THEMIS: Addressing Congestion-Induced Unfairness in Long-Haul RDMA Networks

Menghao Zhang, Zihan Niu, Jue Zhang, *Graduate Student Member, IEEE*, Dongyang Wang, Renjie Xie, Yuan Yang, Mingwei Xu, and Chunming Hu

***Abstract*—RDMA is promising for enhancing the performance of cross-datacenter (DC) services. However, deploying RDMA over long-haul networks introduces severe congestion control unfairness, primarily due to asymmetric congestion feedback delays between inter-DC flows and intra-DC flows. As a result, intra-DC flows often bear the full burden of congestion response, leading to drastically increased flow completion times (FCT). In this work, we identify two key forms of unfairness — near-source and near-destination — depending on whether congestion occurs near the sender or receiver of inter-DC flows. Based on this, we propose THEMIS, a fairness maintenance patch for long-haul RDMA networks. To mitigate near-source unfairness, THEMIS devises a Proactive Notification Point to shorten the congestion feedback loop within a single DC. To alleviate near-destination unfairness, THEMIS introduces a Temporary Reaction Point to temporarily slow down the target inter-DC flow until the sender receives the corresponding congestion feedback. We implement an open-source prototype of THEMIS, and evaluate it on both real-world testbeds and large-scale simulations. Compared to DCQCN, Annulus and BiCC, THEMIS decreases overall FCT by up to 53.9%, 32.9% and 87.7%.**

***Index Terms*—RDMA, long-haul networks, congestion control, fairness.**

I. INTRODUCTION

REMOTE Direct Memory Access (RDMA), benefiting from its kernel-bypass and protocol-offload features, offers low latency, high throughput, and near zero CPU usage for data transmission. It is promising to improve existing TCP network services, especially with the advent of RoCEv2 (RDMA over Converged Ethernet version 2) protocol. Previous studies have explored how RDMA can replace TCP networks within Data Centers (DCs) [1], [2], [3], [4], [5], [6], [7], [8], [9]. In recent years, cross-DC services have emerged to meet the growing demand for big data processing, scientific research, and large language model (LLM) training [10], [11], [12]. These services require efficient exchange of vast amounts of data across geographically distributed DCs. Therefore, it is widely considered that long-haul RDMA network transmission is an unstoppable future trend.

However, extending RDMA to long-haul scenarios is non-trivial. We identify that applying intra-DC congestion control (CC) algorithms to long-haul networks results in severe unfairness issue. This results in significant suppression of intra-DC traffic, with the flow rates failing to converge, therefore severely reducing the overall network throughput and stability. In our simulation experiment, we use average normalized flow completion times (FCTs) to represent flow performance and average normalized FCT ratio of intra-DC/inter-DC flows to quantify unfairness. Taking DCQCN [13] as an example, we find that the long-haul factor significantly increases the FCT of intra-DC flows and increases unfairness by 4.8-10.8×. A closer look into the DCQCN algorithm reveals the underlying root causes of the unfairness issues. When a bottleneck happens on a switch, it marks packets in the congested queue. Upon receiving this congestion mark, the receiver sends congestion feedback to the sender, prompting it to reduce the sending rate and alleviate congestion. However, long-haul scenarios result in significant physical distance differences between the two ends of different RDMA connections. During the period from when congestion starts until the feedback signal arrives at the remote inter-DC traffic sender, only intra-DC traffic is suppressed. Meanwhile, inter-DC traffic continues at its original speed, causing queue buildup at the congestion point. This results in a significant increase in intra-DC FCT, while inter-DC traffic FCT is relatively less affected. In particular, small flows are highly sensitive to latency and are therefore impacted more severely by this unfairness issue.

While recent efforts have been devoted to extending RDMA to cross-DC services [14], [15], [16], [17], [18], they do not fully address the unfairness issue. Swing [15] and Bifrost [16] mainly focus on how to optimize the priority-based flow control (PFC) buffer requirement, as a huge buffer size is required at every switch to achieve the lossless property.

