

RTP Payload Format for Enhanced Variable Rate Codecs (EVRC)  
and Selectable Mode Vocoders (SMV)

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

Abstract

This document describes the RTP payload format for Enhanced Variable Rate Codec (EVRC) Speech and Selectable Mode Vocoder (SMV) Speech. Two sub-formats are specified for different application scenarios. A bundled/interleaved format is included to reduce the effect of packet loss on speech quality and amortize the overhead of the RTP header over more than one speech frame. A non-bundled format is also supported for conversational applications.

Table of Contents

|                                                                     |    |
|---------------------------------------------------------------------|----|
| 1. Introduction .....                                               | 2  |
| 2. Background .....                                                 | 2  |
| 3. The Codecs Supported .....                                       | 3  |
| 3.1. EVRC .....                                                     | 3  |
| 3.2. SMV .....                                                      | 3  |
| 3.3. Other Frame-Based Vocoders .....                               | 4  |
| 4. RTP/Vocoder Packet Format .....                                  | 4  |
| 4.1. Interleaved/Bundled Packet Format .....                        | 5  |
| 4.2. Header-Free Packet Format .....                                | 6  |
| 4.3. Determining the Format of Packets .....                        | 7  |
| 5. Packet Table of Contents Entries and Codec Data Frame Format ... | 7  |
| 5.1. Packet Table of Contents entries .....                         | 7  |
| 5.2. Codec Data Frames .....                                        | 8  |
| 6. Interleaving Codec Data Frames .....                             | 9  |
| 7. Bundling Codec Data Frames .....                                 | 12 |
| 8. Handling Missing Codec Data Frames .....                         | 12 |

|                                                        |    |
|--------------------------------------------------------|----|
| 9. Implementation Issues .....                         | 12 |
| 9.1. Interleaving Length .....                         | 12 |
| 9.2. Validation of Received Packets .....              | 13 |
| 9.3. Processing the Late Packets .....                 | 13 |
| 10. Mode Request .....                                 | 13 |
| 11. Storage Format .....                               | 14 |
| 12. IANA Considerations .....                          | 15 |
| 12.1. Registration of Media Type EVRC .....            | 15 |
| 12.2. Registration of Media Type EVRC0 .....           | 16 |
| 12.3. Registration of Media Type SMV .....             | 17 |
| 12.4. Registration of Media Type SMV0 .....            | 18 |
| 13. Mapping to SDP Parameters .....                    | 19 |
| 14. Security Considerations .....                      | 20 |
| 15. Adding Support of Other Frame-Based Vocoders ..... | 20 |
| 16. Acknowledgements .....                             | 21 |
| 17. References .....                                   | 21 |
| 17.1 Normative .....                                   | 21 |
| 17.2 Informative .....                                 | 22 |
| 18. Author's Address .....                             | 22 |
| 19. Full Copyright Statement .....                     | 23 |

## 1. Introduction

This document describes how speech compressed with EVRC [1] or SMV [2] may be formatted for use as an RTP payload type. The format is also extensible to other codecs that generate a similar set of frame types. Two methods are provided to packetize the codec data frames into RTP packets: an interleaved/bundled format and a zero-header format. The sender may choose the best format for each application scenario, based on network conditions, bandwidth availability, delay requirements, and packet-loss tolerance.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [3].

## 2. Background

The 3rd Generation Partnership Project 2 (3GPP2) has published two standards which define speech compression algorithms for CDMA applications: EVRC [1] and SMV [2]. EVRC is currently deployed in millions of first and second generation CDMA handsets. SMV is the preferred speech codec standard for CDMA2000, and will be deployed in third generation handsets in addition to EVRC. Improvements and new codecs will keep emerging as technology improves, and future handsets will likely support multiple codecs.

The formats of the EVRC and SMV codec frames are very similar. Many other vocoders also share common characteristics, and have many similar application scenarios. This parallelism enables an RTP payload format to be designed for EVRC and SMV that may also support other, similar vocoders with minimal additional specification work. This can simplify the protocol for transporting vocoder data frames through RTP and reduce the complexity of implementations.

### 3. The Codecs Supported

#### 3.1. EVRC

The Enhanced Variable Rate Codec (EVRC) [1] compresses each 20 milliseconds of 8000 Hz, 16-bit sampled speech input into output frames in one of the three different sizes: Rate 1 (171 bits), Rate 1/2 (80 bits), or Rate 1/8 (16 bits). In addition, there are two zero bit codec frame types: null frames and erasure frames. Null frames are produced as a result of the vocoder running at rate 0. Null frames are zero bits long and are normally not transmitted. Erasure frames are the frames substituted by the receiver to the codec for the lost or damaged frames. Erasure frames are also zero bits long and are normally not transmitted.

