Duplicating RTP Streams

draft-begen-avtcore-rtp-duplication-00

IETF 82 – November 2011

Ali C. Begen and Colin Perkins
abegen@cisco.com, csp@cperkins.org
Motivation

- Packet loss is unavoidable due to congestion or network outages
  - It is especially more problematic in multicasting due to large fanout
  - One basic recovery (within a bounded delay and bandwidth) method is to send redundant stream(s)
- A redundant stream can carry FEC-like data or the duplicates of the original source packets
  - Here we are interested in methods where duplicates are used
  - We focus on dual streaming, but triple, quadruple, etc., streaming is also possible
    (although not generally desirable due to very high overhead...)
  - It is not clear how RTP duplication should be done, and how RTCP should be handled
- This document explains how RTP streams are to be duplicated without breaking RTP and RTCP rules
Temporal (Time-Shifted) Redundancy Use Case

- Packets are transmitted twice, each separated by Q time units where Q is the max outage duration that is intended to be repaired
- 5-tuple is the same for both main and redundant streams (If NAPT devices exist, using anything other than an identical 5-tuple can cause spatial redundancy)
- Thus, the streams are in the same RTP session and must use different SSRCs, chosen according to the usual SSRC selection rules
- The RTCP for the redundant stream is generated exactly as-if the redundant stream were a regular media stream
- Open issue: How should we correlate the duplicate streams?
Two streams are sent over diverse paths using
- Separate source interfaces and/or
- Separate destination addresses and/or ports

Thus, the streams are in separate RTP sessions, and choose SSRCs randomly

The RTCP for the redundant stream is generated exactly as-if the redundant stream were a regular media stream
Stream Merging at Receiver

- Media play out delay sized to be longer than the temporal offset of the duplicate stream
- Original and duplicate streams are time synchronised using the usual RTP mechanisms (RTP timestamp, mapped to NTP-format timestamp using RTCP SR/RFC 6051 header extension)
- Packets from the duplicate stream are decoded where lost packets exist in the original stream, and the media is inserted into the original stream; otherwise they’re discarded
- RTP RR (and RTCP XR, if desired) is sent for both original and duplicate stream, treating them entirely separately. Any post-repair loss reports are sent for the original stream after repair.
Stream Merging in Network

- Translator buffering delay sized to be longer than the temporal offset of the duplicate stream
- Original and duplicate streams are time synchronised using the usual RTP mechanisms (RTP timestamp, mapped to NTP-format timestamp using RTCP SR/RFC 6051 header extension)
- Translator suppresses duplicates; it repairs loss by generating missing packets based on the surrounding packet headers in the original stream, and media data from the duplicate stream
- The SSRC of the output stream is the same as the main stream, and will almost certainly be different to the SSRC of the redundant stream
- In case of SRTP, will need to re-authenticate the packets repair by the duplicate stream, and avoid re-encryption, with appropriate signaling of who authenticates the packets
- Also may need to translate RTCP packets flowing upstream
Open Issues

- How to correlate the streams?
  RTCP SDES SRCNAME? SDP SSRC and/or session grouping?

- Can we use standard RTP synchronization tools to align the streams, or should we mandate RTP timestamp alignment?
  The former is more complex, but a cleaner fit with the architecture

- If stream merging happens in network element, how do RTCP reports work?
  If the merging is done by an RTP translator, the merged stream has the same SSRC as the original, and so cannot be reported upon directly
  Should an in-network stream merging device be treated as an RTP mixer to ease RTCP reporting?
Next Steps

- Is there interest in pursuing this work in AVTCORE?