RiCE: Precise Remote In-Network Congestion Elimination in Inter-Datacenter RDMA Networks

Hongyi Li[♠], Chengyuan Huang[♠], Zhiqiang Li[♡], Guangyu Zhao[♡], Lu Lu[♡], Zirui Wan[◇], Jiaqing Dong[♣], Zhuo Tang[†], Guihai Chen[♠], Chen Tian[♠]

State Key Laboratory of Novel Software Technology, Nanjing University[♠]

China Mobile Research Institute[♡], *China Telecom Cloud Computing Research Institute*[◇]

Communication University of China[♣], *Hunan University*[†]

2026 IEEE/ACM International Symposium on Quality of Service (IWQoS) | 979-8-3195-1703-6/26/$31.00 ©2026 IEEE | DOI: 10.1109/IWQoS70441.2026.11661111

***Abstract*—High-performance distributed applications, such as Large Language Model (LLM) training and real-time database synchronization, increasingly span geo-distributed datacenters (DCs), demanding sustained inter-DC high throughput. While RDMA over Converged Ethernet (RoCEv2) offers high performance within local fabrics, extending its lossless abstraction to long-haul inter-DC networks is challenging. Propagating Priority Flow Control (PFC) across high-latency inter-DC links risks triggering widespread deadlocks and congestion spreading. Standard congestion control mechanisms (e.g., DCQCN) fail in this setting due to the excessive feedback delay imposed by the large Round-Trip Time (RTT).**

In this paper, we propose *RiCE*, a novel inter-DC congestion control framework designed to achieve sub-RTT reaction, scalable granularity, and robustness to feedback staleness, while remaining transparent to end-hosts. *RiCE* employs a split-loop architecture to decouple local and remote congestion handling. For local congestion, the sender's DCI gateway generates immediate feedback to throttle traffic before buffers overflow. For remote congestion, *RiCE* deploys a transparent proxy at the remote DCI gateway that enforces rate limiting on a per-path granularity. By exploiting the deterministic downward routing of Clos topologies to aggregate flows sharing the same physical bottleneck, *RiCE* emulates per-flow fairness without requiring prohibitive per-flow queues. Complementing this, a Remote Feedback Filter masks transient noise to prevent premature throttling. We implemented a DPDK-based prototype and conducted extensive evaluations on a physical testbed and large-scale simulations. Results show that *RiCE* achieves up to 3.3× the throughput of DCQCN and reduces flow completion time by up to 50% compared to state-of-the-art solutions.

***Index Terms*—RDMA, Congestion Control, Inter-Datacenter Networks**

I. Introduction

High-performance distributed applications increasingly span multiple geo-distributed datacenters (DCs) [1], [2]. Emerging workloads, such as the distributed training of Large Language Models (LLMs) [3], [4], real-time database synchronization, and large-scale disaster recovery, demand efficient data movement across inter-DC networks. Unlike intra-DC traffic, where microsecond latency is paramount, inter-DC communication prioritizes sustained high throughput to maximize the performance of applications and utilization of expensive long-haul links. Crucially, for practical deployment, it is imperative that the underlying transport mechanism remains transparent to end-hosts, requiring no modifications to hardware drivers or protocol stacks. Consequently, enabling high-performance, seamlessly deployable inter-DC communication has become a fundamental requirement for modern infrastructure.

To meet these stringent demands, RDMA over Converged Ethernet (RoCEv2) [5] is being extended from intra-DC fabrics to inter-DC networks. Unlike traditional TCP/IP, RoCEv2 bypasses the kernel to deliver high throughput with minimal CPU overhead. However, RoCEv2 relies heavily on a lossless network abstraction, typically enforced by Priority Flow Control (PFC) [6] alongside congestion control (e.g., DCQCN [7]), to prevent packet loss. While effective within a single DC, maintaining this lossless abstraction over long-haul links presents significant challenges. Specifically, propagating PFC pause frames across high-latency inter-DC links creates complex cyclic buffer dependencies, significantly elevating the risk of PFC deadlocks. Furthermore, the prolonged pause duration exacerbates congestion spreading (Head-of-Line blocking) and unfairness among flows. Therefore, minimizing PFC triggering over long distances is critical. This necessitates a robust Congestion Control (CC) mechanism capable of proactively regulating traffic to drain switch buffers before the lossless mechanism is invoked, playing a far more pivotal role in inter-DC scenarios than in local intra-DC networks.

Standard intra-DC congestion control mechanisms fail to adapt to this new setting. Native solutions like DCQCN [7] rely on end-to-end feedback loops. In inter-DC scenarios, the large Round-Trip Time (RTT)—often hundreds of times higher than intra-DC RTT—causes feedback to be stale, leading to slow convergence, severe throughput oscillation, and buffer overflows. Although recent inter-DC specific works (e.g., BiCC [8], ATC [9]) attempt to address this by using receiver-side proxies, they suffer from coarse-grained control (e.g., per-destination aggregation) or restrict multi-path routing (ECMP), which compromises fairness and link utilization.

Ideally, a robust inter-DC RDMA congestion control scheme should simultaneously satisfy four stringent design goals derived from these challenges: **(1) Sub-RTT Congestion Reaction** to decouple the control loop from the long inter-DC latency, enabling instantaneous response to local buffer buildups; **(2) Scalable Granularity** to ensure fine-grained fairness without the prohibitive overhead of per-flow queues; **(3) Robustness to Feedback Staleness** to prevent throughput

979-8-3195-1703-6/26/$31.00 ©2026 IEEE

collapse or oscillation caused by delayed remote signals; and **(4) End-Host Transparency** to allow deployment solely at Data Center Interconnect (DCI) gateways without requiring any modifications to RDMA network interfaces (NICs) or protocol stacks. Unfortunately, existing solutions force a trade-off among these objectives, unable to fulfill all of them in complex, dynamic environments.

In this paper, we propose *RiCE*, a novel congestion control framework designed to tackle all four challenges for inter-DC RDMA. *RiCE* fundamentally rethinks the control loop by decoupling the handling of local and remote congestion. It introduces a comprehensive split-loop architecture that isolates the sender from the high latency of the inter-DC link, ensuring that congestion is handled precisely where it occurs.

Specifically, *RiCE* employs a dual-pronged strategy. For local congestion (Figure 1), *RiCE* intercepts congestion signals at the sender's DCI gateway and generates immediate local feedback, throttling the sender before packets overwhelm local buffer. For the more challenging remote congestion, *RiCE* deploys a transparent proxy at the remote DCI gateway. This constitutes our primary contribution: the proxy enforces rate limiting on a per-path granularity. By exploiting the deterministic downward routing inherent in Clos topologies [10], *RiCE* aggregates all flows traversing the same physical bottleneck into a single control instance, effectively emulating per-flow fairness without requiring prohibitive per-flow queues. Complementing this, *RiCE* introduces a Remote Feedback Filter to distinguish transient noise from persistent congestion. By masking short-lived queue spikes at the remote proxy, this filter prevents premature source throttling, maintaining high link utilization even under bursty workloads.

We implemented a fully functional prototype of *RiCE* using DPDK and conducted extensive evaluations using both a physical testbed and large-scale NS-3 simulations. The results demonstrate that *RiCE* significantly outperforms state-of-the-art solutions. In testbed experiments with coexisting congestion, *RiCE* achieves up to 3.3× the throughput of the native DCQCN deployment. In large-scale simulations, *RiCE* reduces the Flow Completion Time (FCT) of inter-DC flows by up to 50% compared to ATC, while maintaining low latency for intra-DC traffic.

