Towards Truly Burst-Aware Evaluation of Data Center Congestion Control

Pragna Mamidipaka
Carnegie Mellon University
Pittsburgh, PA, USA
pmamidip@andrew.cmu.edu

Srikanth Sundaresan
Meta
Menlo Park, USA
ssundaresan@meta.com

Theophilus A. Benson
Carnegie Mellon University
Pittsburgh, USA
theophilus@cmu.edu

Abstract

The performance of datacenter congestion control algorithms (CCAs) is highly sensitive to bursty traffic patterns, yet a significant fidelity gap exists between evaluation workloads and production traffic. Current evaluations primarily rely on synthetic workloads constructed from flow-size CDFs with incast overlaid on top, an approach that, while intuitive, we show produces traffic that is dissimilar to production in its temporal burst clustering. As a result, these workloads fail to exercise the full range of conditions that CCAs encounter in production, and hence, protocols that demonstrate gains in simulation risk diminished performance or unexpected failure modes upon deployment. This motivates the need for a deeper understanding of burstiness for CCA evaluation. To this end, we decompose burstiness into four key dimensions, and use DCTCP as a case study to show distinct behavioral regimes in each dimension. Building on this, we envision a burst-centric evaluation stack: behavioral regime analysis across various CCA classes, and a burst generator to ensure regime coverage along these dimensions, enabling thorough and robust evaluations.

CCS Concepts

• **Networks** → **Network performance evaluation**; **Transport protocols**; **Data center networks**.

Keywords

Congestion Control, Data Center Networks, Bursts, Incast

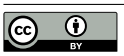
ACM Reference Format:
Pragna Mamidipaka, Srikanth Sundaresan, and Theophilus A. Benson. 2026. Towards Truly Burst-Aware Evaluation of Data Center Congestion Control. In *The 10th Asia-Pacific Workshop on Networking (APNet 2026), August 06–07, 2026, Singapore, Singapore.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3820441.3820468

1 Introduction

Datacenter networking research is highly dependent on synthetic workloads, as real production traffic is rarely shareable. The validity of any performance evaluation, therefore hinges on whether these synthetic workloads adequately cover the operational space. Of all traffic properties, burstiness is the most consequential and the hardest to replicate because it arises simultaneously from application-level patterns (incast) [31, 39], transport-level behavior [7], and host/network artifacts [5, 36]. This is especially critical for Congestion Control Algorithms (CCAs) which are fundamentally reactive, in that, they respond to congestion signals over time, making them uniquely sensitive to the burst structure of the traffic they encounter. CCAs that show strong performance in standard evaluations can exhibit unexpected failure modes under varied burst conditions. DCTCP, for instance, shows flow-level unfairness and divergence under cyclic incast [12], a failure mode undetected under standard evaluations. Crucially, it was shown [12] that this behavior arises from how bursts interact with the CCA state across time. More broadly, we believe that CCAs behave fundamentally differently under different burst structure, yet existing workloads are constructed without the awareness of which CCA regimes they exercise.

Prior work has recognized pieces of this problem in other context. For example, Encore [23] identifies that deriving workloads purely from flow size distributions loses temporal and spatial traffic structure. While this is an important result, one can argue that workloads that overlay an incast traffic pattern solve this problem, by explicitly injecting bursts on top of background traffic. However, we show that this is not enough if the incast parameters are chosen in an ad-hoc manner. As we show in Section 4.2, the temporal burst clustering of the most expressive workloads used in practice differs from that of production traffic.

Our central insight is that CCAs should be evaluated across all combinations of the interactions between their control loop and all dimensions of a burst. This more holistic approach to evaluation of a CCA provides the community with a workload-agnostic lens which allows for comparisons in a fair and open manner. In particular, we observe that a CCA's control loop can be impacted by four distinct dimensions of bursts: *intensity*, *duration*, *synchronicity*, and *inter-arrival time*. CCA behavior can therefore be understood through the lens of how each burst dimension stresses the protocol. As a case study, we characterize DCTCP, identifying various operating regimes along each burst dimension. Throughout, we focus on bursts at the ToR downlink to the receiver, since this is the point at which incast traffic converges and where CCA behavior is most directly stressed. Building on this, we envision three directions that together constitute a burst-centric evaluation stack. First, a *behavioral regime analysis* for various CCA classes, including switch feedback-based, delay-based, and receiver-driven protocols. Second, a *burst generator* that constructs workloads to ensure comprehensive regime coverage across these four dimensions, ensuring that any proposed CCA is rigorously evaluated across its entire operational space. Third, a *burst-centric comparator* that uses these regimes to compare CCAs, or a single CCA across different workloads, on the fair, workload-agnostic basis this approach is meant to enable.



APNet 2026, Singapore, Singapore

ACM ISBN 979-8-4007-2664-4/26/08
https://doi.org/10.1145/3820441.3820468

Our contributions in this paper are:

- We survey and create a taxonomy of current workloads and burst characteristics used to evaluate data center CCA proposals (Section 3).
- We assess and quantify the fidelity gap between a production workload and the state-of-the-art workloads used to evaluate data center CCA proposals (Section 4).
- We define a four-dimensional decomposition of burstiness grounded in CCA control mechanisms and demonstrate the efficacy of the burst dimensions using DCTCP as a case study (Section 5).
- We present a vision for our burst-centric evaluation stack, comprising a behavioral regime analysis, a burst generator, and a burst-centric comparator (Section 6).

2 Bursts in Data Center Networks

2.1 What is a Burst?

At the most fundamental level, a burst is a transient episode during which packets arrive at a network element faster than they can be drained, causing queue or link occupancy to spike. More concretely, we define a given time interval to be bursty if the link utilization for the measurement period, exceeds a given threshold value. Consistent with prior works [21, 44], we use a utilization of 50% as this threshold.

