplexed-Packets. Finally, the Multiplexed-Packets dispatch to their destination gateways. Fig. 2 depicts the Mux in the sender gateway. The D-Alg and the Mini-Header are discussed in the latter part of this paper.

2) Packets De-Multiplexing: The D-Mux in the receiver gateway splits the received Multiplexed-Packets to mini-packets by inspecting the Mini- Header. The D-Mux commences by eliminating the Mini-Header along with the RTP header and then using the D-Alg to return the Compressed-Payloads back to the original size, thus resulting in payloads called Original-Payloads. Subsequently, the D-Mux will attach the RTP header along with the destination addresses to the Original-Payloads. Finally, the constructed packets will be dispatched to their destinations. Fig. 3 depicts the DMux in the receiver gateway.

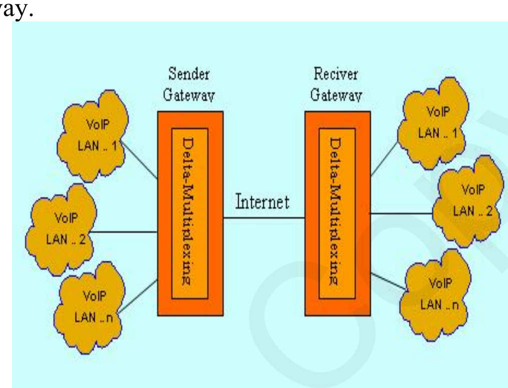

Figure 1. Multiplexing scenario

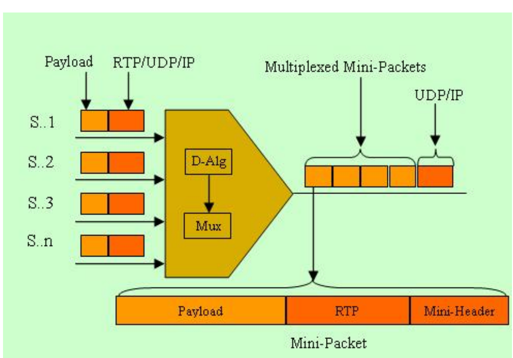

Figure 2. Delta-multiplexing multiplexer, sender gateway

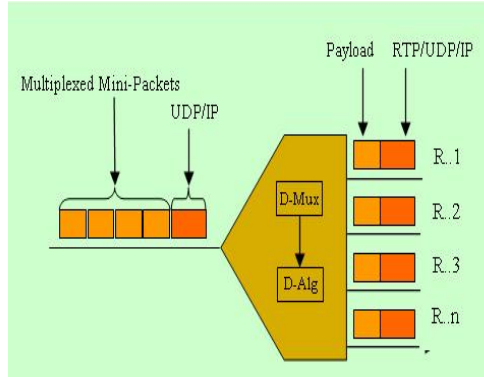

Figure 3. Delta-multiplexing de-multiplexer, receiver gateway

B. D-Alg

The purpose of D-Alg is to reduce the packets payload (frame) size. The essence of the D-Alg is to deal with the frames as integer numbers. Accordingly, D-Alg will subtract the frames from one another, and then packetize and transmit the difference between the frames instead of the full frame size, thus saving bandwidth. For illustrative purpose, suppose the Frame 1 (F1) is 111001100110101 and Frame 2 (F2) is 110011101001001, then the result of subtracting F2 from F1 is 101111101100, which is smaller than the original frame size. In the next section, we will explain the subtraction operation in D-Alg and all its cases in details.

The D-Alg comes in two parts. The first part is performed in the sender gateway with the aim of reducing data size. The second part is performed in the receiver gateway, which targets to return the data to its original size and format.

