RTP Header Extensions for Audio Level Indication

draft-ietf-avtext-client-to-mixer-audio-level

draft-ietf-avtext-mixer-to-client-audio-level

Emil Ivov

Enrico Marocco

Jonathan Lennox

IETF 80

Overview

- Indicate level of an RTP audio stream
  - Client-to-mixer: level of a source
  - Mixer-to-client: levels of contributing sources in a mix
- Now WG items of AVTEXT

Mostly Done

Since the drafts were first presented in AVT they have gone through a fair amount of discussion. Most design issues have been resolved:

- Header extensions build upon RFC 5285

  (RTP Header Extensions)

- Signal levels use dBov as they do in RFC 3389

  (Comfort Noise)

Latest Deltas

- Most of the latest changes were editorial.

- Only exception is the addition of sample level measurement code.

  draft-ietf-avtext-mixer-to-client-audio-level:
  - A.1. AudioLevelCalculator.java
  - A.2. AudioLevelRenderer.java

- Note: code will temporarily be moved to a draft of its own in order to have a more explicit view over what has WG consensus and what does not

WG Call 1

Does the WG consider the additional material (i.e. the code) form the basis of an improvement to the WG task on audio levels, and should therefore be retained in the document?

WG Call 2

Other changes that have taken place are the removal of the existing section 11 (Design choices) from draft-ietf-avtext-mixer-to-client-audio-level. Sometimes such material has been retained as an annex, and sometimes it is discarded.

- Call: Is the WG happy with this deletion?

Other Open Issues

1. Levels in non-audio (e.g., video) streams.

Is the WG happy with leaving these unspecified or should we explicitly declare them nonsense?

2. It has been suggested to reference ITU P.56 [ITU.P56.1993] for level measurement.

   This has been investigated but is probably best left to implementers.

Client-to-Mixer Open Issues

1. Would an additional field indicating the current noise floor level be useful?

    – It gives the level some context, allowing SNR calculation and perhaps simplifying stream equalization.

    – Wouldn't take additional space (assuming no other RTP header extension elements), since the client-to-mixer element is always padded.

    – This information is not always available, but could be optional (second byte).

    – Doesn't make much sense for mixer-to-client.

    – Could alternatively be defined as a separate header extension element.