II. Background

A. Proliferation of Geo-distributed Applications

The evolution of cloud infrastructure has driven a fundamental transition from single-site to geo-distributed deployments. Driven by physical constraints—such as power envelopes and cooling capacities that limit the scale of single facilities—modern high-performance applications are increasingly partitioned across multiple datacenters. This shift has given rise to diverse, bandwidth-hungry workloads traversing the inter-DC network.

For example, (1) data reliability and consistency remain one of the primary drivers for data center interconnect (DCI). Cloud storage services and distributed databases rely heavily on inter-DC network for geo-redundancy and disaster recovery,

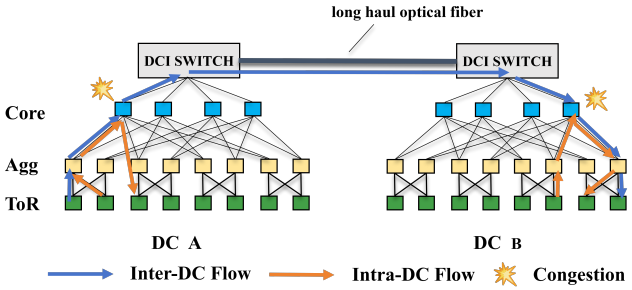


Fig. 1: Cross-datacenter Networks Architecture

generating continuous bulk data replication traffic to ensure high availability [11], [12]. (2) Collaborative computing has emerged as a critical workload. Applications ranging from ultra-scale model training [13], [14] to scientific computing require frequent data exchange between distributed clusters to aggregate computational resources.

B. Cross-Datacenter Networks

Large cloud service providers such as Amazon [1], Meta [2], and Google [3] have established robust cross-datacenter interconnects to support geo-distributed workloads. Typically, these networks employ a distinct "DCI Switch [1] + Long-haul Fiber" architecture, as illustrated in Figure 1. In this topology, dedicated DCI switches aggregate traffic from internal fabrics and forward it over long-haul optical fibers, spanning distances from tens to thousands of kilometers. A critical characteristic of this architecture is the significant propagation delay—approximately 5 μs per kilometer—which results in a Round-Trip Time (RTT) of roughly 1ms for a 100 km link [15]. This high bandwidth-delay product (BDP) environment poses unique challenges for flow control.

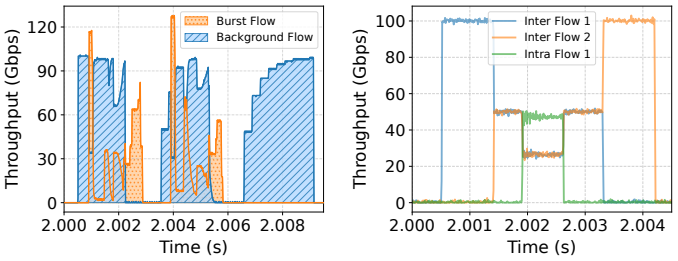
To satisfy stringent performance requirements, the industry is increasingly extending RoCEv2 (RDMA over Converged Ethernet [5]) from intra-DC to inter-DC scenarios [11]. This approach offers two major advantages: First, it inherits the high throughput and low CPU overhead of RDMA via kernel-bypass and zero-copy mechanisms. Second, it maintains protocol homogeneity with the internal network, allowing the use of commodity RDMA NICs (RNICs) without requiring hardware modifications. However, directly applying RoCEv2 to long-haul DCI is perilous. RoCEv2 relies on Priority-based Flow Control (PFC [6]) to ensure a lossless fabric. In high-RTT DCI environments, PFC frames take longer to propagate, exacerbating side effects such as buffer scarcity [16], Head-of-Line (HoL) blocking and deadlock risks [17]. Although end-to-end congestion control algorithms (e.g., DCQCN [7]) attempt to mitigate PFC triggering, their feedback loops are often too sluggish to match the rapid buffer dynamics of DCI switches. This limitation underscores the need for a precise, switch-assisted congestion elimination mechanism.

III. Motivation

A. Mismatch between Intra-DC CC and Inter-DC Latency

Existing intra-DC congestion control algorithms (e.g., DCQCN [7], HPCC [18] and Timely [19]) rely on low-latency

[1]Throughout this paper, we use DCI Switch and Gateway interchangeably.



(a) Bursts in DCQCN (b) Unfairness in ATC

Fig. 2: Throughput in the preliminary experiment

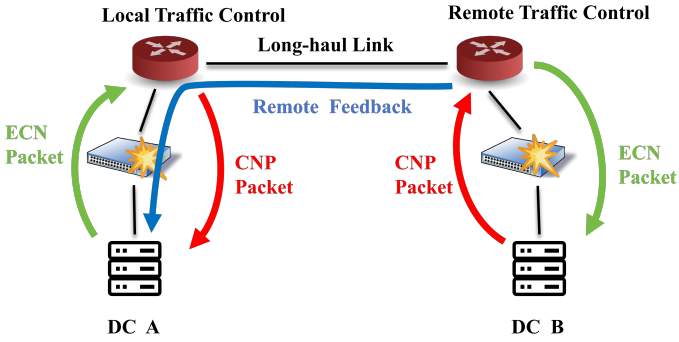


Fig. 3: Overview of *RiCE* Design

feedback loops to match sending rates with network capacity. In cross-datacenter scenarios, however, the significant propagation delay decouples the congestion signal from the instantaneous network state. This feedback lag causes senders to react to stale information, leading to either buffer overflows during congestion onset or prolonged throughput degradation after congestion clears. As shown in our preliminary experiment (Figure 2a), when a transient intra-DC burst competes with a long-lived inter-DC background flow, the inter-DC throughput collapses to near zero and fails to recover immediately. This confirms that end-to-end control is too slow to handle transient congestion effectively in high-RTT environments.

B. Pitfalls of Existing Inter-DC CC Solutions

Recent efforts have introduced switch-based congestion control proxies for inter-DC networks, yet they face intrinsic trade-offs between precision, scalability, and responsiveness.

(1) Coarse-grained Aggregation. Algorithms like BiCC [8] and ATC [9] aggregate inter-DC flows based on destination subnets to reduce state overhead at DCI switches. However, this coarse granularity creates severe fairness issues. By treating a bundle of inter-DC flows as a single entity, these schemes apply uniform rate throttling regardless of the flow count, penalizing inter-DC flows when competing with intra-DC traffic. As illustrated in Figure 2b, an intra-DC flow grabs significantly more bandwidth than individual inter-DC flows within an aggregate, revealing a fundamental inter-intra unfairness. Furthermore, ATC mandates deterministic routing (single path per destination), which disables ECMP load balancing and risks creating hotspots on core links. **(2) Prohibitive Queue Overhead.** Conversely, THEMIS [20] adopts a per-flow proxy architecture at the remote DCI switch. While precise, this approach suffers from state explosion, consuming excessive queues when tracking $O(1M)$ concurrent inter-DC flows [21]. Additionally, THEMIS introduces a proprietary control logic that differs from the native intra-DC algorithm (e.g., DCQCN). This algorithmic heterogeneity makes it difficult to guarantee fair convergence when heterogeneous control loops compete for the same buffer. **(3) Indirect Feedback Latency.** LSCC [22] attempts to bridge the gap by relaying rate limits from the remote to the local DCI switch. However, it relies on a cascaded feedback loop: congestion must first be detected remotely, then signaled to the local switch, which finally generates CNPs. This multi-hop signaling is still bound by the long inter-DC RTT, failing to provide the instantaneous reaction required to arrest bursty congestion in shallow-buffered switches.

