RTP Timestamps for Layered Encodings

Jonathan Lennoxx
Thomas Schierl
Sam Ganesan

draft-lennoxx-avt-rtp-layered-encoding-timestamps-00.txt
Motivation

• RFC 3550 defines a mode where layered encodings are “striped” across multiple RTP sessions.
  – Associated streams use the same SSRC.
  – SSRC collisions are resolved on a base session.
• But it doesn’t say anything about how to use RTP timestamps across the multiple sessions.
Normative Statement

• “When a source is sent as a layered encoding transmitted on multiple sessions, such that the same SSRC identifier is used on each session, the same initial (random) RTP timestamp value MUST be used for every layer.”
  – Since each layer’s timestamps are derived from the same media clock, this implies that packets generated from (e.g.) the same audio sample or video frame have the same RTP timestamp.
  – This only discusses the SSRC alignment case.
Rationale

• A receiver doesn’t have to wait for an RTCP SR in order to associate streams.
  – The motivation for the SSRC association mechanism was to avoid having to wait for RTCP CNAME, for the same reason.

• This is what VIC does, and depends on for stream association.
  – VIC was the only implementation of RFC 3550-style striping of layered encodings, prior to the current generation of layered codecs.
Architectural Implications

• Doesn’t hurt SSRC collision detection.
  – Collision detection is done in the base RTP session anyway.

• RFC 3550-style encryption potentially becomes even more problematic.
  – Can’t use session-level SDP \( k \) lines, would have to use media-level ones, or you can get “two-time pad” problems.
  – You shouldn’t be using RFC 3550-style encryption anyway.
    • No authentication, weak encryption, no replay protection.
  – SRTP (with every currently-defined keying mechanism) is unaffected.
Next steps

• Is the AVT group interested in this work?
• Is there interest in taking it on as a WG item?
• Does anything further need to be added to it?