Journal of Computer Science 1 (2): 164-168, 2005
ISSN 1549-3636

Introducing Song Form Intelligence into Streaming Audio

Kevin Curran
Internet Technologies Research Group
University of Ulster, Magee Campus, Northern Ireland BT48 7JL, UK

Abstract: When receiving streaming media over a low bandwidth wireless connection, users can experience not only packet losses but also extended service interruptions. These dropouts can last for as long as 15 sec. During this time no packets are received and if not addressed, these dropped packets cause unacceptable interruptions in the audio stream. A long dropout of this kind may be overcome by ensuring that the buffer at the client is large enough. However, when using fixed bit rate technologies such as Windows Media or Real Audio, this may only be done by buffering packets for an extended period (10 sec or more) before starting to play the track. During this period, many users are likely to lose interest or become frustrated. This research presents a novel semantic audio error concealment buffering technology, made possible by the song form structure. It enables the audio to start playing within two or three sec, while at the same time using a small proportion of the available bandwidth to fill the client device buffer with received packets but categorised into structures of the song. A pattern matching run-time algorithm works to identify portions of the audio stream and when a dropout does occur relevant section of the buffered audio are inserted so as to create a perfect match for the lost audio. Our algorithm can be shown to increase reliability on bursty wireless networks.

Key words: Streaming Audio, Multimedia, Song Form, Music

INTRODUCTION

It is anticipated that the future will witness the next revolution through telecommunications technology. In the past two decades the communications sector was one of the few constantly growing sectors in industry and a wide variety of new services were created. Digital and powerful communication networks are being discussed, planned or under construction. The new networks will allow services to become tailored for each individual and the interactiveness will open space for yet unknown applications. Services such as audio-on-demand will drastically increase the load on the networks. The spread of the newly created compression standards such as MPEG reflect the current demand for data compression. As these new services become available the demand of audio services through mobiles will increase. The technology for these services is available but suitable standards are yet to be defined. This is due to the nature of mobile radio channels, which are more limited in terms of bandwidth and bit error rates as for example the public telephone network. Therefore new, robust and highly efficient coding algorithms will be necessary. Audio due to its timely nature requires guarantees different in nature with regards to delivery of data from TCP traffic for ordinary HTTP requests. In addition, audio applications increase the set of requirements in terms of throughput, end-to-end delay, delay jitter and synchronization.

The focus of this research is to examine new methods for streaming music over bandwidth constrained networks. One overlooked method to date, which can work alongside existing audio compression schemes, is to take account of the semantics of the music. Songs in general exhibit standard structures that can be used as a forward error checking mechanism. For instance, many songs start with a verse, then proceed to a chorus and then repeat this structure until the end with little variation. Suppose we assume that a song starts streaming with the standard 10 sec buffering interval at the start. Lets also assume that this is sufficient to provide smooth delivery of the initial verse of the song. The chorus follows next with say, 15% losses. In the meantime, an audio pattern-matching algorithm is classifying the song into well-known forms such as chorus and verse. Next time that we approach the verse, the pattern matching algorithm will 'pick up the trail' after a few chords and correctly predict that the streaming song is now re-entering the verse part. The idea here is that whenever packets are lost in the stream, corresponding packets are inserted into the stream from the buffered section of the 'previous' verse thus providing a perfect match. Here we have demonstrated the perfect error concealment algorithm for repairing streaming audio over error prone networks using the semantics of the music.

Related Work: Ghias *et al.* [1] describes a natural way of querying a musical audio database is by humming the tune of a song. Their system allows for querying an audio database by humming. They also have a scheme for representing the melodic information in a song as relative pitch changes. Their work differs in that it simply queries a database rather than serving as real-

time audio pattern matching receiver based error concealment.
Other recent alternative approaches to pattern matching in audio rely on combinatory rather than on signal processing [2]. The audio is searched using techniques derived from the large body of knowledge acquired in the field of pattern matching of biological sequences. Although the degree of flexibility obtained is still inferior to that of the signal processing approach, much faster search algorithms have been obtained. These results are rather encouraging and we plan to investigate further.

Error Concealment: There are a number of techniques for recovery from packet loss on a channel. They may be classified as either sender based repair or receiver-based repair [3]. For streaming transmission style services such as audio broadcasting, latency is of considerably less importance than quality. In addition in the low bandwidth scenarios that we focus upon here, bandwidth is a concern. Thus we feel that a receiver based repair scheme is to be preferred.
For the purposes of space, we do not discuss further sender based forward error checking schemes [4, 5]. Receiver based schemes are initiated by the receiver of an audio stream and require no assistance from the sender. Error concealment attempts to replace a lost packet by a similar packet. This can be achieved as audio signals exhibit large amounts of short-term self-similarity. Taxonomy of various receiver-based recovery techniques is shown in Fig. 1 [6].

