Analysis of Audio Packet Loss in the Internet

Jean-Chrysostome Bolot Hugues Crépin Andres Vega Garcia

INRIA
B. P. 93
06902 Sophia-Antipolis Cedex
France

Abstract. We consider the problem of distributing audio data over networks such as the Internet that do not provide support for real-time applications. Experiments with such networks indicate that audio quality is mediocre in large part because of excessive audio packet losses. In this paper, we show using measurements over the Internet as well as analytic modeling that the number of consecutively lost audio packets is small unless the network load is very high. This indicates that open loop error control mechanisms based on forward error correction would be adequate to reconstruct most lost audio packets.

1 Introduction

We consider the problem of distributing real-time data, and specifically audio data, over networks such as the Internet that do not provide guaranteed resources such as bandwidth or guaranteed performance measures such as maximum delay or maximum loss rate. Two approaches have emerged to support real-time applications over these networks.

One approach is to extend current protocols and switch scheduling disciplines to provide the desired performance guarantees. This approach requires that admission control, policing, reservation, and/or sophisticated scheduling mechanisms be implemented in the network. The design, analysis, and evaluation of such mechanisms is an active research area [11]. Another approach is to adapt applications to the service provided by the network. One way to do this is to control the rate at which packets are sent over a connection, the objective being to limit this rate to the capacity of the connection. In FIFO networks such as the Internet, this capacity changes with time because the number of connections routed through each switch varies and because sources do not send data at a constant rate. The control mechanism must be somehow informed of such changes. One way is for sources of packets to receive feedback about the state of the network. Feedback control mechanisms have been used in the Internet to control sources of non real time traffic (e.g. the window control mechanism in TCP). Recently, they have been advocated and used to control video [10, 4] and audio [19] traffic as well.

Such control mechanisms help reduce packet losses, but they do not prevent them entirely. In fact, the experience accumulated over the Internet indicates that audio packet losses are in large part responsible for the mediocre quality of many audio transmissions over the network.

In this paper, we characterize the packet loss process of audio streams sent over the Internet. We use measurements (in Section 2) as well as analytic modeling (in Section 3). The analytical results tie in well with the measurements. Our main result is that the number of consecutively lost audio packets is small except when the Internet load is high. This suggests that open loop error control mechanisms based on forward error correction would be adequate to reconstruct most lost audio packets. We briefly describe (in Section 4) one such mechanism and report on preliminary evaluations of audio quality obtained with this mechanism.

2 Measurements

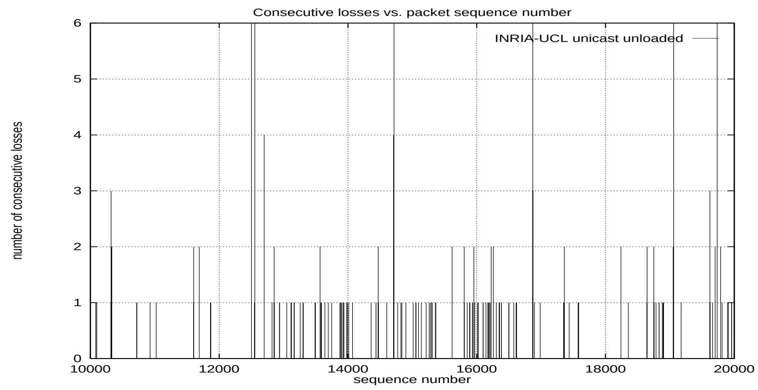
We have measured the loss process of audio packets using the audio tool of IVS (INRIA Videoconference System). IVS [18] is a software system combining a H.261 video codec and a number of audio codecs (PCM, DPCM, ADPCM, and LPC codecs). In all the experiments described in this paper, we used the PCM (Pulse Coded Modulation) coder. Each audio packet sent by this coder includes 320 voice samples, or 40 ms of speech. Therefore, voice packets are sent periodically every 40 ms during talkspurts.