Received 26 October 2025; revised 24 March 2026, 28 June 2026, and 24 August 2026; accepted 6 October 2026; approved by IEEE TRANSACTIONS ON NETWORKING Editor D. Han. Date of publication 13 October 2026; date of current version 28 October 2026. This work was supported in part by the Open Research Fund of State Key Laboratory of Internet Architecture under Grant HLW2025ZD01, in part by the National Natural Science Foundation of China under Grant 62402025 and Grant 623B2062, in part by Shandong Provincial Natural Science Foundation of China under Grant ZR2024LZH011, and in part by the Fundamental Research Funds for the Central Universities. An earlier version of this paper was presented at the Conference of ICNP 2025 [DOI: 10.1109/TON.2026.3743177]. *(Corresponding author: Menghao Zhang.)*

Menghao Zhang and Zihan Niu are with the School of Software, Beihang University, Beijing 100191, China, and also with the State Key Laboratory of Internet Architecture, Tsinghua University, Beijing 100084, China (e-mail: zhangmenghao0503@gmail.com; zihanniu85@gmail.com).

Jue Zhang and Chunming Hu are with the School of Software, Beihang University, Beijing 100191, China (e-mail: zhangjue1@buaa.edu.cn; hucm@buaa.edu.cn).

Dongyang Wang is with ByteDance, Beijing 100098, China (e-mail: wangdongyang.wdy@bytedance.com).

Renjie Xie is with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: rexie@guet.edu.cn).

Yuan Yang and Mingwei Xu are with the State Key Laboratory of Internet Architecture, Tsinghua University, Beijing 100084, China (e-mail: yangyuan_thu@mail.tsinghua.edu.cn; xumw@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TON.2026.3743177

2998-4157 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.

Closely related to the unfairness issue, Annulus [17] is a two-congestion control loop scheme that improves performance when WAN traffic shares the DC network with intra-DC traffic. However, it requires changes to the network protocol stack and can only address part of the unfairness issues (i.e., near-source unfairness). Bilateral Congestion Control (BiCC) [18] alleviates near-source congestion by transmitting near-source congestion feedback through the sender's datacenter interconnection (DCI) switch, and mitigates near-destination congestion by limiting the inflight byte count between the receiver's DCI switch and the receiver to the BDP within DC. However, on one hand, BiCC cannot generate Congestion Notification Packets (CNPs) correctly to reduce the rate of target flows. On the other hand, BiCC uses Virtual Output Queues (VOQ) to temporarily store paused flows with the same destination IP, resulting in significant buffer overhead and lacking the desired precision. To conclude, existing solutions have not fully resolved the unfairness issue and there remains a pressing need for a more comprehensive and effective solution.

We believe that an ideal cross-DC RDMA transmission should require minimal modifications to existing DC infrastructure. Therefore, instead of introducing an entirely new CC algorithm to replace the fine-tuned existing ones [13], [19], we take a more pragmatic approach by proposing an incremental solution named THEMIS to supplement the widely used DCQCN algorithm. THEMIS is inspired by the proxy design concept. This concept reflects a key idea: proxies act as intermediaries that take on the responsibility of monitoring and managing the entire DC's traffic without requiring any modification to the underlying infrastructures. Thus, THEMIS is only deployed at the datacenter interconnection (DCI) switches connecting geographically distributed DCs, functioning like a fairness maintenance patch for long-haul RDMA networks.

At a high level, THEMIS intuitively keeps the distance between the reaction point and the notification point in the CC mechanism confined within a single DC as much as possible. When congestion occurs within the sender's DC (referred to as near-source unfairness), THEMIS at the DCI-switches deployed atop the sender's DC detects congestion marks and immediately sends congestion feedback to the sender to throttle its rate. When congestion occurs within the receiver's DC (referred to as near-destination unfairness), THEMIS receives congestion feedback from the receiver, and temporarily takes responsibility for throttling the target flow.

