Dequeue Rate-Agnostic Switch Buffer Sharing through Packet Queueing Delay

Krishna Agarwal
Eindhoven University of Technology
Eindhoven, Netherlands

Vamsi Addanki
TU Berlin
Berlin, Germany

Habib Mostafaei
Eindhoven University of Technology
Eindhoven, Netherlands

Abstract

Datacenter network switches share packet buffers among all ports to enhance throughput and reduce packet drops. However, declining buffer space per-port-per-bandwidth unit challenges buffer-sharing mechanisms, affecting performance. Recent studies, like ABM (SIGCOMM 2022), suggest hierarchical packet admission schemes to address this, but their complexity hinders efficiency. We propose CBM, a packet delay-based buffer sharing scheme that manages buffer space and controls queue drain rates using a single configurable parameter. Preliminary evaluation shows that CBM improves advanced transport protocol performance, such as PowerTCP, reducing Flow Completion Times (FCTs) by up to 45.07% compared with ABM.

CCS Concepts

• **Networks** → **Data path algorithms**; **Data center networks**.

Keywords

Switch buffer, packet queueing delay, datacenter networks

ACM Reference Format:
Krishna Agarwal, Vamsi Addanki, and Habib Mostafaei. 2024. Dequeue Rate-Agnostic Switch Buffer Sharing through Packet Queueing Delay. In *Proceedings of the CoNEXT Student Workshop 2024 (CoNEXT-SW '24), December 9–12, 2024, Los Angeles, CA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3694812.3699924

1 Introduction

Datacenter switches use shared on-chip packet buffers to improve throughput and reduce packet drops. However, rising buffer costs and the bursty nature of traffic have reduced buffer availability per port-per-bandwidth unit. This trend, along with microsecond-scale bursts [1], has increased the need for better buffer-sharing algorithms to prevent packet drops during congestion [2–6]. However, excessive growth in one queue can harm other queues, leading to issues like starvation and throughput deprivation across independent queues [3].

Datacenter switches use Buffer Management (BM) and Active Queue Management (AQM) to control packet admission and buffer allocation. BM schemes like Complete Sharing (CS) [7] and Dynamic Threshold (DT) [8] improve fairness across output queues and reduce packet loss probability. AQM algorithms like RED [9], CoDel [10], and PIE [11] monitor queue lengths and dynamically adjust sizes based on congestion, dropping packets early to prevent buildup. However, the lack of coordination between BM and AQM schemes creates challenges, such as starvation and inefficiency.

Recent research, such as ABM [2], suggests integrating BM and AQM schemes to address these issues. However, this integration introduces complexities, particularly in calculating drain rates for each queue, which can be prone to estimation errors, especially when available capacity fluctuates or links are shared [11, 12]. For example, ABM calculates drain rates every 30ms, which may not accurately reflect network traffic dynamics, leading to potential inaccuracies. Furthermore, the drain rate inherently depends on the scheduling algorithm, making it challenging to implement a general-purpose technique in practice. For instance, FB [13] estimates the drain rate as the inverse of the number of queues using the port's bandwidth to approximate round-robin scheduling. However, similar techniques do not generalize to other scheduling algorithms, such as weighted round robin or strict-priority scheduling, making them less versatile.

In this paper, we present CBM, a simple and efficient buffer-sharing mechanism for datacenter switches based on packet delay — inspired by Codel [10] — making it dequeue rate-agnostic. CBM optimizes buffer space usage like CS[7] and drops packets either when the buffer is full or when they exceed a dynamically calculated delay target. CBM performs comparably to ABM and outperforms it by up to 45.07% in terms of Flow Completion Times (FCTs) with advanced transport protocols like PowerTCP[14].