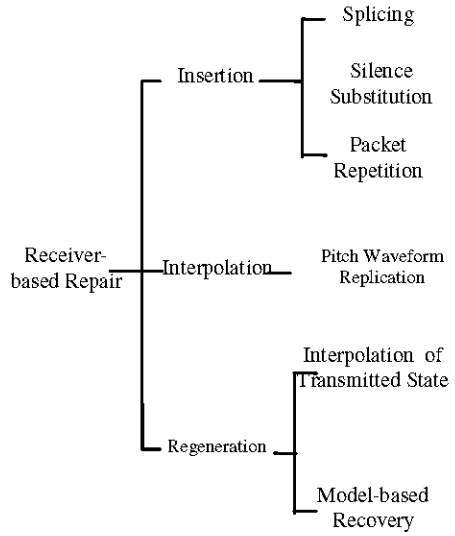

Fig. 1: Error Concealment Techniques

It can be seen that these techniques fall into three groups:

* Insertion-based schemes repair losses by inserting a fill-in packet. This fill-in is commonly silence, noise or repetition of the previous packet.
* Interpolation-based schemes adopt pattern matching and interpolation to arrive at a replacement packet similar to the lost packet. Performances of these schemes are encouraging but they can be difficult to implement and require increased processing power over simpler insertion based schemes.
* Regeneration-based schemes derive the decoder state from packets surrounding the loss and attempt to generate a replacement packet from that. As above, this technique is computationally intensive.

The above concealment schemes however create artifacts, which may be detectable by the user, depending on the amount of audio lost, the type of stream and the actual effectiveness of the concealment algorithm [7]. In addition, since the effectiveness of these concealment schemes depend on the amount and correct interpretation of received data, concealment becomes much harder with bursty loss. The error-concealment framework proposed here intends to overcome these problems by taking the semantics of the audio into account.

Song Form and Structure: The vast majority of songs in Western nations contain a structure known as a song form. A song form is basically a framework, which makes the song listenable [8]. Many people understand the idea of a verse and a chorus. The purpose of a verse is to tell the story or describe the feeling. The chorus is generally the focal point of the song, the central theme. A bridge is a kind of fresh perspective, a small part that may consist of only music, or both lyrics and music, usually placed after the second chorus and often varying in major/minor chords. These are the main parts that are used when describing song form.
Often we can describe songs in terms such as having an VCVC form, or VCBVC. Not all songs however follow a verse, chorus bridge pattern. The oldest song form is often referred to as folk, where there was no chorus, or any other part, for that matter. Verses are always labelled V. So, in describing a folk song form, or any song that has only verses, the song form is VVVV. A large percentage of songs these days however follow a type of song form, which includes a chorus. The chorus is labelled C, so a verse, chorus, verse, chorus type of song is VCVC. We also have a common song form, which includes a bridge, which is labelled B. So a typical form with a bridge might be VVCVCBC. This states that we have a verse, verse, chorus, verse, chorus, bridge and chorus song form. Other less common forms may include a pre-chorus that is a lead up (or build) to the chorus; intros (labeled I) at the very beginning of a song and extros (labeled E) are the lead-outs or endings to a song [8].
There are many variations of these forms however, with some songs starting with the chorus while others have more than one bridge. Any individual artist may have

all kinds of variations. Most songwriters don't start writing by coming up with a song form first. It usually reveals itself as the song is being written. It is, however, a quick and easy language to use when discussing the process with other writers.

Trends in Popular Song Form: Charts vary from country to country and from also between music genres. Over a period of three months, we studied the country music charts presented on an Irish radio country show in November 2002. The findings were that every one of the songs was written in one of the well-established song forms as shown in Fig. 2.

Song form	# of songs in the Top 40
VVV	None
VVCV	8
VCVCV	16
VCBVC	6
VCVV	7
VCBVVC	3

Fig. 2: # of Songs in Country Top 40 with Various Song Form Structures V = Verse, C = Chorus, Nov. 2001

We also found that in the previous three months of song form study, the majority of songs in the top 40 used the simple verse/chorus structure (though some included instrumental sections). The second most popular form was VVCV followed closely by the VCVV structure and the VCBVC structure. As you might expect, there were no VVV songs, although one or two a year are expected to make their way into the charts. (In the analysis above we ignored intros and extros). The study also analysed the length of introductions of each song.

