Delivering Voice over the Internet

Chin-Hwa Kuo, Chia-Lin Chio, Wen-Ching Hsi, and Wei-Hao Chen
Computer and Network (CAN) Laboratory
Department of computer Science and Information Engineering
Tamkang University, Tamshui, Taiwan, CHINA

E-mail: chkuo@mail.tku.edu.tw
Phone: +886 2 2623 8784
Fax: +886 2 2620 9749

ABSTRACT

Due to packet loss and random delay problems in the Internet, the transmission of real-time voice over such an environment is a complex and challenging issue: Many researchers have been devoted to this active research area. In the present paper, we develop a voice-delivering scheme to tackle with the above problems. To overcome the jitter and packet loss, the adaptive voice synchronization scheme is constructed in a feedback configuration. This scheme consists of the following mechanisms: (1) a delay and loss measurement mechanism and (2) a QoS adjustment mechanism. Clearly, voice is a crucial medium for networked multimedia systems. Voice tool is extremely valuable in supporting applications such as computer supported cooperative work (CSCW), Internet telephony, or multimedia instruction on demand (MID). Therefore, the results obtained in the present work are of benefit to many networked multimedia systems.

1. Introduction

The present Internet only offers a point-to-point *best-effort* quality of service, which may present packet delay, delay variation, and packet loss. Much progress has been made in the last several years. The first generation voice conferencing systems, e.g., Nevot [8] and vat [9], were able to deliver good quality of speech in the light network load condition. The second generation voice tools made further improvements, which include (1) low bit rate redundancy packet sending mechanism to overcome packet loss [5][6][12], (2) adaptive playout mechanism to reduce network jitter [3][7], and (3) inter-media synchronization mechanism [4]. In the mean while, many commercial products for voice transmission over the Internet are available, e.g., I-phone, Cool Talk, and Net Meeting. However, up to date, these products may present voice playback difficulties during immediate network load condition.

In the present paper, we design a voice tool to be used over the Internet. The proposed mechanism includes a redundant packet sending mechanism, a network monitoring mechanism, and an adaptive playout scheme. The redundant packet sending mechanism determines the sending packet from a set of compression packet formats to compensate packet loss. The network monitoring mechanism serves the purpose of collecting the present network traffic conditions such as delay, delay jitter, and packet loss. The information will be used in playout time adjustment. The adaptive playout mechanism is used to adjust playout based on the information of network condition. We enhance the adaptive playout scheme proposed by Moon et al. [1], which includes packet loss in the decision of playout time. Thus, the voice playout quality can be further improved.

2. System Overview

We briefly state the related components in our design as follows:

(1) Speech signaling process

The speech signaling process is a pre-process of digitized voice transmission. This may include silence detection, echo cancellation and automatic gain control elements. The implementation of these elements can significantly improve the quality of speech over the network.

(2) Compression and redundant packet generation

To make efficient utilization of network bandwidth, a compression scheme is crucial. Many compression standards are proposed with different computation complexity and compression ratio. To overcome packet loss during transmission, as proposed in [10], different compression schemes may be used to construct a set of packet formats. Then by following different network conditions, different packet formats are sent.

(3) Real-time packet transmission

0-7803-6394-9/00/$10.00 © 2000 IEEE.

We make use of the mechanism of the RTP as the primary vehicle to deliver the real-time multimedia service. Different packet formats are integrated in a packet and then sent by using RFC1889 [11]. In the receiving end, the time stamp can be extracted for further analysis of network condition.

(4) Delay and loss measurement mechanism

Delay and loss measurement mechanism is responsible for the determination of starting packet of a talk spurt, network delay, and redundant packets utilization. Based on each receiving packet, the mechanism updates the network delay statistics profile and records the corresponding sequence as well as packet delay within a designed log array.

(5) QoS adjustment mechanism

Based on the information received from the delay and loss measurement mechanism, the QoS adjustment mechanism determines the available bandwidth and selects a packet sending format from a designed policy table. Then, the corresponding policy number is sent to the sending end to inform the packet sending mechanism. At this time, the mechanism also computes the playout time of the next talk spurt. If a packet does not arrive before playout time, we use duplicate packets to reconstruct those lost packets.

