A Survey of Packet Loss Recovery Techniques for Streaming Audio

Colin Perkins, Orion Hodson, and Vicky Hardman
University College London

Abstract

We survey a number of packet loss recovery techniques for streaming audio applications operating using IP multicast. We begin with a discussion of the loss and delay characteristics of an IP multicast channel, and from this show the need for packet loss recovery. Recovery techniques may be divided into two classes: sender- and receiver-based. We compare and contrast several sender-based recovery schemes: forward error correction (both media-specific and media-independent), interleaving, and retransmission. In addition, a number of error concealment schemes are discussed. We conclude with a series of recommendations for repair schemes to be used based on application requirements and network conditions.

The development of IP multicast and the Internet multicast backbone (Mbone) has led to the emergence of a new class of scalable audio/video conferencing applications. These are based on the lightweight sessions model [1, 2] and provide efficient multiway communication which scales from two to several thousand participants. The network model underlying these applications differs significantly from the tightly coupled approach in use for traditional conferencing systems. The advantage of this new, loosely coupled approach to conferencing is scalability; the disadvantage is unusual channel characteristics which require significant work to achieve robust communication.

In this article we discuss the loss characteristics of such an IP multicast channel and how these affect audio communication. Following this, we examine a number of techniques for recovery from packet loss on the channel. These represent a broad cross-section of the range of applicable techniques, both sender-driven and receiver-based, and have been implemented in a wide range of conferencing applications, giving operational experience as to their behavior. The article concludes with an overview of the scope of applicability of these techniques and a series of recommendations for designers of packet-based audio conferencing applications.

A number of surveys have previously been published in the area of reliable multicast and IP-based audio-video transport. The work by Obraczka [3] and Levine and Garcia-Luna-Aceves [4] is limited to the study of fully reliable transport and does not consider real-time delivery. The survey by Carle and Biersack [5] discusses real-time IP-based audio-video applications and techniques for error recovery in this environment. However, that work neglects receiver-based error concealment techniques and focuses on sender-driven mechanisms for error correction.

Sender-driven and receiver-based repair are complementary techniques, and applications should use both methods to

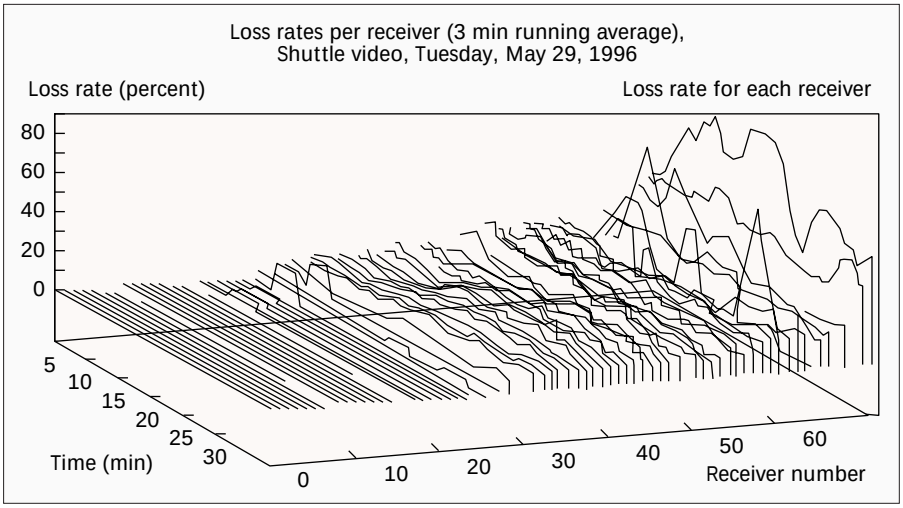


Figure 1. *Observed loss rates in a large multicast conference (from [7]).*

0890-8044/98/$10.00 © 1998 IEEE

achieve the best possible performance. In contrast to previous work, we limit the focus of our article to streaming audio applications, and discuss both sender-driven repair and receiver-based error concealment techniques.

Multicast Channel Characteristics

The concept of IP multicast was proposed by Deering [6] to provide a scalable and efficient means by which datagrams may be distributed to a group of receivers. This is achieved by imposing a level of indirection between senders and receivers: packets are sent to a group address, receivers listen on that same address and the network conspires to deliver packets. Unless provided by an application-level protocol, the senders and receivers are decoupled by the group address: a sender does *not* know the set of hosts which will receive a packet. This indirection is important: routing decisions and recovery from network outages are purely local choices which do not have to be communicated back to the source of packets or to any of the receivers, enhancing scalability and robustness significantly.

Internet conferencing applications, based on IP multicast, typically employ an application-level protocol to provide *approximate* information as to the set of receivers and reception quality statistics. This protocol is the Real-time Transport Protocol (RTP) [8].

The portion of the Internet which supports IP multicast is known as the Mbone. Although some parts of the Mbone operate over dedicated links, the distinguishing feature is the presence of multicast routing support: multicast traffic typically shares links with other traffic. A number of attempts have been made to characterize the loss patterns seen on the Mbone [7, 9–11]. Although these results vary somewhat, the broad conclusion is clear: in a large conference it is inevitable that some receivers will experience packet loss. This is most clearly illustrated by the work of Handley [7], which tracks RTP reception report statistics for a large multicast session over several days. A typical portion of this trace is illustrated in Fig. 1. It can be seen that most receivers experience loss in the range of 2–5 percent, with some smaller number seeing significantly higher loss rates. The overwhelming cause of loss is due to congestion at routers. It is therefore not surprising that there is a correlation between the bandwidth used and the amount of loss experienced [7, 12], and the underlying loss rate varies during the day.

A multicast channel will typically have relatively high latency, and the variation in end-to-end delay may be large. This is clearly illustrated in Fig. 2, which shows the interarrival jitter for a series of packets sent from the University of Oregon to University College London on August 10, 1998. This delay variation is a reason for concern when developing loss-tolerant real-time applications, since packets delayed too long will have to be discarded in order to meet the application's timing requirements, leading to the appearance of loss. This problem is more acute for interactive applications: if interactivity is unimportant, a large playout delay may be inserted to allow for these delayed packets. This problem and algorithms for playout buffer adaptation are studied further in [13–15].

