Internet Engineering Task Force
Internet Draft
draft-ietf-avt-reedsolomon-00.txt
November 3, 1998
Expires: May 2, 1999

Audio Video Transport WG
J.Rosenberg,H.Schulzrinne
Bell Laboratories,Columbia U.

An RTP Payload Format for Reed Solomon Codes

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.

ABSTRACT

This document specifies a payload format for forward error correction of media encapsulated in RTP using Reed Solomon codes. The payload format allows end systems to transmit using arbitrary block lengths and codes. It also allows for the recovery of both the payload and critical RTP header fields. Since FEC is sent as a separate stream, it is backwards compatible with non-FEC capable hosts, so that receivers which do not wish to implement FEC can just ignore the extensions.

1 Introduction

The quality of packet voice on the Internet has been mediocre due, in part, to high packet loss rates. This is especially true on wide-area connections. Unfortunately, the strict delay requirements of real-time multimedia usually eliminate the possibility of retransmissions. It is for this reason that forward error correction (FEC) has been proposed to compensate for packet loss in the Internet [1] [2]. In particular, the use of traditional error correcting codes, such as parity, Reed-Solomon, and Hamming codes, has attracted attention. To support these mechanisms, protocol support is required.

This document defines a payload format for RTP which allows for forward error correction of media based on Reed Solomon (RS) codes. It is similar in design to [3], which specifies forward error correction using exclusive or based parity codes. As with that format, the Reed Solomon format described here is generic. This means that the protocol is (1) independent of the nature of the media being protected, be it audio, video, or otherwise, (2) flexible enough to support a wide variety of RS codes, (3) designed for adaptivity so that the FEC technique can be modified easily without out of band signaling, and (4) supportive of a number of different mechanisms for transporting the FEC packets.

2 Terminology

The following terms are used throughout this document:

1. Media Payload: is a piece of raw, un-protected user data which is to be transmitted from the sender. The media payload would is placed inside of an RTP packet.

2. Media Header: is the RTP header for the packet containing the media payload.

3. Media Packet: The combination of a media payload and media header is called a media packet.

4. FEC Packet: The forward error correction algorithms at the transmitter take the media packets as an input. They output both the media packets that they are passed, and new packets called FEC packets. The FEC packets are formatted according to the rules specified in this document.

5. FEC Header: The FEC header is the header information contained in an FEC packet.

6. FEC Payload: The FEC payload is the payload in an FEC packet.

7. Media Block: The media block is a set of K consecutive media packets which are used to generate the FEC packets.

8. Coding Block: The coding block is a set of N packets, consisting of the K packets from a single media block, plus N-K additional FEC packets generated from the media block.

9. K: K is a variable which represents the number of media packets in a media block.

10. N: N is a variable which represents the total number of packets in a coding block.

11. Reed Solomon Code: A Reed Solomon code is uniquely determined by the values of N and K, as defined above, and the symbol length.

3 Basic Operation

The media packets are broken into blocks of K. K can vary from block to block. The Reed Solomon coding is used to obtain N-K FEC packets which protect the K media packets.

The FEC packets are sent as a separate stream from the media packets. This implies that the FEC packets have their own sequence number space. Although the timestamps for the FEC packets are derived from the media packets, they increment monotonically. Combined together, FEC packet streams work well with RTP header compression. The media packet stream is unaffected by the use of FEC. This allows the two to be sent on a separate multicast group, so that non-FEC receivers can ignore the FEC and just receive the original media. The separation also allows for coherent values for the sequence numbers and timestamps.

This document does not prescibe the definition of "separate streams", but leaves this to applications and higher level protocols to define. For multicast, the separate stream may be implemented by separate multicast groups, different ports in the same group, or by a different SSRC within the same group/port. For unicast, different ports or different SSRC may be used. Each of these approaches has drawbacks and benefits which depend on the application.

At the receiver, arriving FEC and media packets are used to generate a stream of media packets for direct use by the application. This results in a clean separation of error protection from the application.

FEC packets encoded according to this document are indicated through

the payload type in the packet header. These payload types are signaled dynamically.