Nevertheless, designing THEMIS is a non-trivial effort. First, the RoCEv2 protocol specifies that packets do not carry the source Queue Pair Number (sQPN), a field essential for generating congestion feedback, i.e., CNP. This makes it difficult for the DCI-switches to notify the sender correctly. Moreover, as the number of flows in DCs is extremely large, switch memory is insufficient to store QPN mappings for all connections. Second, existing switches only provide binary traffic management functionality (e.g., PFC), which operates at the coarse granularity of priority queues. Therefore, it is unable to support different throttling strategies for different flows on the switch without impacting innocent flows.

To address these challenges, THEMIS designs a Proactive Notification Point (PNP) and a Temporary Reaction Point (TRP) at DCI-switches. For PNP, the host maintains a QP Pool, encoding part of the QPN in the UDP source port and deriving the rest from the hash of destination IP (dIP), so that the DCI-switches can conveniently obtain the sQPN without storing any QPN mappings and promptly generate a CNP to the sender. For TRP, when detecting CNPs from the receiver, it redirects target flows to a pre-filled port and loops them several times, avoiding impacting innocents. We also design a throttling algorithm in smart switch data structures to ensure flow granularity deceleration with controllable buffer overhead.

We implement a prototype of THEMIS in both NS-3 simulation and hardware testbed, and make its source code publicly available at GitHub [20]. We compare THEMIS with DCQCN, Annulus and BiCC under various workloads. Compared to DCQCN, THEMIS reduces the intra-DC FCT by up to 61.9% and decreases the PFC triggers by up to 95%. Although THEMIS slightly reduces the rate of inter-DC flows, it improves fairness and stability, thereby decreasing the overall FCT by up to 53.9%, 32.9% and 87.7%, compared to DCQCN, Annulus, and BiCC. Our ablation experiments show that PNP and TRP independently reduce the normalized FCT of intra-DC flows by up to 55.2% and 44.6%. And one of our optimizations, e.g., alignment with DCQCN, achieves a 12.0% reduction in overall FCT.

In summary, we make following contributions:

- We analyze the root causes of congestion-induced unfairness in long-haul RDMA networks and summarize the shortcomings of existing related works (§II).
- We identify two types of unfairness, and propose THEMIS to address them while maintaining compatibility with existing CC algorithms (§III, §IV, §VI§V).
- We implement an open-source prototype of THEMIS and evaluate THEMIS via testbed experiments and large-scale NS-3 simulations (§VII, §VIII).

Finally, we make some discussions in §IX, describe related works in §X, and conclude this paper in §XI.

II. BACKGROUND AND MOTIVATION

In this section, we give the background on long-haul RDMA networks, and then analyze the unfairness issue when extending current RDMA CC algorithms to long-haul networks.

A. Long-Haul RDMA Networks

Nowadays, DCs have grown significantly to meet the demands of high-performance services. However, for a single DC, there is always an upper limit. Therefore, for the following three reasons, companies are actively developing their cross-DC services. First, the rapid development of various distributed applications has raised higher demands on the capabilities of a single DC, leading to the emergence of cross-DC service requirements [14], [21], [22]. For instance, the demand for storage resources has surged with the rapid growth of application user bases, making it impossible for cloud service providers (CSPs) to rely on a single DC to store user data [23].

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.

Besides, the size and the training data of LLMs are growing rapidly, which requires GPU resources that far exceed the capacity of a single DC [12]. Second, differences in geographic locations result in varying energy, labor, and land costs. CSPs prefer building DCs in cost-effective cities and interconnecting them to achieve greater efficiency [24]. Third, users expect reliable services, prompting CSPs to build DCs in different locations to enhance overall system fault tolerance and prevent service interruptions caused by natural disasters [25]. Unlike TCP, RDMA handles data transmission directly on network hardware and bypasses the kernel network stack, achieving high performance and low CPU occupancy. As a result, adopting RDMA among cross-DC services is promising for meeting the strict performance demands of low latency and high throughput.

