Internet Engineering Task Force (IETF)
Request for Comments: 6464
Category: Standards Track
ISSN: 2070-1721

J. Lennox, Ed.
Vidyo
E. Ivov
Jitsi
E. Marocco
Telecom Italia

December 2011

A Real-time Transport Protocol (RTP) Header Extension for Client-to-Mixer Audio Level Indication

Abstract

This document defines a mechanism by which packets of Real-time Transport Protocol (RTP) audio streams can indicate, in an RTP header extension, the audio level of the audio sample carried in the RTP packet. In large conferences, this can reduce the load on an audio mixer or other middlebox that wants to forward only a few of the loudest audio streams, without requiring it to decode and measure every stream that is received.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6464.

Table of Contents

1. Introduction

In a centralized Real-time Transport Protocol (RTP) [RFC3550] audio conference, an audio mixer or forwarder receives audio streams from many or all of the conference participants. It then selectively forwards some of them to other participants in the conference. In large conferences, it is possible that such a server might be receiving a large number of streams, of which only a few are intended to be forwarded to the other conference participants.

In such a scenario, in order to pick the audio streams to forward, a centralized server needs to decode, measure audio levels, and possibly perform voice activity detection on audio data from a large number of streams. The need for such processing limits the size or number of conferences such a server can support.

As an alternative, this document defines an RTP header extension [RFC5285] through which senders of audio packets can indicate the audio level of the packets' payload, reducing the processing load for a server.

The header extension in this document is different than, but complementary with, the one defined in [RFC6465], which defines a mechanism by which audio mixers can indicate to clients the levels of the contributing sources that made up the mixed audio.

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 RFC 2119 [RFC2119] and indicate requirement levels for compliant implementations.

3. Audio Levels

The audio level header extension carries the level of the audio in the RTP [RFC3550] payload of the packet with which it is associated. This information is carried in an RTP header extension element as defined by "A General Mechanism for RTP Header Extensions" [RFC5285].

The payload of the audio level header extension element can be encoded using either the one-byte or two-byte header defined in [RFC5285]. Figures 1 and 2 show sample audio level encodings with each of these header formats.

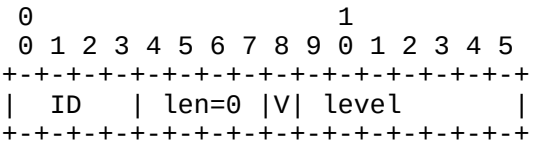

Figure 1: Sample Audio Level Encoding Using the One-Byte Header Format

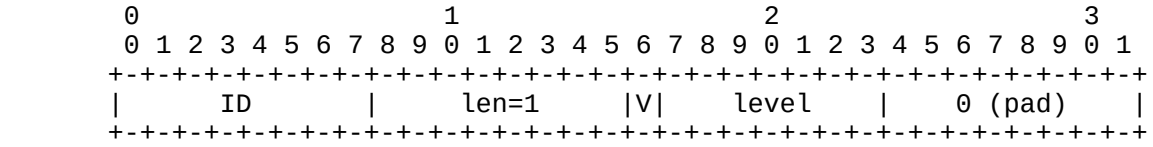

Figure 2: Sample Audio Level Encoding Using the Two-Byte Header Format

Note that, as indicated in [RFC5285], the length field in the one-byte header format takes the value 0 to indicate that 1 byte follows. In the two-byte header format, on the other hand, the length field takes the value of 1.

The magnitude of the audio level itself is packed into the seven least significant bits of the single byte of the header extension, shown in Figures 1 and 2. The least significant bit of the audio level magnitude is packed into the least significant bit of the byte. The most significant bit of the byte is used as a separate flag bit "V", defined below.