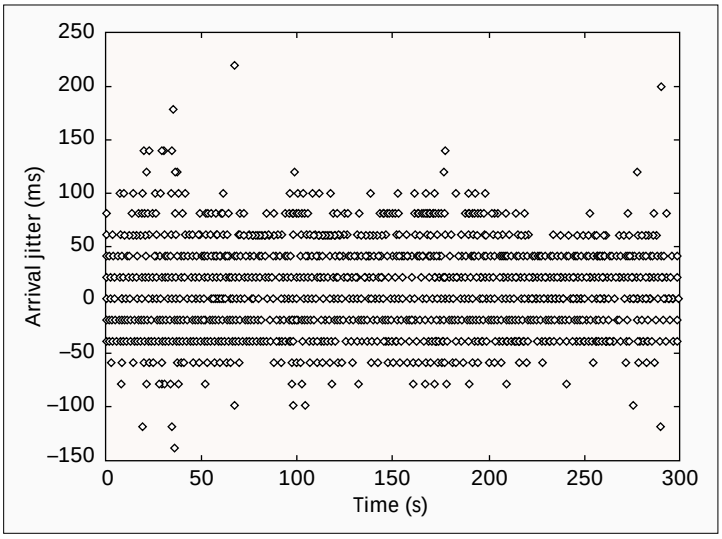


Figure 2. *Observed variation in end-to-end delay as seen by an Mbone audio tool (20 ms timing quantization).*

Unlike other communications media, IP multicast allows for the trade-off between quality and interactivity to be made independently for each receiver in a session, since this is a local choice only and is not communicated to the source of the data. A session may exist with most participants acting as passive observers (high latency, low loss), but with some active participants (low latency, higher loss).

It should be noted that the characteristics of an IP multicast channel are significantly different from those of an asynchronous transfer mode (ATM) or integrated services digital network (ISDN) channel. The techniques discussed herein do not necessarily generalize to conferencing applications built on such network technologies.

The majority of these techniques are applicable to unicast IP, although the scaling and heterogeneity issues are clearly simpler in this case.

Sender-Based Repair

We discuss a number of techniques which require the participation of the sender of an audio stream to achieve recovery from packet loss. These techniques may be split into two major classes: active retransmission and passive channel coding. It is further possible to subdivide the set of channel coding techniques, with traditional forward error correction (FEC) and interleaving-based schemes being used. Forward error correction data may be either media-independent, typically based on exclusive-or operations, or media-specific based on the properties of an audio signal. This taxonomy is summarized in Fig. 3.

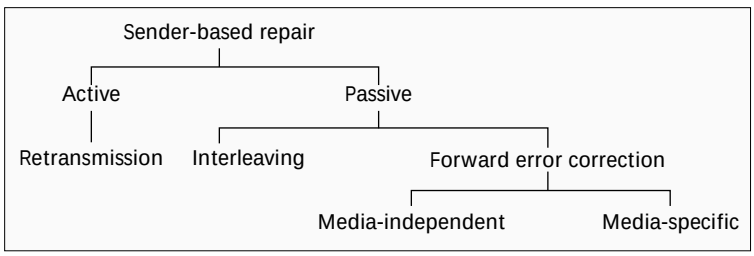
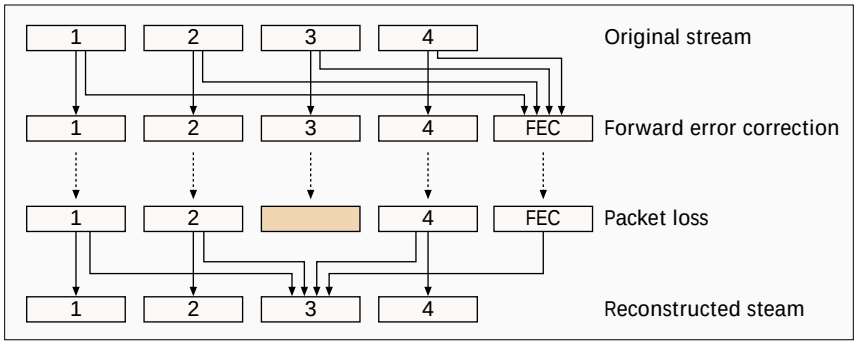


Figure 3. *A taxonomy of sender-based repair techniques.*

In order to simplify the following discussion we distinguish a *unit* of data from a *packet*. A unit is an interval of audio data, as stored internally in an audio tool. A packet comprises one or more units, encapsulated for transmission over the network.

Forward Error Correction

A number of forward error correction techniques have been developed to repair losses of data during transmission. These schemes rely on the addition of repair data to a stream, from which the contents of



■ Figure 4. *Repair using parity FEC.*

lost packets may be recovered. There are two classes of repair data which may be added to a stream: those which are independent of the contents of that stream, and those which use knowledge of the stream to improve the repair process.

Media-Independent FEC — There has been much interest in the provision of media-independent FEC using block, or algebraic, codes to produce additional packets for transmission to aid the correction of losses. Each code takes a codeword of k data packets and generates $n - k$ additional check packets for the transmission of n packets over the network.

A large number of block coding schemes exist, and we discuss only two cases, parity coding and Reed-Solomon coding, since these are currently proposed as an RTP payload [16]. These block coding schemes were originally designed for the detection and correction of errors within a stream of transmitted bits, so the check bits were generated from a stream of data bits. In packet streams we are concerned with the loss of entire packets, so we apply block coding schemes across the corresponding bits in blocks of packets. Hence, the ith bit in a check packet is generated from the ith bit of each of the associated data packets.

In parity coding, the exclusive-or (XOR) operation is applied across groups of packets to generate corresponding parity packets. An example of this has been implemented by Rosenberg [17]. In this scheme, one parity packet is transmitted after every $n - 1$ data packets. Provided there is just one loss in every n packets, that loss is recoverable. This is illustrated in Fig. 4. Many different parity codes may be derived by XORing different combinations of packets; a number of these were proposed by Budge *et al.* and summarized by Rosenberg and Schulzrinne [16].

Reed-Solomon codes [18, 19] are renowned for their excellent error correcting properties, and in particular their resilience against burst losses. Encoding is based on the properties of polynomials over particular number bases. Essentially, RS encoders take a set of codewords and use these as coefficients of a polynomial, $f(x)$. The transmitted codeword is determined by evaluating the polynomial for all nonzero values of x over the number base. While this may sound complicated, the encoding procedure is relatively straightforward, and optimized decoding procedures such as the Berlekamp-Massey algorithm [20, 21] are available. In the absence of packet losses decoding carries the same computational cost as encoding, but when losses occur it is significantly more expensive.

There are several advantages to FEC schemes. The first is that they are media-independent: the operation of the FEC does not depend on the contents of the packets, and the repair is an exact replacement for a lost packet. In addition, the computation required to derive the error correction packets is relatively small and simple to implement. The disadvantages of these schemes are the additional delay imposed, increased bandwidth, and difficult decoder implementation.

