Network Working Group
Request for Comments: 5574
Category: Standards Track

G. Herlein
Independent
J. Valin
Xiph.Org Foundation
A. Heggestad
Creytiv.com
A. Moizard
Antisip

June 2009

RTP Payload Format for the Speex Codec

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

Abstract

Speex is an open-source voice codec suitable for use in VoIP (Voice over IP) type applications. This document describes the payload format for Speex-generated bit streams within an RTP packet. Also included here are the necessary details for the use of Speex with the Session Description Protocol (SDP).

Table of Contents

1. Introduction

Speex is based on the Code Excited Linear Prediction [CELP] encoding technique with support for either narrowband (nominal 8 kHz), wideband (nominal 16 kHz), or ultra-wideband (nominal 32 kHz). The main characteristics can be summarized as follows:

- Free software/open-source
- Integration of wideband and narrowband in the same bit-stream
- Wide range of bit-rates available
- Dynamic bit-rate switching and variable bit-rate (VBR)
- Voice Activity Detection (VAD, integrated with VBR)
- Variable complexity

The Speex codec supports a wide range of bit-rates from 2.15 kbit/s to 44 kbit/s. In some cases however, it may not be possible for an implementation to include support for all rates (e.g., because of bandwidth or RAM or CPU constraints). In those cases, to be compliant with this specification, implementations MUST support at least narrowband (8 kHz) encoding and decoding at 8 kbit/s bit-rate (narrowband mode 3). Support for narrowband at 15 kbit/s (narrowband mode 5) is RECOMMENDED and support for wideband at 27.8 kbit/s (wideband mode 8) is also RECOMMENDED. The sampling rate MUST be 8, 16 or 32 kHz. This specification defines only single channel audio (mono).

2. Terminology

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 RFC2119 [RFC2119] and indicate requirement levels for compliant RTP implementations.

3. RTP Usage for Speex

3.1. RTP Speex Header Fields

The RTP header is defined in the RTP specification [RFC3550]. This section defines how fields in the RTP header are used.

Payload Type (PT): 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, or signaled dynamically out-of-band (e.g., using SDP).

Marker (M) bit: The M bit is set to one on the first packet sent after a silence period, during which packets have not been transmitted contiguously.

Extension (X) bit: Defined by the RTP profile used.

Timestamp: A 32-bit word that corresponds to the sampling instant for the first frame in the RTP packet.

3.2. RTP Payload Format for Speex

The RTP payload for Speex has the format shown in Figure 1. No additional header fields specific to this payload format are required. For RTP-based transportation of Speex-encoded audio, the standard RTP header [RFC3550] is followed by one or more payload data blocks. An optional padding terminator may also be used.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         RTP Header                            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                 one or more frames of Speex ....              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        one or more frames of Speex ....       |    padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: RTP Payload for Speex

3.3. Speex Payload

For the purposes of packetizing the bit stream in RTP, it is only necessary to consider the sequence of bits as output by the Speex encoder [SPEEX], and present the same sequence to the decoder. The payload format described here maintains this sequence.

A typical Speex frame, encoded at the maximum bit-rate, is approximately 110 octets and the total number of Speex frames SHOULD be kept less than the path MTU to prevent fragmentation. Speex frames MUST NOT be fragmented across multiple RTP packets.

The Speex frames must be placed starting with the oldest frame and then continue consecutively in time.

An RTP packet MAY contain Speex frames of the same bit-rate or of varying bit-rates, since the bit-rate for a frame is conveyed in-band with the signal.

The encoding and decoding algorithm can change the bit-rate at any 20 msec frame boundary, with the bit-rate change notification provided in-band with the bit stream. Each frame contains both sampling rate (narrowband, wideband, or ultra-wideband) and "mode" (bit-rate) information in the bit stream. No out-of-band notification is required for the decoder to process changes in the bit-rate sent by the encoder.

The sampling rate MUST be either 8000 Hz, 16000 Hz, or 32000 Hz.

