Circuit Breakers for Multimedia Congestion Control

Varun Singh
Aalto University

Stephen McQuistin
University of Glasgow

Martin Ellis
University of Glasgow

Colin Perkins
University of Glasgow

***Abstract*—Real-time multimedia flows comprise a large, and increasing, fraction of the traffic on the Internet. An important subset of that traffic, primarily due to interactive applications, runs over UDP/IP, and requires applications to implement congestion control to ensure the stability of the network. The IETF is developing congestion control algorithms for such uses as part of the new WebRTC standards, but there is no standard algorithm that can be used at this time. We do not propose a congestion control algorithm. Rather, we propose a circuit breaker for RTP sessions that can detect when an application is causing excessive network congestion, and shut down the transmission. This can be used as an envelope within which congestion control algorithms can operate, providing a safety net to prevent congestion collapse. We present the RTP circuit breaker algorithm, and provide an initial performance evaluation to show that it performs as desired.**

I. Introduction

The Real-time Transport Protocol (RTP) [1] is widely used in interactive telephony, video conferencing, and telepresence applications. These applications are often run on best-effort UDP/IP networks, either running over the public Internet or in private enterprise networks. Due to their high data rate, if congestion control is not implemented such applications have the potential to cause severe congestion in the network, particularly in low capacity access links. This can disrupt both the multimedia quality of experience, and other applications.

There are two strands of work ongoing to address this problem in the Internet Engineering Task Force (IETF). Firstly, the RTP Media Congestion Avoidance Techniques (RMCAT) working group is developing standard congestion control algorithms for unicast RTP-based interactive multimedia applications; this is expected to be a multi-year process. Secondly, we are developing a circuit breaker algorithm that can work with unmodified RTP applications, and determine when those applications are causing excessive congestion and should cease their transmission [2], [3]. New congestion control algorithms defined in the RMCAT working group will need to work inside the envelope of this circuit breaker algorithm. Hence, the circuit breaker algorithm cannot be too aggressive in terminating media flows because it should allow sufficient time for the congestion control algorithm to monitor and respond to congestion signals. The development of the circuit breaker is on a tight schedule, to be ready for inclusion in the initial roll out of the WebRTC framework [4] in web browsers.

In this paper, we present the design of the RTP circuit breaker algorithm, and an initial evaluation of its performance on residential networks. We also study the impact of circuit breakers on point-to-point multimedia calls with varying amounts of cross-traffic on the bottleneck link.

Our first contribution is the definition of the RTP circuit breaker algorithm. This is necessary to allow the safe large-scale roll out of WebRTC applications. Secondly, we present the first evaluations of the RTP circuit breaker performance, considering streaming traffic to residential networks and interactive video traffic. Finally, we present an initial evaluation of the design trade-offs in the RTP circuit breaker, considering possible changes to the TCP throughput equation used, triggering interval, etc., to validate the design choices made.

We structure the remainder of this paper as follows. Section II describes the RTP circuit breaker algorithm. Then, Section III evaluates the performance of the RTP circuit breaker with streaming traffic to residential networks, and Section IV discusses performance with interactive video conferencing. Section V outlines possible changes to the circuit breaker algorithm and their impact, and section VI concludes the paper.

II. Circuit Breakers for Unicast RTP Sessions

RTP transports multimedia data over UDP, and is subject to the uncertainties of the best-effort IP networks due to packet loss, packet reordering, variable queuing delay due to route changes or buffer-bloated queues. Multimedia applications are tolerant to some packet loss, and focus on reducing end-to-end delay. If a small amount of loss occurs, the application either conceals it or uses one of the available error resilience mechanisms (retransmissions, forward error correction), but variation in delay can cause the media to pause and skip frames which is detrimental to the user experience [5]. Therefore, real-time communication applications must either implement congestion control or use a transport that implements congestion control. This motivates the need for UDP congestion control, and the RTP circuit breaker.

Designing effective congestion control algorithms, to adapt transmission of RTP-based media to the available network capacity while maintaining the user experience, is a difficult but important problem. Many congestion control and media adaptation algorithms have been proposed, but to date there is no consensus on the correct approach. TCP is only suitable for interactive multimedia for paths with low RTT ($<$ 100ms) [6], DCCP has problems with NAT traversal [7], and congestion control algorithms for UDP show stability problems [8]. We do not attempt to propose a new congestion control algorithm. Rather, we describe a minimal set of circuit breakers; conditions under which there is general agreement that an RTP flow is causing serious congestion, and should cease transmission. This is one strand of work in the IETF [2]; the other is defining congestion control algorithms in the RMCAT working group.

The RTP circuit breaker algorithm we propose relies on the basic feedback mechanisms defined in the RTP Control Protocol (RTCP) [1]. That is, it solely uses the information available in the RTCP Sender Report (SR) and Receiver Report (RR) packets to detect if the flow is overusing the capacity or causing congestion. RTCP extensions such as [9] are not used, to ensure wide applicability of the circuit breaker. Congestion indicators provided in RTCP SR/RR packets are: the network round trip time (RTT), estimated once per reporting interval; the average timing jitter measured by the receiver over the last reporting interval; the packet loss fraction measured over the last reporting interval; and the cumulative number of packets lost since the start of the session.