The codec chooses the output frame rate based on analysis of the input speech and the current operating mode (either normal or one of several reduced rate modes). For typical speech patterns, this results in an average output of 4.2 kilobits/second for normal mode and a lower average output for reduced rate modes.

#### 3.2. SMV

The Selectable Mode Vocoder (SMV) [2] compresses each 20 milliseconds of 8000 Hz, 16-bit sampled speech input into output frames of one of the four different sizes: Rate 1 (171 bits), Rate 1/2 (80 bits), Rate 1/4 (40 bits), or Rate 1/8 (16 bits). In addition, there are two zero bit codec frame types: null frames and erasure frames. Null frames are produced as a result of the vocoder running at rate 0. Null frames are zero bits long and are normally not transmitted. Erasure frames are the frames substituted by the receiver to the codec for the lost or damaged frames. Erasure frames are also zero bits long and are normally not transmitted.

The SMV codec can operate in six modes. Each mode may produce frames of any of the rates (full rate to 1/8 rate) for varying percentages of time, based on the characteristics of the speech samples and the selected mode. The SMV mode can change on a frame-by-frame basis. The SMV codec does not need additional information other than the codec data frames to correctly decode the

data of various modes; therefore, the mode of the encoder does not need to be transmitted with the encoded frames.

The SMV codec chooses the output frame rate based on analysis of the input speech and the current operating mode. For typical speech patterns, this results in an average output of 4.2 kilobits/second for Mode 0 in two way conversation (approximately 50% active speech time and 50% in eighth rate while listening) and lower for other reduced rate modes. SMV is more bandwidth efficient than EVRC. EVRC is equivalent in performance to SMV mode 1.

### 3.3. Other Frame-Based Vocoders

Other frame-based vocoders can be carried in the packet format defined in this document, as long as they possess the following properties:

- o The codec is frame-based;
- o blank and erasure frames are supported;
- o the total number of rates is less than 17;
- o the maximum full rate frame can be transported in a single RTP packet using this specific format.

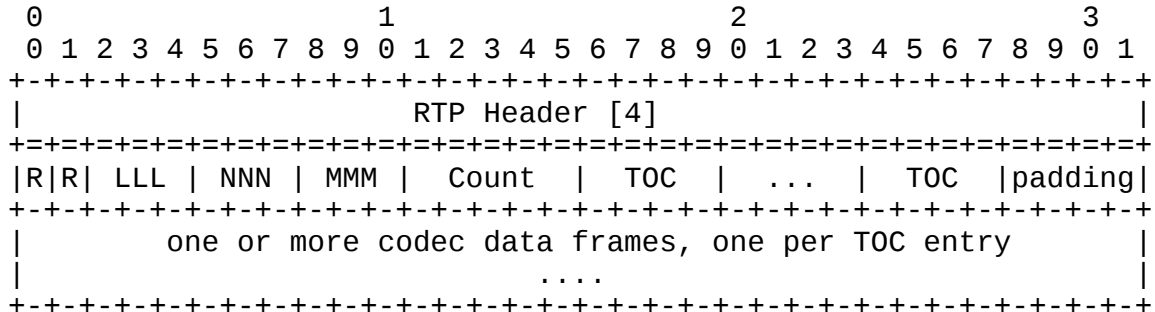
Vocoders with the characteristics listed above can be transported using the packet format specified in this document with some additional specification work; the pieces that must be defined are listed in Section 15.

## 4. RTP/Vocoder Packet Format

The vocoder speech data may be transmitted in either of the two RTP packet formats specified in the following two subsections, as appropriate for the application scenario. In the packet format diagrams shown in this document, bit 0 is the most significant bit.

#### 4.1. Interleaved/Bundled Packet Format

This format is used to send one or more vocoder frames per packet. Interleaving or bundling MAY be used. The RTP packet for this format is as follows:



The RTP header has the expected values as described in the RTP specification [4]. The RTP timestamp is in 1/8000 of a second units for EVRC and SMV. For any other vocoders that use this packet format, the timestamp unit needs to be defined explicitly. The M bit should be set as specified in the applicable RTP profile, for example, RFC 3551 [5]. Note that RFC 3551 [5] specifies that if the sender does not suppress silence, the M bit will always be zero. When multiple codec data frames are present in a single RTP packet, the timestamp is that of the oldest data represented in the RTP packet. The assignment of an RTP payload type for this packet format is outside the scope of this document; it is specified by the RTP profile under which this payload format is used.

The first octet of a Interleaved/Bundled format packet is the Interleave Octet. The second octet contains the Mode Request and Frame Count fields. The Table of Contents (ToC) field then follows. The fields are specified as follows:

Reserved (RR): 2 bits

Reserved bits. MUST be set to zero by sender, SHOULD be ignored by receiver.