C. Design Goals

Based on the limitations identified above, we argue that an ideal inter-DC congestion control mechanism must satisfy the following stringent design goals:

- **Sub-RTT Congestion Reaction:** The mechanism must decouple the control loop from the long inter-DC RTT. It should be capable of reacting to local buffer buildups instantaneously and mitigating remote congestion without the feedback delay penalty inherent in end-to-end schemes.
- **Scalable Granularity:** To ensure fairness between inter-DC and intra-DC flows, the mechanism must maintain fine-grained control (identifying aggressive flows). Crucially, this must be achieved without per-flow queues, avoiding the scalability bottlenecks and queue overhead that plague existing switch-based proxies.
- **Robustness to Feedback Staleness:** The mechanism must provide accurate control signals even when remote feedback is delayed. It needs to filter out noise from stale congestion notifications to prevent the throughput collapse and oscillation caused by reaction lag.
- **End-Host Transparency:** The mechanism must be deployed solely at the DCI gateways, remaining completely transparent to the end-host. Specifically, it should require no modifications to protocol stacks or NIC hardware, ensuring compatibility with existing applications and legacy hardware.

IV. Design

A. Overview

To resolve the mismatch between intra-DC control logic and inter-DC latency, *RiCE* fundamentally decouples the congestion control loop. Instead of a single, sluggish end-to-end feedback loop, *RiCE* introduces a split-control architecture comprising three key components deployed at DCI switches:

- **Local Traffic Control:** Deployed on the source-side DCI switch, this module monitors egress traffic from the local datacenter. It detects local congestion, which occurs within local datacenter, immediately and generates rapid

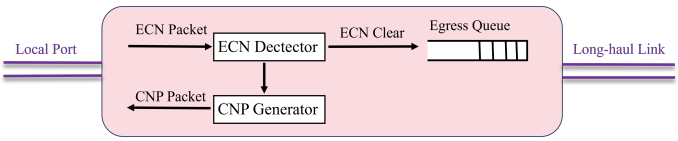


Fig. 4: Local Traffic Control

feedback signals (e.g., CNPs) to throttle local senders, preventing buffer overflows before packets even enter the long-haul link.

- **Remote Traffic Control:** Deployed on the destination-side DCI switch, this module acts as a congestion-control proxy for bottlenecks inside the remote datacenter.[2] It intercepts congestion feedback returned by the remote receiver and locally enforces per-path rate limiting on the DCI-to-DC egress. When downstream bandwidth becomes insufficient, the proxy temporarily buffers arriving inter-DC packets in dedicated queues to absorb short-lived rate mismatches. To preserve fairness without per-flow queues, *RiCE* emulates per-flow control within each path aggregate. If congestion persists and buffering pressure continues to grow, the proxy forwards feedback to senders (via the Remote Feedback Filter), preventing unbounded queue buildup.
- **Remote Feedback Filter:** To shield the sender from transient noise, the remote DCI switch intelligently filters feedback signals. It absorbs congestion notifications caused by short-lived bursts in the remote fabric. The switch only forwards congestion signals to the source sender when it detects sustained congestion within the remote datacenter that cannot be fully absorbed by the proxy mechanism.

Workflow. Figure 3 illustrates the operational flow of *RiCE*, distinguishing between two scenarios: *(1) Local Congestion:* When inter-DC traffic encounters congestion within the local datacenter fabric (indicated by ECN marks) or at the DCI switch itself, the Local Traffic Control module detects these signals and immediately generates CNPs. This triggers a rapid rate reduction at the sender, effectively containing the congestion locally. *(2) Remote Congestion:* When congestion occurs deep within the remote datacenter, the receiver generates CNPs. The congestion-control proxy on the remote DCI switch intercepts these CNPs and reduces its forwarding rate to relieve the remote bottleneck. If the congestion in the remote datacenter is transient, the proxy absorbs the backpressure using its local buffer (e.g., Virtual Output Queues, VOQs). However, in cases of persistent remote congestion, the Feedback Filter allows the CNPs to propagate back to the original sender, ensuring that the source eventually reduces its rate to match the long-term bottleneck capacity.

B. Local Traffic Control

RiCE implements a lightweight local feedback mechanism aligned with existing practices [8], [9]. As shown in Figure 4, upon detecting an intra-DC ECN mark, the local DCI switch immediately returns a CNP to the sender and strips the ECN mark. This atomic operation triggers rapid source throttling while ensuring strict isolation between local and remote congestion domains.

A key optimization in *RiCE* is its stateless design. We leverage the hardware capability of modern NICs (e.g., NVIDIA ConnectX [23]), which automatically enforce minimum CNP intervals at the sender side [24]. Consequently, compared to prior works, the local DCI switch in *RiCE* can simply generate feedback on every congestion event without maintaining per-flow timers or complex state, significantly reducing switch overhead.

Algorithm 1: Remote Traffic Control and Feedback Filter Logic

```
/* Init: N_path ← 1, r ← line_rate                    */
Function Receive_CNP(path) is
    if dec_times == 0 then
        r_dec ← r × (α/2) × (1/N_path);
    r ← r − r_dec;
    dec_times ← dec_times + 1;
    if dec_times == N_path then
        dec_times ← 0;
    Set_Timer(Rate_Increase, path);
Function Rate_Increase(path) is
    /* r_inc is r_ai or r_hai in DCQCN                */
    r_t ← r_t + r_inc × N_path;
    r ← (r + r_t) × 1/2;
    Set_Timer(Rate_Increase, path);
Function Add_Flow(path) is
    {r, r_t} ← {r, r_t} × (1 + 1/N_path);
    N_path = N_path + 1;
Function Remove_Flow(path) is
    {r, r_t} ← {r, r_t} × (1 − 1/N_path);
    N_path = N_path − 1;
Function CNP_Filter(path, cnp) is
    if Duration(qlen > Q_th) ≥ T_sustain then
        forward cnp;
```

C. Remote Traffic Control

As the core engine of *RiCE*, the Remote Traffic Control module functions as a transparent proxy residing on the destination DCI switch. Its primary objective is to shield the source sender from the high latency of the inter-DC link by handling remote congestion locally. Figure 5 illustrates the architectural workflow. To achieve precise control without incurring prohibitive state overhead, we decompose the design into three critical components: **(1) Control Granularity:** We first discuss *RiCE*'s choice of a scalable path-based grouping

[2]We assume DCQCN as the default CC algorithm.

strategy. **(2) Rate Adaptation:** Next, we detail the proxy's rate adjustment algorithm for handling remote congestion and enforcing per-flow fairness. **(3) Flow Lifecycle Management:** Finally, we describe the mechanism for handling flow arrival and departure to maintain system stability under dynamic traffic churn.

Control Granularity. Ideally, congestion control should be performed on a per-flow basis. However, maintaining per-flow state on DCI gateways is impractical due to queue constraints (e.g., $O(128K)$ queues) [25], which is an order of magnitude smaller than the number of concurrent inter-DC flows (e.g., $O(1M)$) [21]. Unlike prior approximations (e.g., BiCC [8], ATC [9]) that rely on coarse-grained per-destination aggregation—which, as discussed in Section III-B, compromises fairness and restricts ECMP load balancing—*RiCE* adopts a scalable per-path granularity. This design choice allows *RiCE* to distinguish congestion states across different physical paths while avoiding the overhead of per-flow tracking.

To implement per-path control efficiently, we exploit a structural characteristic of Clos topologies [10] (e.g., Fat-Tree, Spine-Leaf), the standard architecture for modern datacenters. *In these topologies, while upward routing (Server → Core) uses ECMP, the downward path from a specific top-tier switch (e.g., Spine/Core) to a destination server is unique and deterministic.* Based on this observation, *RiCE* uniquely identifies a downstream path using the tuple:

$$\mathcal{P} = \langle \text{Next-Hop Switch ID}, \text{Destination IP} \rangle.$$

All flows mapping to the same $\mathcal{P}$ share the exact same physical bottleneck path within the remote datacenter. Specifically, if a DCI gateway connects to the local fabric via k next-hops and the datacenter houses h hosts, the total number of distinct downstream paths is strictly bounded by $k \times h$, a constant that is significantly smaller than the number of concurrent inter-DC flows. For instance, assuming a standard DCI deployment with 8 next-hops connecting to a datacenter of 10,000 hosts, the proxy maintains at most 80,000 path instances. Consequently, by enforcing rate limits at this path level, *RiCE* achieves precise congestion management that is both topology-aware and inherently memory-efficient.