The audio level is expressed in -dBov, with values from 0 to 127 representing 0 to -127 dBov. dBov is the level, in decibels, relative to the overload point of the system, i.e., the highest-intensity signal encodable by the payload format. (Note: Representation relative to the overload point of a system is particularly useful for digital implementations, since one does not need to know the relative calibration of the analog circuitry.) For example, in the case of u-law (audio/pcmu) audio [ITU.G711], the 0 dBov reference would be a square wave with values +/- 8031. (This translates to 6.18 dBm0, relative to u-law's dBm0 definition in Table 6 of [ITU.G711].)

The audio level for digital silence -- for a muted audio source, for example -- MUST be represented as 127 (-127 dBov), regardless of the dynamic range of the encoded audio format.

The audio level header extension only carries the level of the audio in the RTP payload of the packet with which it is associated, with no long-term averaging or smoothing applied. For payload formats that contain extra error-correction bits or loss-concealment information, the level corresponds only to the data that would result from the payload's normal decoding process, not what it would produce under error or packet loss concealment. The level is measured as a root mean square of all the samples in the audio encoded by the packet.

To simplify implementation of the encoding procedures described here, Appendix A of [RFC6465] provides a sample Java implementation of an audio level calculator that helps obtain such values from raw linear Pulse Code Modulation (PCM) audio samples.

In addition, a flag bit (labeled "V") optionally indicates whether the encoder believes the audio packet contains voice activity. If the V bit is in use, the value 1 indicates that the encoder believes the audio packet contains voice activity, and the value 0 indicates that the encoder believes it does not. (The voice activity detection algorithm is unspecified and left implementation-specific.) If the V bit is not in use, its value is unspecified and MUST be ignored by receivers. The use of the V bit is signaled using the extension attribute "vad", discussed in Section 4.

When this header extension is used with RTP data sent using the RTP Payload for Redundant Audio Data [RFC2198], the header's data describes the contents of the primary encoding.

> Note: This audio level is defined in the same manner as is audio noise level in the RTP Payload Comfort Noise specification [RFC3389]. In [RFC3389], the overall magnitude of the noise level in comfort noise is encoded into the first byte of the payload, with spectral information about the noise in subsequent bytes. This specification's audio level parameter is defined so as to be identical to the comfort noise payload's noise-level byte.

4. Signaling (Setup) Information

The URI for declaring this header extension in an extmap attribute is "urn:ietf:params:rtp-hdrext:ssrc-audio-level".

It has a single extension attribute, named "vad". It takes the form "vad=on" or "vad=off". If the header extension element is signaled with "vad=on", the V bit described in Section 3 is in use, and MUST be set by senders. If the header extension element is signaled with "vad=off", the V bit is not in use, and its value MUST be ignored by receivers. If the vad extension attribute is not specified, the default is "vad=on".

An example attribute line in the Session Description Protocol (SDP) for a conference might hence be:

```
a=extmap:6 urn:ietf:params:rtp-hdrext:ssrc-audio-level vad=on
```

The vad extension attribute only controls the semantics of this header extension attribute, and does not make any statement about whether the sender is using any other voice activity detection features, such as discontinuous transmission, comfort noise, or silence suppression.

Using the mechanisms of [RFC5285], an endpoint MAY signal multiple instances of the header extension element, with different values of the vad attribute, so long as these instances use different values for the extension identifier. However, again following the rules of [RFC5285], the semantics chosen for a header extension element (including its vad setting) for a particular extension identifier value MUST NOT be changed within an RTP session.

5. Considerations on Use

Mixers and forwarders generally ought not base audio forwarding decisions directly on packet-by-packet audio level information, but rather ought to apply some analysis of the audio levels and trends. This general rule applies whether audio levels are provided by endpoints (as defined in this document), or are calculated at a server, as would be done in the absence of this information. This section discusses several issues that mixers and forwarders may wish to take into account. (Note that this section provides design guidance only, and is not normative.)