We have measured the audio loss process over a number of connections, i.e. of source-destination pairs. Because of lack of space, we focus on one specific connection between INRIA Sophia Antipolis in southern France and University College London (UCL) in the UK. This connection is heavily loaded during daytime. We have examined it both in unicast and multicast mode. In unicast mode, audio packets are sent from a source at INRIA to a destination at UCL using the standard protocol stack of the Internet, i.e. IP/UDP/RTP. In multicast mode, audio packets are sent between the same source destination pair as in unicast mode, but they are sent over the MBone (i.e. the virtual network running on top of the IP layer which provides a multicasting facility in the Internet).

Each audio packet includes a sequence number to detect packet losses. For convenience of presentation, we introduce a boolean variable l_n which is set to 1 if packet n is lost, and 0 otherwise. We record for every packet, i.e. every n, the value of l_n. Many different measures can be used to characterize the loss process of audio packets. The obvious measure is the average loss, or unconditional loss probability, which is the expected value of l_n. We denote $ulp = E[l_n]$. However, ulp does not characterize the burstiness of the loss process, or equivalently the correlation between successive packet losses. One way to capture such correlation is to consider the conditional probability that a packet is lost given that the previous packet was lost. We denote $clp = P[l_{n+1} = 1 | l_n = 1]$. A related measure is the expected number of consecutively lost packets, or packet loss gap [6], which we denote by plg. If the sequence of d_n is stationary and ergodic, then [1] $plg = 1/(1 - clp)$.

Unicast mode

Figure 1 shows the evolutions of the number of consecutively lost packets as a function of n measured at 8:00 am and at 4:00 pm. As expected, the total number



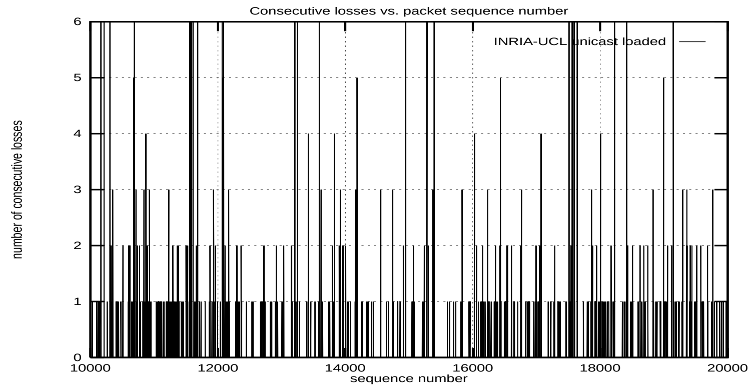
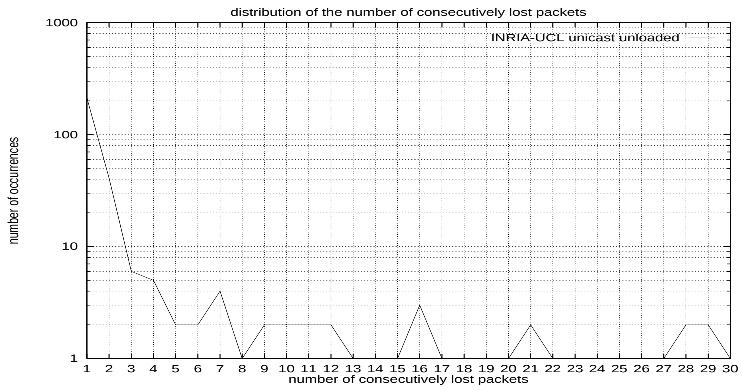


Fig. 1. Evolutions of the number of consecutively lost packets at 8:00 am (top) and 4:00 pm (bottom)

of losses is higher at 4:00 pm, i.e. when the network load is higher. In the top graph, we observe that most losses are isolated except for "almost periodic" loss periods when the number of consecutive losses exceeds 6 (in fact, this number can exceed 100). These loss periods occur around packet number 12500, 14700, 16900, and 19000. The interval between periods is thus approximately 40 ms $\times$ 2200 $\approx$ 88 s. Therefore, it seems likely that these losses would be triggered by the periodic overload caused at routers by the synchronization of IGRP routing update messages every 90 seconds[1] [8]. The observations we make for the top graph are still valid for the bottom graph, i.e. when the network load is high. In particular, it appears that the average number of consecutively lost packets is small.