Rate Adaptation. To ensure fairness between the aggregate inter-DC traffic and native intra-DC flows, *RiCE* adapts the standard DCQCN state machine. To emulate per-flow control, the core challenge is to make a single aggregate flow behave mathematically like a bundle of N concurrent flows. We introduce a state variable, N_{path}, to track the estimated flow count on each path. The adaptation logic follows two principles: *(1) Diluted Rate Reduction:* Upon receiving a CNP, standard DCQCN cuts the flow rate by a factor $\alpha/2$. In our aggregate model, a single CNP typically signifies congestion for only one of the constituent flows. Applying the full cut to the entire aggregate would be overly aggressive. Therefore, *RiCE* attenuates the reduction effect by scaling the decrease factor to: (in Algorithm 1 line 3)

$$r_{dec} = r \times \frac{\alpha}{2} \times \frac{1}{N_{path}} \tag{1}$$

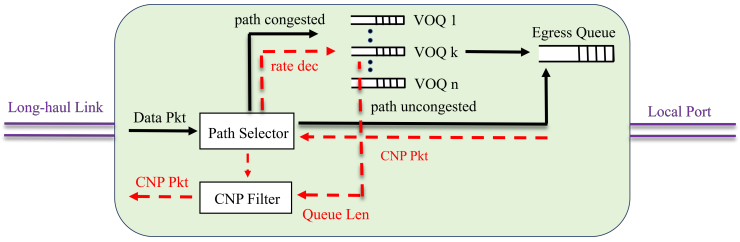


Fig. 5: Remote Traffic Control and Feedback Filter

This ensures that the aggregate rate drops only by the share of a single flow, preserving fairness relative to other traffic. *(2) Amplified Rate Recovery:* Conversely, during congestion-free periods, standard DCQCN increments the rate. Since all N_{path} flows within the aggregate would naturally accelerate simultaneously, the aggregate rate must recover faster to match the collective behavior. Thus, *RiCE* amplifies the increase step of DCQCN's target rate (r_t) by a factor of N_{path}: (in Algorithm 1 line 10)

$$r_t = r_t + r_{inc} \times N_{path} \tag{2}$$

Our design assumes that CNP arrivals are roughly distributed among flows. A potential corner case arises when a single flow generates a burst of consecutive CNPs, causing the aggregate to reduce its rate repeatedly. We argue that this behavior is not a flaw but a beneficial feature of path-level control. Since all flows in the aggregate share the exact same physical bottleneck, heavy congestion experienced by one flow implies that all flows of the same aggregate, which contribute to the same bottleneck, should be throttled. By throttling the aggregate directly, *RiCE* preemptively relieves pressure on the bottleneck, without waiting for feedback from every flow, thus achieving faster reaction to congestion on the same path.

Flow Lifecycle Management. To dynamically track flow count (N_{path}) without maintaining per-flow queues, *RiCE* inspects the Opcode field in the Base Transport Header (BTH) of RoCEv2 packets [26]. RoCEv2 explicitly marks message boundaries: specific opcodes (e.g., `RDMA_WRITE_FIRST`, `RDMA_READ_RESPONSE_FIRST`, ...) denote the start of a transmission, while others (e.g., `RDMA_WRITE_LAST`) denote its completion.

Leveraging this, *RiCE* updates the transmission rate (r) to proactively adapt to traffic churn (Algorithm 1, lines 13-18). Upon detecting the first packet, *RiCE* increments the path counter (N_{path}). Simultaneously, assuming the new flow requires the average bandwidth of existing flows, *RiCE* scales up the rate:

$$r = r \times (1 + \frac{1}{N_{path}}) \tag{3}$$

Conversely, upon detecting the last packet, *RiCE* scales down the rate by a factor of $1 - \frac{1}{N_{path}}$ and subsequently decrements the counter. This mechanism ensures that the per-flow bandwidth allocation remains stable as flows join or leave, effectively guaranteeing fairness between inter-DC and remote intra-DC traffic.

D. Remote Feedback Filter

As discussed in Section III-A, directly forwarding congestion feedback to the sender exposes the control loop to high inter-DC latency, often leading to unnecessary throttling for transient events. Empirical studies show that intra-DC congestion is highly transient, with over 80% of events lasting less than 100 μs [27]. Such micro-bursts are typically resolved locally before a feedback signal could even reach the sender.

To filter this high-frequency noise, *RiCE* employs a CNP Filter at the remote DCI switch. The filter intercepts CNPs generated by the remote receiver, allowing the remote traffic control proxy to handle the congestion first. The proxy absorbs the rate mismatch via per-path queues. Forwarding is triggered only when this local absorption capability is overwhelmed, indicating persistent congestion. Specifically, as detailed in Algorithm 1, the switch monitors the per-path queue length. It forwards a CNP to the sender only if the queue length exceeds a threshold Q_{th} for a continuous duration $T_{sustain}$.

For our 100 Gbps implementation, we configure $Q_{th} = 50$ KB and $T_{sustain} = 10$ μs. Sensitivity analysis indicates that system performance is robust to parameter variations. Generally, Q_{th} should scale linearly with the bandwidth, while $T_{sustain}$ should be slightly larger than the maximum intra-DC RTT to effectively filter out local micro-bursts.

V. Implementation

We prototyped *RiCE* on DPDK, a high-performance kernel-bypass framework that enables packet processing in user space. We implemented both the data and control planes of the DCI switch with 3.2K lines of C code, including the three components detailed in the Design section.

Control plane. The control plane maintains essential state for congestion handling. (1) Local Traffic Control: Generating valid CNPs at the local DCI switch requires the Destination Queue Pair Number (DQPN). We intercept RoCEv2 Connection Management (CM) packets to build a hash table mapping connection tuples to their negotiated QP numbers, enabling accurate CNP synthesis. (2) Remote Traffic Control: As discussed in §IV-C, the remote DCI switch aggregates inter-DC flows into per-path congestion control instances, uniquely identified by $\mathcal{P} = \langle \text{Next-Hop Switch ID}, \text{DIP} \rangle$. To determine the Next-Hop Switch ID (e.g., a specific core switch in a Fat-Tree topology) via ECMP, we employ a 2-tuple hash over the Source IP (SIP) and Destination IP (DIP) of passing packets. This symmetric 2-tuple hashing ensures that both forward data and backward feedback deterministically map to the same control instance $\mathcal{P}$.

Data plane. We designed a high-performance, lock-free packet processing pipeline. (1) Parallelism: We utilize Receive Side Scaling (RSS) to distribute traffic across RX queues, pinning each to a dedicated CPU core to minimize cache thrashing. (2) Lock-free VOQs: Virtual Output Queues (VOQs) are implemented using rte_ring, a lock-free ring buffer in DPDK, ensuring low-latency enqueue/dequeue operations. (3) Pipeline Logic: In the Local Module, the system performs wait-free lookups to generate immediate CNPs upon ECN detection. In the Remote Module, a polling-based scheduler enforces rate limits on VOQs, while the CNP Filter inspects queue depths against Q_{th} to gate feedback forwarding.