Variation in RTT is used as a congestion signal in delay-based congestion control algorithms, and RTT estimates can be used to configure connection timeouts. However, RTCP reporting intervals are large, typically several seconds, and a single RTT estimate per reporting interval is too infrequent to provide useful input to a circuit breaker. Similarly, a single highly aggregated jitter measurement per reporting interval also has insufficient granularity to be useful as a circuit breaker. On the other hand, loss is a strong indicator of congestion in networks where loss occurs predominantly due to queue overflows. Accordingly, we base the circuit breaker conditions on packet loss reports in RTCP SR/RR packets. There are three RTP circuit breakers that should cause a flow to be terminated:

- The **Media Timeout Circuit Breaker** triggers if an endpoint is sending media, but returning RTCP RR packets have a non-increasing highest sequence number received field for two consecutive RTCP reporting intervals.
- The **RTCP Timeout Circuit Breaker** triggers if an endpoint is sending media, but receives no corresponding RTCP RR packet for two consecutive reporting intervals.
- An endpoint that sends media will receive corresponding RTCP RR packets reporting the fraction of packets lost. This can be used to estimate the throughput a TCP connection would achieve over the same path [10]. The **Congestion Circuit Breaker** triggers if the RTP sending rate exceeds the estimated TCP throughput by a factor of ten or more for two consecutive RTCP intervals.[1]

The first two circuit breakers are intended to roughly correspond to TCP timeout conditions; the last is directly congestion related, and based on the principles used in the TFRC protocol [12], which similarly uses a rate-based model using the TCP throughput equation as the basis for its operation. Full details can be found in the IETF Internet-draft [2].

III. Streaming Performance on Residential Links

A key environment where the circuit breaker algorithm must perform well is on residential ADSL and cable modem networks. Indeed, a primary use case for the WebRTC is to allow a user on a standard home broadband connection to chat with others, and streaming is a likely follow-on scenario. In the following we evaluate performance of the RTP circuit breaker using measurements of streaming traffic to residential users.

A. Experimental Methodology

In previous work, we collected end-to-end packet traces and performance metrics for real-time traffic sent to users on residential networks. This data comprises synthetic RTP traffic flows sent from a well-connected server across the public Internet to residential users connected via a mix of ADSL and cable modem connections. There are 3833 RTP packet traces, at a range of bit rates (1–8.5Mbps), with duration varying between one and ten minutes, for a total of approximately 230×10^6 packets. The data was collected in 2009 and 2010, using clients in the UK and Finland. See [13] for details of the datasets, collection methodology, or to download the data.

The data set comprises RTP data packets captured at the receiver on the residential network. It does not capture upstream RTCP reports sent to the source. The RTP circuit breaker algorithm operates on RTCP reports received at the source. This contradiction can easily be resolved since RTCP reports can be generated based on the RTP data packets captured in the data set, and used to simulate the operation of the RTP circuit breaker algorithm. The algorithm to generate the contents of these RTCP packets is deterministic, based on the packet loss rate and timing in the packet traces [1], but the exact timing of the RTCP reports is randomised. Generated packets will be representative of the RTCP packets that would have been present in the original trace, if they had been captured.

We implemented a packet-level simulator that takes a trace of RTP packets, with timing information, and generates a matching trace of RTCP packets.[2] Simulated RTCP packets are sent according to the timing rules in [1]; each is a compound RTCP packet comprising a Reception Report (RR) and a Source Description (SDES) packet. The SDES packet contains dummy data of appropriate size; the RR packet contains reception quality statistics based on the RTP data packets observed in the input packet trace. We post-process the output of the simulator to determine if the congestion circuit breaker would have been triggered for each trace, and to generate other statistics. This models the worst case of a system that does not adapt to congestion; modelling systems that reduce their sending rate in response to congestion is for future study. None of the RTP traces in the dataset show a long enough gap to trigger the media timeout circuit breaker. For the purpose of this experiment we are looking at downstream congestion, so we assume that RTCP packets are delivered reliably and the RTCP timeout circuit breaker is never triggered. We therefore consider only the congestion circuit breaker.

[1] This is a poor quality estimate. Section V discusses the impact of using the more complete TCP throughput model from [11]; other limitations include the need to use of the loss fraction rather than the loss event rate when calculating TCP throughput, since loss event rate is not reported. The constraints of RTCP limit the accuracy of the throughput estimate we can derive; we must work with unmodified RTCP to be useful when interworking with legacy systems.

[2] The traces in *dataset-B* [13] include occasional low time to live (TTL) probes designed to solicit responses from on-path routers that intentionally do not reach the destination host. For the purposes of this work, we re-insert these packets as if they were received mid-way between packets with adjacent sequence numbers. This may potentially slightly under estimate the packet loss rate, but the low TTL probes are rare enough that the effect is not significant.