3. Design of Voice Tool

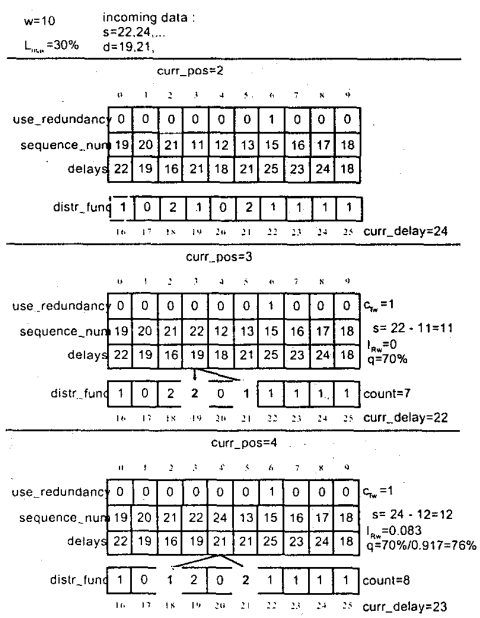

Fig. 1 On-line delay statistics computation method

A simple illustration of playout delay adjustment based on on-line delay statistic is shown in Fig. 1 [1]. The algorithm collects information from arrival packets to estimate playout time. It records each arrival packet delay time and updates the corresponding statistics. For each talk spurt, it computes the statistics from the last w packets to determine the playout delay for a given playout percentage q. This data is then used as the playout time of the next talk spurt. We illustrate the algorithm by using an example. Assuming that $w=10$ and $q=0.7$, with the next consecutive incoming packets equipped with delay to be 19ms, 21ms. As the first packet with a delay of 19ms arrived, the algorithm records it in an array and updates the corresponding value of the distribution array. A counter *count* computes the cumulative value $w*q$ =7 in the distribution array and shifts the curr_delay pointer as current playout delay which satisfies $w*q$. The second packet with a delay of 21ms goes on the same operation. Each time a packet arrives, the estimation value is updated. Thus, the computing result of the last packet during a talk spurt is the playout delay time of the following talk spurt.

3.1 Delay and loss measurement mechanism

The delay and loss measurement mechanism (Fig.2) judges whether the receiving packet is the starting packet of the next talk spurt, and also judges the current network delay state, and then judges whether the system needs to use redundant information of the present packet to reconstruct the lost main information of the past packet. If the network delay state is normal, we refresh the network delay distribution and record the number of the redundant information usage, sequence number, and network delay time of the current packet into the log array (see Fig.1).

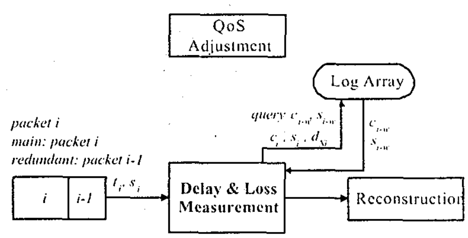

Fig. 2 delay and loss measurement mechanism

As Fig 3 presents, the receiving end first judges the packet to determine if it is the starting packet of a new talk spurt when it receives the packet. The following steps as Fig 4(a)(b) presents the details of a packet, which is a starting talk spurt or not.

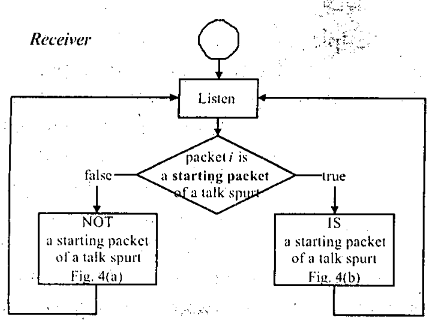

Fig. 3 packet processing flow chart at the receiving end

3.1.1 Not the Starting Packet of the New Talk Spurt

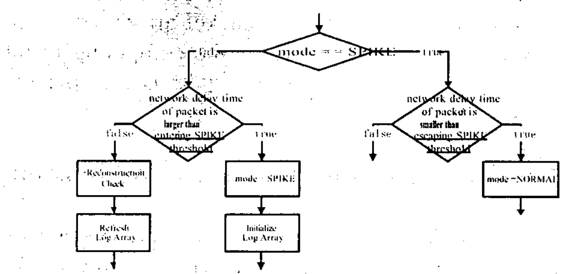

Fig. 4(a) flow chart about packet i is not the starting packet of a new talk spurt

