University of Glasgow | School of Computing Science

RTP Congestion Control: Circuit Breakers for Unicast Sessions

draft-perkins-avtcore-rtp-circuit-breakers-00

Colin Perkins – University of Glasgow
Varun Singh – Aalto University

Background and Goals

- RTP requires applications to be congestion aware, but lacks a standard congestion control algorithm
- Interest in developing and standardising congestion control algorithms for WebRTC
  - These algorithms are new, and will take time to develop and be validated
  - WebRTC, and other, applications need an immediate safety-net, to allow initial deployment before sophisticated congestion control is developed
- This draft defines an envelope within which these algorithms must work
  - Attempt to determine “circuit breaker” conditions for RTP sessions – limits that are not met in normal operation, but can be used to stop errant flows

RTP Background

- RTP data transfer protocol sends audio/visual data
  - Group communication protocol, supporting a variety of network topologies
  - Consider unicast flows only (end-to-end or end-to-RTP-layer-middlebox)
- RTP control protocol (RTCP) used for reporting and some limited session control
  - UDP-based backchannel – unreliable
  - Reception quality feedback reports sent every few *seconds*
  - Rapid feedback extensions exist, but basic mechanism must work without
- Associated signalling channel for high-level control
  - RTSP, SIP, XMPP, WebRTC, etc.
  - Not suitable for congestion control feedback

Congestion Signals for RTP/AVP Flows

- Potential congestion signals available from RTCP:
  - RTT estimate once per reporting interval
  - Jitter estimate once per reporting interval (limited use for video flows)
  - Fraction of packets lost during the reporting interval, plus cumulative number of packets lost over the entire RTP session
- Applicability as RTP circuit breakers:
  - RTT/jitter estimates too infrequent to be useful
  - Packet loss statistics too infrequent for rate adaptation, but useful for detecting overload situations – use as the basis for a limiting condition

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
header |V=2|P|    RC   |   PT=SR=200   |             length            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         SSRC of sender                        |
       +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
sender |              NTP timestamp, most significant word             |
info   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |             NTP timestamp, least significant word             |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         RTP timestamp                         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     sender's packet count                     |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                      sender's octet count                     |
       +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
report |                 SSRC_1 (SSRC of first source)                 |
block  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  1    | fraction lost |       cumulative number of packets lost       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |           extended highest sequence number received           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                      interarrival jitter                      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         last SR (LSR)                         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                   delay since last SR (DLSR)                  |
       +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
report |                 SSRC_2 (SSRC of second source)                |
block  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  2    :                               ...                             :
       +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
       |                  profile-specific extensions                  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

RTP Circuit Breaker Conditions

- Circuit breaker #1: Timeout
  - RTP data packets being sent, but corresponding RTCP RR packets report non-increasing extended highest sequence number received
  - Indication of significant connectivity problem if persistent for ≥ 2 reporting intervals → cease transmission

RTP Circuit Breaker Conditions

- Circuit Breaker #2: Congestion
  - RTP data sent, corresponding RR packets have increasing extended highest sequence number received, but non-zero packet loss fraction
  - Indication of network congestion – estimate equivalent TCP throughput:

$$T = \frac{s}{R\sqrt{\frac{2p}{3}} + (t_{RTO}(3\sqrt{\frac{3p}{8}})p(1+32p^2))}$$

R = round trip time, s = packet size
p = packet loss fraction

where $t_{RTO} \approx 4R$, and cease transmission if RTP sending rate ≥ $10T$ for 2 reporting intervals (based on Padhye *et al*, SIGCOMM 1998)

  - Issue #1: RTCP reports packet loss fraction, not loss event rate
    - Floyd *et al*, SIGCOMM 2000, show the difference is small for steady-state conditions and random loss; using loss fraction more conservative for bursty loss
  - Issue #2: RTT estimate is poor quality
  - Issue #3: measurement timescale is overly long; limits accuracy

Discussion

- Insufficient information for good congestion control using basic RTP/RTCP
  - Extensions, e.g., RTP/AVPF and RTCP XR, required for effective control
  - RTCWeb work will need to assume the presence of these

- Believe reasonable “circuit breaker” conditions can be derived using basic RTP/RTCP
  - Stretches applicability of TCP throughput equation – is this too far beyond breaking point?
  - Should we be using order-of-magnitude comparison to TCP throughput as a limiting condition?