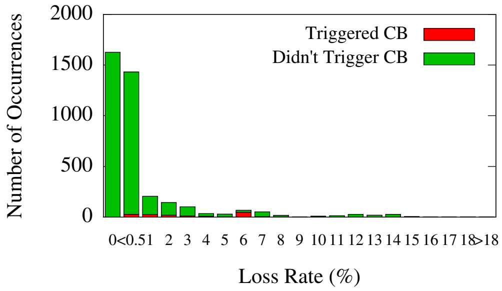


Figure 1: Distribution of traces by packet loss rate

B. Effects of Packet Loss Rates

We simulated RTP circuit breaker performance on 3833 generated RTCP traces corresponding to the measurements collected in *dataset-A* and *dataset-B* of [13]. Of these, 1626 traces have no packet loss, and hence cannot trigger the RTP circuit breaker. The remaining traces each include at least one lost packet. We categorise the traces according to the fraction of packets lost, and plot the distribution of traces by overall packet loss rate in Figure 1. Traces that have no packet loss are shown in the column labelled 0; the column labelled < 0.5 shows traces where the overall packet loss rate was less than 0.5% but at least one packet was lost; the other columns show the count of traces having overall packet loss rate that rounds to the specified integer percentage. Each column is split to show the number of traces that trigger the RTP circuit breaker, and the number that do not. As expected, no sessions with zero packet loss trigger the circuit breaker. However, traces with non-zero packet loss are seen to trigger the RTP circuit breaker in some cases, irrespective of the loss rate. A total of 164 traces out of 3833 trigger the circuit breaker. The circuit breaker tends to be triggered in traces with higher packet loss rate, but this is not uniform, and some low loss traces (including some with loss rate $< 0.5\%$) trigger the circuit breaker. Traces with loss rates exceeding about 10% are likely unusable for interactive multimedia; lower loss rates will impact quality, but may still give usable media depending on the codec, RTP payload format, error concealment algorithm, loss pattern, etc.

These results show that the overall packet loss rate in a trace is an imperfect predictor of whether the RTP circuit breaker will trigger. This is to be expected. The circuit breaker algorithm triggers based on the packet loss rate in a small number of consecutive RTCP reporting intervals. Reporting intervals are short, only a few seconds duration, while the traces are between one and 10 minutes in length. A burst of loss could impact enough RTCP reporting intervals to trigger the circuit breaker, yet be short enough to leave a low overall packet loss rate on a long trace. The pattern of packet loss events must be considered to determine if the circuit breaker triggers, not just the overall packet loss rate.

C. Effects of Packet Loss Patterns

We categorise the traces into three categories: loss free; those with non-bursty loss; and those that exhibit bursty loss using the definition from [14] (with $G_{\min} = 16$ as recommended). Figure 2 shows representative samples of bursty and non-bursty loss. The data comprises 1344 traces with bursty loss, 863 traces with non-bursty loss, and 1626 loss free.

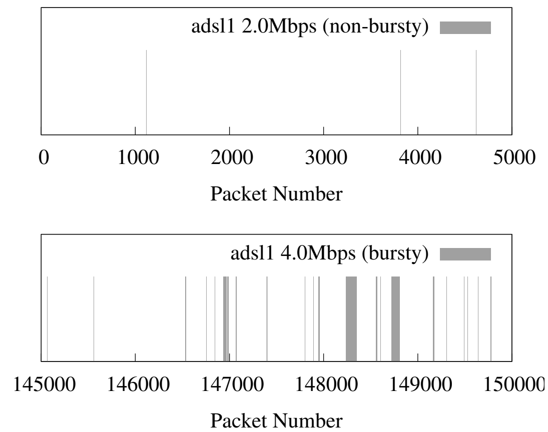


Figure 2: Sample bursty and non-bursty packet loss traces

Table I shows the fraction of sessions that triggered the RTP circuit breaker for each of the three categories of packet loss. As expected, the RTP circuit breaker did not trigger for sessions that are loss free. The RTP circuit breaker also did not trigger for any of the sessions that suffered non-bursty packet loss. However, we note that the RTP circuit breaker is triggered comparatively frequently in sessions that exhibit bursty packet loss.

The residential links over which our traces were captured are subject to three main loss processes: electrical noise on the last mile link; congestion at the edge of the network; and congestion in the core network. Congestion at the edge tends to be bursty, since there is a low degree of statistical multiplexing, and a likelihood that links are over-buffered and drop-tail. Congestion within the core tends to occur on links with a high degree of statistical multiplexing, so tends to affect many flows, but only causes loss of a few packets from each, so it is less visibly bursty. Loss due to noise also tends to be visible only as isolated loss events, due to the use of error correction and interleaving on the last mile link (especially on ADSL links). We therefore suggest that sessions that trigger the RTP circuit breaker are likely those that are congesting the last mile link, causing bursts of packet loss.

D. Effects of Sending Data Rate

The percentage of sessions triggering the RTP circuit breaker is broken down by link type and sending data rate in Table II. We observe that the RTP circuit breaker rarely triggers at low data rates, only occurring at 1Mbps sending rate on link adsl7, and at 2Mbps only on links adsl7, cable1 and fincable0 (the cable1 link has a 2Mbps capacity, so the loss on that link is likely due to congestion since the sending rate matches the link capacity). Overall, only 2.4% of sessions with sending rate of 1Mbps or 2Mbps trigger the RTP circuit breaker.

Loss Pattern	Triggered	Did not trigger
Loss free	0.0%	100.0%
Non-bursty loss	0.0%	100.0%
Bursty loss	12.2%	87.8%

Table I: Sessions triggering circuit breaker by loss pattern

	Sending Data Rate (Mbps)					
Link	1.0	2.0	4.0	5.0	6.0	8.5
adsl1	0%	0%	9%	-	38%	-
adsl2	0%	0%	-	-	-	-
adsl3	0%	0%	-	-	-	-
adsl4	0%	0%	0%	6%	0%	-
adsl5	0%	0%	0%	7%	27%	-
adsl6	0%	0%	19%	0%	52%	-
adsl7	2%	9%	-	29%	-	-
cable1	0%	20%	-	-	-	-
cable2	0%	0%	0%	4%	8%	17%
cable3	0%	0%	-	18%	-	-
cable4	0%	0%	-	2%	-	-
cable5	0%	0%	-	2%	-	-
finadsl0	0%	0%	-	2%	-	-
fincable0	0%	4%	-	100%	-	-

Table II: Fraction of sessions at each sending rate triggering the RTP circuit breaker (link names match [13]; finadsl0 and fincable0 are captured in Finland, others are UK ISPs).

