Backpressure Flow Control

Prateesh Goyal[1], Preey Shah[2], Naveen Kr. Sharma[3], Mohammad Alizadeh[1], Thomas E. Anderson[3]
MIT CSAIL[1], IIT Bombay[2], University of Washington[3]

ABSTRACT

Effective congestion control in data centers is becoming increasingly challenging with rapidly increasing workload demand, ever faster links, small average transfer sizes, extremely bursty traffic, and limited switch buffer capacity. Existing deployed algorithms, such as DCQCN or DCTCP, are not effective in managing buffers and are still far from optimal in many plausible scenarios, particularly for tail latency. Many operators compensate by running their networks at low average utilization, dramatically increasing costs. In this talk, we argue that we have reached the practical limits of end-to-end congestion control. Instead, we propose a new clean slate design based on hop-by-hop per-flow flow control. Our approach reduces buffer occupancy compared to existing schemes, and, achieves near optimal tail latency behavior even under challenging conditions such as high average link utilization and in-cast cross traffic. By contrast with prior hop-by-hop schemes, our approach manages buffers more efficiently and achieves per-flow flow control with limited metadata.

1 INTRODUCTION

Data centers are increasingly dominating the market for all types of high end computing, including enterprise services, parallel computing, large scale data analysis, fault tolerant middleboxes, and global distributed applications [3, 7, 11]. These workloads place enormous pressure on the data center network to deliver, at low cost, ever faster throughput with low tail latency even for highly bursty traffic [6, 16].

Managing buffers efficiently is critical for achieving high performance in data centers. Deep packet buffers can cause head-of-line (HoL) blocking for short flows and increase flow completion times. For RDMA, limiting buffer occupancy is essential to avoid buffer overflows and PFC triggers [15]. Researchers and data center operators have proposed several end-to-end congestion avoidance systems that aim to reduce buffer occupancy. DCTCP [2], Timely [12], DCQCN [17], and HPCC [10] gather information from the network in response to packet transmissions to gate how quickly to send additional packets. These schemes can be effective when congestion is due to so-called elephant flows: large, many-round trip transfers that can be throttled up or down based on network conditions to avoid HoL-blocking for short, latency sensitive flows. However, measurements of data center network workloads suggest that most traffic completes within at most a few round trips. As network speeds increase, an increasing percentage of data center traffic can complete in a single round trip; in effect, to the data center network of the future, almost all traffic will appear to be small and bursty. In aggregate, however, these small flows can still overwhelm downstream switch buffers.

Addressing this puts the network designer into a painful bind. Transfers can be artificially slowed even when bandwidth is available, making the network much less useful for applications with fine-grained parallelism. Network utilization can be kept low, making the network much more costly to operate. Or we can allow tail latencies to become unpredictable and dependent on cross-traffic behavior by other applications.

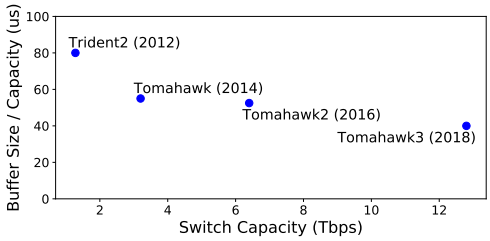


Figure 1: Hardware trends for top of the line data center switches manufactured by Broadcom. Buffer size is not keeping up with increases in switch capacity.

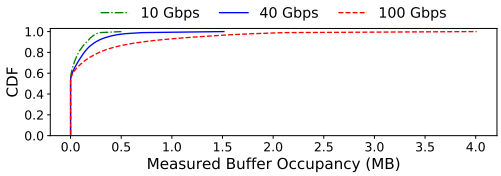


Figure 2: CDF of switch buffer occupancy for DCQCN (without PFC). The workload is scaled for equal utilization at various link speeds. Higher speed switches reduce DCQCN's ability to control buffer occupancy.

Better switch scheduling, such as hierarchical fair queueing, can improve flow completion times for short flows, but scheduling doesn't reduce buffer occupancy on its own. We need to coordinate scheduling and buffer allocation, or end to end performance can suffer.

We next argue that these trends are getting worse over time.

Trend 1: Rapidly increasing link speed Fig. 1 shows the switch capacity of top of the line data center switches manufactured by Broadcom [5, 14]. Switch capacity and link speeds have increased by a factor of 10 over the past six years with no signs of stopping. As a result, much more data can be transmitted within a single round trip; for example, with a 100 Gbps network link and 12 us round trip delay, there can be 150 KB in flight before *any* feedback is received at the endpoint.

Larger bandwidth-delay makes it harder for end-to-end congestion control to manage network buffers. Fig. 2 shows the cumulative distribution function (CDF) of buffer occupancy with DCQCN across different link speeds, using a fat tree topology and the Google workload from Fig. 4 set to 75% load plus 5% incast, but without PFC enabled. DCQCN fails to limit buffer occupancy at higher link speeds, if utilization is kept constant.