Interleave Length (LLL): 3 bits

Indicates the length of interleave; a value of 0 indicates bundling, a special case of interleaving. See Section 6 and Section 7 for more detailed discussion.

Interleave Index (NNN): 3 bits

Indicates the index within an interleave group. MUST have a value less than or equal to the value of LLL. Values of NNN greater than the value of LLL are invalid. Packet with invalid NNN values SHOULD be ignored by the receiver.

Mode Request (MMM): 3 bits

The Mode Request field is used to signal Mode Request information. See Section 10 for details.

Frame Count (Count): 5 bits

The number of ToC fields (and vocoder frames) present in the packet is the value of the frame count field plus one. A value of zero indicates that the packet contains one ToC field, while a value of 31 indicates that the packet contains 32 ToC fields.

Padding (padding): 0 or 4 bits

This padding ensures that codec data frames start on an octet boundary. When the frame count is odd, the sender MUST add 4 bits of padding following the last TOC. When the frame count is even, the sender MUST NOT add padding bits. If padding is present, the padding bits MUST be set to zero by sender, and SHOULD be ignored by receiver.

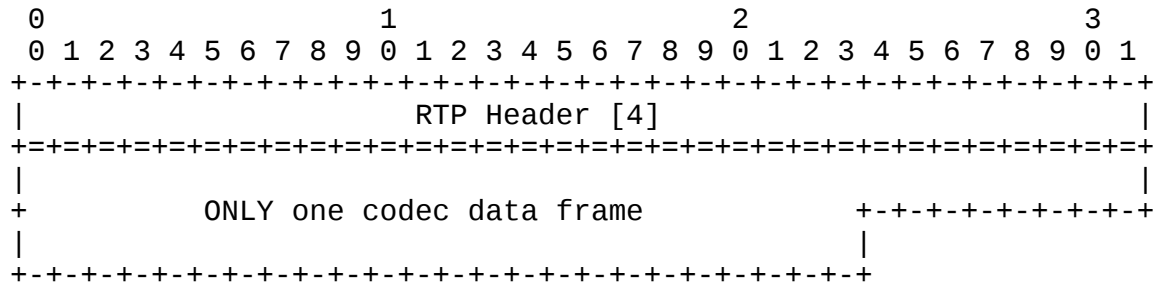
The Table of Contents field (ToC) provides information on the codec data frame(s) in the packet. There is one ToC entry for each codec data frame. The detailed formats of the ToC field and codec data frames are specified in Section 5.

Multiple data frames may be included within a Interleaved/Bundled packet using interleaving or bundling as described in Section 6 and Section 7.

#### 4.2. Header-Free Packet Format

The Header-Free Packet Format is designed for maximum bandwidth efficiency and low latency. Only one codec data frame can be sent in each Header-Free format packet. None of the payload header fields (LLL, NNN, MMM, Count) nor ToC entries are present. The codec rate for the data frame can be determined from the length of the codec data frame, since there is only one codec data frame in each Header-Free packet.

Use of the RTP header fields for Header-Free RTP/Vocoder Packet Format is the same as described in Section 4.1 for Interleaved/Bundled RTP/Vocoder Packet Format. The detailed format of the codec data frame is specified in Section 5.



#### 4.3. Determining the Format of Packets

All receivers SHOULD be able to process both packet formats. The sender MAY choose to use one or both packet formats.

A receiver MUST have prior knowledge of the packet format to correctly decode the RTP packets. When packets of both formats are used within the same session, different RTP payload type values MUST be used for each format to distinguish the packet formats. The association of payload type number with the packet format is done out-of-band, for example by SDP during the setup of a session.

### 5. Packet Table of Contents Entries and Codec Data Frame Format

#### 5.1. Packet Table of Contents entries

Each codec data frame in a Interleaved/Bundled packet has a corresponding Table of Contents (ToC) entry. The ToC entry indicates the rate of the codec frame. (Header-Free packets MUST NOT have a ToC field.)

Each ToC entry occupies four bits. The format of the bits is indicated below:

```

    0 1 2 3
+--+--+--+
|fr type|
+--+--+--+

```

Frame Type: 4 bits

The frame type indicates the type of the corresponding codec data frame in the RTP packet.