The RTP payload MUST be padded to provide an integer number of octets as the payload length. These padding bits are LSB-aligned (Least Significant Bit) in network octet order and consist of a 0 followed by all ones (until the end of the octet). This padding is only required for the last frame in the packet, and only to ensure the packet contents end on an octet boundary.

3.4. Example Speex Packet

In the example below, we have a single Speex frame with 5 bits of padding to ensure the packet size falls on an octet boundary.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      RTP Header                               |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                        ..speex data..                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        ..speex data..               |0 1 1 1 1|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.5. Multiple Speex Frames in an RTP Packet

Below is an example of two Speex frames contained within one RTP packet. The Speex frame length in this example falls on an octet boundary so there is no padding.

The Speex decoder [SPEEX] can detect the bit-rate from the payload and is responsible for detecting the 20 msec boundaries between each 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         RTP Header                            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                     ..speex frame 1..                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       ..speex frame 1..       |      ..speex frame 2..        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      ..speex frame 2..                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4. IANA Considerations

This document defines the Speex media type.

4.1. Media Type Registration

This section describes the media types and names associated with this payload format. The section registers the media types, as per RFC 4288 [RFC4288].

4.1.1. Registration of Media Type Audio/Speex

Media type name: audio

Media subtype name: speex

Required parameters:

rate: RTP timestamp clock rate, which is equal to the sampling rate in Hz. The sampling rate MUST be either 8000, 16000, or 32000.

Optional parameters:

ptime: SHOULD be a multiple of 20 msec [RFC4566]

maxptime: SHOULD be a multiple of 20 msec [RFC4566]

vbr: variable bit-rate - either ’on’, ’off’, or ’vad’ (defaults to ’off’). If ’on’, variable bit-rate is enabled. If ’off’, disabled. If set to ’vad’, then constant bit-rate is used, but silence will be encoded with special short frames to indicate a lack of voice for that period. This parameter is a preference to the encoder.

cng: comfort noise generation - either ’on’ or ’off’ (defaults to ’off’). If ’off’, then silence frames will be silent; if ’on’, then those frames will be filled with comfort noise. This parameter is a preference to the encoder.

mode: Comma-separated list of supported Speex decoding modes, in order of preference. The first is the most preferred and the remaining is in decreasing order of preference. The valid modes are different for narrowband and wideband, and are defined as follows:

* {1,2,3,4,5,6,7,8,any} for narrowband

* {0,1,2,3,4,5,6,7,8,9,10,any} for wideband and ultra-wideband

The ’mode’ parameters may contain multiple values. In this case, the remote party SHOULD configure its encoder using the first supported mode provided. When ’any’ is used, the offerer indicates that it supports all decoding modes. The ’mode’ parameter value MUST always be quoted. If the ’mode’ parameter is not provided, the mode value is considered to be equivalent to ’mode="3,any"’ in narrowband and ’mode="8,any"’ in wideband and ultra-wideband. Note that each Speex frame does contain the mode (or bit-rate) that should be used to decode it. Thus, an application MUST be able to decode any Speex frame unless the SDP clearly specifies that some modes are not supported (e.g., by not including ’mode="any"’). Indicating support for a given set of decoding modes also implies that the implementation support the same encoding modes.

Encoding considerations:

This media type is framed and binary, see Section 4.8 in [RFC4288].

Security considerations: See Section 6.

Interoperability considerations:

None.

Published specification:

RFC 5574.

Applications that use this media type:

Audio streaming and conferencing applications.

Additional information: none.

Person and e-mail address to contact for further information:

Alfred E. Heggestad: aeh@db.org

Intended usage: COMMON

Restrictions on usage:

This media type depends on RTP framing, and hence is only defined for transfer via RTP [RFC3550]. Transport within other framing protocols is not defined at this time.

Author: Alfred E. Heggestad

Change controller:

IETF Audio/Video Transport working group delegated from the IESG.