This observation is confirmed by looking at the frequency distribution of the number of consecutive losses. Figure 2 shows the distributions (i.e. the number of occurrences of n consecutive losses for different n) corresponding to the traces in Figure 1. We observe that the slope of the distribution decreases linearly near



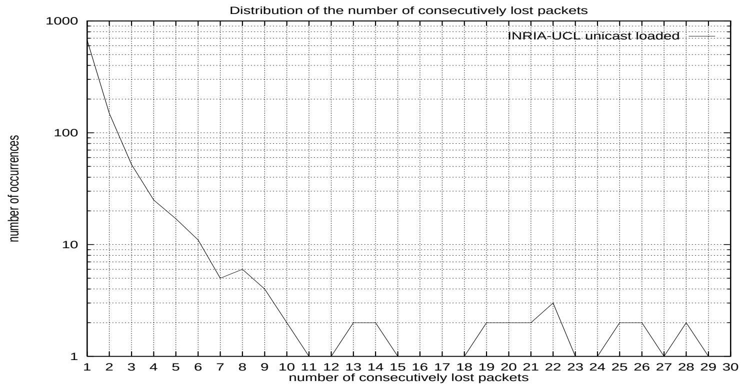
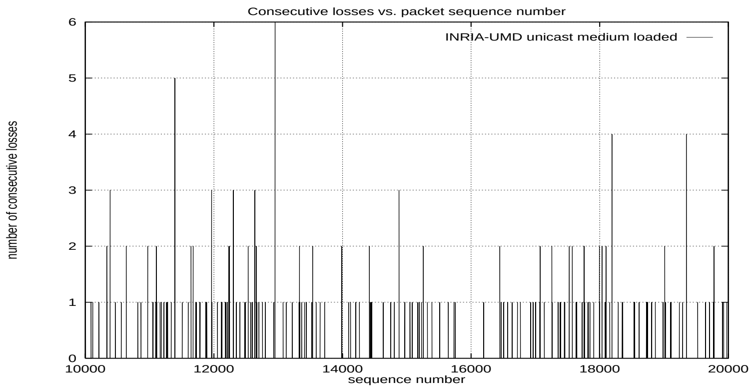


Fig. 2. Frequency distribution of the number of consecutive lost packets at 8:00 am (top) and 4:00 pm (bottom)

the origin. Since the figures are drawn on a log scale, this indicates that the probability distribution decreases geometrically fast away from the origin.

[1] This synchronization problem has very recently been eliminated on the French backbone.

We have examined the loss process of audio packets over connections other than the INRIA-UCL connection. In all cases, we have found that the frequency distribution of the number of consecutively lost packets is similar to that described above. To illustrate this, we next present the evolutions and the frequency distribution of the number of consecutively lost packets measured over a connection between INRIA and the University of Maryland. The measurements were taken at 3:00 pm local time (9:00 am EST) when we expect the transatlantic link between France and the US to be reasonably loaded.