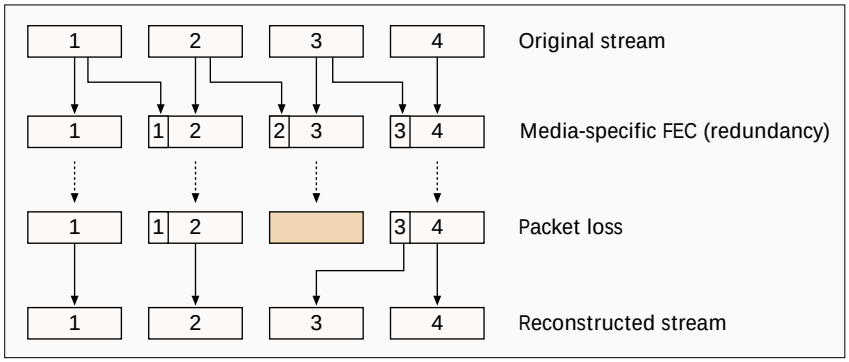
Media-Specific FEC — A simple means to protect against packet loss is to transmit each unit of audio in multiple packets. If a packet is lost, another packet containing the same unit will be able to cover the loss. The principle is illustrated in Fig. 5. This approach has been advocated by Hardman *et al.* [22] and Bolot *et al.* [9] for use on the Mbone, and extensively simulated by Podolsky *et al.* [23].

The first transmitted copy of the audio data is referred to as the *primary encoding*, and subsequent transmissions as *secondary encodings*. It is the sender's decision whether the secondary audio encodings should be the same coding scheme as the primary, although usually the secondary is encoded using a lower-bandwidth, lower-quality encoding than the primary.

The choice of encodings is a difficult problem and depends on both the bandwidth requirements and the computational complexity of the encodings. Erdöl *et al.* [24] consider using short-term energy and zero crossing measurements as their secondary scheme. When loss occurs the receiver then interpolates an audio signal about the crossings using the short-term energy measurements. The advantage of this scheme is that it uses computationally cheap measures and can be coded compactly. However, it can only cover short periods of loss due to the crude nature of the measures. Hardman *et al.* [22] and Bolot *et al.* [9] advocate the use of low-bit-rate analysis-by-synthesis codecs, such as LPC (2.4–5.6 kb/s) and full rate GSM encoding (13.2 kb/s), which, although computationally more demanding, can tolerably cover the loss periods experienced on the Internet.

If the primary encoding consumes considerable processing power, but has sufficient quality and low bandwidth, then the secondary encodings may be the same as the primary. An example of this is the International Telecommunication Union (ITU) G.723.1 [25] codec which consumes a considerable fraction of today's desktop processing power, but has a low bandwidth (5.3/6.3 kb/s).

The use of media-specific FEC incurs an overhead in terms of packet size. For example, the use of 8 kHz PCM μ-law (64 kb/s) as the primary compression scheme and GSM [26] (13 kb/s) as the secondary results in a 20 percent increase in the size of the data portion of each packet. Like media-independent FEC schemes, the overhead of media-specific FEC is variable. How-



■ Figure 5. *Repair using media-specific FEC.*

ever, unlike those schemes, the overhead of media-specific FEC may be reduced without affecting the number of losses which may be repaired; instead, the quality of the repair varies with the overhead. To reduce the overhead approximate repair is used, which is acceptable for audio applications.

It should be noted that it may often not be necessary to transmit media-specific FEC for every packet. Speech signals have transient stationary states that can cover 80 ms. Viswanathan *et al.* [27] describe LPC codecs where units of speech are only transmitted if the parameters of the codec are deemed to have changed sufficiently and achieve a 30 percent saving bandwidth for the same quality. A similar decision could be made about whether to transmit the FEC data, although this is likely to be codec-specific.

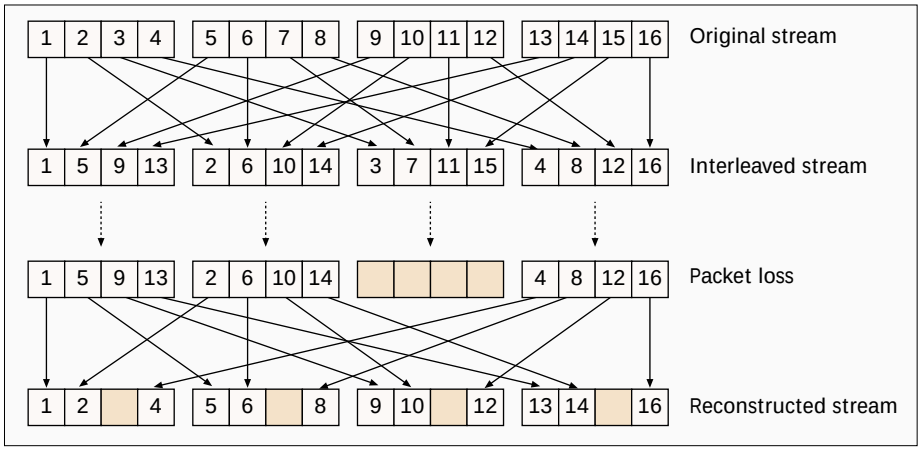


Figure 6. *Interleaving units across multiple packets.*

Unlike many of the other sender-based techniques discussed, the use of media-specific FEC has the advantage of low latency, with only a single-packet delay being added. This makes it suitable for interactive applications, where large end-to-end delays cannot be tolerated. If large end-to-end delay can be tolerated, it is possible to delay the redundant copy of a packet, achieving improved performance in the presence of burst losses [28].

At the time of writing, media-specific FEC is supported by a number of Mbone audio conferencing tools. The standard RTP payload format for media-specific FEC is described in [29].

Congestion Control — The addition of FEC repair data to a media stream is an effective means by which that stream may be protected against packet loss. However, application designers should be aware that the addition of large amounts of repair data when loss is detected will increase network congestion and hence packet loss, leading to a worsening of the problem which the use of FEC was intended to solve.

This is particularly important when sending to large multicast groups, since network heterogeneity causes different sets of receivers to observe widely varying loss rates: low-capacity regions of the network suffer congestion, while high-capacity regions are underutilized.

At the time of writing, there is no standard solution to this problem. There have been a number of contributions which show the likely form the solution will take [30–32]. These typically use some form of layered encoding of data sent at different rates over multiple multicast groups, with receivers joining and leaving groups in response to long-term congestion and with FEC employed to overcome short-term transient congestion.

Such a scheme pushes the burden of adaptation from the sender of a stream to the receivers, which choose the number of layers (groups) they join based on the packet loss rate they observe. Since the different layers contain data sent at different rates, receivers will receive different quality of service depending on the number of layers they are able to join. The precise details of these schemes are beyond the scope of this article; the reader is referred to the above references for further details.

Layered encoding schemes are expected to provide a congestion control solution suitable for streaming audio applications. However, this work is not yet complete, and it is important to give some advice to authors of streaming audio tools as to the behavior which is acceptable, until such congestion control mechanisms can be deployed.