B. RDMA Congestion Control

Unlike TCP networks, RDMA uses Go-Back-N (GBN)[1] as its default packet loss recovery mechanism to simplify network interface card (NIC) logic. However, the GBN mechanism significantly reduces its tolerance to packet loss. To address this, PFC mechanism is introduced to the RoCEv2 protocol. When the depth of switch queue exceeds a predefined threshold, the switch sends a PFC pause frame to upstream devices (switches or NICs), instructing them to halt sending packets temporarily. This prevents packet loss caused by buffer overflow on switches. However, practical deployments reveal that PFC has significant drawbacks [13], [27]. As the PFC pause frame propagates upstream along the link, it may cause a flow that is not even present on its path to be hurt, which is known as the head-of-the-line blocking problem.

To mitigate these problems, various CC algorithms have been designed to reduce the occurrence of congestion and, in turn, the triggering of PFC. Among these, DCQCN [13] has become the *de facto* standard CC mechanism used by major RDMA NIC (RNIC) vendors. CSPs favor DCQCN for its simplicity and effectiveness. In DCQCN, if a switch's queue depth exceeds a pre-set ECN threshold (usually much smaller than the PFC threshold), the switch uses Random Early Detection (RED) to mark packets with ECN. When a NIC receives an ECN-marked packet, it sends a CNP back to the sender. Upon receiving this feedback, the sender reduces its sending rate to alleviate congestion. Other CC algorithms work in a similar way, e.g., HPCC [19] relies on switches adding in-network telemetry (INT) headers to carry detailed network information, and TIMELY [28], along with its improved version Swift [29], focuses on feeding round-trip time (RTT) information back to the sender.

C. Motivating Experiments

CC algorithms, e.g., DCQCN, perform well within DCs but poorly in long-haul RDMA. As shown in Figure 1, switches are divided into two tiers, with a full-mesh connection between tiers (also known as a ToR-Leaf topology). We use the same

[1]Note that ConnectX-5 and newer NICs support hardware retransmission, and ConnectX-6 Dx supports a proprietary selective repeat protocol [26].

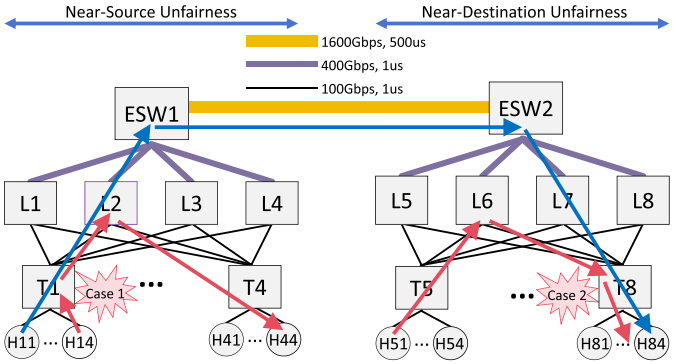


Fig. 1. The topology for motivation experiments.

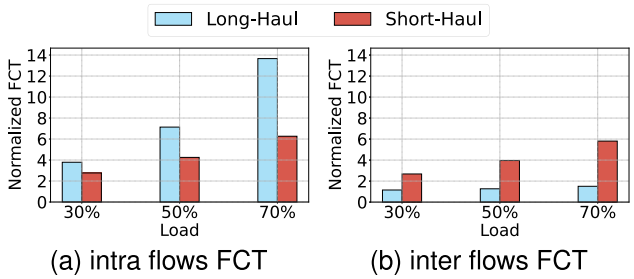


Fig. 2. Normalized FCT of DCQCN in long/short-haul scenarios.

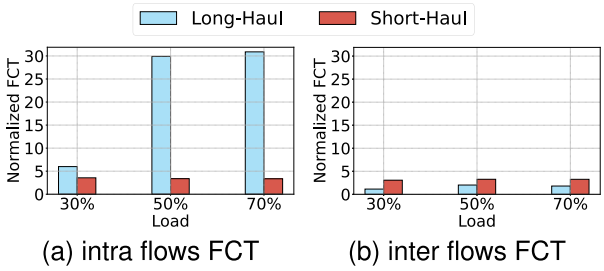


Fig. 3. Normalized FCT of HPCC in long/short-haul scenarios.

experimental setup in §VIII-A. To illustrate the impact of long haul on RDMA transmission, we compare the FCT of DCQCN [13] under inter-DC latencies of 500 microseconds (long-haul) and 1 microsecond (short-haul). As we can see from Figure 2, long-haul transmission increases the FCT of intra-DC flows by up to 119% while reducing the FCT of inter-DC flows by up to 74%. As network load increases, congestion becomes more frequent, exacerbating this unfairness issue.