The RTP circuit breaker triggers more frequently as the sending data rate increases. The worst performance is link fincable0 where 100% of flows trigger the RTP circuit breaker when sending at 5Mbps, however the link capacity here is 5Mbps, so this is not unexpected. The other links are running below capacity at the maximum sending rate, but still see the RTP circuit breaker firing on occasion. We assume, but have no way of knowing, that this is due to our test traffic sharing the link with other traffic, causing transient congestion.

It is clear that there is a strong rate-dependent component in the fraction of sessions triggering the circuit breaker: the higher the sending data rate, the more likely it is that the circuit breaker is triggered. This supports the hypothesis that the non-congestion controlled RTP flows used as test traffic are overloading edge links, causing packet loss at high rates.

E. Summary

Our measurements of the RTP circuit breaker performance with streaming traffic to residential links suggest that the algorithm performs as desired. The RTP circuit breaker tends to trigger when sending at rates close to the link capacity, and when bursty loss is present.

IV. Performance with Interactive Video

In this section, we discuss the performance of the circuit breaker for interactive multimedia sessions such as a video call between two participants. This is expected to be a common scenario for WebRTC. Our video application is built on the open-source libraries gstreamer (http://gstreamer.freedesktop.org/) and x264 (http://www.videolan.org/developers/x264.html) and uses the Akiyo video sequence in VGA frame size at 15 frames-per-second with a 1Mbps target media rate. We multiplex RTP and RTCP packets on the same UDP port [15].

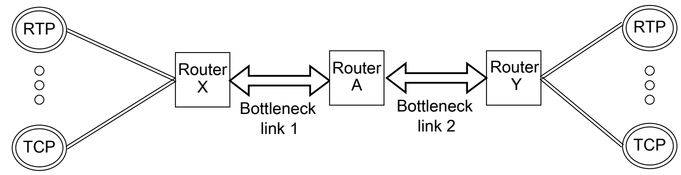


Figure 3: Circuit breaker evaluation for interactive video.

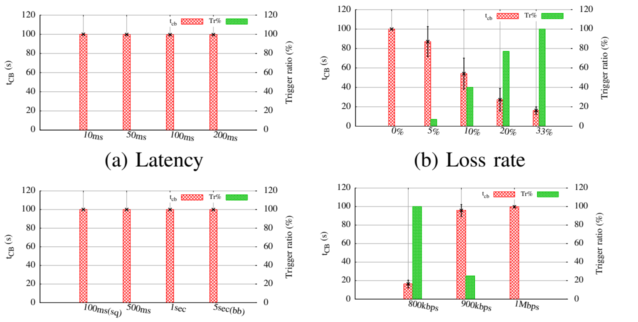
A. Experimental Methodology

Figure 3 shows the evaluation set-up for interactive video. The bottleneck links carry one or more bidirectional RTP media flows with varying amounts of TCP cross traffic. The traffic flows are connected to the edge routers (Router X and Y) over a high capacity and low-delay link. We introduce impairments to the bottleneck links and analyse the performance of the circuit breaker. We use dummynet [16] to emulate the variation in link capacity, latency, intermediate router queue length, and use the Gilbert-Elliott Model to model packet loss.

We evaluate the performance of the circuit breakers in the following scenarios: single RTP flow on a bottleneck link; multiple RTP flows on a bottleneck link; and single RTP flow competing with multiple TCP flows. Performance is tested by introducing impairments on the bottleneck links, then measuring the fraction of sessions that triggered the circuit breaker ($Tr\%$), and the time it takes to trigger the circuit breaker (t_{CB}) after the impairment is introduced. The former is a measure of the effectiveness of the RTP circuit breaker at detecting particular impairments, the latter a measure of the responsiveness of the RTP circuit breaker. The results are averaged and we calculate the 95% confidence interval. In the scenarios when the circuit breaker is triggered, we calculate the average t_{CB} using the call duration for only those calls that triggered the circuit breaker. A test is considered to not trigger the circuit breaker if the session is not halted within 100s of introducing the impairment. In 100s each endpoint sends about 20 RTCP reports (the average RTCP reporting interval is 5s for unicast media sessions), and this allows sufficient amount of time and measurements for the circuit breaker to trigger. To derive statistical significance, each scenario is run multiple times; in total we ran 3000 tests.

B. Effects of Varying Bottleneck Link Characteristics

In this scenario, a single bi-directional RTP stream flows through the bottleneck links. In each test run, we vary only one network characteristic and observe if the change triggers the RTP circuit breaker. We chose a subset of the cases to intentionally trigger the RTP circuit breakers, for example adjusting the bottleneck rate to be lower than the media rate, or simulating a link with high packet loss rate. This is done to test the responsiveness of the RTP circuit breaker. Specifically,



(a) Latency (b) Loss rate (c) Queue length (d) Throughput

Figure 4: A single media flow on a bottleneck link. We vary the bottleneck link characteristics: a) latency, b) loss rate, c) queue length and, d) throughput. Each scenario is run 50 times and the error bars represent the 95% confidence level.

during an ongoing call, at time 20s, we change one of the following: bottleneck link latency, link loss rate, router queue length, or bottleneck throughput. We then observe the RTP streams for the next 100s to determine if, and how quickly, the RTP circuit breaker is triggered.