Receiving end receives the incoming packet with arrival time a_i, and gets the timestamp of the packet (the sending time t_i) to calculate the network delay time of the packet d_{Ni}.

$$d_{Ni} = a_i - t_i$$

According to the network delay time of the packet d_{Ni}., the system checks whether the current network state is NORMAL or SPIKE.

Then it checks whether to proceed with reconstruction or not. Firstly, if examines whether the incoming packet i carries the redundant information of the past packet or not. If the incoming packet i does not carry the redundant information of the past packet, set the value of the redundant information usage of the incoming packet c_i to 0. If the incoming packet i carries the redundant information of the past packet, delay and loss measurement mechanism queries Log Array about the loss situation of the past packets. According to the playout time of the current talk spurt, the system checks if the corresponding main information of the redundant information of the packet i is lost or not. Check whether the redundant information of the packet i reconstructs loss information in time to playback to determine the value of the redundant information usage of the incoming packet c_i.

The delay and loss measurement mechanism queries Log Array to obtain the value of the redundant information usage of the w packets before c_{i-w}, and gets current c_{Tw} from system to refresh c_{Tw}.

$$c_{Tw} = c_{Tw} + c_i - c_{i-w}$$

After that, the system updates the network delay distribution, and records c_i s_i d_{Ni} into Log Array.

The delay and loss measurement mechanism queries the Log Array the sequence number of the last w packets. After receiving s_{i-w}, the mechanism takes the sequence number s_i of the RTP header of the packet. After that we can calculate the actual transmission packet number $_i$ at the sending end as the receiving end gets the last receiving w packets when the receiving end gets packet i. Then we can get the network loss rate l_{Nw} of the recent w packets after receiving packet i.

$$\delta_i = s_i - s_{i-w}$$

$$l_{Nw} = 1 - \frac{w}{\delta_i}$$

After that, we can refresh the smoothed loss rate l_{smooth} that represents long-term network loss rate.

$$l_{smooth} = (1-\alpha) * l_{smooth} + \alpha * l_{Nw}$$

Then the data part will return to reconstruction mechanism to be handled.

3.1.2 The Starting Packet of the New Talk Spurt

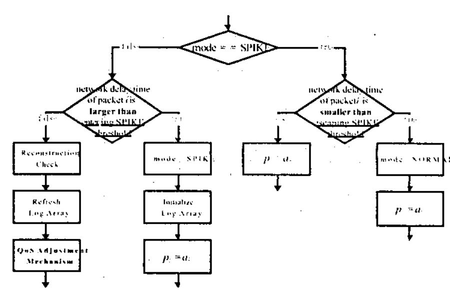

Fig. 4(b) flow chart about packet i is the starting packet of a new talk spurt

After getting c_{Tw} and $_i$ we can calculate the error correction rate e_{Tw} and the loss rate after reconstruction l_{Rw} of the last w packets.

$$e_{Tw} = \frac{c_{Tw}}{\delta_i - w}$$

$$l_{Rw} = l_{Nw}(1 - e_{Tw})$$

Then transferring l_{Nw} l_{Rw} and l_{smooth} to QoS adjustment mechanism and the data part will return to reconstruction mechanism to be handled.

3.2 QoS adjustment mechanism

The Designed QoS adjustment mechanism is similar to [2]. Only as it receives the starting packet of the new talk spurt, the receiving end begins to run QoS adjustment mechanism. As Fig. 5 shows, according to the smoothed network loss rate l_{smooth} that delay and loss measurement mechanism transfers, QoS adjustment mechanism queries the table the corresponding network congestion state in order to determine the bandwidth usage, as table 1 presents.