1) Sender Gateway D-Alg : In the sender gateway, D-Alg reduces the received packet payload by using the previous packet payload. Thus, the D-Alg starts working on the second packet since no packet precedes the first one; hence, the first packet does not change. In order to demonstrate the steps of D-Alg in the sender gateway, we have assumed that the Mux in the sender gateway has extracted three packets payloads. The first is payload A, the second is payload B, and the third is payload C. First, the D-Alg checks whether payload A is greater than payload B or if payload B is greater than payload A as an integer number. Second, the D-Alg then subtracts the smaller payload, either payload A or B, from the other greater payload, payload A or B. Hereafter, the D-Alg compares the result of the subtraction, which is called Sub-Rslt, with payload B. Finally, if the Sub-Rslt is less than payload B, the D-Alg keeps the Sub- Rslt in a new buffer, which is called Compressed-Payload. Otherwise, the D-Alg keeps payload B in the Compressed-Payload buffer. For the next payload (payload C), the D-Alg performs the first to final steps but this time, on payloads B, C, and so on. As discussed before, these payloads, which were subtracted from one another, will be multiplexed by the Mux in the same Multiplexed-Packet. The number of mini-packets inside the Multiplexed-Packet will be discussed later in this paper. The following pseudo code clarifies the D-Alg in the sender gateway.

```
Compressed_Payload_Function (Pld_A, Pld_B)
{
 If (Pld_A >Pld_B) then
  {
      Sub-Rslt = Pld_A - Pld_B
      Else
      Sub_Rslt = Pld_B - Pld_A
  }
 If (Sub_Rslt<Pld_B) then
 {
      Compressed-Payload = Sub-Rslt
      Else
      Compressed_Payload = Pld_B
}}
```

2) *Receiver Gateway D-Alg :* In the receiver gateway, the D-Alg reverts the payload size to its original size, which is called the Original-Payload. To demonstrate the steps in D-Alg, we used the assumption stated in the previous section. First, the D-Alg skips the first packet, packet A, since no packet precedes it. Second, by inspecting the Mini-Header, the D-Alg checks whether payload B has changed in the sender gateway or not. If payload B has not changed in the sender gateway, the D-Alg keeps it as the Original-Payload. Otherwise, if payload B has changed in the sender gateway, the D-Alg needs to revert payload B to its original size. In order to do this, the D-Alg inspects the Mini-Header to determine whether payload A was subtracted from payload B or if payload B was subtracted from payload A in the sender gateway. Finally, if payload A was subtracted from payload B in the sender gateway, the D-Alg adds payload A to payload B. Otherwise, if payload B was subtracted from payload A in the sender gateway, the D-Alg then subtracts payload B from payload A, and the resulting payload is called the Original-Payload. For the next payload (payload C), the D-Alg repeats the first to final steps on payloads B and C and so on. The following pseudo code clarifies the D-Alg in the receiver gateway.

```
De_Compressed_Payload_Function (Pld_A, Pld_B)
{
 If (Mini_Header→ F_Org = 1) then
 Original-Payload = Pld_B
 Else if (Mini_Header→ F_Org = 0) then
 {
    If (Mini-Header→ F-Grt = 1) then
    {
     Original_Payload = Pld_B + Pld_A
     Else if (Mini_Header→ F_Grt = 0) then
     Original_Payload = Pld_A - Pld_B
    }
 }}
```

C. *Mini-Header*

Instead of the 28-byte UDP/IP header, the 12-bit Mini-Header is added to each payload. Fig. 4 displays the Mini-Header fields. The 12-bit Mini-Header reduces the header overhead problem and saves bandwidth. This Mini-Header is used by the receiver gateway to distinguish the Mini-Packets inside the Multiplexed-Packets. Moreover, the Mini-Header is used to return the size of the payload to its original size. The Mini-Header contains the following fields:

- Length (4 bits) – The length of the Mini-Packet payload or the length of the field size depends on the size of the frame codec. For the 14-byte LPC codec frame that we have used on the implementation, 4 bits is enough.
- Stream ID or SID (6 bits) –SID is used by the receiver gateway to identify the Mini-Packet destination address; SID is unique in the single multiplexed connection. The size of the SID field depends on the number of Mini-Packets inside the Multiplexed-Packets. A 6-bit SID is large enough to multiplex 64 Mini-Packets inside one Multiplexed-Packet.
- Original-Payload Flag or F-Org (1 bit) – This flag specifies whether the Mini-Packet payload is the Original-Payload or the Compressed-Payload. A value of one in this field indicates that the Mini-Packet payload is the Original-Payload, and a value of zero indicates that the Mini-Packet payload is the Compressed-Payload. The F-Org is used by the D-Alg to return the size of the Mini-Packet payload to its original size.
- Greater Flag or F-Grt (1 bit) – This flag specifies which Mini-Packet payload is greater, namely, the current Mini-Packet payload or the Mini-Packet payload that precedes the current one. A value of one in this field indicates that the current Mini-Packet payload is greater, and a value of zero indicates that the Mini-Packet payload that precedes the current Mini-Packet payload is greater. The F-Grt is used by the D-Alg to return the size of the Mini-Packet payload to its original size.