VI. Evaluation

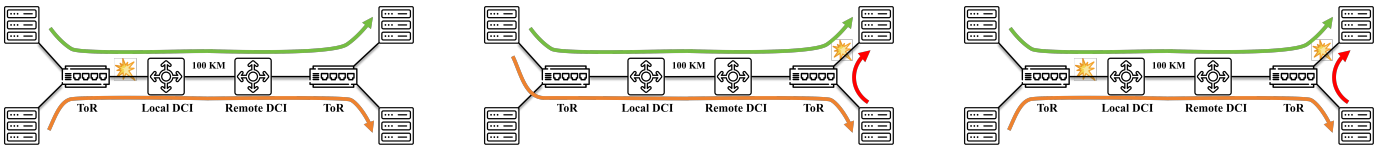
A. Evaluation Setup

Testbed Setup. We constructed a testbed using a dumbbell topology (Figure 6), comprising two ToR switches and two DCI gateways. The DCI gateways are deployed on commodity servers equipped with dual Intel Xeon E5-2650 v4 CPUs (2.2 GHz) and NVIDIA ConnectX-5/6 NICs [23]. To emulate a long-haul inter-DC link, we introduced a 500 μs one-way propagation delay (approx. 100 km physical distance) using DPDK. Traffic is generated using the RDMA perftest toolset [28] to simulate realistic RoCEv2 workloads. To eliminate artifacts from CPU overload in the software prototype, we cap the maximum injection rate of all flows at 60 Gbps. This ensures that observed packet drops are strictly attributable to network congestion rather than forwarding bottlenecks.

Simulation Setup. We conducted large-scale simulations using NS-3 based on the Fat-Tree topology (Figure 1). Each datacenter is configured with a 3-tier architecture comprising 4 Core, 8 Aggregation, and 8 ToR switches, with each ToR connecting two hosts. Network parameters are set to reflect high-performance production environments: the inter-DC link is configured with 400 Gbps bandwidth and 500 μs propagation delay (approx. 100 km), while all intra-DC links operate at 100 Gbps with 1 μs delay. To support lossless transmission, we enabled PFC with a dynamic threshold scheme, triggering pause frames when an ingress queue consumes more than 11% of the free buffer space. Switch buffer sizes are set to 128 MB for DCI gateways (to absorb long-haul delay bandwidth product) and 16 MB for intra-DC switches.

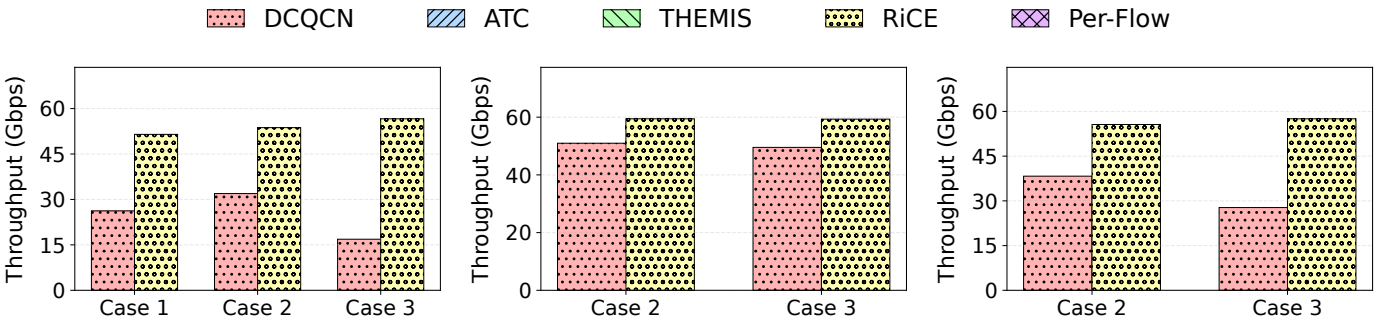
Baselines and Parameter Settings. We compare *RiCE* against the following schemes: (1) DCQCN [7]: The standard RDMA congestion control, serving as the primary baseline for both testbed and simulation experiments. Parameters are tuned based on vendor recommendations [24]. (2) ATC [9] and THEMIS [20]: State-of-the-art inter-DC solutions evaluated in simulations. We omit BiCC results as they mirror ATC's performance in our metrics. (3) Ideal Per-Flow: To validate *RiCE*'s ability to emulate per-flow fairness using per-path granularity, we implemented an idealized baseline. This scheme enforces perfect per-flow rate control at the remote DCI (assuming infinite resources) but excludes the Remote Feedback Filter to isolate the impact of control granularity. ECN thresholds are set according to [18]. For ATC and THEMIS, we use the native suggested parameters.

Workload and Metrics. We utilize the realistic WebSearch workload [29] for both inter- and intra-DC traffic. System performance is evaluated using three key metrics: (1) Normalized Flow Completion Time (FCT): To measure application-level latency performance. (2) DCI Buffer Occupancy: To evaluate the efficacy of the proxy in preventing queue buildup. (3) Inter-DC Link Throughput: To ensure that congestion control does not compromise link utilization.



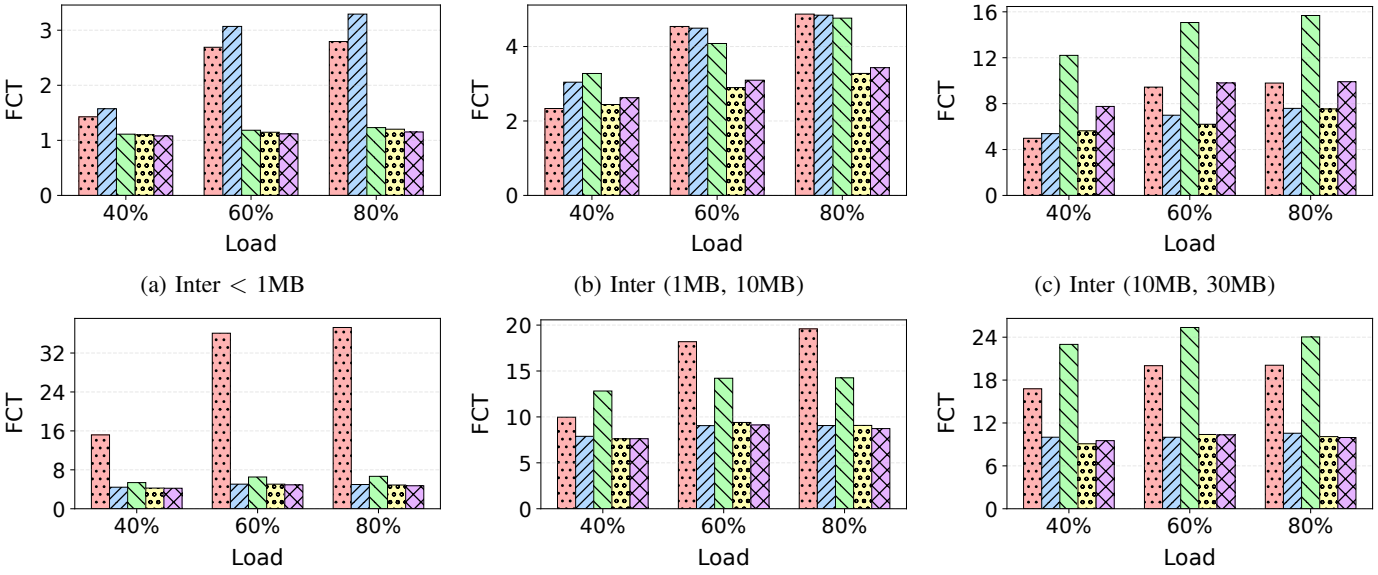
(a) Case 1: local congestion (b) Case 2: remote congestion (c) Case 3: local and remote congestion

Fig. 6: Topology and traffic pattern in testbed experiments



(a) Inter (b) Intra (c) Mix traffic

Fig. 7: Average Throughput of different traffic in testbed experiments



(a) Inter $<$ 1MB (b) Inter (1MB, 10MB) (c) Inter (10MB, 30MB)

(d) Intra $<$ 1MB (e) Intra (1MB, 10MB) (f) Intra (10MB, 30MB)

Fig. 8: Average Normalized FCT of different flow size and workload

B. Testbed Experiment

Throughput under Varied Congestion Scenarios. We evaluate *RiCE*'s throughput performance across three distinct congestion scenarios using the topology depicted in Figure 6. Figure 7 reports the aggregated throughput for inter-DC traffic and the average throughput for intra-DC background traffic.