2.2 Implications of Bursts

Bursts are a challenge in both traditional [12, 17, 33, 44] and AI datacenters [19]. Traditional datacenters power web services, storage systems, and databases that users rely on daily. Bursts in these environments can arise from user queries, background jobs, or host networking stack optimizations. When congestion control techniques fail to handle these bursts, the consequences cascade into slow page loads, timeouts, and degraded user experience - all of which impact SLAs. While traditional datacenters still dominate, AI clusters are a fast growing minority, and these clusters face their own burst-related challenges. Even though packet loss is minimized or avoided altogether in RDMA networks, bursts lead to buffer pressure and latency spikes. Training workloads often generate highly synchronized traffic patterns through collective operations that occur thousands of times per hour. Stragglers can have a multiplicative effect: a single slow flow delays the entire training iteration, leaving expensive GPUs idle. Despite these unique characteristics, recent AI datacenter studies often adopt evaluation methodologies inherited from traditional datacenters, potentially overlooking the impact of burstiness on performance.

2.3 Incast - The Dominant Burst Pattern

Among the various sources of bursts in datacenter networks, incast stands out as the most studied and the most stressful for CCAs [12, 17, 31, 39]. Incast is a many-to-one traffic pattern where multiple senders simultaneously transmit to a single receiver, and is a natural consequence of partition-aggregate workflows. The result is a sharp, concentrated demand spike at the aggregator ToR (Top-of-Rack switch), that exhausts the buffer before any feedback loop can close. The ToR downlink to the receiver is the most operationally significant vantage point for a CCA, as it is where multiple flows converge onto a single last-hop link [17]. Therefore, in this paper we focus exclusively on bursts as observed at the ToR downlink to the receiver.

3 Burst Survey

3.1 DCN Evaluation Survey

To understand the state of evaluation practice, we surveyed datacenter networking papers from SIGCOMM, CoNEXT, NSDI, EuroSys, IMC, HotNets, and OSDI over the last six years. We focused on papers targeting techniques where burstiness is relevant. Speciifically, we focus on CCAs desdign, packet deflection, scheduling, and other CCA-compatible techniques like flow control, AQM etc. For our survey, we included only those papers where burst handling is either in scope of the technique or claimed as a contribution. We focus on the input workloads used for simulations or testbed evaluations in the above papers, and the observations from the survey are presented in Sec 3.2.

3.2 Workload Taxonomy

From our survey, we find that papers can be classified into these four broad categories based on the workloads used in their evaluations.

Canonical workloads are created by sampling flow sizes from an empirical CDF like FB Hadoop [34] or Websearch [8], and assuming Poisson arrivals. They randomly select source and destination hosts to simulate all-to-all traffic with load controlled by varying the Poisson rate. Burstiness is left entirely to chance: bursts may occur incidentally due to the Poisson process, but are not designed in. This is the least representative of all workload classes. 45% of the surveyed papers fall into this category. [6, 13, 14, 16, 22, 29, 35, 37, 38, 40, 42, 43, 47].

Background + Incast: workloads overlay a synthetic incast pattern on top of canonical background traffic. Incast is constructed by selecting a group of senders to simultaneously transmit to a common destination, with parameters such as number of senders and flow size configured manually. Since the parameters are fixed at a single operating point, the workload exercises only one location in the burst dimension space. 17.5% of the surveyed papers fall into this category. [11, 26, 27, 32, 46].

Background + Burst-aware Incast workloads extend the previous category by explicitly varying the incast parameters. This is the most expressive category in the literature, where burstiness is deliberately exercised across a parameter range. 17.5% of the surveyed papers fall into this category [1–3, 20, 28]. However, varying incast parameters in an ad-hoc manner offers no guarantee that the exercised range cover all possible regimes. Typically, the parameters adjusted include the number of incast flows, incast queries per second (QPS), and the size of each incast flow, although some evaluations only vary a subset of these three.

Fully synthetic workloads [15, 18, 24, 41, 45, 48] configure topologies and traffic patterns without reference to any published CDFs. They are common in microbenchmarks and industry papers, and such workloads are also useful for stress-testing specific scenarios. 20% of the surveyed papers fall into this category.

4 The Fidelity Gap

4.1 Millisampler: Our Production Ground Truth

To assess how the above workloads compare to production, we need a production ground truth. We use Meta's Millisampler [17] data, containing link utilization measurements at hosts, at a 1 *ms* time granularity. Deployed at scale across Meta's production datacenter fleet, it provides a representative view of real traffic patterns. Millisampler captures the fine-grained temporal structure of traffic arrival at the host in the form of aggregate byte counts per millisecond interval. This is the quantity we care about: high-utilization intervals, as observed at the ToR downlink to the receiver.

4.2 Do Burst Aware Workloads Reflect Production?

Given that burst-aware workloads are the most expressive category available, the natural question is whether they actually reproduce the structural properties of production traffic.

To answer this, we note the incast parameters or the ranges of incast parameters (number of incast flows, size of each incast flow, incast QPS (queries per second)) used in the burst-aware papers, and present them in Table 1. Some of the papers do not mention all the parameters, but we could derive them from other information given in the paper, such as the incast load. We use NS-3 to simulate a fattree topology with 128 hosts, with DCTCP as the CCA, and generate background traffic by sampling from various flow-size CDFs, and overlay incast traffic with parameters chosen from the ranges in Table 1. We measure packet arrival times at the receiver, bucket them into 1ms granularity, and calculate the link utilization, to allow direct comparison with Millisampler data. We choose 10 Millisampler hosts at random for this comparison. As a comparison metric, we use the r-value [44] of the link utilization time series, modeled as a markov chain. A higher r indicates stronger temporal clustering of high-utilization periods (bursts).