We begin by changing the bottleneck link latency during a call. At time 20s the one-way delay is changed to one of the following: 10ms, 50ms, 100ms, 200ms. The router queue length is set to default 50 packets, and the bottleneck capacity (1.1Mbps) is sufficient to carry the single media flow. We observe that these changes do not trigger the RTP circuit breaker, and all the sessions run for the full 100s after the impairment is applied. This is shown in Figure 4(a).

We next introduce packet loss at routers X and Y in the testbed (Figure 3). We observe that increasing the packet loss rate increases the likelihood of triggering the RTP circuit breaker, showing that the algorithm correctly reacts to increasing packet loss. Further, sessions are terminated by the RTP circuit breaker more quickly at higher loss rate, showing that the algorithm is responsive. Figure 4(b) shows that at 0% loss rate the RTP circuit breakers are not triggered, but as the loss rate rises to 33% loss rate the video call is terminated every time. The main reason for observing this kind of behaviour is the losses mainly affect the single media flow and the RTCP is multiplexed with the media data, therefore it affects the feedback too. However, in later experiments when we introduce cross traffic (Section IV-C) or increase the RTCP feedback rate (Section V-C), fewer calls are terminated.

We vary the queue size at an intermediate router and observe the impact on the circuit breaker. We describe the queue sizes as a function of time, i.e., it is the depth of the queue or the amount of time the packet will remain in the queue before it is discarded. However, in practice the queue size is measured in number of packets. We convert the queue depth (measured in time) to queue length (number of packets) using:

$$\mathrm{QueueSize_{packets}} = \frac{\mathrm{QueueSize_{sec}} \times \mathrm{Throughput_{bps}}}{\mathrm{MTU} \times 8}$$

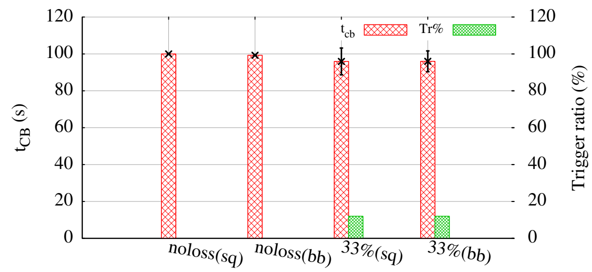


Figure 5: Multiple RTP flows (10 in each direction) compete on a bottleneck link with 0% loss and 33% loss, and with router queue lengths 100ms (short queue, *sq*) and 5s (buffer-bloated queue, *bb*). Each scenario is run 100 times.

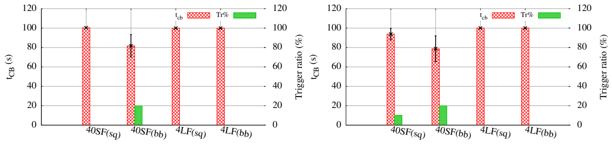
In our experiments the MTU is 1500 bytes. For example, a router with a throughput of 1Mbps and a 1s queue depth would be capable of handling 83 packets (queue length). We experiment with queue depths of 100ms, 500ms, 1s and 5s. The 100ms queue depth represents a short queue, while the 5s queue depth represents a buffer-bloated queue. Since there is only a single flow on the bottleneck link, the variation in the queue lengths increases the end-to-end latency but this does not affect the circuit breaker because the packet burst sizes are relatively small (See Figure 4(c) for details).

Finally, we consider changes in bottleneck bandwidth. In this scenario, we want to observe how quickly the circuit breaker triggers when congestion occurs. The easiest way to create congestion is to limit the capacity on the bottleneck link. The target media bit rate is 1Mbps and the bottleneck throughput is reduced to 800kbps, 900kbps or 1Mbps after 20s. The router queue length is set to 50 packets. Figure 4(d) shows that no circuit breakers are triggered when there is sufficient capacity, and only 20% of the calls are terminated within 100s when there is 90% available capacity. However, all flows are terminated within 20s of reducing the rate to 80% of the bottleneck capacity. The main reason for triggering the circuit breaker is queuing delay, by reducing the bottleneck link capacity the packets are queued longer in the router, however, the queues have finite capacity and drop packets when the queue overflows. Therefore, by terminating the calls, the circuit breaker avoids causing even more queue build up.

C. Effects of RTP Cross Traffic

In this scenario, multiple RTP flows traverse the bottleneck link. The capacity of the bottleneck link is the aggregate sum of the media rates. That is, the bottleneck link has exactly the capacity needed to carry all the flows. We use two different video sequences ("akiyo" and "foreman") and send five media streams, one each at 200kbps, 400kbps, 600kbps, 800kbps, and 1000kbps. The bottleneck is set to 6Mbps, matching the total bit rate sent.

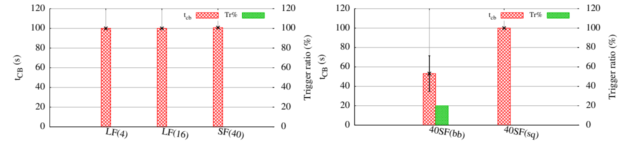
We perform two experiments using this scenario. Firstly, we configure different queue depths, namely short (100ms) and buffer-bloat (5s) queues, and observe the impact on the circuit breaker on each media flow. Figure 5 shows that no circuit



(a) TCP and RTP flows start together

(b) TCP starts 20s before the RTP

Figure 6: Single RTP flow competing with multiple TCP flows. TCP flows can be long (LF) or short (SF); routers can be short queues or buffer-bloated. Each scenario is run 100 times.