Case 1: Local Congestion. This scenario emulates congestion at the local DCI gateway. *RiCE* achieves 96.4% higher throughput compared to DCQCN. The gain stems from the Local Traffic Control module, which intercepts ECN marks and generates immediate backward CNPs locally. In contrast, DCQCN forces feedback to traverse the entire inter-DC link (high RTT), causing slow reaction times that lead to buffer overflows and severe throughput degradation. **Case 2: Remote Congestion.** This scenario introduces contention at the remote datacenter. *RiCE* improves inter- and intra-DC throughput by 68.2% and 16.6%, respectively. This improvement validates the Remote Traffic Control proxy, which shields the long RTT of inter-DC flows. By enforcing rate limits locally at the remote DCI, *RiCE* eliminates the congestion rapidly, while also preventing PFC storms that would otherwise collateral damage intra-DC flows. **Case 3: Coexisting Congestion.** In this worst-

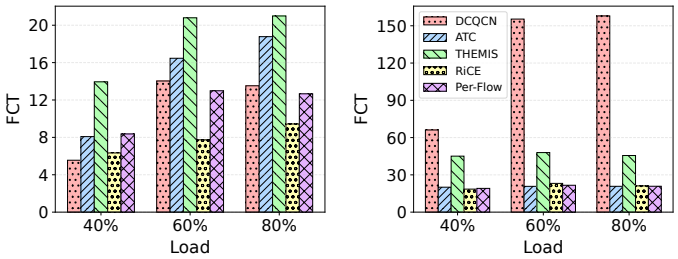
case scenario where both local and remote bottlenecks exist, *RiCE* demonstrates superior resilience, achieving 3.3× the inter-DC throughput of DCQCN. Overall, *RiCE* maintains high link utilization and stability in complex environments where standard end-to-end schemes collapse.

C. Large Scale Simulation

***RiCE* effectively enhances the performance of inter traffic while maintaining low FCT for intra flows.** In the simulation, we fixed the intra traffic load at 60% and varied the inter load from 40% to 80%. The flows were categorized by size into Small (< 1 MB), Medium (1-10 MB), and Large (10-30 MB). Figure 8 presents the average normalized FCT across different flow sizes. Compared to DCQCN, *RiCE* reduces inter FCT by 10%-47.5% and achieves a reduction of up to 80.1% in intra FCT. This significant improvement is attributed to *RiCE*'s shortened congestion feedback loop, which enables rapid mitigation of both local and remote congestion. Against ATC, *RiCE* lowers inter FCT by 23.4%-50% while performing comparably for intra flows. As discussed in Section III, ATC's per-destination granularity implicitly leads to inter-intra unfairness, causing individual inter flows to obtain less bandwidth. Furthermore, its requirement to restrict ECMP leads to load imbalance at core switches, which is evident in its long FCT of small flows under high load. Regarding THEMIS, *RiCE* reduces the average inter and intra FCT by 23.82% and 32.06%, respectively. The algorithm employed by THEMIS tends to inject excessive traffic into the remote datacenter, incurring persistent congestion, which is reflected in the higher FCTs observed for both inter/intra medium and large flows. Finally, when compared to the ideal Per-Flow baseline, *RiCE* exhibits equivalent performance for small and medium inter flows, while maintaining low FCTs for intra traffic. This demonstrates that *RiCE* effectively emulates the precision of per-flow control, and guarantees fairness between inter and intra traffic. Notably, *RiCE* achieves even lower FCTs for large inter flows than the Per-Flow baseline. It stems from the Remote Feedback Filter mechanism, which effectively prevents stale feedback from triggering premature rate reductions at the source.

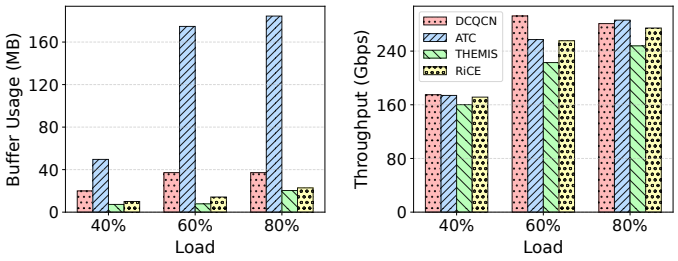
***RiCE* improves tail latency performance.** Illustrated in Figure 9, *RiCE* also demonstrates superior tail latency performance in terms of p99 normalized FCT. Specifically, for inter flows, *RiCE* reduces the p99 FCT by 20.4%, 41.5%, and 57.5% on average across different loads compared to DCQCN, ATC, and THEMIS, respectively. For intra flows, it achieves reductions of 81.24% against DCQCN and 54.75% against THEMIS, while performing comparably to ATC.

***RiCE* ensures high inter-DC throughput and maintains low buffer occupancy at the DCI switches.** As illustrated in Figure 10, we measured both the inter-DC link throughput and the buffer usage at the DCI switches during the simulation. DCQCN achieves high inter throughput due to its delayed response to remote congestion; however, this comes at the expense of intra traffic performance. While ATC achieves relatively high inter throughput, it suffers from excessive buffer occupancy at the DCI switch under heavy loads. This issue arises because a single CNP can trigger the deceleration of all flows targeting the same destination, thereby exacerbating buffer accumulation due to the rate disparity. Conversely, THEMIS maintains low buffer occupancy but yields the lowest inter throughput among the evaluated algorithms. As previously discussed, THEMIS induces long-term congestion within datacenter. Consequently, the senders of inter flows frequently receive rapid feedback from the local DCI switch, forcing them to reduce transmission rates persistently. In contrast, *RiCE* effectively mitigates intra-DC congestion, enabling the DCI switch to drain traffic to remote datacenters swiftly. As a result, *RiCE* achieves high throughput on the inter-DC link while preserving low buffer occupancy at the DCI switch.