IV. Call Set-Up

Multiplexing the VoIP packets between the VoIP gateways requires a small adjustment in the signaling process. The adjustment in the call signaling is summarized in a few steps. First, after initiating the session between the call parties, the Mux in the sender gateway creates an SID for this session. Second, the Mux keeps the SID and the callee address (IP address and port number) in the mapping table. Meanwhile, the sender gateway sends the SID and the address to the receiver gateway. Once the receiver gateway receives the SID and the address, it also keeps them in the mapping table. Finally, the voice packets start transmitting. Table II shows the SID and the address inside the mapping table.

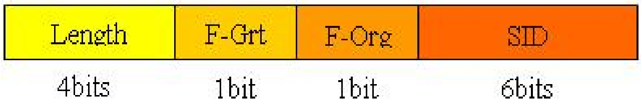

Figure 4. Mini-Header

TABLE II. MAPPING TABLES

Sender Gateway		Receiver Gateway	
SID	Address	SID	Address
10	10.207.160.1:5	10	10.207.160.1:5
15	10.207.160.2:90	15	10.207.160.2:90
16	10.207.160.5:13	16	10.207.160.5:13
20	10.207.160.6:14	20	10.207.160.6:14

V. DELTA-MULTIPLEXING PERFORMANCE ANALYSIS

We have built a simulation for the proposed multiplexing technique, Delta-Multiplexing, to demonstrate its performance. Moreover, this simulation is done to compare the performance of the Delta-Multiplexing technique with conventional techniques that have no multiplexing. We built the simulation based on three main factors, namely, bandwidth efficiency, delay, and network performance. The results showed the high performance of the Delta-Multiplexing technique over other conventional techniques.

A. Bandwidth Efficiency

We have carried out three different experiments by generating 100 streams for each experiment. We have used the 14-byte LPC codec frame attached to the 40-byte RTP/UDP/IP header, with a total size of 54 bytes per packet, in the generated packets. The three experiments are used to compare the bandwidth efficiency between the Delta-Multiplexing technique and conventional techniques. The results showed that the Delta-Multiplexing technique greatly improved bandwidth efficiency as compared to the conventional techniques. In the first experiment, we have compared the consumed bandwidth between the Delta-Multiplexing technique and conventional techniques. Fig. 5 depicts the consumed bandwidth in the two techniques. In the second experiment, we have compared the header overhead between the Delta-Multiplexing technique and the conventional techniques. Fig. 6 depicts the header overhead in the two techniques. In the third experiment, we have shown the total saved bandwidth ratio in the Delta-Multiplexing technique, the saved bandwidth in the payload, and the saved bandwidth in the header. The result showed the total saved bandwidth of between 68% and 72% by multiplexing 10 users in each stream. Fig. 7 depicts the saved bandwidth ratio by using the Delta-Multiplexing technique.

