Duplicating RTP Streams

draft-begen-avtcore-rtp-duplication-01

IETF 83 – March 2012

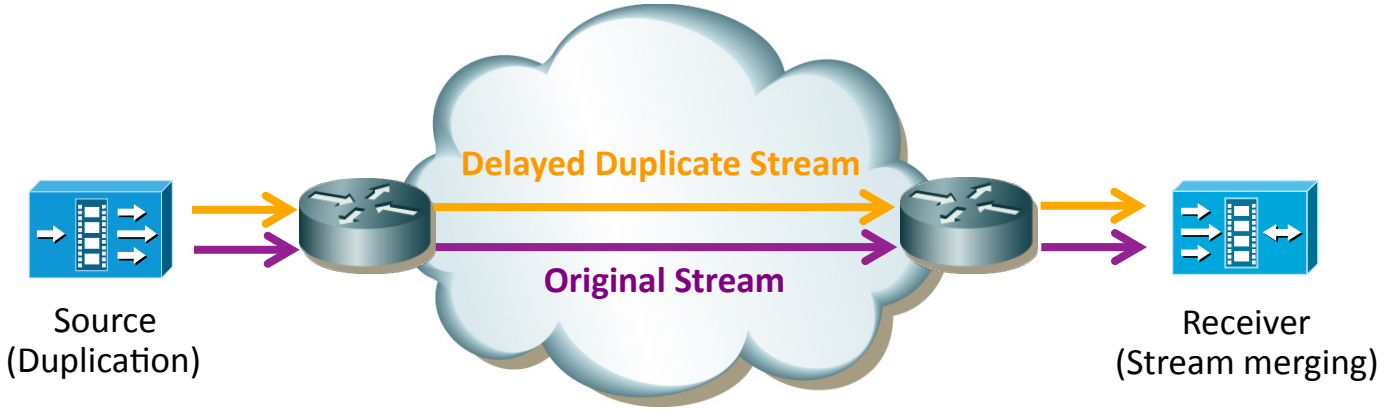
Ali C. Begen and Colin Perkins

abegen@cisco.com, csp@csperkins.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

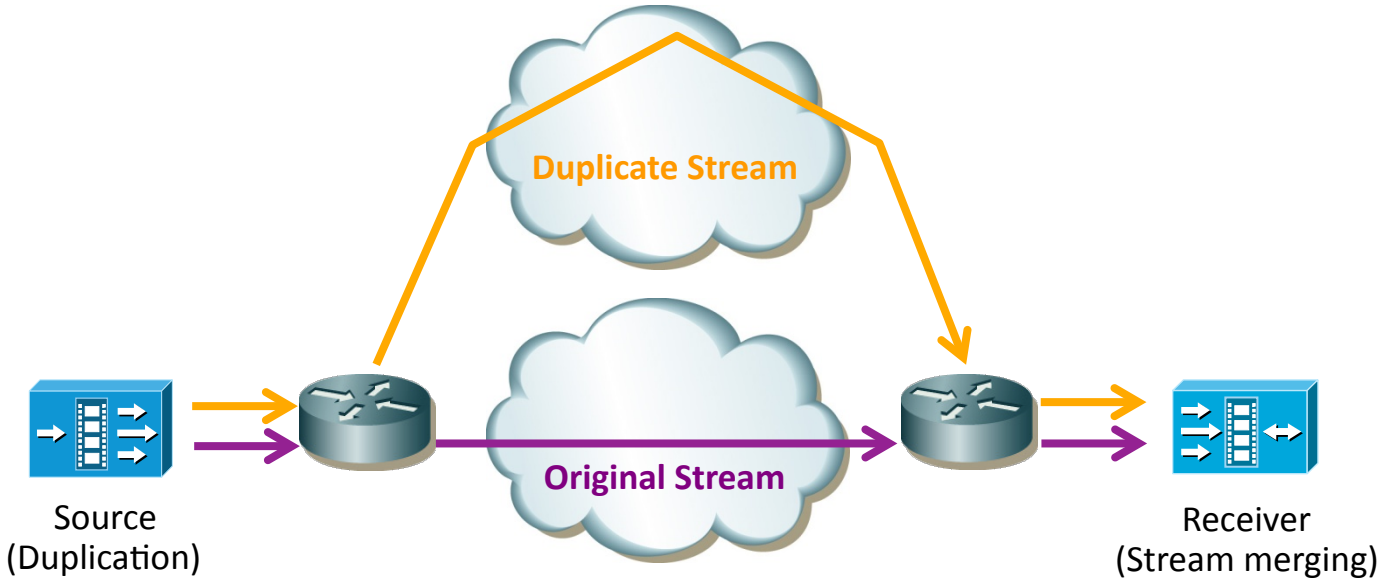


- Packets are transmitted twice, each separated by Q time units where Q is the max outage duration that is intended to be repaired
- 5-tuple is the same for both main and redundant streams (If NAPT devices exist, using anything other than an identical 5-tuple can cause spatial redundancy)
- Thus, the 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

SDP Example

```
v=0
o=ali 1122334455 1122334466 IN IP4 dup.example.com
s=Delayed Duplication
t=0 0
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=source-filter:incl IN IP4 233.252.0.1 198.51.100.1
a=rtpmap:100 MP2T/90000
a=ssrc:1000 cname:ch1@example.com
a=ssrc:1010 cname:ch1@example.com
a=ssrc-group:DUP 1000 1010
a=duplication-delay:100
a=mid:Group1
```

Spatial Redundancy Use Case



- 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

SDP Example

v=0
o=ali 1122334455 1122334466 IN IP4 dup.example.com
s=DUP Grouping Semantics
t=0 0
a=group:DUP S1a S1b
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=source-filter:incl IN IP4 233.252.0.1 198.51.100.1
a=rtpmap:100 MP2T/90000
a=ssrc:1000 cname:ch1@example.com
a=ssrc:1000 srcname:45:a8:f4:19:b4:c3
a=mid:S1a
m=video 30000 RTP/AVP 101
c=IN IP4 233.252.0.2/127
a=source-filter:incl IN IP4 233.252.0.2 198.51.100.1
a=rtpmap:101 MP2T/90000
a=ssrc:1010 cname:ch1@example.com
a=ssrc:1010 srcname:45:a8:f4:19:b4:c3
a=mid:S1b

Open Issues

- 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

- Could we get the draft adopted?