Mark Handley
Vicky Hardman
Isidor Kouvelas
Colin Perkins
University College London

Jean-Chrysostome Bolot
Andres Vega-Garcia
Sacha Fosse-Parisis
INRIA Sophia Antipolis

Payload Format Issues for Redundant Encodings in RTP

draft-perkins-rtp-redundancy-01.txt

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress." To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Distribution of this document is unlimited.

Comments are solicited and should be addressed to the authors and/or the AVT working group's mailing list at rem-conf@es.net.

Abstract

This document describes a payload type for use with the real-time transport protocol (RTP), version 2, for encoding redundant data. The primary motivation for the scheme described herein is the development of audio conferencing tools for use with lossy packet networks such as the Internet Mbone, although this scheme is not limited to such applications.

1 Introduction

If multimedia conferencing is to become widely used by the Internet Mbone community, users must perceive the quality to be sufficiently good for most applications. We have identified a number of problems which impair the quality of conferences, the most significant of which is packet loss over the Mbone. Packet loss is a persistent problem, particularly given the increasing popularity, and therefore increasing load, of the Internet. The disruption of speech intelligibility even at low loss rates which is currently experienced may convince a whole generation of users that multimedia conferencing over the Internet is not viable. The addition of redundancy to the data stream is offered as a solution [1]. If a packet is lost then the missing information may be reconstructed at the receiver from the redundant data that arrives in the following packet(s).

This draft proposes an RTP payload format for the transmission of data encoded in a redundant fashion. Although the primary use of this packet format to date has been in audio applications, the packet format specified is quite general, and is not limited to these applications.

2 Packetisation problem

The main requirements for a redundant encoding scheme under RTP are as follows:

- Packets have to carry a primary encoding and one or more redundant encodings.
- As a multitude of encodings may be used for redundant information, each block of redundant encoding has to have an encoding type identifier.
- As the use of variable size encodings is desirable, each encoded block in the packet has to have a length indicator.
- The RTP header provides a timestamp field that corresponds to the time of creation of the encoded data. When redundant encodings are used this timestamp field can refer to the time of creation of the primary encoding data. Redundant blocks of data will correspond to different time intervals than the primary data. Each block of redundant encoding has to have its own timestamp. To reduce the number of bytes needed to carry the timestamp, it can be computed as the difference of the timestamp for the redundant encoding to timestamp for the primary.

There are two essential means by which redundant audio may be added to the standard RTP specification: a header extension may hold the redundancy, or one, or more, additional payload types may be defined. These are now discussed in turn.

3 Use of RTP Header Extension

The RTP specification [2] states that applications should be prepared to ignore a header extension. Including all the redundancy information for a packet in a header extension would make it easy for applications that do not implement redundancy to discard it and just process the primary encoding data. There are, however, a number of disadvantages with this scheme:

- There is a large overhead from the number of bytes needed for the extension header (4) and the possible padding that is needed at the end of the extension to round up to a four byte boundary (up to 3 bytes). Even for longer duration packets especially when high compression encodings are used the overhead is considerable.
- Use of the header extension limits applications to a single redundant encoding, unless further structure is introduced into the extension. This would result in further overhead.

For these reasons, the use of RTP header extension to hold redundant audio encodings is disregarded.

4 Use Of Additional RTP Payload Types

Currently the RTP profile for audio and video conferences [3] lists a set of payload types and provides for a dynamic range of 32 encodings that may be defined through a conference control protocol. This leads to two possible schemes for assigning additional RTP payload types for redundant audio applications:

1. A dynamic encoding scheme may be defined, for each combination of primary/redundant payload types, using the RTP dynamic payload type range.
2. A fixed payload type may be defined to represent a packet with redundancy. This may then be assigned to either a static RTP payload type, or the payload type for this may be assigned dynamically.

4.1 Dynamic Encoding Schemes

It is possible to define a set of payload types that signify a particular combination of primary and secondary encodings for each of the 32 dynamic payload types provided. This would be a slightly restrictive yet feasible solution for packets with a single block of redundancy as the number of possible combinations is not too large. However the need for multiple blocks of redundancy greatly increases the number of encoding combinations and makes this solution not viable.

A modified version of the above solution could be to decide prior to the beginning of a conference on a set a 32 encoding combinations that will be used for the duration of the conference. All tools in the conference can be initialized with this working set of encoding combinations. Communication of the working set can be made through the use of an external mechanism such as SDR. Setup is complicated as great care needs to be taken in starting tools with identical parameters. This scheme is more efficient as only one byte is used to identify combinations of encodings.

4.2 Payload type to mean packet-with-redundancy

A more flexible solution would be to have only one payload type to signify a packet with redundancy and have each of the encoding blocks in the packet contain it's own payload type field: such a packet acts as a container, encapsulating multiple packets into one.

Such a scheme is flexible, since any number of redundant encodings may be enclosed within a single packet. There is, however, a small overhead since each encapsulated packet must be preceded by a header indicating the type of data enclosed.