2 System Design

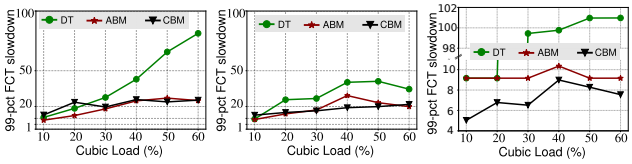
We aim to create a single buffer-sharing algorithm that addresses the limitations of existing methods while incorporating the following desired features: reduced dependency on scheduling algorithms, simple computation, and minimal buffer wastage. The CBM algorithm combines simplicity from CoDel [10] and efficient use of buffer space from CS [7] to calculate the threshold value for packet admission. In an output-queued shared-memory packet switching chip, CBM assigns a threshold $\Theta^i_p(t)$ for dequeuing a packet with priority p and port i for any particular instance of time t. $\Theta^i_p(t)$ calculates the target delay threshold using a configurable value α_p, the port's bandwidth b, and two dynamically changing factors: 1) the number of congested queues of priority p (n_p) and 2) the remaining buffer space $B - Q(t)$ as follows.

$$\Theta^i_p(t) = \frac{\alpha_p * (B - Q(t))}{n_p * b}, \tag{1}$$

where, α_p is the only parameter requiring configuration by the operator. Similar to DT [8], a higher α_p value in CBM leads to a

CoNEXT-SW '24, December 9–12, 2024, Los Angeles, CA, USA

ACM ISBN 979-8-4007-1255-5/24/12
https://doi.org/10.1145/3694812.3699924



(a) `Cubic Flows` (b) `DCTCP Flows` (c) θ-PowerTCP

Figure 1: The behavior of Cubic, DCTCP, and θ-PowerTCP protocol when using different priority queues with the BM algorithm, i.e., DT.

higher likelihood of packet acceptance in a queue. B represents the available buffer per port. $B - Q(t)$ represents the remaining buffer space at time t. n_p denotes the number of congested queues of priority p. We consider a queue to be congested if its length equals or exceeds 90% of its corresponding threshold. b represents the bandwidth per port. We now detail the inner workings of the CBM algorithm, understanding how packets traverse through different stages.

CBM decides whether to dequeue and transmit or to drop and continue to the next packet at the head of the queue. To make such a decision, two critical metrics are computed: 1) sojourn time (representing queuing delay) and 2) target delay. Sojourn time denotes the duration a packet spends in the queue after being enqueued, while target delay indicates the threshold of packet time in the system, which is determined using Eq. 1.

Target delay serves as a guideline for decision-making within the algorithm. If a packet's sojourn time is below the target delay, it is forwarded for transmission. However, if the sojourn time exceeds the target delay, the packet is rejected and dropped, indicating prolonged presence in the network device. Bursty traffic in datacenters primarily arises from unscheduled packets within the first Round-Trip Time (RTT) of a flow since congestion control mechanisms cannot effectively respond within this short timeframe. We assign a higher priority (higher α) to the first RTT packets, reducing the likelihood of dropping them from the buffer, even during bursts like large-scale incasts.

3 Performance Evaluation

We evaluate CBM's performance using NS3 simulations, under realistic workloads. We compare CBM with state-of-the-art buffer management algorithms, ABM and DT. We setup a Leaf-Spine topology with eight spine switches, eight leaf switches, and 256 servers. Each link has 10Gbps capacity and a 4 : 1 oversubscription ratio, similar to the setup in [2], with a 10μs propagation delay. Switches have 9.6KB of buffer space per port per Gbps, matching Broadcom TridentII switch capabilities [15]. We compare the performance of each buffer management algorithm under loss-based (Cubic), ECN-based (DCTCP), and delay-based (θ-PowerTCP) transport protocols. We generate websearch workload superimposed with a synthetic incast workload similar to prior works.

Figure 1 reveals that as Cubic load increases, CBM's FCT slowdown performance improves significantly. At a 50% load, CBM reduces FCT slowdown by 11.80% compared to ABM and by 63.72% compared to DT. Moreover, CBM outperforms ABM and DT in θ-PowerTCP, reducing FCT slowdown by 45.07% at a 10% load and by 28.89% at a 30% load compared to ABM and 93.44% compared to DT at the same load. Additionally, for DCTCP, CBM reduces FCT slowdown by up to 34.73% compared to ABM and by 52.79% compared to DT, showcasing its effectiveness across various priorities in mitigating FCT slowdown.