As shown in Fig. 1, Millisampler hosts yield a mean $r \approx 36$, while the simulated workloads yield a mean $r \approx 7.3$. Recall that $r = 1$ implies the probability of the next burst arriving is the same regardless of whether the previous period contained a burst; in other words, bursts are independent. From this figure, we find that the bursts in the simulated workloads are substantially less correlated than those in Millisampler. In Tables 3 and 2, we analyze the burst transition matrices for three distinct simulated workloads and at three randomly chosen hosts in Millisampler. In Millisampler, bursts are rare (a low $x_{t-1} = 0 \rightarrow x_t = 1$ transition probability) but tend to be long once they begin (a high $x_{t-1} = 1 \rightarrow x_t = 1$ transition probability). By contrast, bursts in the simulated workloads can occur more frequently but are probabilistically shorter. Together, these results expose a fidelity gap: the simulated workloads capture that bursts occur but not how they cluster in time, underrepresenting the sustained, less correlated bursts that dominate production traffic.

5 Burst Dimensions and DCTCP Regimes

To understand the impact of bursts on a CCA, we propose decomposing burstiness into four dimensions. For each dimension, regime boundaries are determined by the properties of a CCA's control mechanism. This yields distinct behavioral regimes per dimension. Crucially, not all dimensions are equally relevant to every CCA. INT based CCAs [25, 28, 41], for example, determine the right sending rates within 1 RTT, effectively reducing regimes along duration dimension. More capable CCAs therefore exhibit fewer regimes overall, as their mechanisms suppress the boundary conditions that would otherwise divide a dimension.

Paper Name	Number of Incast Flows	Individual Flow Size (KB)	Queries Per Second
PowerTCP [3]	10, 255	4-800	1-16
Floodgate [27]	4	45-60	745
Vertigo [1]	50-450	1-180	2000-28000
Backpressure Flow Control [20]	10-2000	10-2000	2000
Practical Packet Deflection [2]	50-450	1-180	2000-70000

Table 1: Incast traffic parameters from a set of papers that use the "Background + Burst-Aware" workload.

	Host 1		Host 2		Host 3	
$p(x_t \mid x_{t-1})$	$x_t = 0$	$x_t = 1$	$x_t = 0$	$x_t = 1$	$x_t = 0$	$x_t = 1$
$x_{t-1} = 0$	0.995	0.005	0.979	0.020	0.969	0.030
$x_{t-1} = 1$	0.727	0.272	0.29	0.709	0.259	0.740

Table 2: Transition matrix for Millisampler hosts with respective r-values = 56, 34, 24.

	Workload 1		Workload 2		Workload 3	
$p(x_t \mid x_{t-1})$	$x_t = 0$	$x_t = 1$	$x_t = 0$	$x_t = 1$	$x_t = 0$	$x_t = 1$
$x_{t-1} = 0$	0.993	0.006	0.916	0.083	0.457	0.542
$x_{t-1} = 1$	0.684	0.315	0.376	0.623	0.371	0.629

Table 3: Transition matrix for burst aware workloads with respective r-values = 48, 7.44, 1.15.

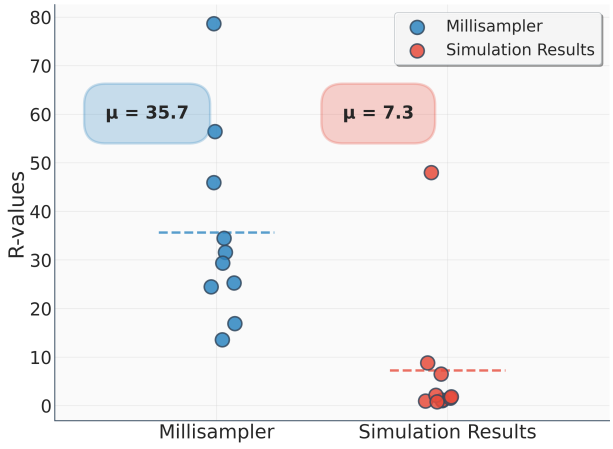


Figure 1: Millisampler hosts exhibit higher r values.

We use DCTCP [7] as a case study, identifying its operating regime boundaries along several dimensions. We note that this analysis is a motivating case study; extending it to INT-based, delay-based and receiver-based CCA classes is a central part of our research vision. We present an outline to do so in Sec 6.1.

5.1 Intensity

Intensity captures the data injected into the network in a single RTT due to a burst. Intensity at burst onset is given by $N \times cwnd_{\text{init}}$, where N is the number of incast flows. Intensity regime boundaries for DCTCP are K (the ECN marking threshold) and B (the buffer capacity allocated to the bottleneck queue). Healthy regime is when the intensity is less than K. On the other hand, if initial injection overshoots K, ECN is triggered, and DCTCP reacts by reducing $cwnd$. In the extreme case where $N \times cwnd_{\text{min}} \geq K$, even full DCTCP reaction cannot bring the queue below K, producing a standing queue above K for the duration of the burst. This characterizes the medium regime. Lastly, bad regime arises when $I \geq B$, i.e., the aggregate injection exceeds available buffer capacity. Packet drops are inevitable, affecting both the background flows and the burst itself, causing retransmissions and elevated flow completion times (FCTs).

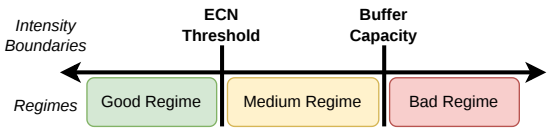


Figure 2: Regime Boundaries of Intensity Dimension.

5.2 Duration

Duration measures how long the burst lasts, denoted by T_{burst} (Burst Completion Time). RTT and T_C(convergence time of DCTCP) are the two boundary conditions in this dimension.

When $T_{burst} > T_C$, DCTCP converges within the burst and the queue settles around K, staying in the healthy regime. When $RTT < T_{burst} < T_C$ (medium regime), DCTCP receives ECN marks, and starts reducing sending rates, but is not able to converge before the burst completes. Repeated bursts of this nature lead to higher FCTs for short flows sharing the queue, since they have to wait at the back of the queue. When $T_{burst} \leq RTT$, DCTCP is unable to react: the burst is either absorbed or dropped before any signal can return to senders. Such short episodes can also adversely affect sending rates of other flows in the network, causing them to lose throughput.