(a) TCP and RTP flows start together

(b) TCP flows starts 20s before the 5 RTP flows

Figure 7: A video call between Helsinki (cable, 10/10) and AWS in Ireland with cross-traffic. Each test is run 25 times.

breaker was triggered despite using different media streams with various bit rates. This shows that occasional transient overload does not cause spurious RTP circuit breaker triggers.

Second, in addition to the different queue depths, we introduce packet losses at the edge router. Figure 5 shows that in about 10% of the cases the RTP circuit breaker was triggered. However, this is a considerably lower rate of triggering than in the single flow scenario (see Section IV-B), since the loss rate is applied on the aggregate, not on a per-flow basis.

D. Effects of TCP Cross Traffic

In this scenario, a single RTP flow competes with TCP cross-traffic on the bottleneck with 3Mbps capacity. The RTP flow occupies about one third of the capacity, leaving 2Mbps to be shared with the TCP flows. The routers are configured to have either short or buffer-bloated queues. We use two forms of TCP flows, either short TCP flows that are modelled to resemble web-traffic, or long TCP flows that represent large file downloads. The TCP cross traffic is emulated by an iperf server running at Router A and identical iperf clients running at the endpoints. The long TCP flow downloads an unbounded amount of data (representing file downloads) and runs in parallel to the media flows. The short TCP flows are modelled as a sequence of web page downloads interleaved with idle periods (on-off traffic). The sizes of the web pages are obtained from a uniform distribution between 100kB and 1.5MB. Lengths of the idle periods are drawn from an exponential distribution with the mean value of 10 seconds. We use 4 long TCP flows or 40 short TCP flows to compete with the RTP flow on the bottleneck.

We test two scenarios. In the first scenario, the long TCP flows start at the same time as the RTP flow, but some of the short TCP flows start in the OFF state. Since the RTP flow starts at the full rate (1Mbps), and the TCP flows begin in the slow start state, the long flow TCP can stabilize and use the remaining bandwidth, thus not triggering the circuit breaker. When competing with multiple short TCP flows, the queue lengths play an important role because the TCP flow reduces its sending rate when it observes a packet loss. When the routers are configured with a short queue, the RTP circuit breaker is not triggered because the router drops incoming packets quickly in response to TCP dynamics (this corresponds to the non-bursty loss cases in Section III). In the case of the buffer-bloated queues, the TCP flows are less responsive due to the long queue, and overshoot the available capacity causing bursts of packet loss. This triggers the circuit breaker in 20% of the cases. Figure 6(a) shows comparative results between short and buffer-bloated queues for long and short TCP flows.

In the second scenario, the TCP flows start 20 seconds before the RTP flow. With the long TCP flows, by the time the RTP flow starts, the TCP flows are already in congestion avoidance state. The arrival of the 1Mbps RTP flow causes queues to build up, inducing packet loss and forcing the long TCP flows decrease their sending rate. This makes room for the RTP flow, and the RTP circuit breaker is not triggered. With short TCP flows, however, the on-off nature of the flows ensures that some are in slow start when the RTP flow starts. This can cause large rate changes, potentially filling the router queues and causing a burst of packet loss. When the size of the queue is short, the router drops the packets often enough that the TCP flows are not too aggressive, however, for the buffer-bloated queue, the TCP flows compete aggressively to fill up the queue which results in long delays and eventually packet loss. The circuit breakers are triggered in both cases; Figure 6(b) shows that the circuit breaker triggered in 20% of cases with buffer-bloat and 10% of the cases with short queues. Again, over buffering causes loss bursts that affect multiple RTCP reporting intervals and trigger the circuit breaker.

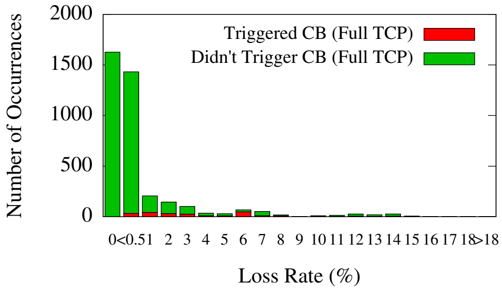
E. Performance on the Public Internet

A video call is initiated between a host on a residential network in Helsinki (cable, 10Mbps down/4Mbps up) and a server running on Amazon Web Service (AWS) in Ireland. We also introduce TCP cross-traffic: a) 4 TCP long flows (LF), b) 16 TCP long flows (LF), and c) 40 TCP short flows (SF) between the two endpoints. In none of the cases is the RTP circuit breaker triggered, most likely because the bottleneck can handle the aggregate capacity of the TCP and RTP flows.

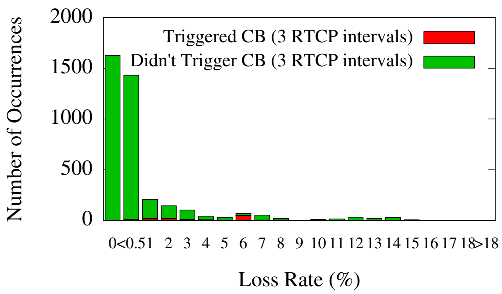
In the second scenario, we introduce 5 RTP flows 20s after starting the 40 short TCP flows. To both endpoints, we add dummynet to emulate a buffer-bloat queue. The circuit breaker is triggered in 20% of the cases and the session do not last more than 50s. This is chiefly because the multiple short TCP flows flood the router's queue and causes excessive delay that triggers the circuit breaker (see Figure 7 for details).