For EVRC and SMV codecs, the frame type values and size of the associated codec data frame are described in the table below:

| Value | Rate    | Total codec data frame size (in octets) |                                        |
|-------|---------|-----------------------------------------|----------------------------------------|
| 0     | Blank   | 0                                       | (0 bit)                                |
| 1     | 1/8     | 2                                       | (16 bits)                              |
| 2     | 1/4     | 5                                       | (40 bits; not valid for EVRC)          |
| 3     | 1/2     | 10                                      | (80 bits)                              |
| 4     | 1       | 22                                      | (171 bits; 5 padded at end with zeros) |
| 5     | Erasure | 0                                       | (SHOULD NOT be transmitted by sender)  |

All values not listed in the above table MUST be considered reserved. A ToC entry with a reserved Frame Type value SHOULD be considered invalid. Note that the EVRC codec does not have 1/4 rate frames, thus frame type value 2 MUST be considered a reserved value when the EVRC codec is in use.

Other vocoders that use this packet format need to specify their own table of frame types and corresponding codec data frames.

## 5.2. Codec Data Frames

The output of the vocoder MUST be converted into codec data frames for inclusion in the RTP payload. The conversions for EVRC and SMV codecs are specified below. (Note: Because the EVRC codec does not have Rate 1/4 frames, the specifications of 1/4 frames does not apply to EVRC codec data frames). Other vocoders that use this packet format need to specify how to convert vocoder output data into frames.

The codec output data bits as numbered in EVRC and SMV are packed into octets. The lowest numbered bit (bit 1 for Rate 1, Rate 1/2, Rate 1/4 and Rate 1/8) is placed in the most significant bit (internet bit 0) of octet 1 of the codec data frame, the second lowest bit is placed in the second most significant bit of the first octet, the third lowest in the third most significant bit of the first octet, and so on. This continues until all of the bits have been placed in the codec data frame.

The remaining unused bits of the last octet of the codec data frame MUST be set to zero. Note that in EVRC and SMV this is only applicable to Rate 1 frames (171 bits) as the Rate 1/2 (80 bits), Rate 1/4 (40 bits, SMV only) and Rate 1/8 frames (16 bits) fit exactly into a whole number of octets.

Following is a detailed listing showing a Rate 1 EVRC/SMV codec output frame converted into a codec data frame:

The codec data frame for a EVRC/SMV codec Rate 1 frame is 22 octets long. Bits 1 through 171 from the EVRC/SMV codec Rate 1 frame are placed as indicated, with bits marked with "Z" set to zero. EVRC/SMV codec Rate 1/8, Rate 1/4 and Rate 1/2 frames are converted similarly, but do not require zero padding because they align on octet boundaries.

## Rate 1 codec data frame

| 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 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |  |
| + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |  |
| 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 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |   |  |
| + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - |  |
| : |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | : |  |
| + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - |  |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |   |   |   |   |   |   |   |   |   |   |  |
| 4 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 7 | 7 | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z |  |
| 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 |   |   |   |   |   |   |   |   |   |   |   |   |   |  |
| + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - |  |

## 6. Interleaving Codec Data Frames

As indicated in Section 4.1, more than one codec data frame MAY be included in a single Interleaved/Bundled packet by a sender. This is accomplished by interleaving or bundling.

Bundling is used to spread the transmission overhead of the RTP and payload header over multiple vocoder frames. Interleaving additionally reduces the listener's perception of data loss by spreading such loss over non-consecutive vocoder frames. EVRC, SMV, and similar vocoders are able to compensate for an occasional lost frame, but speech quality degrades exponentially with consecutive frame loss.

Bundling is signaled by setting the LLL field to zero and the Count field to greater than zero. Interleaving is indicated by setting the LLL field to a value greater than zero.

The discussions on general interleaving apply to the bundling (which can be viewed as a reduced case of interleaving) with reduced complexity. The bundling case is discussed in detail in Section 7.

Senders MAY support interleaving and/or bundling. All receivers that support Interleave/Bundling packet format MUST support both interleaving and bundling.

Given a time-ordered sequence of output frames from the codec numbered  $0..n$ , a bundling value  $B$  (the value in the Count field plus one), and an interleave length  $L$  where  $n = B * (L+1) - 1$ , the output frames are placed into RTP packets as follows (the values of the fields LLL and NNN are indicated for each RTP packet):

First RTP Packet in Interleave group:

LLL=L, NNN=0

Frame 0, Frame  $L+1$ , Frame  $2(L+1)$ , Frame  $3(L+1)$ , ... for a total of  $B$  frames

Second RTP Packet in Interleave group:

LLL=L, NNN=1

Frame 1, Frame  $1+L+1$ , Frame  $1+2(L+1)$ , Frame  $1+3(L+1)$ , ... for a total of  $B$  frames

This continues to the last RTP packet in the interleave group:

$L+1$  RTP Packet in Interleave group:

LLL=L, NNN=L

Frame  $L$ , Frame  $L+L+1$ , Frame  $L+2(L+1)$ , Frame  $L+3(L+1)$ , ... for a total of  $B$  frames

Within each interleave group, the RTP packets making up the interleave group MUST be transmitted in value-increasing order of the NNN field. While this does not guarantee reduced end-to-end delay on the receiving end, when packets are delivered in order by the underlying transport, delay will be reduced to the minimum possible.