This is the preferred solution, since it is both flexible, extensible, and has a relatively low overhead. The remainder of this document describes this solution.

5 RTP payload type for redundant data

The assignment of an RTP payload type for this new packet format is outside the scope of this document, and will not be specified here. An RTP packet containing redundant data shall have a standard RTP header, with payload type indicating redundancy. The other fields of the RTP header relate to the primary data block of the redundant data.

Following the RTP header are a number of additional headers, specified in the figure below, which specify the contents of each of the encodings carried by the packet. Following these additional headers are a number of data blocks, which contain the standard RTP payload data for these encodings. It is noted that all the headers are aligned to a 32 bit boundary, but that the payload data will typically not be aligned.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|F|   block PT  |  timestamp offset         |   block length    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The bits in the header are specified as follows:

F: 1 bit First bit in header indicates whether another encoding block follows. If 1 further redundant data blocks follow, if 0 this is the last data block.

block PT: 7 bits Payload type for this block.

timestamp offset: 14 bits Unsigned offset of timestamp of this block relative to timestamp given in RTP header. The use of an unsigned offset implies that redundant data must be sent after the primary data, and is hence a time to be subtracted from the current timestamp to determine the timestamp of the data for which this block is the redundancy.

block length: 10 bits Length in bytes of the next redundant data block excluding header.

It is noted that this limits the use of redundant data slightly: it is not possible to send redundancy before the primary encoding. This may possibly affect schemes, such as GSM audio, where a low bandwidth coding suitable for redundancy is produced early in the encoding process, and hence could feasibly be transmitted early. The addition of a sign bit would unacceptably reduce the range of the timestamp offset, and increasing the size of the field above 14 bits limits the block length field. It seems that limiting redundancy to be transmitted after the primary will cause fewer problems than limiting the size of the other fields.

It is noted that the block length and timestamp offset are 10 bits, and 14 bits respectively; rather than the more obvious 8 and 16 bits. Whilst such an encoding complicates parsing the header information slightly, and adds some additional processing overhead, there are a number of problems involved with the more obvious choice: An 8 bit block length field is sufficient for most, but not all, possible encodings: for example 80ms PCM and DVI audio packets comprise more than 256 bytes, and cannot be encoded with a single byte length field. It is possible to impose additional structure on the block length field (for example the high bit set could imply the lower 7 bits code a length in words, rather than bytes), however such schemes are complex. The use of a 10 bit block length field retains simplicity and provides an enlarged range, at the expense of a reduced range of timestamp values. A 14 bit timestamp value does, however, allow for 4.5 complete packets delay with 48KHz audio, more at lower sampling rates, and it is felt that this is sufficient.

The primary encoding block should be placed last in the packet. It is therefore required to omit the timestamp and block-length fields from the header of this block, since they may be determined from the RTP header and overall packet length. The header for the primary (final) block comprises only a zero marker bit, and the block payload type information, a total of 8 bits. This is illustrated in below:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|0|   Block PT  |
+-+-+-+-+-+-+-+-+
```

6 Limitations

The RTP marker bit is not preserved. That is, a redundant copy of the RTP marker is not sent, hence if the primary (containing this marker) is lost, the marker is lost. It is not thought that this will cause undue problems: even if the marker bit was transmitted with the redundant information, there would be the possibility of its loss, so applications would still have to be written with this in mind.

7 Example Packet

A redundant audio data packet containing DVI4 (8KHz) primary, and a single block of redundancy encoded using 8KHz LPC is illustrated:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |  sequence number of primary   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              timestamp of primary encoding                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1| block PT=7  |  timestamp offset         |   block length    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0| block PT=5  |                                               |
+-+-+-+-+-+-+-+-+                                               +
|                                                               |
+               LPC encoded redundant data                      +
|                                                               |
+                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
|                                                               |
+                                                               +
|               DVI4 encoded primary data                       |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

8 Author's Addresses

Mark Handley/Vicky Hardman/Isidor Kouvelas/Colin Perkins
Department of Computer Science
University College London
London WC1E 6BT
United Kingdom
Email: M.Handley—V.Hardman—I.Kouvelas—C.Perkins@cs.ucl.ac.uk

Jean-Chrysostome Bolot/Andres Vega-Garcia/Sacha Fosse-Parisis
INRIA Sophia Antipolis
2004 Route des Lucioles, BP 93
06902 Sophia Antipolis
France
Email: bolot@sophia.inria.fr

9 References

[1] Hardman, V. J. and Sasse, M. A. and Handley, M. and Watson, A., Reliable Audio for Use over the Internet, Proceecings INET'95, Honalulu, Oahu, Hawaii, September 1995. http://www.isoc.org/in95prc/

[2] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, RFC 1889, January 1996

[3] H. Schulzrinne, RTP Profile for Audio and Video Conferences with Minimal Control, RFC 1890, January 1996