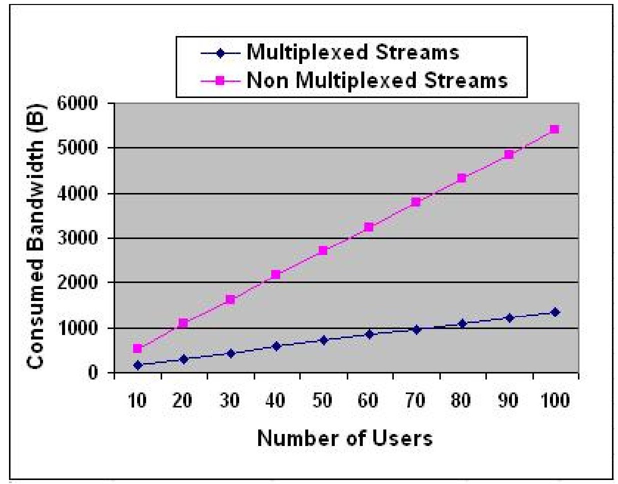

Figure 5. Consumed bandwidth

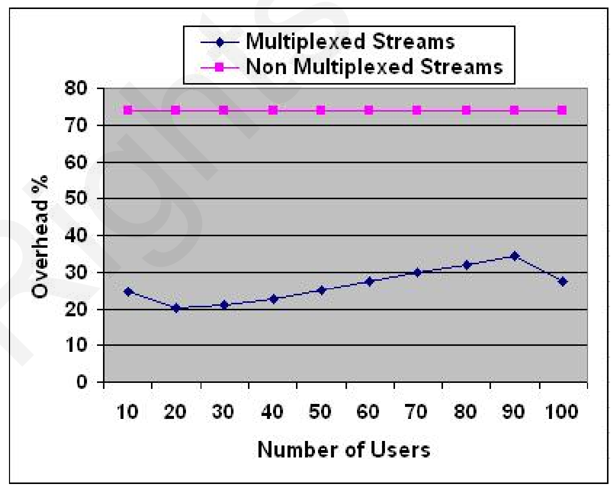

Figure 6. Header overhead

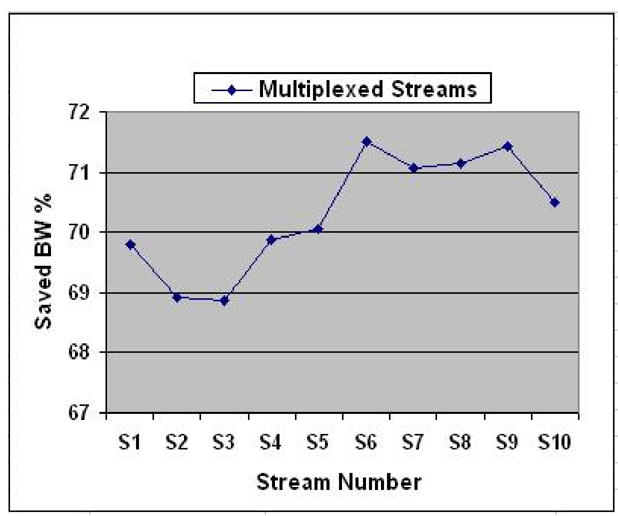

Figure 7. Saved bandwidth ratio

B. Delay

The proposed multiplexing technique, Delta-Multiplexing, has significantly improved bandwidth utilization. Unfortunately, this improvement in bandwidth utilization is a trade-off with the multiplexing period. In such case, when the multiplexing period is increased, the delay will also rise, along with increases in the number of multiplexing packets. Consequently, this decreases the packet overhead and improves bandwidth utilization. Conversely, decreasing the multiplexing period will also decrease the delay. Consequently, this decreases the multiplexing packets and finally increases the overhead, thus reducing bandwidth utilization. However, the Delta-Multiplexing technique reduced the number of injected packets to the network. Thus, the processing delay in the network hops will be reduced since the number of processed packets by the network hops will be lesser, and this compensates the multiplexing period delay.

C. Network Performance

It is clear that the Delta-Multiplexing technique is effective in improving bandwidth efficiency. Moreover, it reduces the number of VoIP packets running between IP telephony GWs, thus reducing network overload and enhancing network performance. The following table, Table 3, summarizes the factors affected by the Delta-Multiplexing technique, which in turn affect network performance.

TABLE III. NETWORK PERFORMANCE

Element	Effect
Bandwidth utilization	Efficient bandwidth utilization
Header overhead	Reduces the header overhead
Number of calls	Increases the number of calls because of efficient bandwidth utilization
Delay	An additional delay can occur because of the processing time and the multiplexing period
Traffic	Reduces the traffic over the network because of the multiplexing of packets between IP-GW
Overload	Reduces the overload because of the reduction in traffic
Congestion	Reduces congestion because of the reduction in traffic
Processing	More hardware performance is required as the processing time becomes longer owing to the multiplexing and D-Alg operations
UDP resources	Saves UDP resources since the Delta-Multiplexing technique combines the multiple streams in one UDP port