Duration determines how many iterations of DCTCP's control loop can execute within a burst - if it is too few, α still has pre-burst memory, and $cwnd$ does not have time to fully adjust. Notably, approximately 60% of production bursts are 2ms or shorter [12]; at high flow counts, T_C exceeds this, making the medium regime common in practice.

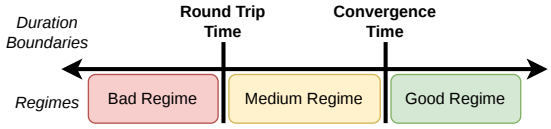


Figure 3: Regime Boundaries of Duration Dimension.

5.3 Synchronicity

Synchronicity refers to the phase alignment of incast senders at the bottleneck link. It can develop due to various factors, like initial jitter J, which leads to a spread of arrival times from the senders. High synchronicity would cause the entire burst to arrive at the queue before any feedback can return, collapsing DCTCP's proportional feedback to a binary signal - nothing is marked before the burst, but a lot of flows will be marked instantaneously. Staggered arrivals allow early ECN marks to return to senders before late senders have committed their full initial window, enabling the queue to build gradually and DCTCP to converge smoothly. However, low synchronicity also increases the possibility of flow disparity, leading to out-of-phase $cwnd$ oscillations among the incast flows. This leads to the problem of divergence at burst boundaries as shown in [12].

5.4 Inter-Arrival Time

Inter-arrival time (IAT) is the temporal gap between successive burst episodes. A higher value of incast QPS (queries per second) gives a lower IAT and vice-versa. A high IAT creates a highly transient traffic pattern with isolated spikes, necessitating faster reaction times for CCAs. Whereas, a lower IAT value can mean that multiple incasts co-exist simultaneously in the network, leading to persistent congestion. Furthermore, it is interesting to examine whether DCTCP reacts differently depending on the interaction phase of two overlapping incasts, for e.g., a second incast starting when the first incast is in steady state vs in transient state.

5.5 Interactions between Dimensions

These burst dimensions can also combine and compound the problematic cases. For example, Intensity and Duration decide the IAT boundaries: at high scale (N), the time required for all flows to complete is longer, so the IAT needed to avoid persistent congestion increases. This is an advantage of decomposing bursts along the dimensions, as it systematically allows us to analyze the possible behaviors.

6 Vision: A Burst Centric Evaluation Stack

Current state-of-the-art evaluation workloads attempt to test CCAs using a protocol-agnostic approach, i.e., workloads are constructed in a similar fashion irrespective of the CCA being tested. However, because a protocol's operating regimes and boundaries are dictated by either its inherent limitations, or mechanisms, evaluation workloads achieve better coverage when explicitly tailored to the CCA being tested.

To solve this, our CCA regime analysis decomposes bursts into four dimensions, with each dimension having certain boundaries

that divide its axis into discrete intervals. We can hence visualize the multi-dimensional burst space partitioned into a finite set of discrete "boxes", where each box corresponds to a behavioral regime of that CCA. The evaluation task is then simplified to pick atleast one workload per box, thus ensuring the workloads are dictated by the properties of the protocol being evaluated.

6.1 Behavioral Regime Analysis

The DCTCP case study demonstrates the existence of boundary conditions that affect protocol behavior. Extending such regime analysis to other CCA classes introduces a challenge: different classes have fundamentally different control mechanisms. The intuition for tackling this is that while the control variables differ across classes, the four burst dimensions remain the same axes of analysis. The analytical task is identifying, for each class, which internal variables are driven by each dimension, and finding the corresponding boundaries. Here, we lay an outline for what the intensity and duration boundaries are for each of the CCA classes.

Delay-Based CCAs - For delay based protocols, the intensity boundaries are dictated by target delay - which acts analogously to an ECN threshold, since both are intended operating points the protocol tries to maintain. The queue capacity still remains the boundary above which loss occurs. Along the duration dimension, the boundaries are the RTT and the protocol's convergence time.

INT-Based CCAs - Protocols leveraging In-Network Telemetry utilize explicit switch or link feedback. Here, the target link utilization acts analogously to an ECN threshold. Because new flows typically begin transmitting at full line rate, there is a chance that initial sending window at burst onset might lead to loss. The duration dimension only has one boundary - the RTT, as INT literature suggests convergence happens within one RTT, so the traditional algorithm convergence time boundary is now just one RTT.

Receiver-Based CCAs - Receiver driven CCAs manage congestion by issuing explicit grants or credits to govern the rate. Every host gets an initial allowance of unscheduled bytes to prevent starvation, and if the volume of the unscheduled bytes from concurrent senders exceeds the buffer, it leads to packet loss. Here as well, the multi-RTT convergence phase is eliminated as the receiver dictates the exact rate and priority allocation via explicit grant/token scheduling.

6.2 Burst Generator

As we see from the DCTCP case study, regime boundaries are a function of incast parameters, CCA configurations (e.g., $cwnd_{\min}$, $cwnd_{init}$), CCA properties (T_c), network configurations (K), network properties (B, RTT), and inter-dimensional dependencies (e.g., IAT regimes depend on intensity and duration). Plugging the CCA, network configuration, and property values into the behavioral expressions (Section 6.1) defines the protocol's boundaries purely in terms of the incast parameters. However, obtaining these values to be plugged in is not very straightforward. Some are directly accessible as configuration parameters. Others, such as T_c and RTT, are emergent properties that depend on the CCA, which require some analytical approximations. Once these boundaries are established, the final step in the burst generator is to map these dimension-based boxes back to concrete evaluation workloads.

The goal of the burst generator is to select specific subsets of incast parameters—number of senders (N), flow size (X), and queries per second (QPS) that guarantee at least one evaluation data point is placed into every discrete behavioral regime. Given that every regime represents a behavioral state for the CCA, if we obtain a set of incast parameter configurations for every regime, we would robustly exercise the CCA's entire operational space.