It has been suggested that one heuristic suitable for determining reasonable behavior for unicast streaming media tools is to adapt the transmission rate to the approximate throughput a TCP/IP stream would achieve over the same path [33]. Since TCP/IP flows are the dominant form of traffic in the Internet, this would be roughly fair to existing traffic. Clearly such a scheme would not work for a multicast flow (although a worst case or average throughput to the set of receivers could be derived and used as the basis for adaptation), and clearly it does not capture the dynamic behavior of the connection, merely the average behavior; but it does provide one definition of reasonable behavior in the absence of real congestion control. In the long term, effective congestion control must be developed.

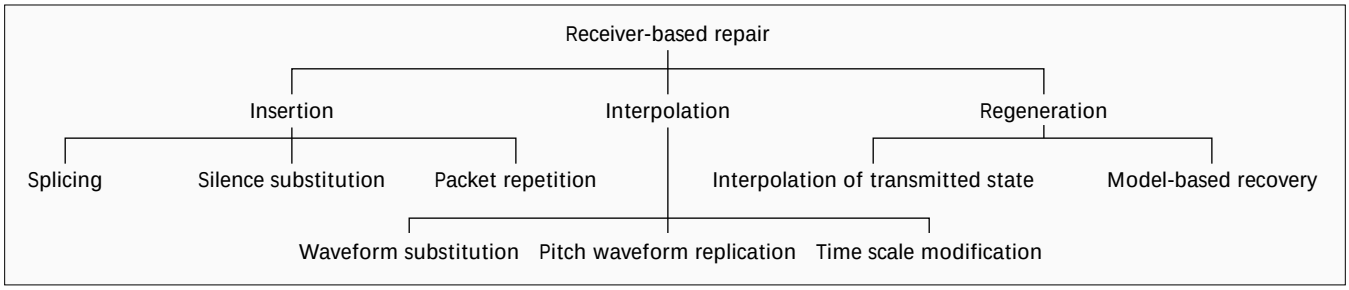
Note that the need for congestion control is *not* specific to FEC encoded audio streams. It should be considered for all streaming media.

Interleaving

When the unit size is smaller than the packet size and end-to-end delay is unimportant, interleaving is a useful technique for reducing the effects of loss [34]. Units are resequenced before transmission so that originally adjacent units are separated by a guaranteed distance in the transmitted stream and returned to their original order at the receiver. Interleaving disperses the effect of packet losses. If, for example, units are 5 ms in length and packets 20 ms (i.e., 4 units/packet), then the first packet would contain units 1, 5, 9, 13; the second units 2, 6, 10, 14; and so on, as illustrated in Fig. 6.

It can be seen that the loss of a single packet from an interleaved stream results in multiple small gaps in the reconstructed stream, as opposed to the single large gap which would occur in a noninterleaved stream. This spreading of the loss is important for two similar reasons: first, Mbone audio tools typically transmit packets which are similar in length to phonemes in human speech. Loss of a single packet will therefore have a large effect on the intelligibility of speech. If the loss is spread out so that small parts of several phonemes are lost, it becomes easier for listeners to mentally patch over this loss [35], resulting in improved perceived quality for a given loss rate. In a somewhat similar manner, error concealment techniques perform significantly better with small gaps, since the amount of change in the signal's characteristics is likely to be smaller.

The majority of speech and audio coding schemes can have their output interleaved and may be modified to improve the effectiveness of interleaving. The disadvantage of interleaving is that it increases latency. This limits the use of this technique for interactive applications, although it performs well for noninteractive use. The major advantage of interleaving is that it does not increase the bandwidth requirements of a stream.



■ Figure 7. *A taxonomy of error concealment techniques.*

Retransmission

Interactive audio applications have tight latency bounds, and end-to-end delays need to be less than 250 ms [36]. For this reason such applications do not typically employ retransmission-based recovery for lost packets. If larger end-to-end delays can be tolerated, the use of retransmission to recover from loss becomes a possibility.

A widely deployed reliable multicast scheme based on the retransmission of lost packets is scalable reliable multicast (SRM) [1]. When a member of an SRM session detects loss, it will wait a random amount of time, determined by its distance from the original source of the lost data, and then multicast a repair request. The retransmission timer is calculated such that, although a number of hosts may miss the same packet, the host closest to the point of failure will most likely timeout first and issue the retransmission request. Other hosts which also see the loss, but receive the retransmission request message, suppress their own request to avoid message implosion.[1] On receiving a retransmission request, any host with the requested data may reply: once again, a timeout is used based on the distance of that host from the sender of the retransmit request, to prevent reply implosion. The timers are calculated such that typically only one request and one retransmission will occur for each lost packet.

While SRM and related protocols are well suited for reliable multicast of data objects, they are not generally suitable for streaming media such as audio. This is because they do not bound the transmission delay and, in the presence of packet loss, may take an arbitrary amount of time. A large number of reliable multicast protocols have been defined (see [4] for a survey) which are similarly unsuitable for streaming media and hence are not studied here. For similar reasons, TCP is not appropriate for unicast streaming audio.

That is not to say that retransmission-based schemes cannot be used for streaming media, in some circumstances. In particular, protocols which use retransmission but bound the number of retransmission requests allowed for a given unit of data may be appropriate. Such retransmission-based schemes work best when loss rates are relatively small. As loss rates increase, the overhead due to retransmission request packets increases. Eventually a cross-over point is reached, beyond which the use of FEC becomes more effective. It has been observed in large Mbone sessions that *most* packets are lost by at least one receiver [7]. Indeed, in their implementation of an SRM-like protocol for streaming audio [37], Xu *et al.* note that "In the worst case, for every multicast packet, at least one receiver does not receive the packet, which means that *every* packet needs to be transmitted to the whole group at least twice." In cases such as this, it is clear that the use of retransmission is probably only appropriate as a secondary technique to repair losses which are not repaired by FEC.

An alternative combination of FEC and retransmission has been studied by Nonnenmacher *et al.* [38]. This work takes the approach of using parity FEC packets to repair multiple losses with a single retransmission, achieving substantial bandwidth savings relative to pure retransmission.

Furthermore, the retransmission of a unit of audio does not need to be identical to the original transmission: the unit can be recoded to a lower bandwidth if the overhead of retransmission is thought to be problematic. There is a natural synchrony with redundant transmission, and a protocol may be derived in which both redundant and retransmitted units may be accommodated. This allows receivers that cannot participate in the retransmission process to benefit from retransmitted units if they are operating with a sufficiently large playout delay.