Length of introduction	# of songs in the Top 40
< 10 seconds	7
11-16 seconds	24
17-20 seconds	9
> 20 seconds	None

Fig. 3: Length of Introductions in Country Top 40

The interesting statistic in Fig. 3 is that twenty-four of the forty songs fell into the second category and no songs had introductions longer than twenty sec. The average introduction length was 14 sec. We also examined the length of time it took the song to reach the chorus as shown in Fig. 4.

There were only six songs in the top 40 that took longer than a minute to get to the chorus. The average length of time was 40 sec. We also looked at the overall length of each song as shown in Fig. 5.

Time to get to the chorus	# of songs in the Top 40
< 30 seconds	5
30-40 seconds	9
41-50 seconds	11
51-60 seconds	9
>60 seconds	6

Fig. 4: Time to Get to the Chorus in Country Top 40

Length of song	# of songs in the Top 40
< 2:00	3
2:01-2:30	10
2:31-4:00	12
4:01-4:30	10
4:31-5:00	4
>5:01	1

Fig. 5: Length of Song in Country Top 40

The average length of a song was 3 minutes. This analysis of song form feeds into the design of the pattern-matching algorithm in order to aid determination of the various semantics of each streaming song. The averages within each category play a large part in the threshold limits set within the algorithm.

Pattern Matching: In the audio compression research community there has being a common consensus that there is no apparent repetitiveness and approximate repetitiveness in audio. Some agree however that repetitiveness occurs but shifted in phase, difficult to recognize in the time occurs but shifted in phase, so difficult to recognize in the time domain [9]. We however have defined earlier what we believe the pattern to be with regards music therefore the trust of this paper is that pattern matching audio (uncompressed and compressed) is possible.

We need an efficient approximate pattern-matching algorithm that is should be able to take into account various forms of errors. Various forms of errors anticipated in a typical pattern-matching scheme include transposition errors, dropout errors and duplication errors. The algorithm that we adopted for this purpose is described by Jackson [8]. This algorithm addresses the problem of string matching with k mismatches. The problem consists of finding all instances of a pattern string $P = p1\ p2\ p3...pm$ in a text string $T = t1\ t2\ t3...tn$ such that there are at most k mismatches (notes that are not the same) for each

instance of P in T. When $k = 0$ (no mismatches) we have the simple string matching problem, solvable in $O(n)$ time. When $k = m$, every substring of T of length m qualifies as a match, since every character of P can be mismatched.

Existing receiver based techniques work for relatively small loss rates <12% and for small packets (4-30 ms). When the loss length approaches the length of a phoneme (5-100 ms) these techniques tend to break down [6].

Semantic Audio Receiver Based Error Concealment: Our system works by 'post-processing' the audio stream into a group of V, B I and C sections (as discussed in section 0). Then a fuzzy pattern-matching algorithm seeks to match later sections of the stream containing errors with buffer-stored sections (e.g. the first chorus section) to allow errors to be concealed with related matching packets in the corresponding buffered section.

As our research is in the early stages, we simplify the procedure of initial song form section recognition by manually identifying the specific sections. We do this by initially sending a header to the client that contains information about sections start-times and lengths of the streaming song to follow. This takes the following form:

I	0.10	V	0.28	C	0.32	V	0.28	C	0.32

Fig. 6: Song Form Structure Header

The song header depicted in Fig. 6 describes a piece of music with the song form IVCVC. It states that there is an introduction section of 10 sec duration followed by a verse of 28 sec, then a chorus of 32 sec, then a verse of 28 sec and finally repeats the chorus of 32 sec.

Evaluation: We concentrate our testing on resource constrained mobile devices, which are set to gain the most benefit from the error concealment algorithm as illustrated in Fig. 7.

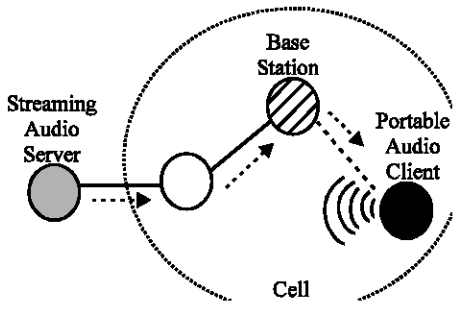

Fig. 7: Mobile Audio Client

The packet losses were simulated by a LOSSY protocol stack element. The LOSSY component simulated reordering and loss of messages (0% loss to 20% loss). The section of code dealing with losses is displayed in Fig. 8.

```
if (!losing_messages && (LossToDate++ % lossrate)
== 0) losing_messages = true;
if (losing_messages && lossCount++ < lossTotal) {
        log.errors ("Simulating loss of message number
"
        +    message.SeqNum ());
        return;
            } else {
                losing_messages = false;
                lossCount = 0;
            }
```