4 Reed Solomon Codes

The detailed operation and theory behind Reed Solomon codes is not important here. A Reed Solomon code takes a group of K data blocks and generates N - K FEC blocks. A receiver needs to receive any K of the N data or FEC blocks in order to recover the K data blocks. Besides K and N, the only parameters of the algorithm are the symbol length, which is the number of bits per symbol used in the algorithm. If the symbol length is l, the size of the data block must be a multiple of l. For more information of Reed Solomon codes, the reader is referred to [4].

Reed Solomon codes are systematic. This means that the K data blocks are not modified as a result of the FEC operation.

5 RTP Media Packet Structure

The media packets and FEC packets are sent as separate streams. The media packets are unaffected by FEC, and are sent in the same fashion they would be sent if there were no FEC.

This lends to a very efficient encoding. When little (or no) FEC is used, there are mostly media packets being sent. This means that the overhead (present in FEC packets only) tracks the amount of FEC in use.

6 FEC Packet Structure

When a packet is to be transmitted which contains FEC data, the RTP header is followed by an FEC header. We first discuss the semantics of the RTP header fields.

The version field is set to 2. The padding bit is computed via the protection operation, defined below. The extension bit is also computed via the protection operation. The SSRC value should generally be the same as the SSRC value of the media stream it protects. It MAY be different if the FEC stream is being demultiplexed via the SSRC value. The CC value is computed via the protection operation. The CSRC list is never present, independent of the value of the CC field. The extension is never present, independent of the value of the X bit. The marker bit is computed via the protection operation.

The sequence number has the standard definition: it is one higher than the sequence number in the previously transmitted FEC packet. The timestamp is set in the following fashion. When the FEC packet is

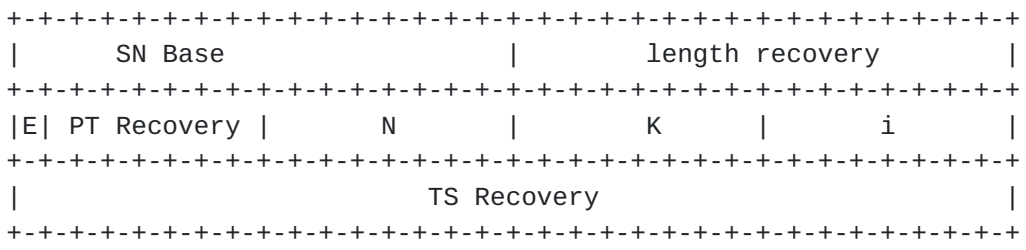
sent, the value of the media RTP timestamp is examined. This value is used as the timestamp of the FEC packet. This results in the TS value in FEC packets to be monotonically increasing, independent of the FEC scheme.

The payload type is obtained through out of band signaling. The signaling protocol MUST establish a symbol length to be associated with the payload type value. End systems which cannot recognize a payload type must discard it. This provides backwards compatibility. The FEC mechanisms can then be used in a multicast group with mixed FEC-capable and FEC-uncapable receivers.

Following the RTP header is the Reed Solomon header.

6.1 Reed Solomon Header

The format of the Reed-Solomon FEC header is shown below.



The length recovery field is used to determine the length of any recovered packets. It is computed via the protection operation applied to the 16 bit natural binary representation of the lengths (in bytes) of the media payload, CSRC list, extension and padding of media packets in the media block (in other words, the CSRC list, extension, and padding, if present, are "counted" as part of the payload). This allows for the FEC procedure to be applied even when the lengths of the media packets are not identical.

The E bit indicates a header extension. Implementations conforming to this version of the specification MUST set this bit to zero.

The PT recovery field is obtained via the protection operation applied to the payload type values of the media packets in the media block.

The TS recovery field is computed via the protection operation

applied to the timestamps of the media packets in the media block. This allows the timestamp to be completely recovered.

The SN base is the sequence number of the first media packet in the media block protected by FEC. The K field is the number of media packets in the media block, minus 1. The N field is the total number of FEC plus data packets in the coding block, minus 1. The i field indicates that this packet is the i+1th FEC packet of the N-K FEC packets in the code. The symbol length of the code is indicated by means of the payload type field in the RTP header.

The payload of the FEC packet is the protection operation applied to the CSRC List plus payloads plus extension plus padding of the media packets in the media block.