5. SDP Usage of Speex

The information carried in the media type specification has a specific mapping to fields in the Session Description Protocol (SDP) [RFC4566], which is commonly used to describe RTP sessions. When SDP is used to specify sessions employing the Speex codec, the mapping is as follows:

- The media type ("audio") goes in SDP "m=" as the media name.
- The media subtype ("speex") goes in SDP "a=rtpmap" as the encoding name. The required parameter "rate" also goes in "a=rtpmap" as the clock rate.
- The parameters "ptime" and "maxptime" go in the SDP "a=ptime" and "a=maxptime" attributes, respectively.
- Any remaining parameters go in the SDP "a=fmtp" attribute by copying them directly from the media type string as a semicolon-separated list of parameter=value pairs.

The tables below include the equivalence between modes and bit-rates for narrowband, wideband, and ultra-wideband. Also, the corresponding "Speex quality" setting (see SPEEX_SET_QUALITY in the Speex Codec Manual [SPEEX]) is included as an indication.

mode	Speex quality	bit-rate
1	0	2.15 kbit/s
2	2	5.95 kbit/s
3	3 or 4	8.00 kbit/s
4	5 or 6	11.0 kbit/s
5	7 or 8	15.0 kbit/s
6	9	18.2 kbit/s
7	10	24.6 kbit/s
8	1	3.95 kbit/s

Table 1: Mode vs. Bit-Rate for Narrowband

mode	Speex quality	wideband bit-rate	ultra wideband bit-rate
0	0	3.95 kbit/s	5.75 kbit/s
1	1	5.75 kbit/s	7.55 kbit/s
2	2	7.75 kbit/s	9.55 kbit/s
3	3	9.80 kbit/s	11.6 kbit/s
4	4	12.8 kbit/s	14.6 kbit/s
5	5	16.8 kbit/s	18.6 kbit/s
6	6	20.6 kbit/s	22.4 kbit/s
7	7	23.8 kbit/s	25.6 kbit/s
8	8	27.8 kbit/s	29.6 kbit/s
9	9	34.2 kbit/s	36.0 kbit/s
10	10	42.2 kbit/s	44.0 kbit/s

Table 2: Mode vs. Bit-Rate for Wideband and Ultra-Wideband

The Speex parameters indicate the decoding capabilities of the agent, and what the agent prefers to receive.

The Speex parameters in an SDP Offer/Answer exchange are completely orthogonal, and there is no relationship between the SDP Offer and the Answer.

Several Speex specific parameters can be given in a single a=fmtp line provided that they are separated by a semicolon:

```
a=fmtp:97 mode="1,any";vbr=on
```

Some example SDP session descriptions utilizing Speex encodings follow.

5.1. Example Supporting All Modes, Prefer Mode 4

The offerer indicates that it wishes to receive a Speex stream at 8000 Hz, and wishes to receive Speex ’mode 4’. It is important to understand that any other mode might still be sent by remote party: the device might have bandwidth limitation or might only be able to send ’mode="3"’. Thus, applications that support all decoding modes SHOULD include ’mode="any"’ as shown in the example below:

```
m=audio 8088 RTP/AVP 97
a=rtpmap:97 speex/8000
a=fmtp:97 mode="4,any"
```

5.2. Example Supporting Only Modes 3 and 5

The offerer indicates the mode he wishes to receive (Speex ’mode 3’). This offer indicates mode 3 and mode 5 are supported and that no other modes are supported. The remote party MUST NOT configure its encoder using another Speex mode.

```
m=audio 8088 RTP/AVP 97
a=rtmap:97 speex/8000
a=fmtp:97 mode="3,5"
```

5.3. Example with Variable Bit-Rate and Comfort Noise

The offerer indicates that it wishes to receive variable bit-rate frames with comfort noise:

```
m=audio 8088 RTP/AVP 97
a=rtmap:97 speex/8000
a=fmtp:97 vbr=on;cng=on
```

5.4. Example with Voice Activity Detection

The offerer indicates that it wishes to use silence suppression. In this case, the vbr=vad parameter will be used:

```
m=audio 8088 RTP/AVP 97
a=rtmap:97 speex/8000
a=fmtp:97 vbr=vad
```