6.3 Burst-Centric Comparator

The ability to determine the incast parameters that stress each of a CCA's regimes across all burst dimensions provides a unique lens through which to compare CCAs, or to compare a single CCA across different workloads. In particular, this regime-centric approach lets us evaluate CCAs more thoroughly, stressing aspects of their control loop that arbitrary workloads do not always exercise. Throughout, we use flow completion time (FCT) as the performance metric. More broadly, the comparator is the central piece that reframes how the community evaluates congestion control: by anchoring comparison to each protocol's own behavioral regimes rather than to a fixed, protocol-agnostic workload, it turns CCA evaluation into a principled, behavior-driven exercise rather than an artifact of the particular traffic chosen. Below we highlight two use-cases for the comparator:

To compare two independent CCAs, CCA_1 and CCA_2, we extract their regimes (Section 6.1) and then generate bursts that stress the union of their regimes (Section 6.2). If CCA_1 achieves a lower FCT than CCA_2 across the union of regimes, then CCA_1 strictly dominates and is the more performant protocol. This criterion is the analogue of first-order stochastic dominance, which requires one protocol to be no worse than the other in every regime. In practice, however, strict dominance is rare: each CCA typically attains a lower FCT in some regimes and a higher FCT in others, yielding only a partial order. Extending the comparator to this partial-dominance setting is an area for future work, in which we plan to explore weaker variants of stochastic dominance, e.g., second-order or almost stochastic dominance, that can still induce a meaningful ordering when per-regime FCT outcomes cross, for instance by weighting regimes according to their prevalence in a target deployment.

To compare workloads, we examine both the distribution of data points within a CCA's regimes and the coverage that the different workloads achieve across those regimes. Specifically, to assess how predictive a CCA's evaluation results are of its behavior in a given production environment, we compare the distribution of data points across regimes for a synthetic workload W_1 and a production workload W_2; the more closely W_1's distribution matches that of W_2, the more representative the synthetic evaluation is of production performance.

7 Related Works

Prior work relevant to our agenda falls into two threads: workload realism, and analytical CCA modeling.

On **workload realism**, the work most closely related to ours in terms of motivation is Encore [23], which demonstrates that flows in production are neither temporally independent nor homogeneous across hosts, and hence argues against using flow-size CDFs to

generate workloads. We go a step further: showing how even the workloads that overlay incast on top of such background flows are insufficient, because the parameters governing the overlay must be chosen with awareness of the coverage they provide. Work on generating representative network traces [10] raises similar concerns as Encore [23], but focuses on WAN settings, where traffic is inherently different: RTTs are orders of magnitude larger, shallow buffers and the synchronized fan-in patterns consequential for datacenter CCAs are absent. Another work [30] uses active learning techniques to automate the performance assessment of internet CCAs. However, they treat the CCAs as a blackbox, and hence lack explainable failure boundaries, which are insightful for CCA development.

On **analytical CCA modeling**, the DCTCP analysis [7, 9] provides the foundation we build upon, deriving DCTCP's operating regimes. More broadly, the Contracts framework [4] offers a unified lens on CCA steady-state behavior, characterizing tradeoffs between robustness, fairness, congestion, and generality across a wide class of protocols. However, both of these works characterize CCAs at steady state, under sustained, well-formed traffic, and they do not address how the burst structure specifically drives protocols into distinct operating regimes.

Our work bridges these two threads to enable robust CCA evaluation under burstiness. By deriving the regime boundaries of a CCA along each burst dimension, we break down the operational search space, eliminating the need for exhaustive parameter sweeps otherwise required.

8 Conclusion

Bursty traffic is a defining characteristic of modern data center networks, yet existing evaluation workloads fail to capture its multidimensional nature along with the correct ranges. Through our analysis and case study of DCTCP, we demonstrate that workload burst characteristics decide the effective operating point of a CCA, highlighting the limitations of current evaluation practices. To address this gap, we outline a burst-centric evaluation stack that systematically characterizes, generates, and compares CCA-aware evaluation workloads, hence enabling robust congestion control evaluation for data center networks.

Acknowledgments

We thank Christopher Canel, Santiago Vargas, and the anonymous reviewers for their constructive and insightful comments.

References

[1] Sepehr Abdous, Erfan Sharafzadeh, and Soudeh Ghorbani. 2021. Burst-tolerant datacenter networks with Vertigo. In *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies* (Virtual Event, Germany) *(CoNEXT '21)*. Association for Computing Machinery, New York, NY, USA, 1–15. doi:10.1145/3485983.3494873

[2] Sepehr Abdous, Erfan Sharafzadeh, and Soudeh Ghorbani. 2023. Practical Packet Deflection in Datacenters. *Proc. ACM Netw.* 1, CoNEXT3, Article 25 (Nov. 2023), 25 pages. doi:10.1145/3629147

[3] Vamsi Addanki, Oliver Michel, and Stefan Schmid. 2022. PowerTCP: Pushing the Performance Limits of Datacenter Networks. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*. USENIX Association, Renton, WA, 51–70. https://www.usenix.org/conference/nsdi22/presentation/addanki

[4] Anup Agarwal, Venkat Arun, and Srinivasan Seshan. 2026. Contracts: A Unified Lens on Congestion Control Robustness, Fairness, Congestion, and Generality. In *Proceedings of New Ideas in Networked Systems (NINES)*.