(a) Inter-P99 (b) Intra-P99

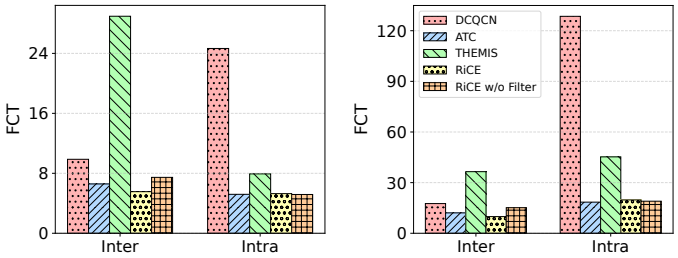
Fig. 9: P99 Normalized FCT of different workload



(a) Buffer Usage (b) Inter-DC throughput

Fig. 10: Buffer Usage of DCI switch and Inter-DC throughput

D. Micro-benchmark



(a) Avg. FCT (b) P99 FCT

Fig. 11: Normalized FCT in large flows dominated scenarios

Efficacy of Remote Feedback Filter. The Remote Feedback Filter module enhances performance by filtering CNP packets, thereby preventing the senders from undergoing premature rate reduction. This mechanism is particularly beneficial for long flows. In the context of cross-datacenter AI training

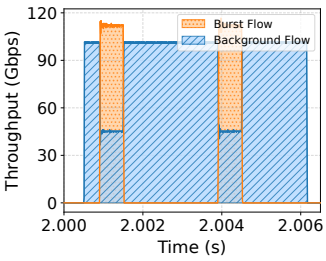


Fig. 12: Burst Responsiveness

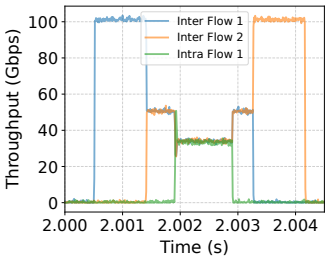


Fig. 13: Inter-Intra Fairness

and inference, traffic is typically characterized by long flows ranging from 10x-100x MB [30]. To evaluate the efficacy of the Remote Feedback Filter module, we constructed a specialized scenario dominated by such long inter flows. We denote the variant of our algorithm with this module removed as "*RiCE* w/o Filter". Figure 11 presents the average and p99 normalized FCT for both inter and intra traffic. Compared to the implementation without CNP filter, *RiCE* reduces the average and p99 FCT of inter flows by 25.6% and 34.9%, respectively, while incurring only a negligible increase in the FCT of intra flows. The result demonstrates that in cross-datacenter training and inference workloads, which are dominated by long flows, *RiCE* substantially improves the performance by selectively filtering CNPs at the remote DCI switch. In fact, even in the large-scale simulations discussed earlier, this filtering mechanism contributed to a 22.5% reduction in the p99 FCT of inter flows.

Responsiveness to bursts and Fairness. We further evaluated *RiCE*'s responsiveness and fairness under the same simulation scenarios described in Section III, where DCQCN and BiCC exhibited suboptimal performance. As shown in Figure 12, when bursty intra traffic is introduced, the throughput of background inter flows suffers only a temporary decrease and immediately recovers. This behavior highlights *RiCE*'s capability to rapidly respond to and mitigate congestion. Furthermore, Figure 13 illustrates that after the injection of intra traffic, the two inter flows and the single intra flow achieve an equal share of the link bandwidth. It confirms that *RiCE* guarantees fairness between inter and intra traffic.

VII. Discussion

Deployability on Commodity Switch Hardware. The per-path queuing mechanism of *RiCE* aligns naturally with the architecture of modern Deep-Buffer DCI gateways. Leading chipsets, such as the Broadcom Jericho series [25] and Cisco Silicon One [31], natively support hierarchical Virtual Output Queues (VOQs) and usually provide GB-level on-chip buffers. These devices typically support up to 128K VOQs. Although it remains a concern to maintain per-flow queues (e.g., $O(1M)$ concurrent flows), it can now offer sufficient scalability for per-path control granularity in *RiCE*, which is orders of magnitude smaller than the flow granularity. We plan to implement a hardware prototype on P4-programmable switches or programmable line cards in future work. Additionally, no specific modifications to existing flow control (e.g., PFC) are needed, and *RiCE* is orthogonal to these schemes.

Extensibility to other CC algorithms. The algorithm adapted for DCQCN in Section IV can be extended to other signal-based CC algorithms, such as HPCC [18] and TIMELY [19]. For HPCC, the local DCI switch can generate rapid feedback based on the In-band Network Telemetry (INT) header, while the remote DCI switch adjusts rates based on the INT information in ACKs. For TIMELY, the local DCI switch can generate feedback, enabling the sender to utilize the intra RTTs to adjust its rates. The remote DCI switch can mark data packets with timestamps. Upon receiving ACKs, the system calculates RTTs within the remote DC and adjusts the rates accordingly. Moreover, the aggregation logic used to emulate per-flow control is equally applicable to these CC algorithms. For every rate adjustment, the magnitude should be scaled down to $1/N_{path}$ of the original (note that this differs slightly from DCQCN, as rate increase in HPCC and TIMELY is ACK-driven and each ACK corresponds to a single flow). Consistent with Algorithm 1, *RiCE* records the initial state before the first rate reduction. Subsequent reductions are calculated relative to this initial state until the rate has been decreased N_{path} times, at which point the state should be updated.

VIII. Related Work

Intra-Datacenter Congestion Control. Existing algorithms are predominantly optimized for low-latency, short-RTT intra-DC environments. (1) Sender-driven schemes, such as DCQCN [7], TIMELY [19], and HPCC [18], rely on ECN or RTT signals to adjust rates. In cross-DC scenarios, the feedback loop is stretched by the long propagation delay, causing slow convergence and buffer overflows before the sender can react. (2) Receiver-driven schemes, including NDP [32], Homa [33], ExpressPass [34] and Aeolus [35], achieve near-perfect load balancing by decoupling scheduling from transmission. However, they face significant deployment hurdles: ExpressPass and NDP require custom switch hardware (e.g., packet trimming) not supported by commodity DCI gateways. Homa relies on priority queues for Shortest Remaining Processing Time (SRPT) scheduling, which can lead to starvation or priority inversion in cross-DC environments. Moreover, their proactive request-grant mechanisms are sensitive to the high latency of DCI links, potentially lowering link utilization.

Inter-Datacenter Transport. Several protocols address congestion over Wide Area Networks (WAN). (1) TCP-based Optimization: Schemes like GEMINI [36] and Annulus [37] use dual-loop controls to manage local and remote congestion separately. However, they are tailored for TCP/IP stacks, which tolerate high packet loss and rely on deep buffers. They are incompatible with the strict lossless requirements and shallow-buffered switches typical of RDMA fabrics. (2) Inter-DC RDMA: Recent works like BiCC [8] and ATC [9] attempt to adapt RoCEv2 for DCI. However, as extensively analyzed in Section III, they rely on coarse-grained per-destination aggregation, leading to frequent ECMP collision on the path.

RDMA over Lossy Networks. Protocols like IRN [38] and SRNIC [39] relax the lossless constraint of RoCE, relying on efficient selective retransmission to handle packet drops. While

promising for intra-DC, applying them to long-haul DCI poses challenges: the high bandwidth-delay product (BDP) necessitates maintaining massive reordering buffers and bitmaps at the receiver. Furthermore, on long-fat pipes, even a low packet loss rate can trigger frequent retransmissions, significantly degrading goodput. Consequently, *RiCE* adheres to a lossless design to maximize throughput and minimize tail latency.

IX. Conclusion

In this paper, we presented *RiCE*, a congestion control framework tailored for inter-datacenter RDMA. *RiCE* decouples the control loop into three synergistic components: Local Traffic Control for immediate reaction to local buffer buildup; Remote Traffic Control, which exploits per-path granularity to emulate per-flow fairness without maintaining per-flow queues; and a Remote Feedback Filter that masks transient noise to prevent premature throttling. Extensive evaluations on a DPDK testbed and large-scale simulations demonstrate that *RiCE* effectively bridges the gap between lossless transport and long-haul latency, achieving up to 3.3× higher throughput than DCQCN while significantly reducing flow completion times.

Acknowledgments

This project is partially supported by the National Key R&D Program of China (2024YFB2906700), the National Natural Science Foundation of China under Grant Numbers 62502194, 62325205 and U25B2035, the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM103), the "111 Center" (No. B26023), and the Nanjing University-China Mobile Communications Group Co.,Ltd. Joint Institute. Chengyuan Huang and Lu Lu are the corresponding authors.

References

[1] P. A. S. Alam and G. Dumont, "Awsre: Invent. enterprise fundamentals: design your account and vpc architecture for enterprise operating models," 2016.

