

One Control to Rule Them All: Coupled Congestion Control for RTP Media

Safiqul Islam(student), Michael Welzl(faculty) and Stein Gjessing(faculty)
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 can therefore benefit from using only a single congestion control instance when they share the same path. These benefits include the ability to perfectly control the rate allocation between flows. We present a mechanism for coupling congestion control for real-time media that is simpler and easier to deploy than the conceptually similar “Congestion Manager”, and we show its benefits by coupling multiple congestion controlled flows sharing 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 between 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 address, transport protocol, and the transport protocol’s port numbers. When – as it will be possible with the new WebRTC¹ 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. This means that congestion management can perhaps be readily applied for WebRTC.

The new “RTP Media Congestion Avoidance Techniques” (RMCAT) IETF Working Group intends to develop 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, WebRTC’s benefit 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 (cf. [1] and [2]). Being able to make use of such 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) [3].

We have opted for an approach [4] that minimizes the amount of necessary changes to existing applications. It involves a central storage element called “Flow State Exchange” (FSE). The elements of the proposed architecture for coupled congestion control are: the Flow State Exchange (FSE), Shared Bottleneck Detection (SBD) and Flows. The FSE is a storage element that can be implemented in two ways: active and passive. In the active version, it initiates communication with flows and SBD. However, in the passive version, it does not actively initiate communication with flows and SBD; its only active role is internal state maintenance (e.g., an implementation could use soft state to remove a flow’s data after long periods of inactivity).

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. In the active version, the FSE additionally calculates the rates for all the other flows in the FG and actively informs their congestion controllers with a callback function. Using information about/from the currently active flows, SBD updates the FSE with the correct Flow State Identifiers (FSIs). Groups of flows which should be controlled together have a common “Flow Group Identifier” (FGI), which can be set by a “Shared Bottleneck Detection” (SBD) module based on measurements or knowledge about multiplexing (as with WebRTC). An SBD module can be a part of one of the applications using the FSE, or it can be a standalone entity. Work on SBD is currently ongoing; this paper only focuses on the FSE, assuming that flows are sharing the same bottleneck.

¹The IETF counterpart of the W3C WebRTC standard is called RTCWEB. For simplicity, we will use the term “WebRTC” for the whole set of standards in this document.

III. PERFORMANCE EVALUATION OF THE FSE

We implemented the FSE in ns-2 and simulated the behavior of congestion controlled flows using a dumbbell topology. The current implementation only supports two rate-based protocols: Rate Adaptation Protocol (RAP) (because it is a simple rate-based Additive-Increase, Multiplicative-Decrease (AIMD) mechanism, hence representing a whole class of TCP-like mechanisms) and TCP Friendly Rate Control (TFRC) (because it is the only standardized congestion control mechanism aimed at supporting media flows) [5].

Jain's fairness index is used to calculate the expected gains in fairness where a fairness index of 1 denotes that all n concurrent flows get a fair share of the total available bandwidth whereas a fairness index of $1/n$ means that one of the n flows gets the entire available bandwidth.

Figure 1 illustrates the fairness index for 2 RAP flows as the RTT ratio is varied; the positive influence of the fairness for the FSE-controlled flows is noticeable. Results were similar for 3, 4 and 5 flows. Figure 2 also shows the influence of the fairness index when the number of flows with homogeneous RTTs is varied. We also ran the same tests for TFRC flows, and the results were similar. The FSE ensures the fairness by distributing the aggregated sending rates fairly to all the flows.

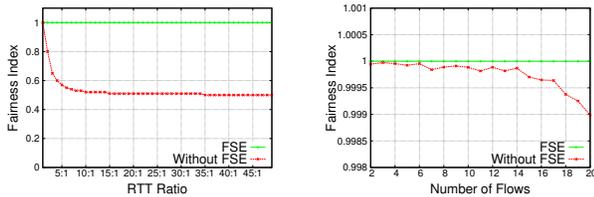


Fig. 1. Fairness index for 2 RAP flows as the RTT ratio is varied, with and without FSE

Fig. 2. Fairness index as the number of RAP flows is varied, with and without FSE

To achieve prioritization, one of the requirements of RM-CAT, the FSE can calculate and assign rates based on the priority. Figure 3 shows how two 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.

The loss ratio for FSE-controlled vs. non-FSE-controlled RAP flows is illustrated in Figure 4. As can be seen, the loss ratio gain with the FSE increases with the number of flows. Results were less favorable with TFRC; this is a subject of current investigation.

This paper only describes results from the active version of our algorithm, which is designed to be the simplest possible method to assign rates according to priorities. However, in [4] we also propose a passive variant of the FSE which is easier to use (there are no callbacks) and lets bulk transfers immediately use the bandwidth that is not used by application-limited flows.

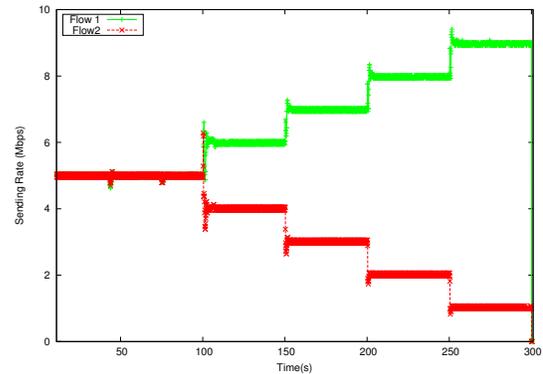


Fig. 3. 2 Flows with different priorities over time coupled via FSE

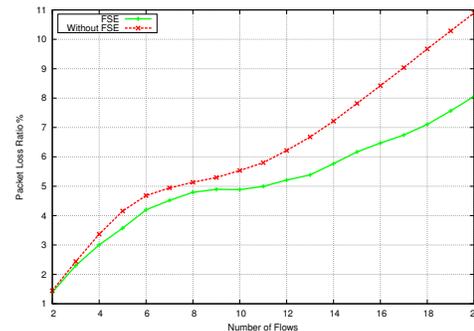


Fig. 4. Loss ratio percentage as the number of flows is increased. It is clear that loss ratio percentage for FSE controlled flows decreases as the number of flows is varied

IV. ACKNOWLEDGEMENTS

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

REFERENCES

- [1] B. Y. Muhammad Murtaza Yousaf, Michael Welzl, "Accurate shared bottleneck detection based on svd and outlier detection," University of Innsbruck, Institute of Computer Science, Tech. Rep. DPS NSG Technical Report 1, August 2008.
- [2] S. Hassayoun, J. Iyengar, and D. Ros, "Dynamic window coupling for multipath congestion control," in *Network Protocols (ICNP), 2011 19th IEEE International Conference on*, oct. 2011, pp. 341–352.
- [3] H. Balakrishnan, H. S. Rahul, and S. Seshan, "An integrated congestion management architecture for internet hosts," in *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, ser. SIGCOMM '99. New York, NY, USA: ACM, 1999, pp. 175–187. [Online]. Available: <http://doi.acm.org/10.1145/316188.316220>
- [4] M. Welzl, S. Islam, and S. Gjessing. (2013) Coupled congestion control for rtp media. Internet draft [work in progress]. [Online]. Available: <http://tools.ietf.org/html/draft-welzl-rmcat-coupled-cc-01>
- [5] J. Widmer, R. Denda, and M. Mauve, "A survey on tcp-friendly congestion control," *Network, IEEE*, vol. 15, no. 3, pp. 28–37, 2001.