[5] Saksham Agarwal, Arvind Krishnamurthy, and Rachit Agarwal. 2023. Host Congestion Control. In *Proceedings of the ACM SIGCOMM 2023 Conference* (New York, NY, USA) *(ACM SIGCOMM '23)*. Association for Computing Machinery, New York, NY, USA, 275–287. doi:10.1145/3603269.3604878

[6] Albert Gran Alcoz, Alexander Dietmüller, and Laurent Vanbever. 2020. SP-PIFO: Approximating Push-In First-Out Behaviors using Strict-Priority Queues. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. USENIX Association, Santa Clara, CA, 59–76. https://www.usenix.org/conference/nsdi20/presentation/alcoz

[7] Mohammad Alizadeh, Albert Greenberg, David A. Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. 2010. Data center TCP (DCTCP). In *Proceedings of the ACM SIGCOMM 2010 Conference* (New Delhi, India) *(SIGCOMM '10)*. Association for Computing Machinery, New York, NY, USA, 63–74. doi:10.1145/1851182.1851192

[8] Mohammad Alizadeh, Albert Greenberg, David A. Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. 2010. Data center TCP (DCTCP). *SIGCOMM Comput. Commun. Rev.* 40, 4 (Aug. 2010), 63–74. doi:10.1145/1851275.1851192

[9] Mohammad Alizadeh, Adel Javanmard, and Balaji Prabhakar. 2011. Analysis of DCTCP: stability, convergence, and fairness. In *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems* (San Jose, California, USA) *(SIGMETRICS '11)*. Association for Computing Machinery, New York, NY, USA, 73–84. doi:10.1145/1993744.1993753

[10] Tobias Bühler, Roland Schmid, Sandro Lutz, and Laurent Vanbever. 2022. Generating representative, live network traffic out of millions of code repositories. In *Proceedings of the 21st ACM Workshop on Hot Topics in Networks* (Austin, Texas) *(HotNets '22)*. Association for Computing Machinery, New York, NY, USA, 1–7. doi:10.1145/3563766.3564084

[11] Qizhe Cai, Mina Tahmasbi Arashloo, and Rachit Agarwal. 2022. dcPIM: near-optimal proactive datacenter transport. In *Proceedings of the ACM SIGCOMM 2022 Conference* (Amsterdam, Netherlands) *(SIGCOMM '22)*. Association for Computing Machinery, New York, NY, USA, 53–65. doi:10.1145/3544216.3544235

[12] Christopher Canel, Balasubramanian Madhavan, Srikanth Sundaresan, Neil Spring, Prashanth Kannan, Ying Zhang, Kevin Lin, and Srinivasan Seshan. 2024. Understanding Incast Bursts in Modern Datacenters. In *Proceedings of the 2024 ACM on Internet Measurement Conference* (Madrid, Spain) *(IMC '24)*. Association for Computing Machinery, New York, NY, USA, 674–680. doi:10.1145/3646547.3689028

[13] Shawn Shuoshuo Chen, Keqiang He, Rui Wang, Srinivasan Seshan, and Peter Steenkiste. 2024. Precise Data Center Traffic Engineering with Constrained Hardware Resources. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. USENIX Association, Santa Clara, CA, 669–690. https://www.usenix.org/conference/nsdi24/presentation/chen-shawn

[14] Wenxue Cheng, Kun Qian, Wanchun Jiang, Tong Zhang, and Fengyuan Ren. 2020. Re-architecting Congestion Management in Lossless Ethernet. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. USENIX Association, Santa Clara, CA, 19–36. https://www.usenix.org/conference/nsdi20/presentation/cheng

[15] Inho Cho, Ahmed Saeed, Joshua Fried, Seo Jin Park, Mohammad Alizadeh, and Adam Belay. 2020. Overload Control for µs-scale RPCs with Breakwater. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*. USENIX Association, 299–314. https://www.usenix.org/conference/osdi20/presentation/cho

[16] Xinle Du, Tong Li, Guangmeng Zhou, Zhuotao Liu, Hanlin Huang, Xiangyu Gao, Mowei Wang, Kun Tan, and Ke Xu. 2025. PRED: Performance-oriented Random Early Detection for Consistently Stable Performance in Datacenters. In *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*. USENIX Association, Philadelphia, PA, 1–20. https://www.usenix.org/conference/nsdi25/presentation/du

[17] Ehab Ghabashneh, Yimeng Zhao, Cristian Lumezanu, Neil Spring, Srikanth Sundaresan, and Sanjay Rao. 2022. A microscopic view of bursts, buffer contention, and loss in data centers. In *Proceedings of the 22nd ACM Internet Measurement Conference* (Nice, France) *(IMC '22)*. Association for Computing Machinery, New York, NY, USA, 567–580. doi:10.1145/3517745.3561430

[18] Hamid Ghasemirahni, Alireza Farshin, Mariano Scazzariello, Gerald Q. Maguire, Dejan Kostić, and Marco Chiesa. 2024. FAJITA: Stateful Packet Processing at 100 Million pps. *Proc. ACM Netw.* 2, CoNEXT3, Article 14 (Aug. 2024), 22 pages. doi:10.1145/3676861

[19] Soudeh Ghorbani, Yimeng Zhao, Srikanth Sundaresan, Ying Zhang, Yijing Zeng, Abhigyan Sharma, Prashanth Kannan, and Cristian Lumezanu. 2025. Congestion Patterns in a Large-scale RDMA Datacenter. In *Proceedings of the 2025 ACM Internet Measurement Conference* (USA) *(IMC '25)*. Association for Computing Machinery, New York, NY, USA, 944–951. doi:10.1145/3730567.3764494

[20] Prateesh Goyal, Preey Shah, Kevin Zhao, Georgios Nikolaidis, Mohammad Alizadeh, and Thomas E. Anderson. 2022. Backpressure Flow Control. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*. USENIX Association, Renton, WA, 779–805. https://www.usenix.org/conference/nsdi22/presentation/goyal

[21] Daniel Halperin, Srikanth Kandula, Jitendra Padhye, Paramvir Bahl, and David Wetherall. 2011. Augmenting data center networks with multi-gigabit wireless links. *SIGCOMM Comput. Commun. Rev.* 41, 4 (Aug. 2011), 38–49. doi:10.1145/2043164.2018442

[22] Shuihai Hu, Wei Bai, Gaoxiong Zeng, Zilong Wang, Baochen Qiao, Kai Chen, Kun Tan, and Yi Wang. 2020. Aeolus: A Building Block for Proactive Transport in Datacenters. In *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication* (Virtual Event, USA) *(SIGCOMM '20)*. Association for Computing Machinery, New York, NY, USA, 422–434. doi:10.1145/3387514.3405878