The use of retransmission allows for an interesting trade-off between the desired playback quality and the desired degree of latency inherent in the stream. Within a large session, the amount of latency which can be tolerated varies greatly for different participants: some users desire to participate closely in a session, and hence require very low latency, whereas others are content to observe and can tolerate much higher latency. Those participants who require low latency must receive the media stream without the benefit of retransmission-based repair (but may use FEC). Others gain the benefit of the repair, but at the expense of increased delay.

Error Concealment

We consider a number of techniques for error concealment which may be initiated by the receiver of an audio stream and do not require assistance from the sender. These techniques are of use when sender-based recovery schemes fail to correct all loss, or when the sender of a stream is unable to participate in the recovery.

Error concealment schemes rely on producing a replacement for a lost packet which is similar to the original. This is possible since audio signals, in particular speech, exhibit large amounts of short-term self-similarity. As such, these techniques work for relatively small loss rates ($\lesssim$ 15 percent) and for small packets (4–40 ms). When the loss length approaches the length of a phoneme (5–100 ms) these techniques break down, since whole phonemes may be missed by the listener.

It is clear that error concealment schemes are not a substitute for sender-based repair, but rather work in tandem with it. A sender-based scheme is used to repair most losses, leaving a small number of isolated gaps to be repaired. Once the effective loss rate has been reduced in this way, error concealment forms a cheap and effective means of patching over the remaining loss.

A taxonomy of various receiver-based recovery techniques is given in Fig. 7. It can be seen that these techniques split into three categories:

- *Insertion*-based schemes repair losses by inserting a fill-in packet. This fill-in is usually very simple: silence or noise

[1] *The SRM protocol is designed to scale to very large groups. If request suppression were not used, a lost packet near the source would trigger simultaneous retransmission requests from many group members, which could overwhelm the sender (consider the effects in a group with many hundreds, or thousands, of members).*

are common, as is repetition of the previous packet. Such techniques are easy to implement but, with the exception of repetition, have poor performance.

- *Interpolation*-based schemes use some form of pattern matching and interpolation to derive a replacement packet which is expected to be similar to the lost packet. These techniques are more difficult to implement and require more processing when compared with insertion-based schemes. Typically performance is better.
- *Regeneration*-based schemes derive the decoder state from packets surrounding the loss and generate a replacement for the lost packet from that. This process is expensive to implement but can give good results.

The following sections discuss each of these categories in turn. This is followed by a summary of the range of applicability of these techniques.

Insertion-Based Repair

Insertion-based repair schemes derive a replacement for a lost packet by inserting a simple fill-in. The simplest case is splicing, where a zero-length fill-in is used; an alternative is silence substitution, where a fill-in with the duration of the lost packet is substituted to maintain the timing of the stream. Better results are obtained by using noise or a repeat of the previous packet as the replacement.

The distinguishing feature of insertion-based repair techniques is that the characteristics of the signal are not used to aid reconstruction. This makes these methods simple to implement, but results in generally poor performance.

Splicing — Lost units can be concealed by splicing together the audio on either side of the loss; no gap is left due to a missing packet, but the timing of the stream is disrupted. This technique has been evaluated by Gruber and Strawczynski [39] and shown to perform poorly. Low loss rates and short clipping lengths (4–16 ms) faired best, but the results were intolerable for losses above 3 percent.

The use of splicing can also interfere with the adaptive playout buffer required in a packet audio system, because it makes a step reduction in the amount of data available to buffer. The adaptive playout buffer is used to allow for the reordering of misordered packets and removal of network timing jitter, and poor performance of this buffer can adversely affect the quality of the entire system.

It is clear, therefore, that splicing together audio on either side of a lost unit is not an acceptable repair technique.

Silence Substitution — Silence substitution fills the gap left by a lost packet with silence in order to maintain the timing relationship between the surrounding packets. It is only effective with short packet lengths (< 4 ms) and low loss rates (< 2 percent) [40], making it suitable for interleaved audio over low-loss paths.

The performance of silence substitution degrades rapidly as packet sizes increase, and quality is unacceptably bad for the 40 ms packet size in common use in network audio conferencing tools [22]. Despite this, the use of silence substitution is widespread, primarily because it is simple to implement.

Noise Substitution — Since silence substitution has been shown to perform poorly, an obvious next choice is noise substitution, where, instead of filling in the gap left by a lost packet with silence, background noise is inserted instead.

A number of studies of the human perception of interrupted speech have been conducted, for example, that by Warren [41]. These have shown that *phonemic restoration*, the ability of the human brain to subconsciously repair the missing segment of speech with the correct sound, occurs for speech repair using noise substitution but not for silence substitution.

In addition, when compared to silence, the use of white noise has been shown to give both subjectively better quality [35] and improved intelligibility [41]. It is therefore recommended as a replacement for silence substitution.

As an extension for this, a proposed future revision of the RTP profile for audio-video conferences [42] allows for the transmission of *comfort noise* indicator packets. This allows the communication of the loudness level of the background noise to be played, allowing for better fill-in information to be generated.

Repetition — Repetition replaces lost units with copies of the unit that arrived immediately before the loss. It has low computational complexity and performs reasonably well. The subjective quality of repetition can be improved by gradually fading repeated units. The GSM system, for example, advocates the repetition of the first 20 ms with the same amplitude followed by fading the repeated signal to zero amplitude over the next 320 ms [43].

The use of repetition with fading is a good compromise between the other poorly performing insertion-based concealment techniques and the more complex interpolation-based and regenerative concealment methods.

Interpolation-Based Repair

A number of error concealment techniques exist which attempt to interpolate from packets surrounding a loss to produce a replacement for that lost packet. The advantage of interpolation-based schemes over insertion-based techniques is that they account for the changing characteristics of a signal.

Waveform Substitution — Waveform substitution uses audio before, and optionally after, the loss to find a suitable signal to cover the loss. Goodman *et al.* [44] studied the use of waveform substitution in packet voice systems. They examined both one- and two-sided techniques that use templates to locate suitable pitch patterns either side of the loss. In the one-sided scheme the pattern is repeated across the gap, but with the two-sided schemes interpolation occurs. The two-sided schemes generally performed better than one-sided schemes, and both work better than silence substitution and packet repetition.

Pitch Waveform Replication — Wasem *et al.*, [45] present a refinement on waveform substitution by using a pitch detection algorithm either side of the loss. Losses during unvoiced speech segments are repaired using packet repetition and voiced losses repeat a waveform of appropriate pitch length. The technique, known as pitch waveform replication, was found to work marginally better than waveform substitution.

Time Scale Modification — Time scale modification allows the audio on either side of the loss to be stretched across the loss. Sanneck *et al.* [46] present a scheme that finds overlapping vectors of pitch cycles on either side of the loss, offsets them to cover the loss, and averages them where they overlap. Although computationally demanding, the technique appears to work better than both waveform substitution and pitch waveform replication.

