

# A Packet-Loss-Recovery Scheme for Continuous-Media Streaming Over the Internet

H. P. Sze, S. C. Liew, *Senior Member, IEEE*, and Y. B. Lee

**Abstract**—This letter proposes a new retransmission-based loss-recovery scheme for reliable streaming of continuous-media data over the Internet. The proposed scheme integrates two techniques, namely gap and time-out detection, to detect packet loss. The integrated scheme is ideal for situations in which it is difficult for end users to assess network characteristics (e.g., delay jitter) and for situations in which network characteristics may change drastically over the duration of a streaming session.

**Index Terms**—Packet loss recovery, retransmission, streaming protocol.

## I. INTRODUCTION

**T**IME-SENSITIVE multimedia contents such as audio and video are routinely transmitted over the Internet nowadays. The current Internet does not yet provide comprehensive quality of service (QoS) control. It will be some time before equipment capable of providing QoS becomes prevalent in the Internet because of the large installed base of legacy equipment. With no network-based QoS guarantee, packet loss is inevitable when transmission errors or traffic congestion occurs. To ensure playback quality, one needs to incorporate loss-recovery schemes at the transmitter and receiver ends.

One common loss-recovery scheme is to retransmit the lost packets. For the retransmission to be successful, retransmitted packet must arrive at the receiver in time for playback. Otherwise, the retransmission will have simply wasted the previous bandwidth since late packets will be discarded anyway. To minimize this problem, a buffer is usually set up at the receiver side to prefetch a certain amount of data before playback begins. The buffered data provides additional time to absorb network jitter and delay due to retransmission. Obviously the buffer cannot be too large or excessive startup delay will result. The issue can be phrased as follows: Given a limited buffer size, how can we maximize the number of retransmission attempts for a lost packet before its presentation time so as to reduce the overall data-loss rate during playback?

The retransmission delay is generally determined by two factors, namely network delay and loss-detection delay. Clearly, network delay in today's Internet is not controllable by the application, and minimizing the loss-detection delay is the primary means to improve retransmission performance. For streaming applications, the most common loss-detection techniques are gap detection (GD) and timeout detection (TO).

Manuscript received October 9, 2000. The associate editor coordinating the review of this paper and approving it for publication was Dr. D. Petr.

The authors are with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.

Publisher Item Identifier S 1089-7798(01)02813-7.

Gap detection checks for sequence number gaps in the received packets [1]–[3] and declares packets to be lost when a gap in the received sequence numbers is found. The GD mechanism depends on the continuous transmission nature of audio/video. It could perform poorly under two scenarios. First, the detection delay depends on the inter-packet time, which could be large for low-bitrate streams. Second, packet loss in the Internet often occurs in bursts. Burst loss results in larger gaps and therefore a longer detection delay.

By contrast, the time-out approach [4] detects packet loss by estimating the packet arrival time. If a packet does not arrive by a preset deadline, then it is assumed to be lost. The TO approach is not affected by burst loss and works well for low-bitrate streams. But it does not function so well as GD when the delay jitter is large.

Whether GD or TO achieves a shorter detection delay depends on the network characteristics, which may vary dynamically over time. Typical users may not be able to make the best choice as to which is the better approach. We propose an integrated scheme that combines the virtues of both schemes. We also present in this paper a new way to estimate deadlines for use in TO, and a new scheme to enable GD for lost retransmitted packets.

## II. THE INTEGRATED ERROR-RECOVERY SCHEME

We assume a server-push model for streaming in which the sender periodically transmits data packets to the receiver according to the media bitrate. The receiver sends control messages to the server only when needed (e.g., to initiate pause/resume). The sender retransmits packets when it receives negative acknowledgment (NACK) from the receiver, which are triggered by the detection of packet loss. Both GD and TO loss detection are used concurrently in our scheme. A packet is considered lost when either one of the detection schemes declares it to be lost. A retransmission request, or NACK, is then issued immediately.