VI. CONCLUSION

One of the hindrances of VoIP is inefficient bandwidth utilization. However, there are many techniques that can improve bandwidth utilization. In this paper, we proposed a new multiplexing technique called Delta-Multiplexing. The purpose of this technique is to exploit the available bandwidth with intention of improve the network performance. The Delta-Multiplexing technique consists of two parts. First, VoIP packets are multiplexed in the same route in a single UDP/IP header, hence reducing the header overhead and saving bandwidth. Second, the payload size is reduced through transmitting the difference between consecutive packets, thus saving more bandwidth. Accordingly, the Delta-Multiplexing technique saved between 68% and 72% as compared to conventional techniques. The bandwidth utilization also significantly improved through the reduction of the header overhead and the payload size. Furthermore, the Delta-Multiplexing technique reduces the running VoIP packets over the network, therefore improving the overall network performance and enhancing voice quality.

REFERENCES

[1] G. Thomsen and Y. Jani, "Internet telephony: Going like crazy," IEEE Spectrum, pp. 52–58, May 2000.

[2] Hui Min Chong; Matthews, H.S., "Comparative analysis of traditional telephone and voice-over-Internet protocol (VoIP) systems," Electronics and the Environment, 2004. Conference Record. 2004 IEEE International Symposium on , vol., no., pp. 106–111, May 10–13, 2004.

[3] Bo Li; Hamdi, M.; Dongyi Iang; Xi-Ren Cao; Hou, Y.T., "QoS enabled voice support in the next generation Internet: issues, existing approaches and challenges," Communications Magazine, IEEE , vol. 38, no. 4, pp. 54–61, April 2000.

[4] J Gerard , Waldron, and W. Rachel, "Voice-over-IP: The Future of Communications", April 29, 2002.

[5] A. S. Amin, and H. M. El-Sheikh, "SCALABLE VoIP GATEWAY," ISBN 978-89-5519-131-8 93560, ICACT2007, February 12–14, 2007.

[6] Subbiah, B.; Sengodan, S.; Rajahalme, J., "RTP payload multiplexing between IP telephony gateways," Global Telecommunications Conference, 1999. GLOBECOM '99 , vol. 2, no., pp.1121-1127 vol.2, 1999.

[7] Sze, H.P.; Liew, S.C.; Lee, J.Y.B.; Yip, D.C.S., "A multiplexing scheme for H.323 voice-over-IP applications," Selected Areas in Communications, IEEE Journal on, vol. 20, no. 7, pp. 1,360–1,368, September 2002.

[8] H. Schulzrinne and S. Casner, " RTP Profile for Audio and Video Conferences with Minimal Control," in Proc. IETF RFC 3551, July 2003.

[9] V. Anca, L.Hans Peter, S.Harald, and H.Jan, "VoIP over DVB-RCS A Radio Resource and QoS Perspective", Nera Broadband Satellite, December 2004.

[10] S. Casner and V. Jacobson, “Compressing IP/UDP/RTP headers for lowspeed serial links,” in Proc. IETF RFC 2508, February 1999.

[11] VIVALDI PROJECT, "Advancing interactive Broadband satellite access by optimal convergence of session based services over DVB-RCS", Final Version 14.08.2006.

[12] Sangwan, A.; Chiranth, M.C.; Jamadagni, H.S.; Sah, R.; Venkatesha Prasad, R.; Gaurav, V., "VAD techniques for real-time speech transmission on the Internet," High Speed Networks and Multimedia Communications 5th IEEE International Conference on , vol., no., pp. 46–50, 2002.

[13] Venkatesha Prasad, R.; Sangwan, A.; Jamadagni, H.S.; Chiranth, M.C.; Sah, R.; Gaurav, V., "Comparison of voice activity detection algorithms for VoIP," Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on , vol., no., pp. 530–535, 2002.

[14] Wei Chu; Ahmad, M.O.; Swamy, M.N.S., "Modified silence suppression algorithms and their performance tests," Circuits and Systems, 2005. 48th Midwest Symposium on , vol., no., pp. 436-439 vol. 1, August 7–10, 2005.

[15] T. Hoshi, K. Tanigawa, and K. Tsukada, "Proposal of a method of voice stream multiplexing for IP telephony systems," in Proc. IWS '99, February 1999, pp. 182–188.