A System for Buffer Management in Voice Communication

Ikram Ud Din, Saeed Mahfooz, Noor ul Amin and Muhammad Zeeshan

1Department of Information Technology, University of Haripur, Pakistan
2Department of Computer Science, University of Peshawar, Pakistan
3Department of Information Technology, Hazar University, Mansehra, Pakistan

Abstract: The transmission of voice and video packets through Internet suffers from delay which is needed to be minimized at the time of PlayOut. The packets are received from the network with a varying delay. This delay is called jitter. To surmount the mentioned problems, a system is designed in this work. This technique handles packets in the buffer at the receiver end. The receiver receives packets from the network with a varying delay. To manage delay at the receiver side, the packets are divided into chunks by the system. Size of the chunk is 50 packets. All chunks are of the same size. In case the last chunk is not filled due to less number of packets, then dummy bits are added to the chunk so that the chunk reaches to the maximum negotiated value. This method is called bit padding. The proposed system is simulated in OPNET Modeler 14.0 and compared with the normal flow of voice traffic.

Key words: CJM · Jitter · Delay · Voice · Buffer

INTRODUCTION

Because of the rapid increase of the Internet users, networks are overloaded promptly and still there is exponential increase in the number of users on daily basis. The transmission of voice and video applications is affected extremely with this enormous growth in the number of users. Voice and video packets' transmission has stringent end-to-end delay bounds, therefore, their performance is affected by end-to-end delay and variation in delay which is called jitter. If a packet is received late and the receiver has to display it, then a gap occurs. Three approaches are used for jitter minimization, such as source-based approach [1], Network-Node-based Approach [2-3] and Destination-End-based Approach. The first two techniques are not of our concern, therefore, we will discuss only third technique here.

Destination-end-Based Approach: For reducing gap in PlayOut, a very simple and familiar operation is used called a PlayOut buffer. In this method all incoming packets at the receiver side are queued in a buffer for a period of time before playing out, but this approach introduces latency due to additional buffer delay [4].

A voice or video packet is said to be lost and useless if it arrives after its time is expired. Therefore, a buffer at the receiver is introduced for continuous PlayOut. This buffer allows data to be coded with a variable bit rate; therefore, it improves the overall quality of voice and video [5].

It is clear that most packet communication paths, which are used for voice and video traffic, use jitter buffer located at their end points and mid points. Jitter buffers are implemented in routers, media gateways, or border controllers. Jitter buffers can also be implemented at user devices and customer principle equipment. The aim of jitter buffer is to decrease the effects of delay and variable latency in the transmission of real time applications. The quality of real time traffic can be largely improved by jitter buffers [6]. End-to-end delay means the total time taken by a packet to reach from source to the final destination [7].

Related Work: Many researchers have proposed their own methods for buffer management and delay minimization in real time transmission.

Naylor and his co-author [8] proposed the technique for jitter management which is called I-Policy (Ignored Policy) and E-Policy (Expand Policy). According
to the I-Policy, a packet is discarded when it is received late. In E-Policy, the receiver waits for the late packet and then continues to play these packets. Consequently, all packets are delayed after the late arrived packet. I-Policy has no mechanism for late packets. In E-Policy, the receiver waits for the late packet and hence, increases the waiting time for packets in the buffer, which is a drawback for both of the above mentioned methods.

Zhang and his co-authors [9] have studied jitter management algorithms both for voice and video applications. They have focused on prediction algorithms and evaluated these algorithms using OPNET simulation. They have also compared their results with I-Policy and E-Policy. Zhang and his co-authors’ contribution has concluded that applying prediction algorithm, that is, Least Mean Square (LMS) for jitter management minimizes delay and improves the throughput.

The work of Vitalio A. Reguera and his co-authors [10] evaluates the result of active queue management (AQM) techniques on the QoS on VoIP applications. Their proposed algorithm is based on the fixed point approach for estimation of the users' satisfaction. The empirical evidences and theoretical predictions confirm that the use of AQM presents better QoS than the conventional queue management schemes used in Internet.

Voice packet is normally compressed when it is sent across a voice over IP network or telephony over local area network (ToL). Voice packets are generated at a constant interval when they are sent across a network. However, these packets lose their intervals when they travel on the network.

To overcome the above problem, a Jitter Buffer Adjustment Algorithm was proposed by Mark Grosberg and his co-authors [11] to adjust the depth of the buffer with the help of jitter buffer controller. A cache of the preceding jitter values is maintained by the jitter buffer controller. Two variables called depth and rise are used by the jitter buffer controller to adjust depth of the jitter buffer. A more precise depth is used by these two variables to reduce variation of jitter rates. Many other researchers have proposed some methods for buffer management as in [12-17].

**Working of the Proposed System:** Illustration of the proposed system is shown in Fig. 1. The receiver receives packets from network with a varying delay. To handle delay at the receiver side, the proposed approach divides packets into chunks. The process of chunking and size of the chunks are the two key tasks in this proposed approach [18].

**Explanation**

**Step 1:** The method shows that the system reads packets arriving from the network and then stores them in buffer in the form of chunks.

**Step 2:** As the voice packets arrive, the system collects them and stores in the buffer in the form of equal size chunks.

**Step 3:** The system plays first packet of the first chunk, then second packet of the first chunk and so on. When all packets of the first chunk are played then first packet of the second chunk is started to play, then second packet of the second chunk is played and so on. Thus, all chunks are played out completely.

**Step 4:** As soon as the buffer becomes empty (that is, processing of all packets is completed), then the system is stopped and the process is over.

**Simulation Environment and Results:** The scenario has been tested for each of the following routing protocols: RIP, IGRP, OSPF and EIGRP. Usually network administrators configure RIP or OSPF at routers, but for voice traffic receiving, IGRP (a Cisco proprietary protocol) performs the best [19] as shown in Fig. 2. If we look at Fig. 3, so, for voice traffic end-to-end delay, IGRP also acts well as compared to RIP, OSPF and another Cisco
proprietary protocol: EIGRP [19]. Fig. 4 illustrates the end-to-end delay in voice transmission in which the chunk-based approach is tremendously better than the normal flow of data. The jitter is observed in Fig. 5, in which the proposed algorithm has also incredible supremacy over the normal PlayOut. If one looks at the figures then he/she can compare and determine that the proposed technique is quite better.

CONCLUSION

The communication industries go through a vital improvement very rapidly. Voice communications through cellular communication appliances and Internet became very popular all over the world during the past decade. Before transmitting voice packets, they are encoded and are decoded before PlayOut. During transmission these
packets may also encounter propagation delay. Thus, because of encoding/decoding and propagation delay, these packets in turn face end-to-end delay and jitter which exacerbate voice quality.

To overcome the mentioned problems, an algorithm is proposed in this work. This algorithm treats packets in buffer at the receiver end. The receiver receives packets from network with a varying delay. To handle delay at the receiver side, the proposed approach divides packets into chunks. The average size of the chunk is 50 packets. All chunks are of the same size.

REFERENCES