[23] Sijiang Huang, Lingfeng Peng, Mowei Wang, Yashe Liu, Zhenhua Liu, Xin Wang, and Yong Cui. 2023. Datacenter Network Deserves Better Traffic Models. In *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks* (Cambridge, MA, USA) *(HotNets '23)*. Association for Computing Machinery, New York, NY, USA, 124–130. doi:10.1145/3626111.3628209

[24] Gautam Kumar, Nandita Dukkipati, Keon Jang, Hassan M. G. Wassel, Xian Wu, Behnam Montazeri, Yaogong Wang, Kevin Springborn, Christopher Alfeld, Michael Ryan, David Wetherall, and Amin Vahdat. 2020. Swift: Delay is Simple and Effective for Congestion Control in the Datacenter. In *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication* (Virtual Event, USA) *(SIGCOMM '20)*. Association for Computing Machinery, New York, NY, USA, 514–528. doi:10.1145/3387514.3406591

[25] Yuliang Li, Rui Miao, Hongqiang Harry Liu, Yan Zhuang, Fei Feng, Lingbo Tang, Zheng Cao, Ming Zhang, Frank Kelly, Mohammad Alizadeh, and Minlan Yu. 2019. HPCC: high precision congestion control. In *Proceedings of the ACM Special Interest Group on Data Communication* (Beijing, China) *(SIGCOMM '19)*. Association for Computing Machinery, New York, NY, USA, 44–58. doi:10.1145/3341302.3342085

[26] Hwijoon Lim, Jaehong Kim, Inho Cho, Keon Jang, Wei Bai, and Dongsu Han. 2023. FlexPass: A Case for Flexible Credit-based Transport for Datacenter Networks. In *Proceedings of the Eighteenth European Conference on Computer Systems* (Rome, Italy) *(EuroSys '23)*. Association for Computing Machinery, New York, NY, USA, 606–622. doi:10.1145/3552326.3587453

[27] Kexin Liu, Chen Tian, Qingyue Wang, Hao Zheng, Peiwen Yu, Wenhao Sun, Yonghui Xu, Ke Meng, Lei Han, Jie Fu, Wanchun Dou, and Guihai Chen. 2021. Floodgate: taming incast in datacenter networks. In *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies* (Virtual Event, Germany) *(CoNEXT '21)*. Association for Computing Machinery, New York, NY, USA, 30–44. doi:10.1145/3485983.3494854

[28] Kexin Liu, Zhaochen Zhang, Chang Liu, Yizhi Wang, Vamsi Addanki, Stefan Schmid, Qingyue Wang, Wei Chen, Xiaoliang Wang, Jiaqi Zheng, Wenhao Sun, Tao Wu, Ke Meng, Fei Chen, Weiguang Wang, Bingyang Liu, Wanchun Dou, Guihai Chen, and Chen Tian. 2025. Pyrrha: Congestion-Root-Based Flow Control to Eliminate Head-of-Line Blocking in Datacenter. In *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*. USENIX Association, Philadelphia, PA, 379–405. https://www.usenix.org/conference/nsdi25/presentation/liu-kexin

[29] Yuan Liu, Wenxin Li, Yulong Li, Lide Suo, Xuan Gao, Xin Xie, Sheng Chen, Ziqi Fan, Wenyu Qu, and Guyue Liu. 2025. Fork: A Dual Congestion Control Loop for Small and Large Flows in Datacenters. In *Proceedings of the Twentieth European Conference on Computer Systems* (Rotterdam, Netherlands) *(EuroSys '25)*. Association for Computing Machinery, New York, NY, USA, 446–459. doi:10.1145/3689031.3696101

[30] Parsa Pazhooheshy, Soheil Abbasloo, and Yashar Ganjali. 2025. Mahak: An Automated and Efficient Assessment Framework for Internet Control Algorithms . In *2025 IEEE 33rd International Conference on Network Protocols (ICNP)*. IEEE Computer Society, Los Alamitos, CA, USA, 1–13. doi:10.1109/ICNP65844.2025.11192397

[31] Amar Phanishayee, Elie Krevat, Vijay Vasudevan, David G. Andersen, Gregory R. Ganger, Garth A. Gibson, and Srinivasan Seshan. 2008. Measurement and analysis of TCP throughput collapse in cluster-based storage systems. In *Proceedings of the 6th USENIX Conference on File and Storage Technologies* (San Jose, California) *(FAST'08)*. USENIX Association, USA, Article 12, 14 pages.

[32] Konstantinos Prasopoulos, Ryan Kosta, Edouard Bugnion, and Marios Kogias. 2025. SIRD: A Sender-Informed, Receiver-Driven Datacenter Transport Protocol. In *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*. USENIX Association, Philadelphia, PA, 451–471. https://www.usenix.org/conference/nsdi25/presentation/prasopoulos

[33] Arjun Roy, Hongyi Zeng, Jasmeet Bagga, George Porter, and Alex C. Snoeren. 2015. Inside the Social Network's (Datacenter) Network. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication* (London, United Kingdom) *(SIGCOMM '15)*. Association for Computing Machinery, New York, NY, USA, 123–137. doi:10.1145/2785956.2787472

[34] Arjun Roy, Hongyi Zeng, Jasmeet Bagga, George Porter, and Alex C. Snoeren. 2015. Inside the Social Network's (Datacenter) Network. *SIGCOMM Comput. Commun. Rev.* 45, 4 (Aug. 2015), 123–137. doi:10.1145/2829988.2787472

[35] Ahmed Saeed, Varun Gupta, Prateesh Goyal, Milad Sharif, Rong Pan, Mostafa Ammar, Ellen Zegura, Keon Jang, Mohammad Alizadeh, Abdul Kabbani, and Amin Vahdat. 2020. Annulus: A Dual Congestion Control Loop for Datacenter and WAN Traffic Aggregates. In *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication* (Virtual Event, USA) *(SIGCOMM '20)*. Association for Computing Machinery, New York, NY, USA, 735–749. doi:10.1145/3387514.3405899