Regeneration-Based Repair

Regenerative repair techniques use knowledge of the audio compression algorithm to derive codec parameters, such that audio in a lost packet can be synthesized. These tech-

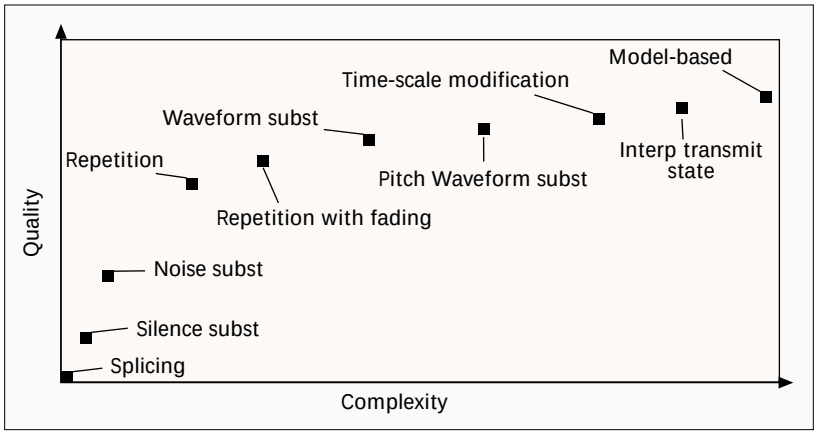


Figure 8. *Rough quality/complexity trade-off for error concealment.*

niques are necessarily codec-dependent but perform well because of the large amount of state information used in the repair. Typically, they are also somewhat computationally intensive.

Interpolation of Transmitted State — For codecs based on transform coding or linear prediction, it is possible that the decoder can interpolate between states. For example, the ITU G.723.1 speech coder [25] interpolates the state of the linear predictor coefficients either side of short losses and uses either a periodic excitation the same as the previous frame, or gain matched random number generator, depending on whether the signal was voiced or unvoiced. For longer losses, the reproduced signal is gradually faded. The advantages of codecs that can interpolate state rather than recoding the audio on either side of the loss is that there is are no boundary effects due to changing codecs, and the computational load remains approximately constant. However, it should be noted that codecs where interpolation may be applied typically have high processing demands.

Model-Based Recovery — In model-based recovery the speech on one, or both, sides of the loss is fitted to a model that is used to generate speech to cover the period loss. In recent work by Chen and Chen [47], interleaved μ-law encoded speech is repaired by combining the results of autoregressive analysis on the last received set of samples with an estimate of the excitation made for the loss period. The technique works well for two reasons: the size of the interleaved blocks (8/16 ms) is short enough to ensure that the speech characteristics of the last received block have a high probability of being relevant. The majority of low-bit-rate speech codecs use an autoregressive model in conjunction with an excitation signal.

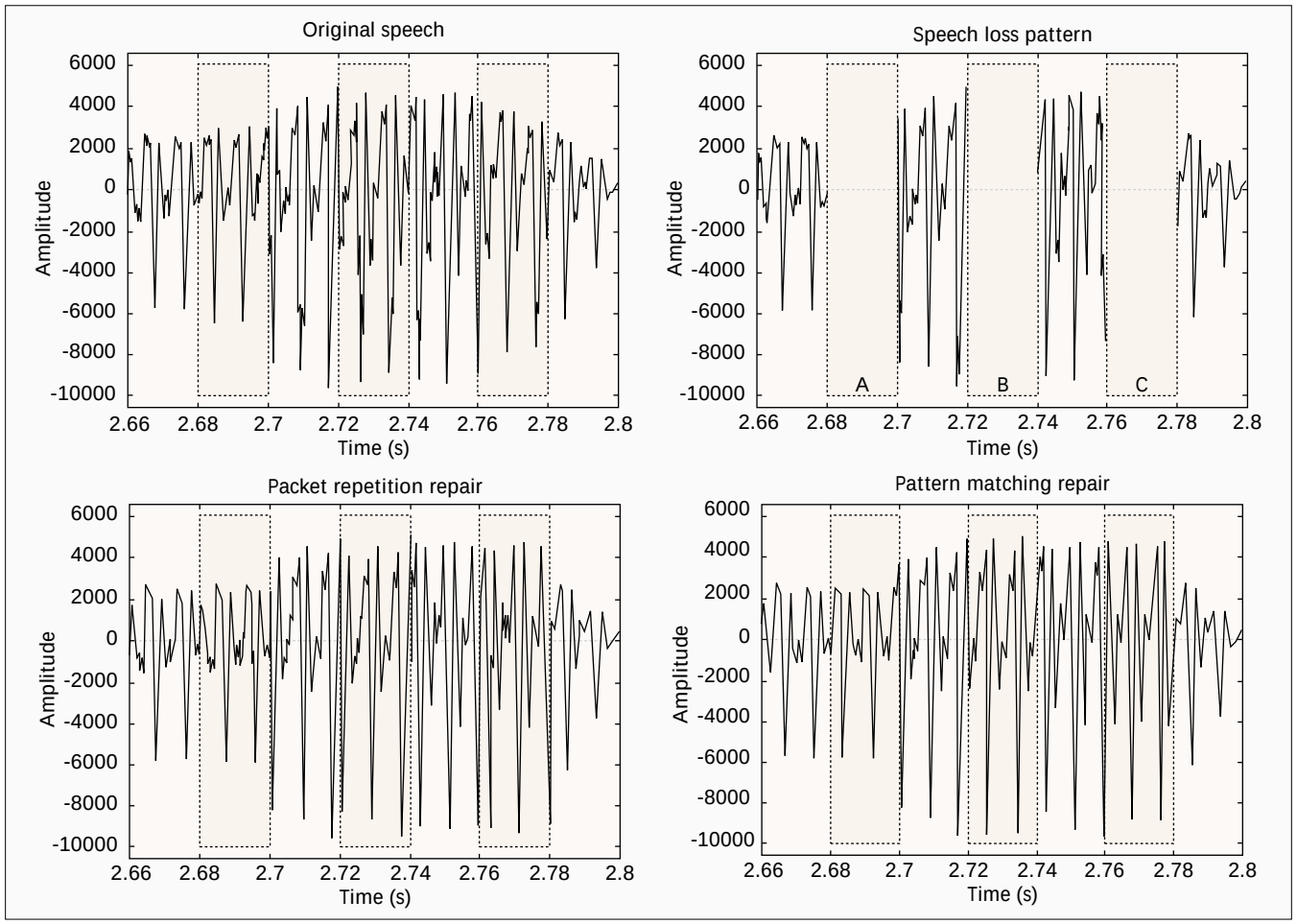


Figure 9. *(a) Sample error concealment techniques: original audio signal; (b) sample error concealment techniques: the loss pattern; (c) sample error concealment techniques; packet repetition; (d) sample error concealment techniques: one sided waveform substitution.*

Summary

It is difficult to obtain an accurate characterization of the performance and complexity of error concealment techniques since the measurements which may be performed are, due to the nature of the repair, subjective. However, based on our experience, we believe that Fig. 8 provides a reasonable illustration of the quality/complexity trade-off for the different repair techniques discussed.