V. Discussion

In the following we discusses possible changes to the circuit breaker algorithm, and explore their impact on performance



(a) Using full TCP Model



(b) Trigger after three RTCP reporting intervals

Figure 8: Distribution of traces by loss rate using modified RTP circuit breaker

Loss Pattern	Triggered	Did not trigger
Loss free	0.0%	100.0%
Non-bursty loss	0.2%	99.8%
Bursty loss	19.3%	80.7%

Table III: Sessions triggering circuit breaker by loss pattern (full TCP model)

	Sending Data Rate (Mbps)					
Link	1.0	2.0	4.0	5.0	6.0	8.5
adsl1	0%	1%	14%	-	42%	-
adsl2	0%	0%	-	-	-	-
adsl3	0%	0%	-	-	-	-
adsl4	3%	5%	0%	26%	0%	-
adsl5	0%	4%	7%	20%	31%	-
adsl6	0%	1%	26%	0%	56%	-
adsl7	10%	9%	-	29%	-	-
cable1	0%	33%	-	-	-	-
cable2	0%	0%	0%	6%	8%	21%
cable3	18%	13%	-	29%	-	-
cable4	2%	0%	-	2%	-	-
cable5	2%	0%	-	4%	-	-
finadsl0	0%	0%	-	6%	-	-
fincable0	16%	16%	-	100%	-	-

Table IV: Fraction of sessions at each sending rate triggering the RTP circuit breaker (full TCP model).

using the residential streaming traces studied in Section III.

A. Choice of TCP Throughput Equation

We repeat the evaluation from Section III using the more complete TCP throughput model from [11], rather than the simple TCP model [10] used in the original evaluation. The distribution of traces that trigger this modified circuit breaker by packet loss rate is shown in Figure 8a. We also show the fraction of sessions triggering the modified circuit breaker by loss pattern (Table III) and sending rate (Table IV).

Comparing Figure 8a with Figure 1 the distributions look similar. Comparing Table IV with Table II gives a clearer picture, however, showing that considerably more low-rate sessions trigger the RTP circuit breaker when using the more complete TCP model. Table III gives more insight: the number of bursty sessions triggering the circuit breaker almost doubles with this TCP model. The more complete TCP model is more sensitive to bursty packet loss.

B. Choice of Triggering Interval

We also repeat the evaluation using a modified version of the RTP circuit breaker that takes three RTCP reporting intervals to trigger, rather than the usual two intervals. The distribution of traces triggering this circuit breaker is shown in Figure 8b. We also show the fraction of session triggering the circuit breaker by loss pattern (Table V) and sending rate (Table VI).

As with the standard circuit breaker, no sessions with non-bursty loss trigger this modified circuit breaker. The number of sessions with bursty loss is slightly down, at 10.1% rather than 12.2% of the sessions. Comparing Table V with Table I and Figure 8b with Figure 1 we see that fewer sessions trigger the circuit breaker, especially at high rates. This shows that increasing the number of reporting intervals needed to trigger the RTP circuit breaker makes it less responsive, as expected. It is not clear, however, that this is an improvement. The circuit breaker should trigger with bursty loss and at high rates.

C. Choice of RTCP Interval

The RTP Profile for RTCP-based Feedback [9] allows endpoints to send RTCP reports early in some cases, provided the average reporting interval is respected. This can be useful to indicate the early onset of congestion, or the receiver may want to send rapid feedback about significant congestion events, allowing the congestion control algorithm to be more responsive. Early reports may be sent in a compound RTCP packet, or using the Reduced-Size RTCP extension [17].

Sending reduced size RTCP saves bandwidth, but since such packet do not contain an SR/RR packet, they do not count towards resetting the RTCP Timeout in the circuit breaker. Sending early feedback as a compound packet will use more bandwidth, but allows the RTCP timeout circuit breaker to function. We compare the performance of the circuit breaker for an RTP flow with a standard RTCP interval (5 ± 2.5s) and another flow with a shorter RTCP interval (2.5 ± 1.25s) using compound RTCP packets. Figure 9 shows the percentage of sessions that triggered the circuit breaker for the two cases, and the average time it took to trigger the circuit breaker. The number of times the circuit breaker is triggered increases with the loss rate, as expected. However, by using the shorter RTCP interval the sessions last relatively longer, and fewer sessions are terminated compared to the standard RTCP interval. Loss

Loss Pattern	Triggered	Did not trigger
Loss free	0.0%	100.0%
Non-bursty loss	0.0%	100.0%
Bursty loss	10.1%	89.9%

Table V: Sessions triggering circuit breaker by loss pattern (3 RTCP intervals)

	Sending Data Rate (Mbps)					
Link	1.0	2.0	4.0	5.0	6.0	8.5
adsl1	0%	0%	7%	-	24%	-
adsl2	0%	0%	-	-	-	-
adsl3	0%	0%	-	-	-	-
adsl4	0%	0%	0%	6%	0%	-
adsl5	0%	0%	0%	5%	19%	-
adsl6	0%	0%	15%	0%	48%	-
adsl7	2%	9%	-	26%	-	-
cable1	0%	13%	-	-	-	-
cable2	0%	0%	0%	4%	0%	17%
cable3	0%	0%	-	12%	-	-
cable4	0%	0%	-	2%	-	-
cable5	0%	0%	-	0%	-	-
finadsl0	0%	0%	-	2%	-	-
fincable0	0%	2%	-	100%	-	-

Table VI: Fraction of sessions at each sending rate triggering the RTP circuit breaker (3 RTCP intervals).

events that would have been reported in consecutive RTCP reports, triggering the RTP circuit breaker, are now split across more reporting intervals, with loss hitting non-consecutive reports, so the circuit breaker does not trigger.

