Coupled Congestion Control for WebRTC

Safiqul Islam, Michael Welzl and Stein Gjessing
Networks and Distributed Systems Group, Department of Informatics
University of Oslo, Norway
Email: {safiquli, michawe, steing}@ifi.uio.no

***Abstract*—Congestion occurs at a bottleneck along an Internet path. Multiple flows between the same sender and receiver pairs can benefit from using only a single congestion control instance when they share the same bottleneck. These benefits include the ability to control the rate allocation between flows and reduced overall delay (multiple congestion control instances cause more queuing delay than one since each has no knowledge of the congestion episodes experienced by the others). We present a mechanism for coupling congestion control for real-time media and show its benefits by coupling multiple congestion controlled flows that share the same bottleneck.**

I. Introduction

Multiple congestion controlled flows (e.g., TCP) between the same two hosts usually have separate congestion control instances, even when the path used by them is the same. There may be several reasons for this separation. For example, one cannot always be sure if the path is indeed the same – routing mechanisms like Equal-Cost Multi-Path (ECMP) may assign different flows to different paths to achieve load balancing, even when they have the same destination IP address.

Routers or other middle-boxes usually identify flows using a five-tuple of source and destination IP addresses, transport protocol, and the transport protocol's source and destination port numbers. When – as it will be possible with the new WebRTC standard for interactive communication between web browsers – multiple flows are multiplexed over a single UDP port pair, they are normally regarded as a single flow inside the network and therefore treated in the same way. In such a setup, congestion management can be readily applied.

The new "RTP Media Congestion Avoidance Techniques" (RMCAT) IETF Working Group develops standards for RTP-based interactive real-time media. WebRTC being the major use case for these standards, RMCAT will also standardize methods for coupled congestion control, with the goal of having the best possible control over the send rate allocation. Here, we describe the first proposal for RMCAT's coupled congestion control and show its feasibility and some of its benefits.

II. The Flow State Exchange (FSE)

RMCAT's congestion control should be applicable but not limited to WebRTC. This means that we may need to jointly control flows that reside within a single application (a web browser, in case of WebRTC) or in multiple applications. In the latter case, the benefit for WebRTC of knowing that packets from multiple flows will be routed in the same way is lost. There are, however, measurement based methods to determine whether multiple flows share a bottleneck in the network; being able to make use of measurements when necessary, and supporting various intra- as well as inter-application scenarios calls for a congestion management architecture that is much simpler than, e.g., the well-known Congestion Manager (CM) [1].

We have opted for an approach [2] that minimizes the amount of necessary changes to existing congestion control algorithms. It involves a central storage element called the "Flow State Exchange" (FSE). The elements of our architecture for coupled congestion control are: the FSE and Shared Bottleneck Detection (SBD) [3].

Every time a flow's congestion control mechanism would normally update its sending rate, the flow instead updates information in the FSE and performs a query on the FSE, leading to a sending rate that can be different from what the congestion controller originally determined. The FSE additionally calculates the rates for all the other flows based on their aggregate behavior in the same Flow Group (FG) and actively informs their congestion controllers using a callback function. A Flow Group consists of flows which should be controlled together, i.e. they have a common network bottleneck as indicated by the SBD element. We limit the aggregate rate growth (in the absence of congestion) of N flows to I/N, where I is the flow's increase factor. In order to avoid overreacting to congestion, the FSE emulates the behavior of a single flow by proportionally reducing the aggregate rate on congestion. Additionally, we set a timer that prohibits flows other than the flow that just reduced its rate from changing their rate for two RTT periods (of the flow that reduced its rate). We decided to use 2 RTTs so that other flows do not react to the same loss interval. We assume a loss interval to persist for up to one RTT and added another RTT to compensate for fluctuations in the measured RTT value.

III. Performance Evaluation of the FSE

We implemented the FSE in ns-2 and simulated the behavior of congestion controlled flows using a dumbbell network topology(bottleneck capacity 10 Mbit/s, RTT 100 ms, packet size 1000 bytes, and half-BDP queue of 62 packets). We also tested other queue lengths and saw consistently lower queuing delay with FSE, see [4]. For simplicity, unless otherwise mentioned, senders always had enough data to send.[1] All tests

[1] This may not be a totally unreasonable assumption for modern multimedia systems, which may be able to closely track the available bandwidth (cf. [5]). However, the actual behavior is codec-dependent and hard to characterize.

reported here were carried out 10 times with different randomly picked start times over the first second. The produced results had such a small standard deviation (the worst case was 0.2%) that we opted against showing error bars for the sake of clarity.

The current implementation supports four protocols: Rate Adaptation Protocol (RAP) [6] (because it is a simple rate-based Additive Increase – Multiplicative Decrease (AIMD) scheme, hence representing a whole class of TCP-like mechanisms); TCP Friendly Rate Control (TFRC) [7] (because it is the only standardized congestion control mechanism aimed at supporting media flows); Low Extra Delay Background Transport (LEDBAT) [8] (because it is a delay-based mechanism and the congestion control of RMCAT is currently under development, and will probably be delay based); and Network-Assisted Dynamic Adaptation (NADA) [9] (because it is work in progress in RMCAT). This paper only shows results for RAP and TFRC, however we provide recommendations in section IV for the congestion controllers which require special consideration.