Trend 2: Buffer size is not scaling with switch capacity Fig. 1 shows that the switch buffer size relative to its capacity has decreased by a factor of 2 (from 80 us to 40 us) over the past six years. With smaller buffers relative to link speed, buffers now fill up more quickly, making it more difficult for

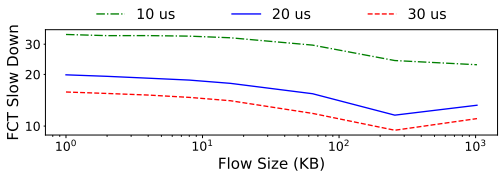


Figure 3: The effect of the switch buffer capacity ratio on the 99^{th} percentile flow completion times (FCT) with DCQCN. Reducing buffer size hurts tail latency.

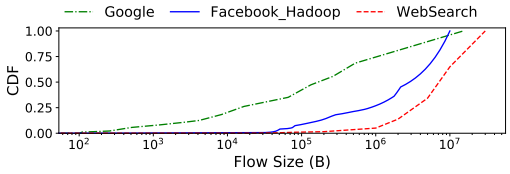


Figure 4: Cumulative bytes contributed by different flow sizes for three different industry workloads.

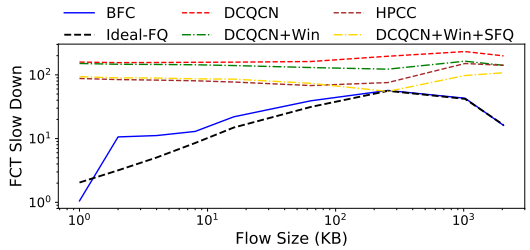
end-to-end congestion control to manage those buffers. Using 100 Gbps links and the same topology and workload as Fig. 2, Fig. 3 shows the effect on flow completion tail latency (99^{th} percentile) as a function of the buffer space ratio. If the buffer space ratio continues to go down due to chip design constraints, the ability of DCQCN to manage tail latency is likely to worsen.

Trend 3: Flow sizes remain small Fig. 4 shows the cumulative bytes contributed by flows with different sizes in three industry data center workloads. For each of these workloads, the majority of flows are smaller than the bandwidth-delay product (150 KB on 100 Gbps links with 12us end-to-end RTT). Instead, the graph shows the byte-weighted cumulative distribution. For example, the large majority of bytes in the Google workload are in flows that fit inside a single round trip. Although data processing jobs are typically considered to have large flows, almost all Facebook Hadoop traffic is likely to fit within a single round trip within the next few years, with websearch graph analysis to follow shortly thereafter.

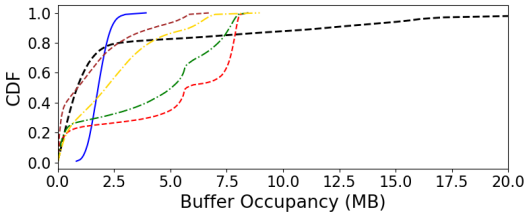
There are several consequences. Even if we develop protocols to gather precise information about the state of the bottleneck switch and return it to the sender, that information could well be out of date by the time the sender could act on it. Even if congestion is stable, end-to-end adaptation is inherently iterative. The rate for any particular flow depends on secondary bottlenecks encountered by other flows as they simultaneously change their rate. Equilibrium sharing is reached with DCQCN/HPCC only after many round trips. Flows could be intentionally slowed down to provide more consistency over time, but this would come at a cost in substantial added latency.

2 BACKPRESSURE FLOW CONTROL

Unlike existing work on improving end-to-end congestion control, we propose a different approach, to revisit the idea of per-hop per-flow flow control. The key challenge for data center networks, in our view, is to efficiently allocate buffer space at congested network switches. This becomes easier and simpler when control actions are taken per flow and per hop. Per-hop per-flow flow control is of course not a new idea, having been introduced, and discarded, with ATM networks over



(a) 99^{th} percentile FCT



(b) Buffer Occupancy

Figure 5: 99^{th} flow completion time and buffer occupancy on the Google workload with incast traffic.

two decades ago [4, 9]. These earlier schemes didn't focus on managing buffers and suffered from buffer exhaustion. Moreover, these schemes required per-flow state at each switch even for quiescent flows, an amount of state that would not be practical in today's data center networks. Our principal insight is to show that per-hop per-flow flow control can be approximated with a small amount of switch state and a modest amount of signalling overhead. We note that our approach is approximate, and therefore it is not completely loss-free.