Receivers MAY signal the maximum number of codec data frames (i.e., the maximum acceptable bundling value  $B$ ) they can handle in a single RTP packet using the OPTIONAL maxptime RTP mode parameter identified in Section 12.

Receivers MAY signal the maximum interleave length (i.e., the maximum acceptable LLL value in the Interleaving Octet) they will accept using the OPTIONAL maxinterleave RTP mode parameter identified in Section 12.

The parameters maxptime and maxinterleave are exchanged at the initial setup of the session. In one-to-one sessions, the sender MUST respect these values set by the receiver, and MUST NOT interleave/bundle more packets than what the receiver signals that it can handle. This ensures that the receiver can allocate a known amount of buffer space that will be sufficient for all interleaving/bundling used in that session. During the session, the sender may decrease the bundling value or interleaving length (so that less buffer space is required at the receiver), but never exceed

the maximum value set by the receiver. This prevents the situation where a receiver needs to allocate more buffer space in the middle of a session but is unable to do so.

Additionally, senders have the following restrictions:

- o MUST NOT bundle more codec data frames in a single RTP packet than indicated by maxptime (see Section 12) if it is signaled.
- o SHOULD NOT bundle more codec data frames in a single RTP packet than will fit in the MTU of the underlying network.
- o Once beginning a session with a given maximum interleaving value set by maxinterleave in Section 12, MUST NOT increase the interleaving value (LLL) to exceed the maximum interleaving value that is signaled.
- o MAY change the interleaving value, but MUST do so only between interleave groups.
- o Silence suppression MUST only be used between interleave groups. A ToC with Frame Type 0 (Blank Frame, Section 5.1) MUST be used within interleaving groups if the codec outputs a blank frame. The M bit in the RTP header is not set for these blank frames, as the stream is continuous in time. Because there is only one time stamp for each RTP packet, silence suppression used within an interleave group would cause ambiguities when reconstructing the speech at the receiver side, and thus is prohibited.

Given an RTP packet with sequence number S, interleave length (field LLL) L, interleave index value (field NNN) N, and bundling value B, the interleave group consists of this RTP packet and other RTP packets with sequence numbers from  $S - N \bmod 65536$  to  $S - N + L \bmod 65536$  inclusive. In other words, the interleave group always consists of L+1 RTP packets with sequential sequence numbers. The bundling value for all RTP packets in an interleave group MUST be the same.

The receiver determines the expected bundling value for all RTP packets in an interleave group by the number of codec data frames bundled in the first RTP packet of the interleave group received. Note that this may not be the first RTP packet of the interleave group if packets are delivered out of order by the underlying transport.

## 7. Bundling Codec Data Frames

As discussed in Section 6, the bundling of codec data frames is a special reduced case of interleaving with LLL value in the Interleave Octet set to 0.

Bundling codec data frames indicates that multiple data frames are included consecutively in a packet, because the interleaving length (LLL) is 0. The interleaving group is thus reduced to a single RTP packet, and the reconstruction of the codec data frames from RTP packets becomes a much simpler process.

Furthermore, the additional restrictions on senders are reduced to:

- o MUST NOT bundle more codec data frames in a single RTP packet than indicated by maxptime (see Section 12) if it is signaled.
- o SHOULD NOT bundle more codec data frames in a single RTP packet than will fit in the MTU of the underlying network.

## 8. Handling Missing Codec Data Frames

The vocoders covered by this payload format support erasure frames as an indication when frames are not available. The erasure frames are normally used internally by a receiver to advance the state of the voice decoder by exactly one frame time for each missing frame. Using the information from packet sequence number, time stamp, and the M bit, the receiver can detect missing codec data frames from RTP packet loss and/or silence suppression, and generate corresponding erasure frames. Erasure frames MUST also be used in storage format to record missing frames.

## 9. Implementation Issues

### 9.1. Interleaving Length

The vocoder interpolates the missing speech content when given an erasure frame. However, the best quality is perceived by the listener when erasure frames are not consecutive. This makes interleaving desirable as it increases speech quality when packet loss occurs.

On the other hand, interleaving can greatly increase the end-to-end delay. Where an interactive session is desired, either Interleaved/Bundled packet format with interleaving length (field LLL) 0 or Header-Free packet format is RECOMMENDED.

When end-to-end delay is not a primary concern, an interleaving length (field LLL) of 4 or 5 is RECOMMENDED as it offers a reasonable compromise between robustness and latency.