7 Protection Operation

The protection operation involves taking a variety of fields from the various headers, adding the payloads, appending the whole thing together, padding zeroes, and then computing the FEC across the resulting binary block. The result is then placed into the FEC packet.

For each media block consisting of K media packets, K binary arrays are generated (one for each packet) by appending the following fields from the header and payload together:

- Padding Bit (1 bit)
- Extension Bit (1 bit)
- CC bits (3 bits)
- Marker bit (1 bit)
- Payload Type (7 bits)
- Timestamp (32 bits)
- Natural binary representation of the length of the CSRC List plus padding plus payload plus extension of the media packet (16 bits)
- CSRC List (if CC is 1), else null (variable)
- Header Extension (if X is 1), else null (variable)
- the payload (variable)

o Padding, if present (variable)

If the lengths of the binary arrays are not equal, they are padded with zeroes to be the length of the longest binary array. If the resulting binary arrays have a length which is not a multiple of the symbol length, they are all padded further until they are a multiple of the symbol length.

The Reed Solomon encoding operation is then applied to the K binary arrays, generating N-K FEC arrays. Each FEC array is used to generate a single FEC packet.

The first bit in the FEC packet binary array is written into the Padding Bit of the FEC packet. The second bit in the FEC packet binary array is written into the Extension bit of the FEC packet. The next three bits of the FEC packet binary array are written into the CC field of the FEC packet. The next bit of the FEC packet binary array is written into the marker bit of the FEC packet. The next 7 bits of the FEC packet binary array are written into the PT recovery field in the FEC packet header. The next 32 bits of the FEC packet binary array are written into the TS recovery field in the packet header. The next 16 bits are written into the Length Recovery field in the FEC packet header. The remaining bits are set to be the payload of the FEC packet.

8 Recovery Procedures

The FEC packets allow end systems to recover from the loss of media packets. All of the header fields of the missing packets, including CSRC lists, extensions, padding bits, marker and payload type, are recoverable. This section describes the procedure for performing this recovery.

Recovery requires two distinct operations. The first determines when recovery should be attempted, based on the packets which have arrived. The second is to actually perform the reconstruction.

The determination of when to recover is straightforward. For each coding block, once K of the N packets in the block have arrived, recovery can be attempted. Each FEC packet contains an FEC base, and the value of K an N. Using these, it is easy to determine when K of N packets have been received.

One approach for an implementation of this logic is to use linked lists of packets. Each list is associated with two numbers: an SN base and a value of K. When an FEC packet arrives, if there is no list associated with its SN base field, a new list is created. Otherwise, the FEC packet is placed in the list associated with that SN

base field. When a media packet arrives, its sequence number is checked against each list. If its SN is between the SN base and SN base plus K, the media packet is also placed in the list. Once a list has K packets in it, recovery can be peformed.

8.1 Reconstruction

Reconstruction is possible when any K of the packets (media or FEC) from the coding block have arrived. Let T be the list of packets (FEC and media) which can be combined to recover some media packet xi. The procedure is as follows:

1. For each of the media packets in T, compute the binary array as described in the previous section.

2. For the FEC packets in T, compute the binary array in the same fashion, except always set the CSRC list, extension, and padding to null.

3. If the resulting K binary arrays are not of equal length, pad them with zeroes to be the length of the longest binary array.

4. If the lengths are not a multiple of the symbol length, pad them until they are a multiple.

5. Apply the Reed Solomon operation to the K binary arrays. This will result in N binary arrays, one of which is the recovery array corresponding to the packet to be recovered.

6. Create a new packet with the standard 12 byte header and no payload.

7. Set the version of the new packet to 2.

8. Set the Padding bit in the new packet to the first bit in the recovery array.

9. Set the Extension bit in the new packet to the second bit in the recovery array.

10. Set the CC field to the next three bits in the recovery array.

11. Set the marker bit in the new packet to the next bit in the recovery array.

12. Set the payload type in the new packet to the next 7 bits

in the recovery array.

13. Set the SN field in the new packet to xi.

14. Set the TS field in the new packet to the next 32 bits in the recovery array.

15. Take the next 16 bits of the recovery array. Whatever the natural binary number this corresponds to, take that many bytes from the recovery array and append them to the new packet. This represents the CSRC list, extension, payload, and padding.