### A. Loss Detection of Ordinary Packets

To enable gap detection, the sender stamps each packet with a sequence number before transmission. The receiver then detects packet loss by checking for sequence gaps. For TO detection, assume the sender transmits a new data packet (ordinary packet) once every  $k$  seconds during a session. To initiate a session, the receiver sends a request to the sender. After that, there is no need for it to send further requests for subsequent data packets and the sender will continue to push subsequent packets out periodically. Let  $t_1$  be the time at which the receiver sends

out the initiation request. To determine the packet arrival deadlines, we could imagine that requests for specific packets were indeed generated by the receiver once every  $k$  seconds. The *virtual* request time for packet  $i$  is given by

$$t_i = t_1 + (i - 1)k, \quad i = 2, 3, \dots \quad (1)$$

Let  $A_i$  be the actual arrival time of packet  $i$ . Then the difference, denoted by  $\delta_i$ ,

$$\delta_i = A_i - t_i \quad (2)$$

is the round-trip time (RTT) experienced by packet  $i$ . We call this virtual RTT (VRTT) to differentiate it from the real network RTT. Obviously, the VRTT will vary from packet to packet due to delay jitters. To avoid frequent false alarms of packet loss, the receiver needs to set a deadline with sufficient margin to cater for delay jitters.

We could use a similar deadline estimation problem as in the TCP scheme. The sender under TCP is responsible for the estimation of RTT based on known transmission times and acknowledgment from the receiver. By contrast, our proposed scheme relies on the receiver to estimate the VRTT from data packet arrivals because neither the transmission times are known nor acknowledgment are available. Using derivations similar to TCP [5], the arrival deadline for packet  $(i + 1)$  can be estimated by

$$TO_{i+1} = t_{i+1} + \tilde{\delta}_{i+1} + \beta\sigma_i \quad (3)$$

where  $\tilde{\delta}_{i+1}$  is the estimated VRTT of packet  $(i + 1)$ ;  $\sigma_i$  is the smoothed estimation error of VRTT computed after receiving packet  $i$ ; and  $\beta$  is the timeout multiplier with a typical value set to 4 according to TCP.

### B. Lost Detection of Retransmitted Packets

For detection of the loss of retransmitted packets, the TO scheme as described in Section A can be used directly replacing VRTT with actual RTT. But retransmitted packets are not continuous and it is less obvious how to perform gap detection.

To accomplish gap detection, we introduce a retransmission sequence number (RSN) in every ordinary and retransmitted packets. Unlike ordinary sequence number generated by the sender, RSN's are generated by the receiver. In particular, starting from a value of zero, the RSN is incremented by one every time a retransmission request is issued and is stored in the negative acknowledgment (NACK) packet. Once the sender receives the NACK, the retransmitted packet and all subsequent ordinary packets will be marked with the new RSN until the next NACK packet arrives. That is, the sender embeds the same RSN value into the transmitted packets until it receives a NACK, at which point the RSN is updated and the next transmitted packet, which is a retransmission of the lost packet, carries the new RSN.

To detect loss of a retransmitted packet, a receiver can simply inspect the RSN embedded in each received packet to look for RSN increases. Note that whenever the RSN is incremented and a NACK packet is issued, the retransmitted packet corresponding to the NACK should be the first packet to arrive at the

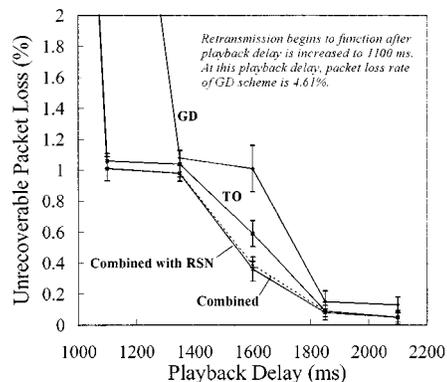


Fig. 1. Performance comparison of various schemes in low-bitrate streaming and low network jitter.

receiver with the new RSN. So, if the RSN increase is observed in a packet other than the expected-arrived retransmitted packet, then the receiver can assume that the retransmitted packet has been lost.

## III. EXPERIMENTAL RESULTS

To compare the performance of the proposed error-recovery scheme with the GD and TO schemes, we have conducted experiments on a three-node testbed consisting of three machines, serving as the sender, router, and receiver. The sender is connected to the router via a FastEthernet link, and the router in turn connects to the receiver via another FastEthernet link. The use of high-speed network links avoids possible congestion at the link level, allowing us to simulate various link parameters such as bandwidth, delay, and packet loss simply by throttling, delaying, or discarding packets at the router.