Other CC algorithms [19], [28], [29] also suffer from this unfairness issue, thus perform poorly. HPCC relies on INT to measure per-RTT queue occupancy and compute the target sending rate. With more accurate feedback and rate computation, HPCC outperforms DCQCN in the short-haul scenario. However, in long-haul RDMA, the INT feedback is already stale when it reaches the sender, resulting in the unfairness issue similar to that of DCQCN, as shown in Figure 3. TIMELY [28] is initially designed for intra-DC networks, where the baseline RTT is on the order of microseconds, making absolute delay thresholds effective indicators of queue buildup. In long-haul RDMA, the baseline RTT can be several orders of magnitude larger (milliseconds) and

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.

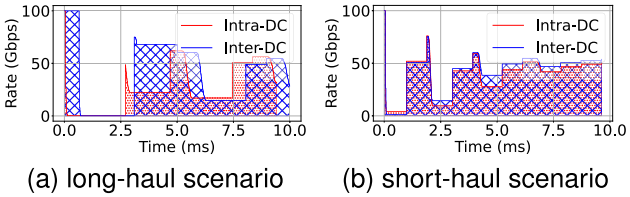


Fig. 4. Sender rate of DCQCN in long/short-haul scenarios.

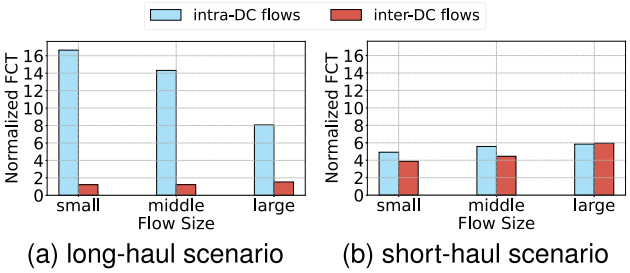


Fig. 5. Normalized FCT for different flow size in long/short-haul scenarios.

varies significantly across flows and paths. As a result, it is impossible to set its thresholds T_{low} and T_{high} in long-haul RDMA networks; therefore, we do not include it in our experiments for comparison.

A closer look into these CC algorithms reveals the underlying root causes of the unfairness issue. Long-haul scenarios result in significant physical distance differences between the two ends of different flows, causing the senders of intra-DC flows to receive congestion feedback much earlier than the senders of inter-DC flows. Consequently, the senders of intra-DC flows have to shoulder the entire responsibility for congestion before the senders of inter-DC flows receive any congestion feedback, leading to unfairness. Furthermore, we classify this unfairness into two types: one caused by congestion within the DC hosting the sender of the inter-DC flow, referred to as near-source unfairness (Case 1 in Figure 1); the other caused by congestion within the DC hosting the receiver of the inter-DC flow, referred to as near-destination unfairness (Case 2 in Figure 1). To further illustrate the unfairness, we take near-source congestion as an example, using the topology in Figure 1. We set up two 100Mb flows, one from H14 to H44, representing an intra-DC flow, and the other from H11 to H84, representing an inter-DC flow. To ensure controlled congestion, we configured T1 to connect only to L2, ensuring that congestion occurs exclusively at the T1 switch. Figure 4 illustrates how the NIC sending rates of intra/inter-DC flows are affected by the time difference in receiving CNPs. When the inter-DC latency is 500 microseconds (Figure 4a), the inter-DC flow maintains a higher sending rate most of the time due to the delayed congestion feedback. In contrast, when the inter-DC latency is only 1 microsecond (Figure 4b), both intra/inter-DC flows receive CNPs almost simultaneously, resulting in nearly synchronized sending rates.