16. Set the SSRC of the new packet to the SSRC of the media stream its protecting

This procedure will completely recover both the header and payload of an RTP packet.

9 Use with Redundant Encodings

One can consider an FEC packet as a "redundant coding" of the media. Because of this, the payload format for encoding of redundant audio data [5] can be used to carry the FEC data along with the media. The procedure for this is simple. In some media packet, the payload type is set to the value for redundant encodings. The secondary coder is then set to be the FEC data. This means that the PTI of the secondary coder is set to the PTI value which indicates FEC. The block length of the secondary coder is set to the length of the FEC header and payload. The timestamp offset is set to the difference between the media timestamp and the timestamp from the FEC packet. The secondary coder payload includes the FEC header and FEC payload.

This procedure only works if an FEC packet is sent after the last of the media packets in the media block has been sent. Otherwise, the timestamp offset would be negative, which is not allowed.

Using the redundant encodings payload format also implies that the marker bit cannot be recovered.

An advantage of this approach is a reduction in the overhead for sending FEC packets.

10 Conclusion

This document has presented a new RTP payload format which allows for Reed Solomon based forward error correction of audio visual media. It is flexible, allowing nearly any Reed Solomon Code to be used. It is

also backwards compatible with existing RTP implementations. Receivers which cannot understand FEC can discard the FEC packets, and still receive the media packets.

11 Security Considerations

The use of FEC has implications on the usage and changing of keys for encryption. As the FEC packets do consist of a separate stream, there are a number of permutations on the usage of encryption. In particular:

- The FEC stream may be encrypted, while the media stream is not.
- The media stream may be encrypted, while the FEC stream is not.
- The media stream and FEC stream are both encrypted, but using different keys.
- The media stream and FEC stream are both encrypted, but using the same key.

The first three of these would require any application level signaling protocols to be aware of the usage of FEC, and to thus exchange keys for it and negotiate its usage on the media and FEC streams separately. In the final case, no such additional mechanisms are needed. Applications utilizing encryption SHOULD encrypt both streams, however. Encrypting just one may make certain known-plaintext attacks possible.

However, the changing of keys becomes problematic. For example, if two packets a and b are sent, and FEC packet f(a,b) is sent, and the keys used for a and b are different, which key should be used to decode f(a,b)? In general, old keys will likely need to be cached, so that when the keys change for the media stream, the old key is kept, and used, until it is determined that the key has changed on the FEC packets as well.

Another issue with the use of FEC is its impact on network congestion. Adding FEC in the face of increasing network losses is a bad idea, as it can lead to increased congestion and eventual congestion collapse if done on a widespread basis. As a result, implementors MUST NOT substantially increase the amount of FEC in use as network losses increase.

12 Full Copyright Statement

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implmentation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works.

However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

13 Author's Addresses

Jonathan Rosenberg
Lucent Technologies, Bell Laboratories
101 Crawfords Corner Rd.
Holmdel, NJ 07733
Rm. 4C-526
email: jdrosen@bell-labs.com

Henning Schulzrinne
Columbia University
M/S 0401
1214 Amsterdam Ave.
New York, NY 10027-7003
email: schulzrinne@cs.columbia.edu

14 Bibliography

[1] J.-C. Bolot and A. Garcia, "The case for fec-based error control for packet audio in the internet," Multimedia Systems , 1997.

[2] C. Perkins and C. Perkins, "Options for repair of streaming media," Request for Comments (Informational) 2354, Internet Engineering Task Force, June 1998.

[3] J. Rosenberg and H. Schulzrinne, "An RTP payload format for generic forward error correction," Internet Draft, Internet Engineering Task Force, Aug. 1998. Work in progress.

[4] L. Rizzo, "Erasure codes for computer communication protocols," technical report, Universita di Pisa, Pisa, Italy, Jan. 1997.

[5] C. Perkins, I. Kouvelas, V. Hardman, M. Handley, and J. Bolot, "RTP payload for redundant audio data," Request for Comments (Proposed Standard) 2198, Internet Engineering Task Force, Sept. 1997.