## 9.2. Validation of Received Packets

When receiving an RTP packet, the receiver SHOULD check the validity of the ToC fields and match the length of the packet with what is indicated by the ToC fields. If any invalidity or mismatch is detected, it is RECOMMENDED to discard the received packet to avoid potential severe degradation of the speech quality. The discarded packet is treated following the same procedure as a lost packet, and the discarded data will be replaced with erasure frames.

On receipt of an RTP packet with an invalid value of the LLL or NNN fields, the RTP packet SHOULD be treated as lost by the receiver for the purpose of generating erasure frames as described in Section 8.

On receipt of an RTP packet in an interleave group with other than the expected frame count value, the receiver MAY discard codec data frames off the end of the RTP packet or add erasure codec data frames to the end of the packet in order to manufacture a substitute packet with the expected bundling value. The receiver MAY instead choose to discard the whole interleave group.

## 9.3. Processing the Late Packets

Assume that the receiver has begun playing frames from an interleave group. The time has come to play frame x from packet n of the interleave group. Further assume that packet n of the interleave group has not been received. As described in Section 8, an erasure frame will be sent to the receiving vocoder.

Now, assume that packet n of the interleave group arrives before frame x+1 of that packet is needed. Receivers should use frame x+1 of the newly received packet n rather than substituting an erasure frame. In other words, just because packet n was not available the first time it was needed to reconstruct the interleaved speech, the receiver should not assume it is not available when it is subsequently needed for interleaved speech reconstruction.

## 10. Mode Request

The Mode Request signal requests a particular encoding mode for the speech encoding in the reverse direction. All implementations are RECOMMENDED to honor the Mode Request signal. The Mode Request signal SHOULD only be used in one-to-one sessions. In multi-party sessions, any received Mode Request signals SHOULD be ignored.

In addition, the Mode Request signal MAY also be sent through non-RTP means, which is out of the scope of this specification.

The three-bit Mode Request field is used to signal the receiver to set a particular encoding mode to its audio encoder. If the Mode Request field is set to a valid value in RTP packets from node A to node B, it is a request for node B to change to the requested encoding mode for its audio encoder and therefore the bit rate of the RTP stream from node B to node A. Once a node sets this field to a value, it SHOULD continue to set the field to the same value in subsequent packets until the requested mode is different. This design helps to eliminate the scenario of getting the codec stuck in an unintended state if one of the packets that carries the Mode Request is lost. An otherwise silent node MAY send an RTP packet containing a blank frame in order to send a Mode Request.

Each codec type using this format SHOULD define its own interpretation of the Mode Request field. Codecs SHOULD follow the convention that higher values of the three-bit field correspond to an equal or lower average output bit rate.

For the EVRC codec, the Mode Request field MUST be interpreted according to Tables 2.2.1.2-1 and 2.2.1.2-2 of the EVRC codec specifications [1].

For SMV codec, the Mode Request field MUST be interpreted according to Table 2.2-2 of the SMV codec specifications [2].

## 11. Storage Format

The storage format is used for storing speech frames, e.g., as a file or e-mail attachment.

The file begins with a magic number to identify the vocoder that is used. The magic number for EVRC corresponds to the ASCII character string "#!EVRC\n", i.e., "0x23 0x21 0x45 0x56 0x52 0x43 0x0A". The magic number for SMV corresponds to the ASCII character string "#!SMV\n", i.e., "0x23 0x21 0x53 0x4d 0x56 0x0A".

The codec data frames are stored in consecutive order, with a single TOC entry field, extended to one octet, prefixing each codec data frame. The ToC field is extended to one octet by setting the four most significant bits of the octet to zero. For example, a ToC value of 4 (a full-rate frame) is stored as 0x04.

Speech frames lost in transmission and non-received frames MUST be stored as erasure frames (frame type 5, see definition in Section 5.1) to maintain synchronization with the original media.

## 12. IANA Considerations

Four new MIME sub-types as described in this section have been registered by the IANA.

The MIME-names for the EVRC and SMV codec are allocated from the IETF tree since all the vocoders covered are expected to be widely used for Voice-over-IP applications.

### 12.1. Registration of Media Type EVRC

Media Type Name: audio

Media Subtype Name: EVRC

Required Parameter: none

Optional parameters:

The following parameters apply to RTP transfer only.

ptime: Defined as usual for RTP audio (see RFC 2327).

maxptime: The maximum amount of media which can be encapsulated in each packet, expressed as time in milliseconds. The time SHALL be calculated as the sum of the time the media present in the packet represents. The time SHOULD be a multiple of the duration of a single codec data frame (20 msec). If not signaled, the default maxptime value SHALL be 200 milliseconds.

maxinterleave: Maximum number for interleaving length (field LLL in the Interleaving Octet). The interleaving lengths used in the entire session MUST NOT exceed this maximum value. If not signaled, the maxinterleave length SHALL be 5.