5.5. Example with Multiple Sampling Rates

The offerer indicates that it wishes to receive Speex audio at 16000 Hz with mode 10 (42.2 kbit/s) or, alternatively, Speex audio at 8000 Hz with mode 7 (24.6 kbit/s). The offerer supports decoding all modes.

```
m=audio 8088 RTP/AVP 97 98
a=rtmap:97 speex/16000
a=fmtp:97 mode="10,any"
a=rtmap:98 speex/8000
a=fmtp:98 mode="7,any"
```

5.6. Example with Ptime and Multiple Speex Frames

The "ptime" SDP attribute is used to denote the packetization interval (i.e., how many milliseconds of audio is encoded in a single RTP packet). Since Speex uses 20 msec frames, ptime values of multiples of 20 denote multiple Speex frames per packet. It is recommended to use ptime values that are a multiple of 20.

If ptime contains a value that is not multiple of 20, the internal interpretation of it should be rounded up to the nearest multiple of 20 before the number of Speex frames is calculated. For example, if the "ptime" attribute is set to 30, the internal interpretation should be rounded up to 40 and then used to calculate two Speex frames per packet.

In the example below, the ptime value is set to 40, indicating that there are two frames in each packet.

```
m=audio 8088 RTP/AVP 97
a=rtpmap:97 speex/8000
a=ptime:40
```

Note that the ptime parameter applies to all payloads listed in the media line and is not used as part of an a=fmtp directive.

Care must be taken when setting the value of ptime so that the RTP packet size does not exceed the path MTU.

5.7. Example with Complete Offer/Answer Exchange

The offerer indicates that it wishes to receive Speex audio at 16000 Hz or, alternatively, Speex audio at 8000 Hz. The offerer does support ALL modes because no mode is specified.

```
m=audio 8088 RTP/AVP 97 98
a=rtmap:97 speex/16000
a=rtmap:98 speex/8000
```

The answerer indicates that it wishes to receive Speex audio at 8000 Hz, which is the only sampling rate it supports. The answerer does support ALL modes because no mode is specified.

```
m=audio 8088 RTP/AVP 99
a=rtmap:99 speex/8000
```

6. Implementation Guidelines

Implementations that support Speex are responsible for correctly decoding incoming Speex frames.

Each Speex frame does contain all needed information to decode itself. In particular, the ’mode’ and ’ptime’ values proposed in the SDP contents MUST NOT be used for decoding: those values are not needed to properly decode a RTP Speex stream.

7. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550], and any appropriate RTP profile. 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 encodings using compression techniques that have non-uniform receiver-end computational load. The attacker can inject pathological datagrams into the stream that are complex to decode and cause the receiver to be overloaded. However, this encoding does 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.

8. Acknowledgments

The authors would like to thank Equivalence Pty Ltd of Australia for their assistance in attempting to standardize the use of Speex in H.323 applications, and for implementing Speex in their open-source OpenH323 stack. The authors would also like to thank Brian C. Wiles <brian@streamcomm.com> of StreamComm for his assistance in developing the proposed standard for Speex use in H.323 applications.

The authors would also like to thank the following members of the Speex and AVT communities for their input: Ross Finlayson, Federico Montesino Pouzols, Henning Schulzrinne, Magnus Westerlund, Colin Perkins, and Ivo Emanuel Goncalves.

Thanks to former authors of this document; Simon Morlat, Roger Hardiman, and Phil Kerr.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

9.2. Informative References

[CELP] Schroeder, M. and B. Atal, "Code-excited linear prediction(CELP): High-quality speech at very low bit rates", Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vol 10, pp. 937-940, 1985, <http://www.ntis.gov/>.

[RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.

[SPEEX] Valin, J., "The Speex Codec Manual", <http://www.speex.org/docs/>.

Authors' Addresses

Greg Herlein
Independent
2034 Filbert Street
San Francisco, California 94123
United States

EMail: gherlein@herlein.com

Jean-Marc Valin
Xiph.Org Foundation

EMail: jean-marc.valin@usherbrooke.ca

Alfred E. Heggestad
Creytiv.com
Biskop J. Nilssonsgt. 20a
Oslo 0659
Norway

EMail: aeh@db.org

Aymeric Moizard
Antisip
5 Place Benoit Crepu
Lyon,   69005
France

EMail: jack@atosc.org