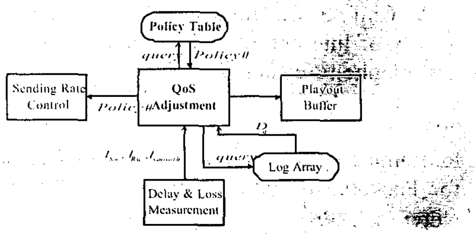

Fig. 5 QoS adjustment mechanism

Table 1 Network Congestion State vs. Bandwidth Usage

Loss Rate	State	Bandwidth	New Bandwidth
< L% L_{max} = a%	Unloaded	Increase	bandwidth = bandwidth+
L% ~ U% L_{max} = b%	Loaded	Maintain	bandwidth = bandwidth
> U% L_{max} = c%	Congested	Reduce	bandwidth = bandwidth*

Choose the larger between l_{Nw} and l_{smooth} to query the table the corresponding number of redundancy, as table 2 presents.

Table 2 Loss Rate vs. Redundant Information Usage

Loss Rate	Primary	1st Redundancy	2nd Redundancy
< L'%			
L'% ~ U'%			
> U'%			

By the usage of bandwidth and redundant information, the mechanism queries policy table all possible combination satisfying the conditions, it chooses the combination with maximum MOS value of main information and redundant information, and then it sends the policy number to the sending end.

Table 3 policy table

Redundancy	Primary	1st redundant	2nd redundant	Bandwidth	Policy
0	PCM	-	-	64	1
0	ADPCM	-	-	32	2
0	GSM	-	-	13	3
1	PCM	ADPCM	-	96	4
1	PCM	GSM	-	77	5
1	PCM	LPC	-	68.8	6
1	ADPCM	GSM	-	45	7
1	ADPCM	LPC	-	36.8	8
1	GSM	LPC	-	17.8	9
2	PCM	ADPCM	GSM	109	10
2	PCM	ADPCM	LPC	100.8	11
2	PCM	GSM	GSM	90	12
2	PCM	GSM	LPC	81.8	13
2	PCM	LPC	LPC	73.6	14
2	ADPCM	GSM	GSM	58	15
2	ADPCM	GSM	LPC	49.8	16
2	ADPCM	LPC	LPC	41.6	17
2	GSM	GSM	GSM	39	18
2	GSM	GSM	LPC	30.8	19
2	GSM	LPC	LPC	22.6	20

According to the loss rate after reconstruction l_{Rw} and the system allowable loss rate at that network congestion state L_{max}, we can get the playout percentage q.

$$L_{\max} \geq l_{Tw}$$
$$L_{\max} \geq l_{Rw} + (1 - l_{Rw}) * l_{Pw}$$
$$L_{\max} \geq l_{Rw} + l_{Pw} - l_{Rw} * l_{Pw}$$
$$L_{\max} \geq l_{Rw} + (1 - q) - l_{Rw} * (1 - q)$$
$$q(1 - l_{Rw}) \geq 1 - L_{\max}$$
$$q \geq \frac{1 - L_{\max}}{1 - l_{Rw}}$$

Then it recursively queries Log Array the distribution in order to get the playout delay D_q of the playout percentage q.

After getting D_q, the QoS adjustment mechanism calculates the total end-to-end delay time of the next talk spurt by the redundant information usage n.

$$d_{Ti} = D_q + d_{Ri}$$

The packet playout method adopts an absolute time approach, that is, the system determines playout time of the other arriving packets of the same talk spurt after the first packet of each talk spurt reaches the receiving end. If the current network congestion state is SPIKE, the playout time of the next talk spurt is the time that the first packet of the next talk spurt reaches the receiving end. If the current network congestion state is NORMAL, the playout time of the next talk spurt is D_q if there is no error recovery mechanism, it is D_q plus d_{Ri} if there is error recovery mechanism.

By the playout mode of absolute time approach, playout time (p_i) of the first packet of each talk spurt is

IF (mode == SPIKE)

$$p_1 = a_1$$

ELSE (mode == NORMAL)

$$p_1 = t_1 + D_q + d_{R1}$$

playout time of the packets except the first one of each talk spurt is

$$p_i = p_i + t_j - t_i$$

4. Simulation results

We performed voice tool testing from Tamkang University, which locates at north of Taiwan, to National Sun Yet-Sen University, which locates at south of Taiwan, on the Internet. The selecting parameters of the proposed algorithm are as follows: = 0.998002 L = L' = 5 U = U' = 25 = 4.8kbps and = 0.875. We made use of 20 ms duration to form a packet size. The results are shown in Fig. 6 for a test which consists of 38,000 packets.