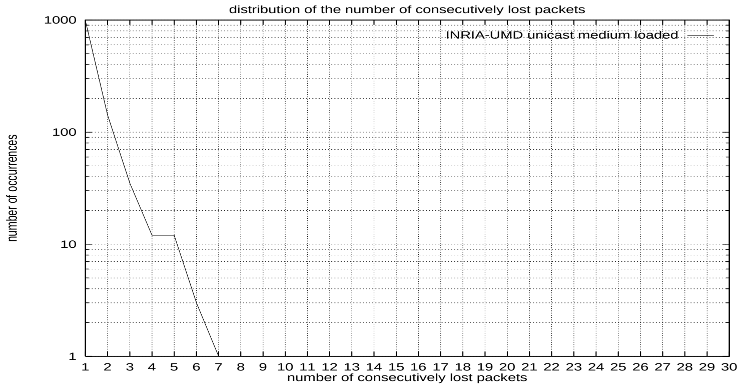
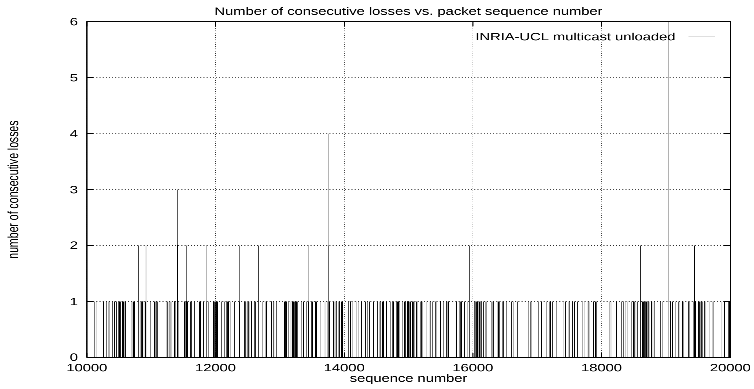


Fig. 3. Evolutions (top) and frequency distribution (bottom) of the number of consecutive lost packets measured over the INRIA-UMD connection

Multicast connections

The results obtained for multicast connections are essentially the same as those obtained for the unicast connections. Figure 4 shows the evolutions of the number of consecutively lost packets between INRIA and UCL as a function of n in multicast mode at 8:00 am and at 4:00 pm.



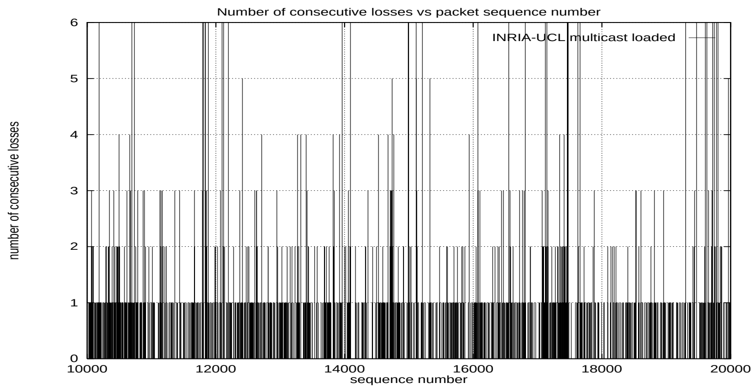
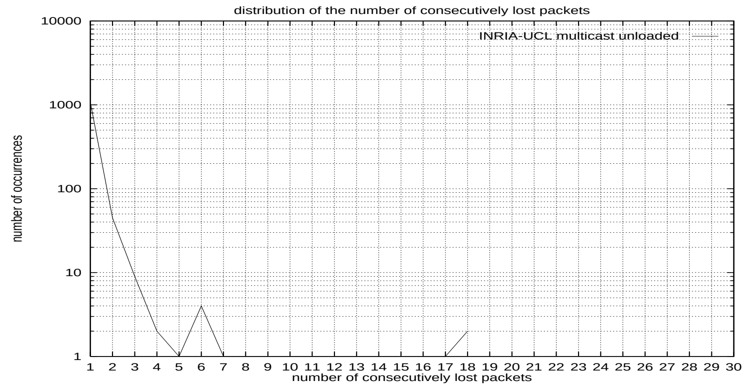


Fig. 4. Evolutions of the number of consecutively lost packets at 8:00 am (top) and 4:00 pm (bottom)

The observations we made in Figure 1, namely that most losses appear to be isolated[2], are still valid in Figure 4. These observations are confirmed by the shapes of the frequency distributions of the number of consecutive losses.