[2] Y. Sverdlik, "Facebook rethinks in-region data center interconnection," 2018.

[3] J. da Silva, "Google turns to nuclear to power AI data centres," 2024. [Online]. Available: https://www.bbc.com/news/articles/c748gn94k95o

[4] "Multi-datacenter training: OpenAI's ambitious plan to beat Google's infrastructure," 2024. [Online]. Available: https://newsletter.semianalysis.com/p/multi-datacenter-training-openais

[5] "Supplement to infiniband architecture specification volume 1, release 1.2.2, annex a17: Rocev2 (ip routable roce)," InfiniBand Trade Association, Tech. Rep., 2014.

[6] *IEEE 802.1Qbb: Priority Based Flow Control*, Std., 2011.

[7] Y. Zhu, H. Eran, D. Firestone, C. Guo, M. Lipshteyn, Y. Liron *et al.*, "Congestion control for large-scale rdma deployments," in *ACM SIGCOMM*, 2015.

[8] Z. Wan, J. Zhang, M. Yu, J. Liu, J. Yao, X. Zhao, and T. Huang, "Bicc: Bilateral congestion control in cross-datacenter rdma networks," in *IEEE INFOCOM*, 2024.

[9] Z. Wan, J. Zhang, Y. Su, H. Pan, M. Yu, Y. Li, and T. Huang, "Re-architecting traffic control in cross-datacenter rdma networks," *IEEE Transactions on Networking*, 2025.

[10] A. Singh, J. Ong, A. Agarwal, G. Anderson, A. Armistead, R. Bannon *et al.*, "Jupiter rising: A decade of clos topologies and centralized control in google's datacenter network," in *ACM SIGCOMM*, 2015.

[11] W. Bai, S. S. Abdeen, A. Agrawal, K. K. Attre, P. Bahl, A. Bhagat *et al.*, "Empowering azure storage with RDMA," in *USENIX NSDI*, 2023.

[12] R. Miao, L. Zhu, S. Ma, K. Qian, S. Zhuang, B. Li *et al.*, "From luna to solar: the evolutions of the compute-to-storage networks in alibaba cloud," in *ACM SIGCOMM*, 2022.

[13] "How to Connect Distributed Data Centers Into Large AI Factories with Scale-Across Networking," 2025. [Online]. Available: https://developer.nvidia.com/blog/how-to-connect-distributed-data-centers-into-large-ai-factories-with-scale-across-networking

[14] "Broadcom Ships Jericho4, Enabling Distributed AI Computing Across Data Centers," 2025. [Online]. Available: https://www.broadcom.com/company/news/product-releases/63361

[15] M. Filer, J. Gaudette, Y. Yin, D. Billor, Z. Bakhtiari, and J. L. Cox, "Low-margin optical networking at cloud scale [invited]," *Journal of Optical Communications and Networking*, 2019.

[16] Y. Chen, C. Tian, J. Dong, S. Feng, X. Zhang, C. Liu, P. Yu, N. Xia, W. Dou, and G. Chen, "Swing: Providing long-range lossless rdma via pfc-relay," *IEEE TPDS*, 2023.

[17] C. Guo, H. Wu, Z. Deng, G. Soni, J. Ye, J. Padhye, and M. Lipshteyn, "Rdma over commodity ethernet at scale," in *ACM SIGCOMM*, 2016.

[18] Y. Li, R. Miao, H. H. Liu, Y. Zhuang, F. Feng, L. Tang *et al.*, "Hpcc: high precision congestion control," in *ACM SIGCOMM*, 2019.

[19] R. Mittal, V. T. Lam, N. Dukkipati, E. Blem, H. Wassel, M. Ghobadi *et al.*, "Timely: Rtt-based congestion control for the datacenter," in *ACM SIGCOMM*, 2015.

[20] Z. Niu, M. Zhang, J. Zhang, R. Xie, Y. Yang, and X. Hu, "THEMIS: Addressing Congestion-Induced Unfairness in Long-Haul RDMA Networks ," in *IEEE ICNP*, 2025.

[21] "AliStorage Traffic Trace." [Online]. Available: https://github.com/alibaba-edu/High-Precision-Congestion-Control/tree/master/traffic_gen

[22] M. Long, J. Han, W. Wang, J. Yang, and K. Xue, "Lscc: Link-segmented congestion control for rdma in cross-datacenter networks," in *IEEE/ACM IWQoS*, 2024.

[23] "Ethernet Network Adapters - ConnectX NICs," 2024. [Online]. Available: https://www.nvidia.com/en-us/networking/ethernet-adapters/

[24] "DCQCN Parameters," 2024. [Online]. Available: https://enterprise-support.nvidia.com/s/article/dcqcn-parameters

[25] "BCM88800 Traffic Management Architecture," 2021. [Online]. Available: https://docs.broadcom.com/doc/88800-DG1-PUB

[26] "Infiniband architecture volume 1: General specifications," InfiniBand Trade Association, Tech. Rep. Release 1.2.1, 2008.

[27] H. Zheng, C. Huang, X. Han, J. Zheng, X. Wang, C. Tian, W. Dou, and G. Chen, "μmon: Empowering microsecond-level network monitoring with wavelets," in *ACM SIGCOMM*, 2024.

[28] "perftest: Infiniband verbs performance tests," https://github.com/linux-rdma/perftest, 2022.

[29] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar *et al.*, "Data center tcp (dctcp)," in *ACM SIGCOMM*, 2010.

[30] W. Li, X. Liu, Y. Li, Y. Jin, H. Tian, Z. Zhong, G. Liu, Y. Zhang, and K. Chen, "Understanding communication characteristics of distributed training," in *Asia-Pacific Workshop on Networking (APNet)*, 2024.

[31] "Cisco ASR 1000 Series Aggregation Services Routers," 2024. [Online]. Available: https://www.cisco.com/c/en/us/products/routers/asr-1000-series-aggregation-services-routers/index.html

[32] M. Handley, C. Raiciu, A. Agache, A. Voinescu, A. W. Moore, G. Antichi, and M. Wójcik, "Re-architecting datacenter networks and stacks for low latency and high performance," in *ACM SIGCOMM*, 2017.

[33] B. Montazeri, Y. Li, M. Alizadeh, and J. Ousterhout, "Homa: a receiver-driven low-latency transport protocol using network priorities," in *ACM SIGCOMM*, 2018.

[34] I. Cho, K. Jang, and D. Han, "Credit-scheduled delay-bounded congestion control for datacenters," in *ACM SIGCOMM*, 2017.

[35] S. Hu, W. Bai, G. Zeng, Z. Wang, B. Qiao, K. Chen, K. Tan, and Y. Wang, "Aeolus: A building block for proactive transport in datacenters," in *ACM SIGCOMM*, 2020.

[36] G. Zeng, W. Bai, G. Chen, K. Chen, D. Han, Y. Zhu, and L. Cui, "Congestion control for cross-datacenter networks," in *IEEE ICNP*, 2019.

[37] A. Saeed, V. Gupta, P. Goyal, M. Sharif, R. Pan, M. Ammar *et al.*, "Annulus: A dual congestion control loop for datacenter and wan traffic aggregates," in *ACM SIGCOMM*, 2020.

[38] R. Mittal, A. Shpiner, A. Panda, E. Zahavi, A. Krishnamurthy, S. Ratnasamy, and S. Shenker, "Revisiting network support for rdma," in *ACM SIGCOMM*, 2018.

[39] Z. Wang, L. Luo, Q. Ning, C. Zeng, W. Li, X. Wan *et al.*, "SRNIC: A scalable architecture for RDMA NICs," in *USENIX NSDI*, 2023.