Fig. 8: Code to Simulate Lost Messages

We perform the streaming evaluation over our three-test bed network as depicted in Fig. 9. We used a Windows 98 Pentium Pro 500 MHz PC laptop client, with 128 MB Ram connected via a 2 MB wireless LAN to a Windows 2000 Pentium III 800 MHz Server with 128 MB RAM.

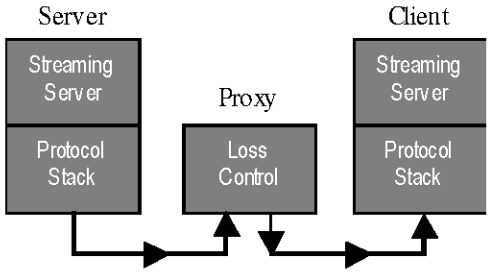

Fig. 9: Test Bed for Evaluation

The songs that we streamed had an obvious song form structure which gained from semantic audio error concealment and we do acknowledge that this will not be the case with a large majority of music. We also streamed an instrumental piece so the repeated chorus and verse were exact matches for previous sections. The song in question contained the structure IVCVCV The average drop loss ratio was set to 15%.

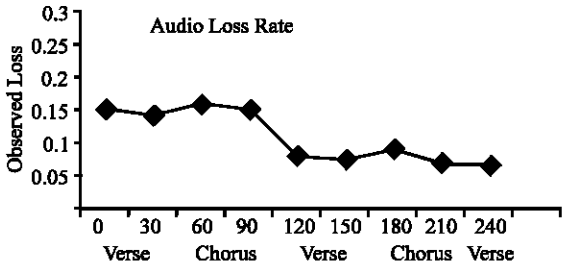

Fig. 10: Random Audio Drop Test

Figure 10 demonstrates that the loss rate significantly deteriorates after the first verse and chorus as expected. Once the first verse and chorus have been mapped and stored in the buffer, later errors are masked by insertion of corresponding correct-fit packets from the buffer. This experiment validates the use of semantic receiver based error concealment mechanisms. Further improvements can be expected by fine tuning the code and upgrading of the hardware.

Future Work: We wish to investigate techniques, which allow the tracking of pitch in audio streams so as to ascertain movements in song form structure. At present we adopt primitive methods for automatically tracking pitch in songs. One promising technique is Autocorrelation, which isolates and tracks the peak energy levels of the signal that is a measure of the pitch.

REFERENCES

1. Ghias, A., J. Logan and D. Chamberlin, 1995. Query by Humming. ACM Multimedia 95-Electronic Proceedings, San Francisco, California, Nov. 5-9, pp: 29-38.
2. Yates, R. and G. Navarro, 2002. New and faster filters for multiple approximate string matching. Random Structures and Algorithms (RSA), 20: 23-49.
3. Hardman, V. and J. Crowcroft, 1995. Reliable audio for use over the internet. Proc. INET '95, June, pp: 171-178.
4. Bolot, J., S. Parisis and D. Towsley, 1999. Adaptive FEC-based error control for internet telephony. In: Proceedings of IEEE INFOCOM '99, Mar. 3, pp: 1453-1460.
5. Sanneck, I. and N. Le, 2000. Speech property-based FEC for internet telephony applications. Proc. SPIE-The International Society for Optical Engineering, Jan., 3969: 38-51.
6. Perkins, C., O. Hodson and V. Hardman, 1998. A survey of packet-loss recovery techniques for streaming audio. IEEE Network Magazine, June, 12: 64-79.
7. Papadopoulos, C. and G. Parulkar, 1996. Retransmission-based error control for continuous media applications. Proc. NOSSDAV 1996, pp: 45-52.
8. Jackson, I., 2002. Song forms and terms-a quick study. http://www.irenejackson.com/form.html
9. Atallah, M. and Y. Genin, 1996. Pattern matching image compression: Algorithmic and empirical results. Proc. International Conference on Image Processing, 3: 349-352, Lausanne, Switzerland.
10. Yates, R. and C. Perleberg, 1992. Fast and practical approximate string matching. Combinatorial Pattern Matching, Third Annual Symposium, pp: 185-192.