[2] Interestingly, the "almost periodic" loss periods we observed earlier have disappeared. This is because the path followed by the audio packets on the MBone between a source and a destination has little to do with the path that would be taken by packets between the same source and the same destination using the standard IP routing. Furthermore, much of the queueing in the MBone is done at the gateways running *mrouted* daemons. Many such gateways are workstations which are therefore not affected by phenomena such as the routing update synchronization visible in "standard" IP routers.



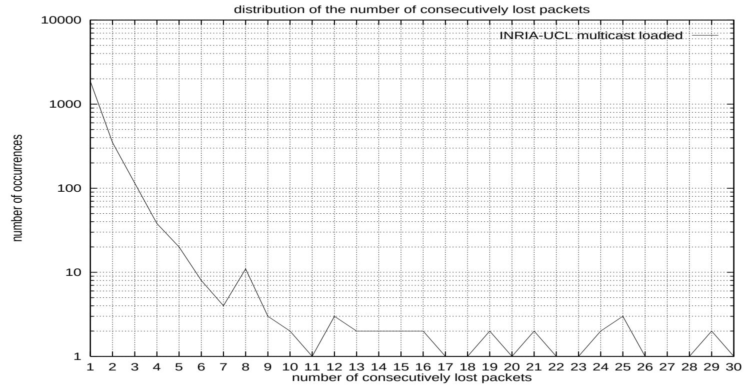


Fig. 5. Distribution of the number of consecutive lost packets at 8:00 am (top) and 4:00 pm (bottom)

In summary, the results presented in this section indicate that the number of consecutively lost packets is small especially when the network load is low or moderate. This is in agreement with previous experimental results obtained with non-audio UDP packets over many different connections in the Internet [3].

Note, however, that we have not defined what is meant by "low" or "moderate" load above. This is because it is essentially impossible for us to control the load of the network. We controlled it in a simple (but obviously very rough) way by modifying the time at which experiments were carried out.

In the next section, we present a model that captures the impact of the Internet traffic on a flow of audio packets sent from a source to a destination. The model can be solved analytically in part, and it provides a mathematical basis for the conclusions above. Furthermore, it can be used to evaluate the impact of the network load on the audio packet loss process in a more flexible way than can be done with experiments.

3 A Discrete-Time Model

The model is based on the result obtained in [3], where it is shown that the impact of the Internet traffic on a periodic stream of packets can be approximated by that of a batch Bernoulli traffic. Therefore, we model our experiments with a simple single-server queueing model with 2 input streams, where one stream represents the audio traffic and the other stream represents the Internet traffic. We note that PCM-coded voice audio streams are not expected to be continuous because of silence detection at the coder. We have considered models where the audio stream is modeled as an on/off process. However, the results regarding the loss process are essentially identical to those obtained with the periodic model. Our model is shown in Figure 6. The delay D represents the fixed part of the total end to end delay of audio packets, i.e. the propagation and transmission delays.

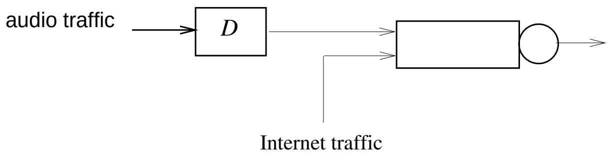


Fig. 6. A model for our experiments

The time axis is assumed to be slotted. Packets arrive at the beginning of a slot and are serviced at the end of a slot. Audio packets arrive every T slots, and Internet packets arrive every slot. Packet sizes are expressed in multiple of a base size equal to P bytes which corresponds to a service time equal to one slot. For convenience, we refer to this base size unit also as a slot. Audio packets have a fixed size of A slots (i.e. $A \times P$ bytes). In practice, a slot is equal to 32 bytes, and hence $A = 10$. Internet packets have independent and identically distributed sizes drawn from general distribution. Specifically, we denote by b_i the probability that the size of the Internet packet is $i \times P$ bytes. In [3], it is shown how measures of end to end delays of audio packets can be used to find the value of b_i. Our measures of end to end delays indicate, not surprisingly, that the Internet packet size distribution is a multimodal distribution. In our model, we take the size of Internet packets to be equal to either 1 slot (this models 32-byte Telnet packets), or a multiple of 16 slots (this models 512-byte FTP and SMTP packets). We assume that the buffer can hold $K > A$ slots.