It is clear from the algorithm, and was also confirmed in our simulations, that the FSE achieves precise fairness among the flows. This is important, as it is a requirement for WebRTC [10] – but because coupling congestion controllers should help avoid competition at the bottleneck, we expected reduced queuing delay and packet loss, while achieving throughput that is close to the throughput of a single flow.

Sending very little obviously produces a small queue and reduces packet loss; however, because we try to emulate the behavior of one flow, it should not have a significantly smaller throughput than a single flow. As expected, in all tests, the link utilization with the FSE was at most equal or smaller than without the FSE. However, link utilization of the FSE-controlled RAP flows is *higher* than the link utilization of a single RAP flow. In contrast, for the FSE-controlled TFRC flows, link utilization is in some cases less than the link utilization of one flow, but the difference appears rather marginal (3% less in the worst case in our tests).

Figures 1 and 2 illustrate that the FSE achieves a consistent reduction of the average queuing delay both for TFRC and RAP.

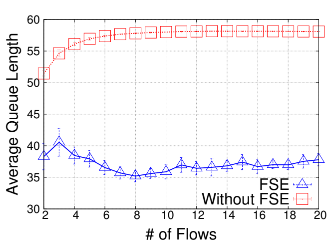

Fig. 1. Average queue length (TFRC)

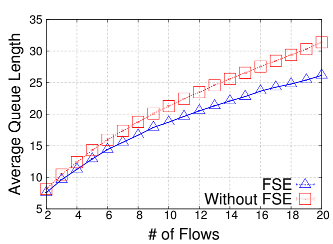

Fig. 2. Average queue length (RAP)

To achieve prioritization, one of the requirements of RMCAT, the FSE can calculate and assign rates based on a priority. FSE-controlled flows change their rates based on the assigned priorities over time. This means that a high priority flow can easily get the desired rate from the FSE without requiring any further changes in its congestion controller.

IV. Recommendations

Our approach minimizes the amount of necessary changes to the existing congestion controllers. Table I provides recommendations to the necessary changes to that specific congestion controllers of the flows.

CC	Description
TFRC	We recommend to incorporate the receiver side's calculation to compensate for the lower value of p. From [7], TFRC increases by at most 0.22 packets per RTT, as a result of the deterministic length of loss intervals measured by the receiver. When TFRC uses a lower rate than planned, the loss interval gets artificially prolonged at the receiver, which then calculates a lower value for the loss event ratio p, which in turn provokes a faster rate increase at the sender.
NADA	Nada calculates the reference rate upon receiving an acknowledgment, and then based on the reference rate, it calculates an encoding rate and a sending rate for the flows. This creates a problem when we actively change the reference rate, but not the sending and encoding rate. Hence, we recommend to change the sending rate and encoding rate along with the reference rate.

TABLE I
RECOMMENDATIONS FOR USING THE FSE WITH THE EXISTING CONGESTION CONTROL ALGORITHMS

V. Acknowledgements

This work is partially supported by the European Union through the FP7-ICT project RITE under contract number 317700.

References

[1] H. Balakrishnan and S. Seshan, "The Congestion Manager," RFC 3124, June 2001.

[2] M. Welzl, S. Islam, and S. Gjessing, "Coupled congestion control for RTP media," Internet-draft draft-welzl-rmcat-coupled-cc-04 (work in progress)", 2014.

[3] D. A. Hayes, S. Ferlin, and M. Welzl, "Practical passive shared bottleneck detection using shape summary statistics," in *2014 IEEE 39th Conference on Local Computer Networks (LCN)*. IEEE, 2014, pp. 150–158.

[4] S. Islam, M. Welzl, S. Gjessing, and N. Khademi, "Coupled congestion control for RTP media," Technical Report 440, March 2014, https://www.duo.uio.no/handle/10852/39177.

[5] M. Nagy, V. Singh, J. Ott, and L. Eggert, "Congestion control using fec for conversational multimedia communication," *arXiv preprint arXiv:1310.1582*, 2013.

[6] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the internet," in *IEEE INFOCOM*, 1999.

[7] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *ACM SIGCOMM*, 2000. [Online]. Available: http://doi.acm.org/10.1145/347059.347397

[8] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuehlewind, "Low extra delay background transport (ledbat)," *IETF draft*, 2010.

[9] X. Zhu, R. Pan, M. A. Ramalho, S. Mena, C. Ganzhorn, P. E. Jones, and S. D. Aronco, "NADA: A unified congestion control scheme for real-time media," draft-zhu-rmcat-nada-06 (work in progress), 2015.

[10] C. Holmberg, S. Hakansson, and G. Eriksson, "Web real-time communication use-cases and requirements," Internet-draft draft-ietf-rtcweb-use-cases-and-requirements-12.txt (work in progress), 2013.