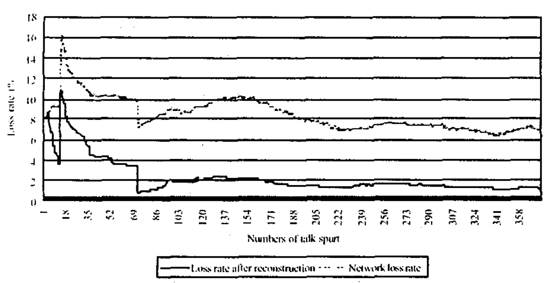

Fig. 6 Testing result of voice tool

The common network packet loss rate is around 20% to 30% the above link (Tamkang University to National Sun Yet-Sen University). As shown in Fig. 6, it indicates that the developed QoS adjustment mechanism reduces the network packet loss rate (ℓ_{Nw}) to 10% (denoted by the dotted line). When the redundant packet reconstruction scheme operates, the packet loss rate (l_{Rw}) decreases to 1% to 2% (denoted by solid line).

5. Conclusions

It goes without saying that due to packet delay, delay jitter, and packet loss problems on the Internet, the transmission of real-time voice over such an environment is a complex and challenging issue. In this paper, we integrate three mechanisms into a whole. (1) QoS adjustment mechanism determines the packet format and information combination strategy by the collecting information. (2) The adaptive playout mechanism, which takes packet loss into account, deals with the voice packet transmission over the Internet. (3) The redundant information sending mechanism is used for solving packet loss problem on the Internet. In our experimental testing on the Internet, the proposed voice tool effectively reduces the packet loss rate from the range of 20 to 30% to the range of 1to 2%. Therefore, very convincing results is achieved.

REFERENCES

1. Sue B. Moon, Jim Kurose, Don Towsley, "Packet Audio Playout Delay Adjustment: Performance Bounds and Algorithms", *ACM Multimedia*, Vol. 6, Jan 1998, pp. 40-48.
2. K. V. Chin, S. C. Hui and S. Foo, "Enhancing the quality of Internet voice communication for Internet telephony systems", *Journal of Network and Computer Applications*, Vol. 21, July 1998, pp. 203-218.
3. R. Ramjee et al, "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Network", *Proc. IEEE Infocom'94*, pp.680-688, Jun 1994.
4. N. Shivakumar, C. J. Sreenan, B. Narendran,P. Agrawal, "The Concord Algorithm for Synchronization of Networked Multimedia Streams", *Proc.ICMCS'95*, pp.31-40.
5. Jean-C. Bolot, Andres V-Garcia, "The Case for FEC-based Error Control for Packet Audio in the Internet", *ACM Multimedia Systems*, 1997
6. Vicky Hardman, Mark Handley, "Reliable Audio for Use over the Internet", *Proc. INET '95, Honolulu, HI*, pp.171-178, Jun 1995.
7. W. Montgomery, "Techniques for Packet Voice Synchronization", *IEEE J. on Selected Area in Communications, Sol. SAC-6, No.1*, pp.1022-1028, Dec 1983.
8. H. Schulzrinnne, "Voice Communication across the Internet: A Network Voice Terminal", *Research report, Dept. Electrical and Computer Engineering, University of Massachusetts*,
9. vat - LBNL Audio Conferencing Tool, http://www-nrg.ee.lbl.gov/vat/
10. Jean-C. Bolot, Andres V-Garcia, "Control Mechanism for Packet Audio in the Internet", *Proc. IEEE Infocom.'96*, pp.232-239
11. H. Schulzrinne, S. Casner, R. Fredreck, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", *RFC 1889, Internet Engineering Task Force*, Feb. 1. 1996.
12. Jean-C. Bolot, Sacha Fosse-Parisis, and Don Towsley, "Adaptive FEC-Based Error Control for Internet Telephony", *IEEE Infocom'99*, New York, NY,March 1999