Two sets of experiments have been performed on this testbed. To increase the transmission delay of packets, an artificial delay of 500 ms is introduced at the router. The first experiment simulates audio streaming at a bitrate of 128 kbps. Data packets are 1 KB (payload) each, translating into a packet inter-departure time of 62 ms. To simulate delay jitter, we set the router bandwidth to 500 Kbps and introduce an on-off interference source injecting an average data rate of 128 Kbps into the router. The peak rate is 450 kbps, with average ON and OFF periods equal to 34 and 85 ms, respectively. Thus, when the interference source is on, the aggregated data rate of the audio stream and the interfering stream exceeds the router bandwidth and delay builds up. The standard deviation of packet delay in this configuration is 5.5 ms. Packet loss is simulated by discarding packets randomly at the router at a rate of 10%. Occasionally, burst loss occurs due to queue overflow at the router.

Fig. 1 summarizes the results of the first experiment, plotting the residual packet loss rate against playback delay (from sending of first request packet to start of playback). The error bars are 95% confidence intervals. In all cases, the residual loss rate (packet loss despite of retransmission mechanism) decreases with the playback delay. But we observe that the proposed scheme consistently achieves lower loss rate compared to the GD and TO schemes. Note that GD has the largest residual loss rate due to the low audio bitrate.

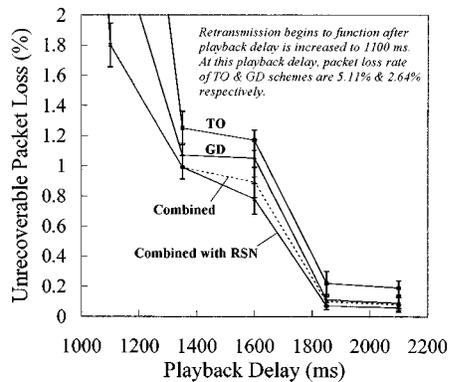


Fig. 2. Performance comparison of various schemes in high-bitrate streaming and high network jitter.

The second experiment simulates video streaming at a bitrate of 300 kbps. Data packets are again 1 KB (payload) each, translating into a packet inter-departure time of 27 ms. The router bandwidth is still set to 500 Kbps, with two interference sources each averaging at 100 kbps to increase the jitter. This time the peak rate of each interference traffic is 350 kbps, and the average ON and OFF periods are 48 ms and 120 ms, resulting in a standard deviation of packet delay equal to 14 ms. As shown in Fig. 2, again the proposed scheme consistently outperforms the other two schemes. In this second experiment, the use of RSN is shown to further reduce the unrecoverable loss rate. Note that the GD scheme performs better than the TO scheme in this configuration, due to the increased video bitrate.

Overall, from the results of both experiment sets, we observe that the percentage decrease in unrecoverable loss rate between the worst scheme in each set and the proposed scheme with RSN

is more than 60% for over half of the data points. This shows that the new scheme is effective in reducing unrecoverable packet loss.

#### IV. CONCLUSIONS

We have investigated a new retransmission-based loss-recovery scheme for reliable streaming over the Internet. The proposed loss-detection scheme combines the virtues of both gap detection and timeout detection schemes to shorten the loss-detection delay. It automatically adopts the better of the two schemes in a dynamic fashion. This new loss-recovery scheme is ideal for situations in which it is difficult for users to assess network characteristics (e.g., delay jitter) and make a wise choice as to whether the gap detection or timeout detection should be adopted. It is also particularly suitable for situations in which network characteristics may change drastically over the duration of a streaming session.

#### REFERENCES

- [1] I. Rhee, "Retransmission-based error control for interactive video applications over the internet," in *Proc. Multimedia Computing and Systems*, 1998, pp. 118–127.
- [2] B. Dempsey, J. Liebeherr, and A. Weaver, "On retransmission-based error control for continuous media traffic in packet-switching networks," *Computer Networks and ISDN Syst.*, vol. 28, no. 5, pp. 719–736, Mar. 1996.
- [3] S.-H. Lee and S. Lee, "Retransmission scheme for MPEG streams in mission critical multimedia applications," in *Proc. Euromicro Conf.*, 1998, pp. 574–580.
- [4] P. Wu and S. C. Liew, "A streaming-protocol retransmission scheme without client-server clock synchronization," *IEEE Commun. Lett.*, vol. 3, July 1999.
- [5] V. Jacobson, "Congestion avoidance and control," in *ACM Proc. SIGCOMM '88*, Stanford, CA.