Throughout the rest of the paper, we assume that an audio packet arrives as the last packet during a slot. Note that if the audio packet arrives first, it is always admitted. Thus, we obtain the $D + D^X/D/1/K$ model shown in Figure 7.

Many models similar to that in Figure 7 have been studied assuming an infinite buffer size. The finite buffer case has been considered in [13]. In this

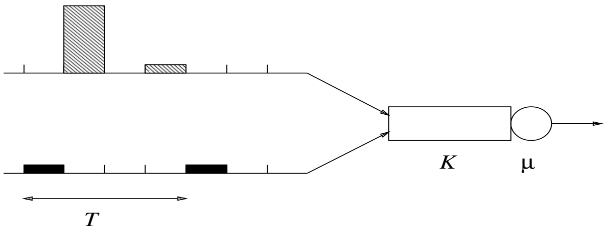


Fig. 7. The $D + D^X/D/1/K$ model

paper, we use an approach based on state transition matrices similar to that used in [16]. The state of the system is defined as the workload (expressed in terms of slots) before the arrival of packet(s) in a slot. Thus the state ranges over $[0...K-1]$. The $K \times K$ transition matrix for the Internet stream assuming no audio arrival is

$$Q = \begin{bmatrix} b_0 + b_1 & b_2 & \ldots & b_{K-1} & 1 - \sum_{i=0}^{K-1} b_i \\ b_0 & b_1 & \ldots & b_{K-1} & 1 - \sum_{i=0}^{K-1} b_i \\ 0 & b_0 & \ldots & b_{K-1} & 1 - \sum_{i=0}^{K-2} b_i \\ & & \ddots & & \vdots \\ 0 & 0 & \ldots & b_0 & 1 - b_0 \end{bmatrix}$$

The $K \times K$ transition matrix assuming an audio arrival is

$$Q\prime = \begin{bmatrix} 0\ 0 \ldots b_0 & b_1 & \ldots & b_{K-A-1} & 1 - \sum_{i=0}^{K-A-1} b_i \\ 0\ 0 \ldots 0 & b_0 & \ldots & b_{K-A-2} & 1 - \sum_{i=0}^{K-A-2} b_i \\ 0\ 0 \ldots 0 & 0 & \ldots & b_{K-A-3} & 1 - \sum_{i=0}^{K-A-3} b_i \\ \ddots & & \ldots & & \vdots \\ 0\ 0 \ldots & & \ldots & & 1 \end{bmatrix}$$

Note that if only $x < A$ slots are available when an audio packet arrives, then all x slots are filled with whatever fraction of the audio packet can fit in. Thus, the audio packet is not entirely discarded.

The transition matrix between two audio arrivals is $P = Q\prime Q^{T-1}$. The steady state probability vector π seen by the first bit in a slot with an audio arrival is the solution of $\pi = \pi P$. The distribution of the state ξ seen by an audio packet is thus $\xi_i = \sum_{k=0}^{i} \pi_k b_{i-k}$

We can now derive various performance measures. The unconditional loss probability for audio packets is given by $ulp = \sum_{i=K-A+1}^{K} \xi_i$. To compute the conditional loss probability clp, we note that the state of the system after an audio packet loss is always $K-1$. Thus clp is obtained as the sum of the $A-1$ rightmost elements on the bottom row of the transition matrix $Q^{T-1}R$ where R is the matrix describing the transition caused by the arrivals preceding an audio packet. We do not write R for lack of space, but it is easy see that it is almost identical to Q. The average number of consecutively lost packets is then $plg = 1/(1-clp)$.