Moreover, we find that in the long-haul scenario, intra-DC small flows (<10KB) experience severe slowdown, reaching up to 16× under 70% workload, as shown in Figure 5. In contrast, the slowdown of large intra-DC flows (>100KB) in the long-haul scenario is only about half as much, showing no significant difference compared with their performance in the short-haul scenario. Small flows suffer the most from unfairness, primarily due to their high sensitivity to latency. Therefore, we believe that an ideal solution to the unfairness issue should reduce the control-loop latency from the extremely high inter-DC scale to a level comparable to intra-DC latency.

D. Existing Studies Fall Short

Recent efforts have attempted to extend RDMA to support cross-DC scenarios; however, they cannot fully address the unfairness issue. Bifrost [16] and Swing [15] mainly focus on reducing the buffer size required by the PFC mechanism. Specifically, due to the delay of PFC pause frames, switches need a buffer of 2BDP (bandwidth delay product) to prevent buffer overflow. Additionally, due to the delay of PFC resume frames, switches require an extra 2BDP space to ensure no throughput loss. On one hand, Bifrost [16] observes that PFC resume frames could be sent earlier, eliminating the need for the additional 2BDP space to prevent throughput loss. On the other hand, Swing [15] adds PFC-Relay devices in long-haul links to share the 2BDP space requirement across multiple devices. They successfully reduce the total PFC buffer space requirement for DCI switches and enhance the performance of long-haul RDMA. However, neither of them notices the unfairness issue in long-haul RDMA networks. RoCC [30] and PACC [31] improve intra-datacenter RDMA performance by performing precise rate calculations in switches. Although their ideas are inspirational, they rely on modifying all switches, incurring high deployment obstacles. Besides, their methods are dedicated to intra-DC scenarios, which cannot be migrated directly to long-haul RDMA scenarios.

To the best of our knowledge, Annulus [17] is the first representative work highly related to the unfairness issue. It relies on L3-routed Quantized Congestion Notification (QCN) [32] to shorten feedback time from near-source bottleneck and directly eliminates the near-source unfairness. However, on one hand, as Annulus can only be deployed on Top-of-Rack (ToR) switches, it does not deal with the long latency of congestion feedback issued from bottlenecks near the receiver, and thus cannot properly address near-destination congestions. On the other hand, since Annulus requires L2 learning ability and Congestion Notification Message (CN Message) [32] format modification, it needs changes in the network protocol and thus the DC infrastructure (including switch and RNIC hardware). Recently BiCC [18] is designed to alleviate the hybrid traffic congestion in long-haul RDMA networks. To alleviate near-source congestion, BiCC leverages the sender-side DCI switch to transmit near-source feedback. However, BiCC does not specify implementation details, such as how to get the necessary sQPN field and how to send CNP correctly. To limit the queue size caused by near-destination hybrid traffic congestion and reduce average FCT, BiCC constrains inflight bytes between the receiver-side DCI switch and receiver to the BDP of intra-DC, and uses VOQs to temporarily store traffic with the same destination IP at the receiver-side DCI switch. Since flows with the same destination IP may

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.

take different paths, IP-level rate limiting is less accurate and effective compared to flow-level rate limiting.

Conventional wisdom that addresses this unfairness in TCP networks is also not suitable for RDMA. In TCP networks, this issue is typically referred to as *RTT fairness*. The underlying cause is fundamentally the same as the one we discussed, although the outcomes differ. TCP CC algorithms aim to increase the congestion window (cWND) by one segment per RTT. Therefore, for each received ACK, the cWND increases by $\frac{1}{M}$ of a segment (where M is the size of cWND). However, as short-RTT flows receive ACKs earlier, they will increase their cWNDs more quickly and ultimately occupy more network resources. Prague [33] identifies this fundamental cause and proposes to slow down the growth rate of the cWND for short-RTT flows. It simply sets a minimum value of 25ms for rtt_virt of all flows to reduce the gap between the RTTs of long and short flows. And it uses the value of $\frac{rtt_virt}{rtt_real}$ to instruct changes in the cWND, which essentially reduces the cwnd growth/decrease rate of short-RTT flows. GTCP [34] allows inter-DC flows to dynamically switch between a sender-based mode for high throughput and a receiver-driven mode to protect intra-DC traffic, based on intra-DC congestion signals. GEMINI [35] uses delay signals to bound WAN traffic and control end-to-end latency, and leverages ECN signals to manage shallow-buffered DCN queues, while adjusting the window based on RTT. However, on one hand, they are implemented within the Linux TCP stack and cannot be easily extended to hardened RNICs. On the other hand, most commodity RNICs [36] do not support window-based CC.

