Congestion Control Aware Queuing

Maximilian Bachl, Joachim Fabini, Tanja Zseby
Technische Universität Wien
firstname.lastname@tuwien.ac.at

ABSTRACT

Recent model-based congestion control algorithms such as BBR use repeated measurements at the end-point to build a model of the network connection and use it to achieve optimal throughput with low queuing delay. Conversely, applying this model-based approach to Active Queue Management (AQM) has so far been underinvestigated. We propose an AQM scheduler based on fair queuing, which adapts the buffer size depending on the needs of each flow without requiring active participation from the endpoint. We implement this scheduler for the Linux kernel and show that it interacts well with the most common congestion control algorithms and can significantly increase throughput compared to CoDeL while avoiding overbuffering.

ACM Reference Format:
Maximilian Bachl, Joachim Fabini, Tanja Zseby. 2019. Congestion Control Aware Queuing. In *Proceedings of Workshop on Buffer Sizing*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In the last decades various Active Queue Management (AQM) mechanisms have been proposed to minimize excessive standing queues in the Internet. One of the most influential recent efforts is CoDeL [5] whose goal is that the queuing delay at the bottleneck link is at least once under 5 ms in a moving window of 100 ms.

While it is important to keep the queuing delay constrained, it is also necessary to ensure that overly "aggressive" flows cannot benefit by "stealing" less aggressive flows' bandwidth. Thus researchers and engineers have developed *Fair Queuing (FQ)* mechanisms [1, 6] to isolate different flows' queues (*flow queuing*) so that for example a delay-sensitive live video call cannot be impaired by a concurrent bulk transfer which takes all the available bandwidth.

Recent approaches have tried to combine AQM with FQ. [7] developed *fq_codel*, a queuing discipline (qdisc) that uses FQ and lets CoDeL manage each queue. [3] expand upon this and create the *cake* qdisc that also adds features such as not only per-flow queuing but also per-host queuing for even increased fairness. Furthermore they also include bandwidth shaping into their solution and aim to create one qdisc that is easy to configure and can be easily deployed on home routers and offers all features in one solution.

While we do not want to make statements about the general performance of CoDeL, we show that fq_codel and cake do not optimally use available bandwidth in common network configurations for common Congestion Control Algorithm (CCA)s. This becomes especially prevalent for links with a high bandwidth or a large Round-Trip Time (RTT) but is already noticeable for common scenarios, such as a link with 100 Mbit/s and an RTT of 50 ms.

We show that the impaired performance is a result of keeping the queuing delay under 5 ms, which hinders CCAss such as Reno or Cubic from reaching maximum throughput. The problem is that these qdiscs aim to keep the delay under the threshold no matter the effect on throughput and do not take the congestion control of the flow into account.

As a remedy, we conceive an AQM mechanism that explicitly measures the behavior of the Congestion Control (CC) of a flow and dynamically changes the buffer so that

(1) link utilization is optimized and
(2) queuing delay is kept at a minimum that is required to achieve optimum throughput considering the CC

Finally, we develop a prototype of our qdisc and show that it can achieve these objectives for the most common loss-based CCAs, Reno [4] and Cubic [2]. These CCAs work by continuously increasing the number of bytes that are allowed to be in the network if no packet loss is experienced and by sharply decreasing this number if a packet is lost. With Cubic being the default CC in all major OS TODO: citation needed, we especially emphasize our evaluation on improving its performance. Contrasting to the aforementioned CCAs, recently proposed BBR [?] does not continuously increase and then sharply decrease when packets are lost but instead uses periodic measurements to estimate the available bandwidth as well as the minimal RTT and then tries to stay at this

Workshop on Buffer Sizing, December 2-3, 2019, Stanford, California, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

point of optimal bandwidth and minimal delay. BBR is thus considered to be *model-based*. We demonstrate that our qdisc also behaves well in interaction with CCAs BBR.

2 CONCEPT

Algorithm 1 Procedure which determines when to synchronize the neural network weights from the global weights in case of congestion control. We assume that the window at time step t is stored in a list l_{windows} and that the number of bytes sent in each time step is stored in l_{sent}. Alternatively to this procedure one can simply synchronize the weights after each $t_{\max}$ actions where $t_{\max}$ is an arbitrarily chosen integer greater than 0.

```
1:  function updateWeights?
2:      i ← 0
3:      while i < #(l_actions) do
4:          w ← ⌊l_windows[i] + l_actions[i]⌋
5:          for j ← i..#(l_actions) − 1 do
6:              w ← w − l_sent[j]
7:              if w ≤ 0 ∧ i + 1 ≥ 10 then          ▷ at least 10
8:                  i ← j
9:                  break
10:             end if
11:         end for
12:         if i = #(l_windows) − 1 then
13:             return true
14:         end if
15:     end while
16:     return false
17: end function
```

ACKNOWLEDGEMENTS

We thank Gernot Vormayr for providing us with the `py-virtnet` toolkit for building virtual networks.

REFERENCES

[1] Eric Dumazet. 2013. pkt_sched: fq: Fair Queue packet scheduler [LWN.net]. https://lwn.net/Articles/565421/

[2] Sangtae Ha, Injong Rhee, and Lisong Xu. 2008. CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS Operating Systems Review* 42, 5 (July 2008), 64–74. https://doi.org/10.1145/1400097.1400105

[3] T. Høiland-Jørgensen, D. Täht, and J. Morton. 2018. Piece of CAKE: A Comprehensive Queue Management Solution for Home Gateways. In *2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*. 37–42. https://doi.org/10.1109/LANMAN.2018.8475045

[4] V. Jacobson. 1988. Congestion Avoidance and Control. In *Symposium Proceedings on Communications Architectures and Protocols (SIGCOMM '88)*. ACM, New York, NY, USA, 314–329. https://doi.org/10.1145/52324.52356 event-place: Stanford, California, USA.

[5] Kathleen Nichols and Van Jacobson. 2012. Controlling Queue Delay. *Queue* 10, 5 (May 2012), 20:20–20:34. https://doi.org/10.1145/2208917.2209336

[6] M. Shreedhar and G. Varghese. 1996. Efficient fair queuing using deficit round-robin. *IEEE/ACM Transactions on Networking* 4, 3 (June 1996), 375–385. https://doi.org/10.1109/90.502236

[7] D. Taht, Jim Gettys, T. Hoeiland-Joergensen, Toke Hoeiland-Joergensen, Eric Dumazet, J. Gettys, E. Dumazet, and P. McKenney. 2018. The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm. https://tools.ietf.org/html/rfc8290