Encoding considerations:

This type is defined for transfer of EVRC-encoded data via RTP using the Interleaved/Bundled packet format specified in Sections 4.1, 6, and 7 of RFC 3558. It is also defined for other transfer methods using the storage format specified in Section 11 of RFC 3558.

Security considerations:

See Section 14 "Security Considerations" of RFC 3558.

Public specification:

The EVRC vocoder is specified in 3GPP2 C.S0014. Transfer methods are specified in RFC 3558.

Additional information:

The following information applies for storage format only.

Magic number: #!EVRC\n (see Section 11 of RFC 3558)

File extensions: evc, EVC

Macintosh file type code: none

Object identifier or OID: none

Intended usage:

COMMON. It is expected that many VoIP applications (as well as mobile applications) will use this type.

Person & email address to contact for further information:

Adam Li

adamli@icsl.ucla.edu

Author/Change controller:

Adam Li

adamli@icsl.ucla.edu

IETF Audio/Video Transport Working Group

## 12.2. Registration of Media Type EVRC0

Media Type Name: audio

Media Subtype Name: EVRC0

Required Parameters: none

Optional parameters: none

Encoding considerations: none

This type is only defined for transfer of EVRC-encoded data via RTP using the Header-Free packet format specified in Section 4.2 of RFC 3558.

Security considerations:

See Section 14 "Security Considerations" of RFC 3558.

Public specification:

The EVRC vocoder is specified in 3GPP2 C.S0014. Transfer methods are specified in RFC 3558.

Additional information: none

Intended usage:

COMMON. It is expected that many VoIP applications (as well as mobile applications) will use this type.

Person & email address to contact for further information:

Adam Li  
adamli@icsl.ucla.edu

Author/Change controller:

Adam Li  
adamli@icsl.ucla.edu  
IETF Audio/Video Transport Working Group

### 12.3. Registration of Media Type SMV

Media Type Name: audio

Media Subtype Name: SMV

Required Parameter: none

Optional parameters:

The following parameters apply to RTP transfer only.

ptime: Defined as usual for RTP audio (see RFC 2327).

maxptime: The maximum amount of media which can be encapsulated in each packet, expressed as time in milliseconds. The time SHALL be calculated as the sum of the time the media present in the packet represents. The time SHOULD be a multiple of the duration of a single codec data frame (20 msec). If not signaled, the default maxptime value SHALL be 200 milliseconds.

maxinterleave: Maximum number for interleaving length (field LLL in the Interleaving Octet). The interleaving lengths used in the entire session MUST NOT exceed this maximum value. If not signaled, the maxinterleave length SHALL be 5.

Encoding considerations:

This type is defined for transfer of SMV-encoded data via RTP using the Interleaved/Bundled packet format specified in Section 4.1, 6, and 7 of RFC 3558. It is also defined for other transfer methods using the storage format specified in Section 11 of RFC 3558.

Security considerations:

See Section 14 "Security Considerations" of RFC 3558.

Public specification:

The SMV vocoder is specified in 3GPP2 C.S0030-0 v2.0.  
Transfer methods are specified in RFC 3558.

**Additional information:**

The following information applies to storage format only.

Magic number: #!SMV\n (see Section 11 of RFC 3558)

File extensions: smv, SMV

Macintosh file type code: none

Object identifier or OID: none

**Intended usage:**

COMMON. It is expected that many VoIP applications (as well as mobile applications) will use this type.

**Person & email address to contact for further information:**

Adam Li

adamli@icsl.ucla.edu

**Author/Change controller:**

Adam Li

adamli@icsl.ucla.edu

IETF Audio/Video Transport Working Group

**12.4. Registration of Media Type SMV0**

Media Type Name: audio

Media Subtype Name: SMV0

Required Parameter: none

Optional parameters: none

Encoding considerations: none

This type is only defined for transfer of SMV-encoded data via RTP using the Header-Free packet format specified in Section 4.2 of RFC 3558.

**Security considerations:**

See Section 14 "Security Considerations" of RFC 3558.

**Public specification:**

The SMV vocoder is specified in 3GPP2 C.S0030-0 v2.0. Transfer methods are specified in RFC 3558.

Additional information: none

**Intended usage:**

COMMON. It is expected that many VoIP applications (as well as mobile applications) will use this type.

Person & email address to contact for further information:

Adam Li  
adamli@icsl.ucla.edu

Author/Change controller:

Adam Li  
adamli@icsl.ucla.edu  
IETF Audio/Video Transport Working Group

### 13. Mapping to SDP Parameters

Please note that this section applies to the RTP transfer only.