First of all, audio levels generally ought to be measured over longer intervals than that of a single audio packet. In order to avoid false-positives for short bursts of sound (such as a cough or a dropped microphone), it is often useful to require that a participant's audio level be maintained for some period of time before considering it to be "real"; i.e., some type of low-pass filter ought to be applied to the audio levels. Note, though, that such filtering must be balanced with the need to avoid clipping of the beginning of a speaker's speech.

Additionally, different participants may have their audio input set differently. It may be useful to apply some sort of automatic gain control to the audio levels. There are a number of possible approaches to achieving this, e.g., by measuring peak audio levels, by average audio levels during speech, or by measuring background audio levels (average audio levels during non-speech).

6. Security Considerations

A malicious endpoint could choose to set the values in this header extension falsely, so as to falsely claim that audio or voice is or is not present. It is not clear what could be gained by falsely claiming that audio is not present, but an endpoint falsely claiming that audio is present, or falsely exaggerating its reported levels, could perform a denial-of-service attack on an audio conference, so as to send silence to suppress other conference members' audio, or could dominate a conference by seizing its speaker-selection algorithm. Thus, if a device relies on audio level data from untrusted endpoints, it SHOULD periodically audit the level information transmitted, taking appropriate corrective action against endpoints that appear to be sending incorrect data. (However, as it is valid for an endpoint to choose to measure audio levels prior to encoding, some degree of discrepancy could be present. This would not indicate that an endpoint is malicious.)

In the Secure Real-time Transport Protocol (SRTP) [RFC3711], RTP header extensions are authenticated but not encrypted. When this header extension is used, audio levels are therefore visible on a packet-by-packet basis to an attacker passively observing the audio stream. As discussed in [SRTP-VBR-AUDIO], such an attacker might be able to infer information about the conversation, possibly with phoneme-level resolution. In scenarios where this is a concern, additional mechanisms MUST be used to protect the confidentiality of the header extension. This mechanism could be header extension encryption [SRTP-ENCR-HDR], or a lower-level security and authentication mechanism such as IPsec [RFC4301].

7. IANA Considerations

This document defines a new extension URI in the RTP Compact Header Extensions subregistry of the Real-Time Transport Protocol (RTP) Parameters registry, according to the following data:

```
Extension URI: urn:ietf:params:rtp-hdrext:ssrc-audio-level
Description:   Audio Level
Contact:       jonathan@vidyo.com
Reference:     RFC 6464
```

8. References

8.1. Normative References

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

[RFC2198] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J., Vega-Garcia, A., and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", RFC 2198, September 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.

[RFC5285] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", RFC 5285, July 2008.

8.2. Informative References

[ITU.G711] International Telecommunication Union, "Pulse Code Modulation (PCM) of Voice Frequencies", ITU-T Recommendation G.711, November 1988.

[RFC3389] Zopf, R., "Real-time Transport Protocol (RTP) Payload for Comfort Noise (CN)", RFC 3389, September 2002.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC6465] Ivov, E., Ed., Marocco, E., Ed., and J. Lennox, "A Real-time Transport Protocol (RTP) Header Extension for Mixer-to-Client Audio Level Indication", RFC 6465, December 2011.

[SRTP-ENCR-HDR] Lennox, J., "Encryption of Header Extensions in the Secure Real-Time Transport Protocol (SRTP)", Work in Progress, October 2011.

[SRTP-VBR-AUDIO] Perkins, C. and JM. Valin, "Guidelines for the use of Variable Bit Rate Audio with Secure RTP", Work in Progress, July 2011.

Authors' Addresses

Jonathan Lennox (editor)
Vidyo, Inc.
433 Hackensack Avenue
Seventh Floor
Hackensack, NJ 07601
US

EMail: jonathan@vidyo.com

Emil Ivov
Jitsi
Strasbourg 67000
France

EMail: emcho@jitsi.org

Enrico Marocco
Telecom Italia
Via G. Reiss Romoli, 274
Turin 10148
Italy

EMail: enrico.marocco@telecomitalia.it