The computation required to perform the more advanced repair techniques increases greatly relative to the simpler repair options. However, the improvement in quality achieved by these schemes is incremental at best. For this reason, the use of packet repetition with fading is recommended as offering a good compromise between achieved quality and excessive complexity. For comparison, an example using packet repetition and waveform substitution can be seen in Fig. 9.

Several of these techniques can be applied using data from one or both sides of the loss. Many audio and speech coders assume continuity of the decoder state. When a loss occurs, it may not be possible to decode audio data on both sides of the loss for use in the repair since the decoded audio after the loss may start from an inappropriate state. In addition, two-sided operations incur greater processing overhead and usually represent a marginal improvement. In the majority of cases one-sided repair is sufficient.

Recommendations

In this final section, we suggest which of these techniques should be considered for IP multicast applications in some common scenarios. We discuss the trade-off between achieving good performance with acceptable cost/complexity.

Noninteractive Applications

For one-to-many transmissions in the style of radio broadcasts, latency is of considerably less importance than quality. In addition, bandwidth efficiency is a concern since the receiver set is likely to be diverse and the group may include members behind low-speed links. The use of interleaving is compatible with both of these requirements and is strongly recommended.

Although interleaving drastically reduces the audible effects of lost packets, some form of error concealment will still be needed to compensate. In this case the use of a simple repair scheme, such as repetition with fading, is acceptable and will give good quality.

Retransmission-based repair is not appropriate for a multicast session, since the receiver set is likely to be heterogeneous. This leads to many retransmission requests for different packets and a large bandwidth overhead due to control traffic. For unicast sessions retransmission is more acceptable, particularly in low-loss scenarios.

A media-independent FEC scheme will perform better than a retransmission-based repair scheme, since a single FEC packet can correct many different losses and there is no control traffic overhead. The overhead due to the FEC data itself still persists, although this may be acceptable. In particular, FEC-protected streams allow for exact repair, while repair of interleaved streams is only approximate.

Interactive Applications

For interactive applications, such as IP telephony, the principal concern is minimizing end-to-end delay. It is acceptable to sacrifice some quality to meet delay requirements, provided that the result is intelligible.

The delay imposed by the use of interleaving, retransmission, and media-independent FEC is not acceptable for these applications. While media-independent FEC schemes do exist that satisfy the delay requirements, these typically have high bandwidth overhead and are likely to be inappropriate for this reason.

Our recommendation for interactive conferencing applications is that media-specific FEC is employed, since this has low latency and tunable bandwidth overhead. Repair is approximate due to the use of low-rate secondary encodings, but this is acceptable for this class of application when used in conjunction with receiver-based error concealment.

Error Concealment

Receivers must be prepared to accept some loss in an audio stream. The overhead involved in ensuring that all packets are received correctly, in both time and bandwidth, is such that some loss is unavoidable. Once this is accepted, the need for error concealment becomes apparent. Many current conferencing applications use silence substitution to fill the gaps left by packet loss, but it has been shown that this does not provide acceptable quality. A significant improvement is achieved by the use of packet repetition, which also has the advantages of being simple to implement and having low computational overhead. The other error concealment schemes discussed provide incremental improvements, with significantly greater complexity. Accordingly, we recommend the use of packet repetition since it is a simple and effective means of recovering from the low-level random packet loss inherent in the Mbone.

Acknowledgments

This work has benefited from the insightful comments of the reviewers and discussion with members of the networked multimedia research group at UCL. In particular, we wish to thank Jon Crowcroft and Roy Bennett for their helpful comments. We are grateful to Mark Handley for permission to use Fig. 1.

The authors are supported by the U.K. EPSRC project RAT (GR/K72780), the EU Telematics for research project MERCI (#1007), and British Telecommunications plc (ML72254).

References

[1] S. Floyd *et al.*, "A reliable multicast framework for light-weight sessions and application level framing," *IEEE/ACM Trans. Networking*, Dec. 1997.
[2] V. Jacobson, "Multimedia conferencing on the Internet," *SIGCOMM Symp. Commun. Architectures and Protocols*, tutorial slides, Aug. 1994.
[3] K. Obraczka, "Multicast transport mechanism: A survey and taxonomy," to appear, *IEEE Commun. Mag.*, 1998.
[4] B. N. Levine and J. J. Garcia-Luna-Aceves, "A comparison of reliable multicast protocols," *ACM Multimedia Sys.*, Aug. 1998.
[5] G. Carle and E. W. Biersack, "Survey of error recovery techniques for IP-based audio-visual multicast applications," *IEEE Network*, vol. 11, no. 6, Nov./Dec. 1997, pp. 24–36.
[6] S. Deering, "Multicast Routing in a Datagram Internetwork," Ph.D. thesis, Stanford University, Palo Alto, CA, Dec. 1991.
[7] M. Handley, "An examination of Mbone performance," USC/ISI res. rep. ISI/RR-97-450, Apr. 1997.
[8] H. Schulzrinne *et al.*, "RTP: A transport protocol for real-time applications," IETF Audio/Video Transport WG, RFC1889, Jan. 1996.
[9] J.-C. Bolot and A. Vega-Garcia, "The case for FEC based error control for packet audio in the Internet," to appear, *ACM Multimedia Sys.*
[10] J.-C. Bolot and A. Vega-Garcia, "Control mechanisms for packet audio in the Internet," *Proc. IEEE INFOCOM '96*, 1996.
[11] M. Yajnik, J. Kurose, and D. Towsley, "Packet loss correlation in the Mbone multicast network," *Proc. IEEE Global Internet Conf.*, Nov. 1996.
[12] O. Hermanns and M. Schuba, "Performance investigations of the IP multicast architecture," *Comp. Networks and ISDN Syst.*, vol. 28, 1996, pp. 429–39.
[13] J.-C. Bolot, "End-to-end packet delay and loss behavior in the internet," *Proc. ACM SIGCOMM '93*, San Francisco, Sept. 1993, pp. 289–98.
[14] S. B. Moon, J. Kurose, and D. Towsley, "Packet audio playout delay adjustment algorithms: performance bounds and algorithms," Res. rep., Dept. of Comp. Sci., Univ. of MA at Amherst, Aug. 1995.
[15] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide-area networks," *Proc. IEEE INFOCOM*, Toronto, Canada, June 1994.