The Internet load , i.e. the traffic intensity of the Internet stream, is given by expected value of b_i. Figure 8 shows the variations of plg as a function of the Internet load, computed from the model above. We observe that plg stays

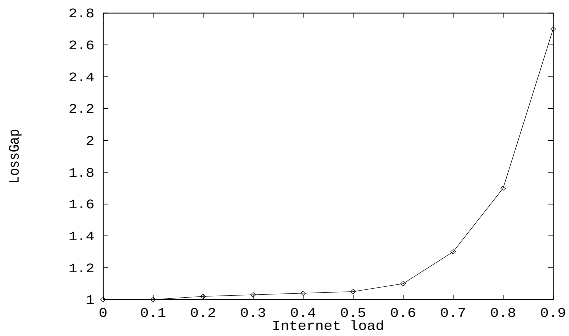


Fig. 8. Variations of plg as a function of Internet load

close to 1 unless the load generated by the Internet traffic exceeds 0.8. This is confirmation that we should expect the average number of consecutive losses to be small except when the load in the network Not surprisingly, our results tie in well with others in [16] obtained with a similar model which indicate that losses for a periodic stream (e.g. an audio stream) are isolated as long as the periodic stream uses less than 10% of the server capacity. They are also in agreement with other results in the literature [14, 16, 3].

Interestingly, packet losses are generally thought to be “bursty”. However, it is important to consider carefully the hypotheses made regarding the traffic model [5]. Furthermore, many analytic studies consider the aggregate loss process at a node, i.e. the loss process from all input streams. In the model above, we evaluate the loss process of the audio stream *only*.

4 Packet Loss Recovery

An error control mechanism is required if the number of lost audio packets is higher than that tolerated by the listener at the destination. Typical mechanisms fall in one of two classes. Automatic Repeat Request (ARQ) mechanisms are closed-loop mechanisms based on the retransmission of the packets that were not received at the destination. Forward Error Correction (FEC) mechanisms are open-loop mechanisms based on the transmission of redundant information along with the original information so that (at least some of) the lost original data can be recovered from the redundant information. ARQ mechanisms are not acceptable for live audio applications such as audioconferencing over the Internet because they dramatically increase end to end latency. Furthermore, they are not well suited to multicast environments such as the MBone.

FEC is an attractive alternative to ARQ for providing reliability without increasing latency. This is particularly important for applications with real time constraints over high speed networks (e.g. [17, 2]). However, the potential of FEC mechanisms to recover from losses depends crucially on the characteristics of the packet loss process in the network. Clearly, FEC mechanisms are more effective when lost packets are dispersed throughout the stream of packets sent from a source to a destination. Our results above then indicate that FEC is particularly well suited for live audio applications over the Internet.

The simplest way to add redundancy to an audio packet is to add no redundancy at all. Indeed, it is possible to recover at the destination from packet losses without any redundant information. For example, a lost packet can be replaced by silence or by noise. A better way is to reconstruct it by duplicating the previous packet (assuming this packet was actually received) [9].

A large variety of more elaborate FEC mechanisms have been proposed in the literature. Many such mechanisms involve exclusive-OR operations, the idea being to send every nth packet a redundant packet obtained by exclusive-ORing the other n packets [17]. This mechanism can recover from a single loss in a n packet message. It is a very simple mechanism, but it increases the send rate of the source by a factor of $1/n$, and it adds latency since n packets have to be received before the lost packet can be reconstructed.

Within the MICE project (a European project devoted to multimedia research) [12], we have developed a novel mechanism for loss recovery. Consider for example the case when audio is sent using PCM encoding. In our mechanism, packet n includes in addition to its encoded PCM samples, a highly compressed version of packet $n-1$. This redundant information would typically be obtained using linear predictive coding (LPC) [15] of packet $n-1$. Clearly, this mechanism can recover from isolated losses. It can also recover from two consecutive losses. One way of doing this is to combine it with packet repetition. Another way is to use as redundant information in packet n a LPC version of packet $n-2$.