Outcome: In distinct priority scenarios, CBM occasionally outperforms both ABM and DT, particularly in the case of θ-PowerTCP, where it surpasses both algorithms across all load conditions.

4 Conclusion

In this paper, we introduced CBM, a novel switch buffer-sharing scheme inspired by the packet delay concept used in CoDel. CBM is specifically designed to address the challenges of sharing on-chip buffers across queues in network devices. With only one tunable parameter, CBM outperforms most existing buffer-sharing algorithms in reducing FCTs. Our experiments, conducted with advanced congestion control mechanisms like PowerTCP, highlight CBM's effectiveness in improving network performance.

As a next step, we envision implementing CBM on programmable switches, extending CBM to handle PFC thresholds, and further exploring its generalization to multiple priorities and traffic classes.

References

[1] Ehab Ghabashneh, Yimeng Zhao, Cristian Lumezanu, Neil Spring, Srikanth Sundaresan, and Sanjay Rao. A microscopic view of bursts, buffer contention, and loss in data centers. In *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, page 567–580, 2022.

[2] Vamsi Addanki, Maria Apostolaki, Manya Ghobadi, Stefan Schmid, and Laurent Vanbever. Abm: active buffer management in datacenters. In *Proceedings of the ACM SIGCOMM 2022 Conference*, SIGCOMM '22, page 36–52, 2022.

[3] Vamsi Addanki, Maciej Pacut, and Stefan Schmid. Credence: Augmenting datacenter switch buffer sharing with ml predictions. In *21st USENIX symposium on networked systems design and implementation (NSDI 24)*, 2024.

[4] Vamsi Addanki, Wei Bai, Stefan Schmid, and Maria Apostolaki. Reverie: Low pass Filter-Based switch buffer sharing for datacenters with RDMA and TCP traffic. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pages 651–668, Santa Clara, CA, April 2024. USENIX Association.

[5] Sijiang Huang, Mowei Wang, and Yong Cui. Traffic-aware buffer management in shared memory switches. *IEEE/ACM Transactions on Networking*, 30(6):2559–2573, 2022.

[6] Hamidreza Almasi, Rohan Vardekar, and Balajee Vamanan. Protean: Adaptive management of shared-memory in datacenter switches. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, pages 1–10, 2023.

[7] William Aiello, Alex Kesselman, and Yishay Mansour. Competitive buffer management for shared-memory switches. *ACM Trans. Algorithms*, 5(1), dec 2008.

[8] A.K. Choudhury and E.L. Hahne. Dynamic queue length thresholds for shared-memory packet switches. *IEEE/ACM Transactions on Networking*, 6(2):130–140, 1998.

[9] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.

[10] Kathleen Nichols and Van Jacobson. Controlling queue delay: A modern aqm is just one piece of the solution to bufferbloat. *Queue*, 10(5):20–34, 2012.

[11] Rong Pan, Preethi Natarajan, Chiara Piglione, Mythili Suryanarayana Prabhu, Vijay Subramanian, Fred Baker, and Bill VerSteeg. Pie: A lightweight control scheme to address the bufferbloat problem. In *2013 IEEE 14th International Conference on High Performance Switching and Routing (HPSR)*, pages 148–155, 2013.

[12] Habib Mostafaei and Georgios Smaragdakis. Per priority data rate measurement in data plane. In *Proceedings of the 6th on European P4 Workshop*, EuroP4 '23, page 9–15, 2023.

[13] Maria Apostolaki, Vamsi Addanki, Manya Ghobadi, and Laurent Vanbever. Fb: A flexible buffer management scheme for data center switches. *arXiv preprint arXiv:2105.10553*, 2021.

[14] Vamsi Addanki, Oliver Michel, and Stefan Schmid. {PowerTCP}: Pushing the performance limits of datacenter networks. In *19th USENIX symposium on networked systems design and implementation (NSDI 22)*, pages 51–70, 2022.

[15] Broadcom Inc. Trident2 / bcm56850 series. https://www.broadcom.com/products/ethernet-connectivity/switching/strataxgs/bcm56850-series. Accessed: 13 March 2024.