In this talk, we present BFC (Backpressure Flow Control), a protocol for per-hop per-flow flow control, that uses a small amount of metadata to manage buffers efficiently and achieve near-optimal tail-latency performance for typical data center workloads. In BFC, a switch only tracks flows with packets in the switch. On a flow arrival the switch dynamically assigns an unused physical queue (if there is one) to the flow. This dynamic assignment reduces the number of collisions (flows sharing the same physical queue) in comparison to static assignments such as in stochastic fair queueing (SFQ). Based on the occupancy of physical queues, a switch pauses/resumes individuals flows at the upstream. Dynamic physical queue assignment and aggressive pause/resume reduces HoL blocking while keeping the buffer occupancy low.

We evaluate BFC using ns3 [1] on synthetic traces drawn to be consistent with measured workloads from Google and Facebook data centers [13], for a multi-level Clos network topology. Fig. 5 shows the 99^{th} percentile FCT (flow completion time) and buffer occupancy at the switch on the Google workload. The average load was 60% in this experiment with 5% incast traffic. For BFC and DCQCN+Win+SFQ we used 32 physical queues per port in accordance with existing switches. To understand how close is BFC to ideal, we simulated ideal fair queuing with infinite buffer (Ideal-FQ). Unlike DCQCN and HPCC, BFC achieves close to the optimal tail latency performance across a wide range of flow transfer sizes. The tail buffer occupancy in BFC is lower than for any other scheme. Please see [8] for further design details and evaluation of BFC.

REFERENCES

[1] [n. d.]. Network Simulator 3. https://www.nsnam.org. ([n. d.]).

[2] Mohammad Alizadeh, Albert Greenberg, David A Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. 2010. Data Center TCP (DCTCP). In *SIGCOMM*.

[3] Amazon. [n. d.]. Amazon Web Services. https://aws.amazon.com/s3/. ([n. d.]).

[4] Thomas E Anderson, Susan S Owicki, James B Saxe, and Charles P Thacker. 1993. High-speed switch scheduling for local-area networks. *ACM Transactions on Computer Systems (TOCS)* 11, 4 (1993), 319–352.

[5] Broadcom. [n. d.]. StrataXGS. https://www.broadcom.com/products/ethernet-connectivity/switching/strataxgs. ([n. d.]).

[6] Jeffrey Dean and Luiz André Barroso. 2013. The tail at scale. *Commun. ACM* 56, 2 (2013), 74–80.

[7] Google. [n. d.]. Google Cloud Platform. https://cloud.google.com. ([n. d.]).

[8] Prateesh Goyal, Preey Shah, Naveen Kr Sharma, Mohammad Alizadeh, and Thomas E Anderson. 2019. Backpressure Flow Control. *arXiv preprint arXiv:1909.09923* (2019).

[9] NT Kung and Robert Morris. 1995. Credit-based flow control for ATM networks. *IEEE network* 9, 2 (1995), 40–48.

[10] Yuliang Li, Rui Miao, Hongqiang Harry Liu, Yan Zhuang, Fei Feng, Lingbo Tang, Zheng Cao, Ming Zhang, Frank Kelly, Mohammad Alizadeh, et al. 2019. HPCC: high precision congestion control. In *Proceedings of the ACM Special Interest Group on Data Communication*. ACM, 44–58.

[11] Microsoft. [n. d.]. Microsoft Azure. https://azure.microsoft.com/. ([n. d.]).

[12] Radhika Mittal, Nandita Dukkipati, Emily Blem, Hassan Wassel, Monia Ghobadi, Amin Vahdat, Yaogong Wang, David Wetherall, David Zats, et al. 2015. TIMELY: RTT-based Congestion Control for the Datacenter. In *ACM SIGCOMM Computer Communication Review*, Vol. 45. ACM, 537–550.

[13] Behnam Montazeri, Yilong Li, Mohammad Alizadeh, and John Ousterhout. 2018. Homa: A receiver-driven low-latency transport protocol using network priorities. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. ACM, 221–235.

[14] The Next Platform. [n. d.]. FLATTENING NETWORKS – AND BUDGETS – WITH 400G ETHERNET. https://www.nextplatform.com/2018/01/20/flattening-networks-budgets-400g-ethernet/. ([n. d.]). January 20, 2018.

[15] Robert Williams and Bahadir Erimli. 2005. Method and apparatus for performing priority-based flow control. (Oct. 18 2005). US Patent 6,957,269.

[16] David Zats, Tathagata Das, Prashanth Mohan, Dhruba Borthakur, and Randy Katz. 2012. DeTail: reducing the flow completion time tail in datacenter networks. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*. ACM, 139–150.

[17] Yibo Zhu, Haggai Eran, Daniel Firestone, Chuanxiong Guo, Marina Lipshteyn, Yehonatan Liron, Jitendra Padhye, Shachar Raindel, Mohamad Haj Yahia, and Ming Zhang. 2015. Congestion control for large-scale RDMA deployments. *ACM SIGCOMM Computer Communication Review* 45, 4 (2015), 523–536.