The impact of this mechanism on the quality of the audio delivered over the Internet is being carefully evaluated by our MICE colleagues at UCL. They have obtained preliminary results which show that audio quality as measured by intelligibility is much improved (the details of the study are in [7]) even when a relatively low quality LPC coder is used to obtain the redundant information. These results provide added motivation to design even more efficient loss recovery schemes.

References

1. F. Baccelli, P. Brémaud, *Palm Probabilities and Stationary Queues*, Lecture Notes in Statistics, vol. 41, Springer, Heidelberg, 1987.
2. E. W. Biersack, "Performance evaluation of FEC in ATM networks", *Proc. ACM Sigcomm '92*, pp. 248-257, Baltimore, MD, Aug. 1992.
3. J-C. Bolot, "End-to-end packet delay and loss behavior in the Internet", *Proc. ACM Sigcomm '93*, San Fransisco, CA, pp. 189-199, Aug. 1993.
4. J-C. Bolot, T. Turletti, "A rate control scheme for packet video in the Internet", *Proc. IEEE Infocom '94*, Toronto, Canada, pp. 1216-1223, June 1994.
5. I. Cidon, A. Khamisy, M. Sidi, "Analysis of packet loss processes in high-speed networks", *IEEE Trans. Info. Theory*, vol. 39, no. 1, pp. 98-108, Jan. 1993.
6. J. Ferrandiz, A. Lazar, "Monitoring the packet gap of real-time packet traffic", *Queueing Systems*, vol. 12, pp. 231-242, Dec. 1992.
7. V. Hardman et al., "Reliable audio for use over the Internet", to appear in *Proc. INET '95*, Hawai, June 1995.
8. V. Jacobson, S. Floyd, "The synchronization of periodic routing messages", *Proc. ACM Sigcomm '93*, San Fransisco, CA, pp. 33-44, Sept. 1993.
9. N. Jayant, "Effects of packet losses in waveform-coded speech", *IEEE Trans. Comm.*, vol. COM-29, pp. 101-109, Feb. 1981.
10. K. Jeffay, D. L. Stone, T. Talley, F. D. Smith, "Adaptive, best-effort delivery of digital audio and video across packet-switched networks", *Proc. NOSSDAV '92*, San Diego, CA, Nov. 1992.
11. J. Kurose, "Open issues and challenges in providing QoS guarantees in high-speed networks", *Computer Comm. Review*, vol. 23, no. 1, pp. 6-15, 1993.
12. The MICE home page is found at URL http://www.cs.ucl.ac.uk/mice. See also http://www.inria.fr/rodeo/personnel/huitema/audio.html
13. M. Murata, Y. Oie, T. Suda, "Analysis of a discrete-time single-server queue with bursty inputs for traffic control in ATM networks", *IEEE JSAC*, vol. 8, no. 3, pp. 447-458, April 1990.
14. I. Norros, J. Virtamo, "Who loses cells in the case of burst scale congestion", *Proc. ITC 13*, Copenhagen, pp. 829-833, June 1991.
15. L. R. Rabiner, R. W. Schafer, *Digital Processing of Speech Signals*, Prentice Hall, 1978.
16. H. Schulzrinne, J. Kurose, D. Towsley, "Loss correlation for queues with bursty input streams", *Proc. IEEE ICC '92*, Chicago, IL, pp. 219-224, 1992.
17. N. Shacham, P. McKenney, "Packet recovery in high-speed networks using coding and buffer management", *Proc. IEEE Infocom '90*, San Fransisco, CA, pp. 124-131, May 1990.
18. T. Turletti, "A H.261 software codec for videoconferencing over the Internet", INRIA Research Report 1834, January 1993.
19. N. Yin, M. G. Hluchyj, "A dynamic rate control mechanism for source coded traffic in a fast packet network", *IEEE JSAC*, vol. 9, no. 7, pp. 1003-1012, Sept. 1991

This article was processed using the LTEX macro package with LLNCS style