[16] J. Rosenberg and H. Schulzrinne, "An RTP payload format for generic forward error correction," IETF Audio/Video Transport WG, work in progress (Internet-draft), July 1998.
[17] J. Rosenberg, "Reliability enhancements to NeVoT," Dec. 1996.
[18] H. F. Mattson and G. Solomon, "A new treatment of Bose-Chaudhuri codes," *J. SIAM*, vol. 9, no. 4, Dec. 1961, pp. 654–69.
[19] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. SIAM*, vol. 8, no. 2, June 1960, pp. 300–4.
[20] E. R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, 1968.
[21] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Info. Theory*, vol. IT-15, 1969, pp. 122–27.
[22] V. Hardman *et al.*, "Reliable audio for use over the Internet," *Proc. INET '95*, 1995.
[23] M. Podolsky, C. Romer and S. McCanne, "Simulation of FEC-based error control for packet audio on the Internet," *Proc. IEEE INFOCOM '98*, San Francisco, CA, Apr. 1998.
[24] N. Erdöl, C. Castelluccia, and A. Zilouchian, "Recovery of missing speech packets using the short-time energy and zero-crossing measurements," *Trans. Speech and Audio Processing*, vol. 1, no. 3, July 1993, pp. 295–303.
[25] ITU Rec. G.723.1, "Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s," Mar. 1996.
[26] M. Mouly and M.-B. Pautet, *The GSM System for Mobile Communications*, Europe Media Duplication, Lassay-les-Chateaux, France, 1993.
[27] V. R. Viswanathan *et al.*, "Variable frame rate transmission: A review of methodology and application to narrow-band LPC speech coding," *IEEE Trans. Commun.*, vol. COM-30, no. 4, Apr. 1982, pp. 674–87.
[28] I. Kouvelas *et al.*, "Redundancy control in real-time Internet audio conferencing," *Proc. AVSPN '97*, Aberdeen, Scotland, Sept. 1997.
[29] C. S. Perkins *et al.*, "RTP Payload for redundant audio data," IETF Audio/Video Transport WG, RFC2198, 1997.
[30] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," *Proc. ACM SIGCOMM '96*, Stanford, CA., Aug. 1996.
[31] L. Rizzo and V. Vicisano, "A reliable multicast data distribution protocol based on software fec techniques," *Proc. 4th IEEE Wksp. Arch. and Implementation of High Perf. Commun. Sys.* (HPCS '97), 1997.
[32] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer," *Proc. IEEE INFOCOM '98*, 1998.
[33] C. S. Perkins and O. Hodson, "Options for repair of streaming media," IETF Audio/Video Transport WG, RFC2354, June 1998.
[34] J. L. Ramsey, "Realization of optimum interleavers," *IEEE Trans. Info. Theory*, vol. IT-16, May 1970, pp. 338–45.
[35] G. A. Miller and J. C. R. Licklider, "The intelligibility of interrupted speech," *J. Acoust. Soc. Amer.*, vol. 22, no. 2, 1950, pp. 167–73.
[36] P. T. Brady, "Effects of transmission delay on conversational behavior on echo-free telephone circuits," *Bell Sys. Tech. J.*, vol. 50, Jan. 1971, pp. 115–34.
[37] R. X. Xu *et al.*, "Resilient multicast support for continuous media applications," *Proc. 7th Int'l. Wksp. Network and Op. Sys. Support for Digital Audio and Video* (NOSSDAV '97), Washington Univ., St. Louis, MO, May 1997.
[38] J. Nonnenmacher, E. Biersack, and D. Towsley, "Parity-based loss recovery for reliable multicast transmission," *Proc. ACM SIGCOMM '97*, Cannes, France, Sept. 1997.
[39] J. G. Gruber and L. Strawczynski, "Subjective effects of variable delay and clipping in dynamically managed voice systems," *IEEE Trans. Commun.*, vol. COM-33, no. 8, Aug. 1985, pp. 801–8.
[40] N. S. Jayant and S. W. Christenssen, "Effects of packet losses in waveform coded speech and improvements due to an odd-even sample-interpolation procedure," *IEEE Trans. Commun.*, vol. COM-29, no. 2, Feb. 1981, pp. 101–9.
[41] R. M. Warren, *Auditory Perception*, Pergamon Press, 1982.
[42] H. Schulzrinne, "RTP profile for audio and video conferences with minimal control," IETF Audio/Video Transport WG, work in progress, Mar. 1997.
[43] ETSI Rec. GSM 6.11, "Substitution and muting of lost frames for full rate speech channels," 1992.
[44] D. J. Goodman *et al.*, "Waveform substitution techniques for recovering missing speech segments in packet voice communications," *IEEE Trans. Acoustics, Speech, and Sig. Processing*, vol. ASSP-34, no. 6, Dec. 1986, pp. 1440–48.
[45] O. J. Wasem *et al.*, "The effect of waveform substitution on the quality of PCM packet communications," *IEEE Trans. Acoustics, Speech, and Sig. Processing*, vol. 36, no. 3, Mar. 1988, pp. 342–48.
[46] H. Sanneck et al., "A new technique for audio packet loss concealment," *IEEE Global Internet 1996*, IEEE, Dec. 1996, pp. 48–52.
[47] Y. L. Chen and B. S. Chen, "Model-based multirate representation of speech signals and its application to recovery of missing speech packets," *IEEE Trans. Speech and Audio Processing*, vol. 15, no. 3, May 1997, pp. 220–31.

Biographies

COLIN PERKINS (C.Perkins@cs.ucl.ac.uk) received the B. Eng. degree in electronic engineering from the University of York in 1992. In 1995 he received a D. Phil. from the University of York, Department of Electronics, where his work involved software reliability modeling and analysis. Since then he has been a research fellow at University College London, Department of Computer Science. His work at UCL has included development of the Robust-Audio Tool (RAT), audio transcoder/mixer design and implementation, and local conference coordination issues.

ORION HODSON (O.Hodson@cs.ucl.ac.uk) received a B.Sc. in physics and theory from the University of Birmingham, England, in 1993, and an M.Sc. in computation neuroscience from the University of Stirling, Scotland, in 1995. He is currently a Ph.D. candidate in the Computer Science Department of University College London. His research interests include voice over IP networks, multimedia conferencing, and real-time applications.

VICKY HARDMAN is a lecturer in computer science at University College London. She has a Ph.D. in speech over packet networks from Loughborough University of Technology, England, where she subsequently worked as a research assistant. Her research interests include multicast conferencing, speech over packet networks, audio in virtual reality environments, and real-time multimedia applications.