[36] Erfan Sharafzadeh, Sepehr Abdous, and Soudeh Ghorbani. 2023. Understanding the impact of host networking elements on traffic bursts. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. USENIX Association, Boston, MA, 237–254. https://www.usenix.org/conference/nsdi23/presentation/sharafzadeh

[37] Naveen Kr. Sharma, Chenxingyu Zhao, Ming Liu, Pravein G Kannan, Changhoon Kim, Arvind Krishnamurthy, and Anirudh Sivaraman. 2020. Programmable Calendar Queues for High-speed Packet Scheduling. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. USENIX Association, Santa Clara, CA, 685–699. https://www.usenix.org/conference/nsdi20/presentation/sharma

[38] Lide Suo, Yiren Pang, Wenxin Li, Renjie Pei, Keqiu Li, Xiulong Liu, Xin He, Yitao Hu, and Guyue Liu. 2024. PPT: A Pragmatic Transport for Datacenters. In *Proceedings of the ACM SIGCOMM 2024 Conference* (Sydney, NSW, Australia) *(ACM SIGCOMM '24)*. Association for Computing Machinery, New York, NY, USA, 954–969. doi:10.1145/3651890.3672235

[39] Vijay Vasudevan, Amar Phanishayee, Hiral Shah, Elie Krevat, David G. Andersen, Gregory R. Ganger, Garth A. Gibson, and Brian Mueller. 2009. Safe and effective fine-grained TCP retransmissions for datacenter communication. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication* (Barcelona, Spain) *(SIGCOMM '09)*. Association for Computing Machinery, New York, NY, USA, 303–314. doi:10.1145/1592568.1592604

[40] Zirui Wan, Jiao Zhang, Haoran Wei, Zhuo Jiang, Xiaolong Zhong, Wenfei Wu, Huaping Zhou, Tian Pan, and Tao Huang. 2024. RECC: Joint Congestion Control Based on RTT and ECN for High-speed RDMA Networks. *Proc. ACM Netw.* 2, CoNEXT4, Article 31 (Nov. 2024), 18 pages. doi:10.1145/3696402

[41] Weitao Wang, Masoud Moshref, Yuliang Li, Gautam Kumar, T. S. Eugene Ng, Neal Cardwell, and Nandita Dukkipati. 2023. Poseidon: An Efficient Congestion Control using Deployable INT for Data Center Networks. https://www.usenix.org/system/files/nsdi23-wang-weitao.pdf

[42] Xinyu Wu, Zhuang Wang, Weitao Wang, and T. S. Eugene Ng. 2023. Augmented Queue: A Scalable In-Network Abstraction for Data Center Network Sharing. In *Proceedings of the ACM SIGCOMM 2023 Conference* (New York, NY, USA) *(ACM SIGCOMM '23)*. Association for Computing Machinery, New York, NY, USA, 305–318. doi:10.1145/3603269.3604858

[43] Siyu Yan, Xiaoliang Wang, Xiaolong Zheng, Yinben Xia, Derui Liu, and Weishan Deng. 2021. ACC: automatic ECN tuning for high-speed datacenter networks. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference* (Virtual Event, USA) *(SIGCOMM '21)*. Association for Computing Machinery, New York, NY, USA, 384–397. doi:10.1145/3452296.3472927

[44] Qiao Zhang, Vincent Liu, Hongyi Zeng, and Arvind Krishnamurthy. 2017. High-resolution measurement of data center microbursts. In *Proceedings of the 2017 Internet Measurement Conference* (London, United Kingdom) *(IMC '17)*. Association for Computing Machinery, New York, NY, USA, 78–85. doi:10.1145/3131365.3131375

[45] Yiwen Zhang, Gautam Kumar, Nandita Dukkipati, Xian Wu, Priyaranjan Jha, Mosharaf Chowdhury, and Amin Vahdat. 2022. Aequitas: admission control for performance-critical RPCs in datacenters. In *Proceedings of the ACM SIGCOMM 2022 Conference* (Amsterdam, Netherlands) *(SIGCOMM '22)*. Association for Computing Machinery, New York, NY, USA, 1–18. doi:10.1145/3544216.3544271

[46] Yiran Zhang, Qingkai Meng, Chaolei Hu, and Fengyuan Ren. 2024. Revisiting Congestion Control for Lossless Ethernet. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. USENIX Association, Santa Clara, CA, 131–148. https://www.usenix.org/conference/nsdi24/presentation/zhang-yiran

[47] Zhiyu Zhang, Shili Chen, Ruyi Yao, Ruoshi Sun, Hao Mei, Hao Wang, Zixuan Chen, Gaojian Fang, Yibo Fan, Wanxin Shi, Sen Liu, and Yang Xu. 2024. vPIFO: Virtualized Packet Scheduler for Programmable Hierarchical Scheduling in High-Speed Networks. In *Proceedings of the ACM SIGCOMM 2024 Conference* (Sydney, NSW, Australia) *(ACM SIGCOMM '24)*. Association for Computing Machinery, New York, NY, USA, 983–999. doi:10.1145/3651890.3672270

[48] Zhaochen Zhang, Feiyang Xue, Keqiang He, Zhimeng Yin, Gianni Antichi, Jiaqi Gao, Yizhi Wang, Rui Ning, Haixin Nan, Xu Zhang, Peirui Cao, Xiaoliang Wang, Wanchun Dou, Guihai Chen, and Chen Tian. 2025. Enabling Virtual Priority in Data Center Congestion Control. In *Proceedings of the Twentieth European Conference on Computer Systems* (Rotterdam, Netherlands) *(EuroSys '25)*. Association for Computing Machinery, New York, NY, USA, 396–412. doi:10.1145/3689031.3717463