The information carried in the MIME media type specification has a specific mapping to fields in the Session Description Protocol (SDP) [6], which is commonly used to describe RTP sessions. When SDP is used to specify sessions employing the EVRC or EMV codec, the mapping is as follows:

- o The MIME type ("audio") goes in SDP "m=" as the media name.
- o The MIME subtype (payload format name) goes in SDP "a=rtpmap" as the encoding name.
- o The parameters "ptime" and "maxptime" go in the SDP "a=ptime" and "a=maxptime" attributes, respectively.
- o The parameter "maxinterleave" goes in the SDP "a=fmtp" attribute by copying it directly from the MIME media type string as "maxinterleave=value".

Some examples of SDP session descriptions for EVRC and SMV encodings follow below.

Example of usage of EVRC:

```
m=audio 49120 RTP/AVP 97
a=rtpmap:97 EVRC/8000
a=fmtp:97 maxinterleave=2
a=maxptime:80
```

Example of usage of SMV

```
m=audio 49122 RTP/AVP 99
a=rtpmap:99 SMV0/8000
a=fmtp:99
```

Note that the payload format (encoding) names are commonly shown in upper case. MIME subtypes are commonly shown in lower case. These names are case-insensitive in both places. Similarly, parameter names are case-insensitive both in MIME types and in the default mapping to the SDP a=fmtp attribute.

#### 14. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [4], and any appropriate profile (for example [5]). This implies that confidentiality of the media streams is achieved by encryption. Because the data compression used with this payload format is applied end-to-end, encryption may be performed after compression so there is no conflict between the two operations.

A potential denial-of-service threat exists for data encoding using compression techniques that have non-uniform receiver-end computational load. The attacker can inject pathological datagrams into the stream which are complex to decode and cause the receiver to become overloaded. However, the encodings covered in this document do not exhibit any significant non-uniformity.

As with any IP-based protocol, in some circumstances, a receiver may be overloaded simply by the receipt of too many packets, either desired or undesired. Network-layer authentication may be used to discard packets from undesired sources, but the processing cost of the authentication itself may be too high. In a multicast environment, pruning of specific sources may be implemented in future versions of IGMP [7] and in multicast routing protocols to allow a receiver to select which sources are allowed to reach it.

Interleaving may affect encryption. Depending on the used encryption scheme there may be restrictions on, for example, the time when keys can be changed. Specifically, the key change may need to occur at the boundary between interleave groups.

#### 15. Adding Support of Other Frame-Based Vocoders

As described above, the RTP packet format defined in this document is very flexible and designed to be usable by other frame-based vocoders.

Additional vocoders using this format **MUST** have properties as described in Section 3.3.

For an eligible vocoder to use the payload format mechanisms defined in this document, a new RTP payload format document needs to be published as a standards track RFC. That document can simply refer to this document and then specify the following parameters:

- o Define the unit used for RTP time stamp;
- o Define the meaning of the Mode Request bits;
- o Define corresponding codec data frame type values for ToC;
- o Define the conversion procedure for vocoders output data frame;
- o Define a magic number for storage format, and complete the corresponding MIME registration.

## 16. Acknowledgements

The following authors have made significant contributions to this document: Adam H. Li, John D. Villaseñor, Dong-Seek Park, Jeong-Hoon Park, Keith Miller, S. Craig Greer, David Leon, Nikolai Leung, Marcello Liroy, Kyle J. McKay, Magdalena L. Espelien, Randall Gellens, Tom Hiller, Peter J. McCann, Stinson S. Mathai, Michael D. Turner, Ajay Rajkumar, Dan Gal, Magnus Westerlund, Lars-Erik Jonsson, Greg Sherwood, and Thomas Zeng.

## 17. References

### 17.1 Normative

- [1] 3GPP2 C.S0014, "Enhanced Variable Rate Codec, Speech Service Option 3 for Wideband Spread Spectrum Digital Systems", January 1997.
- [2] 3GPP2 C.S0030-0 v2.0, "Selectable Mode Vocoder, Service Option for Wideband Spread Spectrum Communication Systems", May 2002.
- [3] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [4] Schulzrinne, H., Casner, S., Jacobson, V. and R. Frederick, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.
- [5] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", RFC 3551, July 2003.
- [6] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

## 17.2 Informative

- [7] Deering, S., "Host Extensions for IP Multicasting", STD 5, RFC 1112, August 1989.

## 18. Author's Address

Adam H. Li  
Image Communication Lab  
Electrical Engineering Department  
University of California  
Los Angeles, CA 90095  
USA

Phone: +1 310 825 5178  
EMail: [adamli@icsl.ucla.edu](mailto:adamli@icsl.ucla.edu)

## 19. Full Copyright Statement

Copyright (C) The Internet Society (2003). 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 implementation 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.

## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