VI. Conclusions

We proposed a set of circuit breaker conditions that can be applied to RTP media flows. Such flows typically do not implement congestion control at this time, and are likely to cause congestion if deployed on the Internet. We carried out a series of experiments based on real-world traces and on a test-bed emulating real-world conditions. Our preliminary results show that the proposed RTP circuit breaker performs well, triggering in cases of bursty loss and in sessions that are congesting the links, and does not trigger in low-loss and non-bursty scenarios.

The RTP circuit breaker algorithms aims to stop flows that significantly exceed an approximation of the TCP friendly throughput for the observed loss rate. It protects the network, but does not explicitly consider user experience. Our expectation, however, is that the threshold is such that user experience will be poor if the circuit breaker triggers. Future work will evaluate media quality at the time the circuit breaker triggers, with a range of media types and codecs, to confirm this. We will also validate the factor-of-ten multiple of the TCP throughout as the circuit breaker threshold, and use a synthetic loss model [18] to explore performance in higher loss regimes.

The data set we use is limited. Call flows between residential users, and uploads from residential links have not yet been

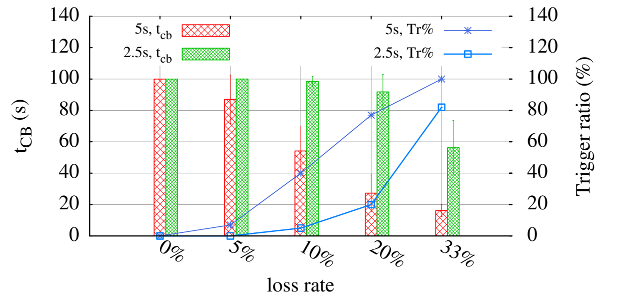


Figure 9: Impact of using a shorter RTCP interval on the circuit breaker. Each scenario is run 50 times and the error bars represent the 95% confidence level.

explored, and while the data is likely representative of residential links in the UK and Finland, the infrastructure may differ significantly in other regions. We also have not yet considered wireless and mobile users. Evaluating the performance of the RTP circuit breaker based on results of a wider measurement study would be desirable. Overall, though, our initial results show that the RTP circuit breaker is working as designed.

Acknowledgements

This work was funded in part by the UK EPSRC, the Cisco University Research Program Fund, and by Ericsson. Varun Singh is funded in part by Future Internet Graduate School and EIT-ICT labs.

References

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," IETF, 2003, RFC 3550.

[2] C. S. Perkins and V. Singh, "Multimedia congestion control: Circuit breakers for unicast RTP sessions," IETF, Feb. 2013, work in progress.

[3] V. Singh and J. Ott, "Evaluating Congestion Control for Interactive Real-time Media," IETF, July 2013, work in progress.

[4] C. Jennings, T. Hardie, and M. Westerlund, "Real-time communications for the web," *IEEE Comm. Magazine*, vol. 51, no. 4, April 2013.

[5] M. Zink, O. Künzel, J. Schmitt, and R. Steinmetz, "Subjective Impression of Variations in Layer Encoded Videos," in *Proc. IWQoS*, 2003.

[6] E. Brosh, S. A. Baset, D. Rubenstein, and H. Schulzrinne, "The delay-friendliness of TCP," *Proc. ACM SIGMETRICS*, vol. 36, no. 1, 2008.

[7] M. Schier and M. Welzl, "Using DCCP: Issues and improvements," in *Proc. IEEE ICNP*, 2012.

[8] V. Singh, A. A. Lozano, and J. Ott, "Performance analysis of receive-side real-time congestion control for WebRTC," in *Proc. Intl. Packet Video Workshop*, Dec. 2013.

[9] J. Ott, S. Wenger, N. Sato, C. Burmeister, and J. Rey, "Extended RTP Profile for RTCP-Based Feedback," RFC 4585, IETF, Jul. 2006.

[10] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *ACM CCR*, July 1997.

[11] J. Padhye *et al.*, "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. ACM SIGCOMM*. ACM, August 1998.

[12] S. Floyd *et al.*, "Equation-based congestion control for unicast applications," in *Proc. ACM SIGCOMM*, August 2000.

[13] M. Ellis, C. S. Perkins, and D. P. Pezaros, "End-to-end and network-internal measurements of real-time traffic to residential users," in *Proc. ACM MMSys*, San Jose, CA, USA, Feb. 2011.

[14] T. Friedman, R. Caceres, and A. Clark, "RTP control protocol extended reports)," IETF, November 2003, RFC 3611.

[15] C. Perkins and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port," RFC 5761, IETF, Apr. 2010.

[16] M. Carbone and L. Rizzo, "Dummynet revisited," *ACM CCR*, Jan 2010.

[17] I. Johansson and M. Westerlund, "Support for Reduced-Size RTCP: Opportunities and Consequences," RFC 5506, IETF, Apr. 2009.

[18] M. Ellis, D. P. Pezaros, T. Kypraios, and C. S. Perkins, "Modelling packet loss in RTP-based streaming video for residential users," in *Proc. Conference on Local Computer Networks*. IEEE, October 2012.