III. THEMIS DESIGN OVERVIEW

THEMIS aims to minimize unfairness after congestion occurs while maintaining compatibility with existing CC algorithms and DC infrastructure. To achieve the above objectives, THEMIS follows these three principles:

- For near-source/destination bottlenecks, the reaction time of both inter-DC and intra-DC traffic should be roughly consistent.
- Minimum hardware modifications or additional hardware functionality requirements should be imposed on existing DC devices (e.g., switches, NICs).
- THEMIS should operate without interfering with the original end-to-end CC (e.g., DCQCN).

A. Themis Workflow

Observation. The DCI-switch is capable of collecting relevant information about inter-DC traffic. Whether for the ECN marks placed by the congested switches or the congestion feedback (i.e., CNP) sent from the host, the DCI-switch can receive this information within the propagation delay of a single DC. As a result, DCI-switch can serve as a sweet point to address both near-source and near-destination unfairness. An alternative approach is to react at every potential congestion point in the network, e.g., ToR/Leaf switches. However, such a design requires modifications across all switches in large-scale DCs and faces significant deployment challenges in practice. Therefore, we choose to deploy THEMIS at DCI-switches, and also provide experimental comparisons with the ToR deployment solution in §VIII-C.

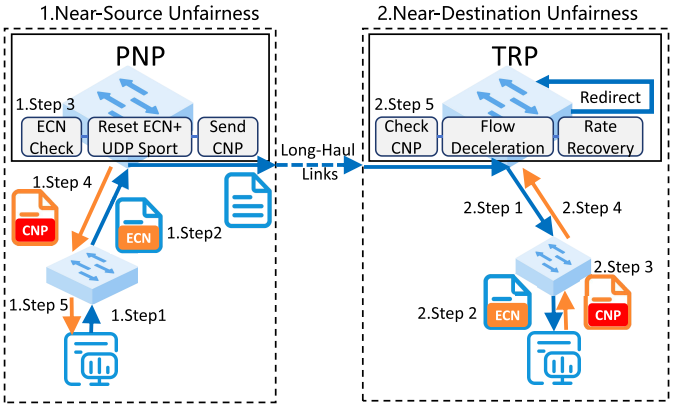


Fig. 6. Workflow overview of THEMIS.

Workflows. Figure 6 shows the workflow overview of THEMIS. When a near-source bottleneck happens, switches mark data packets with ECN. PNP clears the ECN mark and proactively sends a CNP to the corresponding sender, thereby mitigating the significant feedback delay caused by the long-haul and addressing near-source unfairness. As for near-destination unfairness, when the receiver receives ECN-marked data packets, it responds with CNPs. TRP can check these CNPs and decelerate the corresponding inter-DC flow temporarily before the sender starts to slow down.

B. Challenges and Requirements

However, implementing THEMIS at the DCI-switches presents two main challenges. First, RDMA packets only carry the destination QPN (dQPN) [37]. However, if we want the DCI-switch to send a CNP, we must ensure that the DCI-switch can obtain the sQPN[2] as soon as the first data packet arrives, to prevent a situation where the DCI-switch wants to send a CNP but lacks the sQPN. In BiCC, since the DCI-switch cannot obtain the sQPN, the sender is unable to determine which flow the CNP sent from the DCI-switch corresponds to. In addition, considering that large-scale DC may generate tens of thousands of flows per second, it is crucial for THEMIS to be designed with efficiency to minimize resource overhead at the DCI-switches.

Second, implementing flow control at the flow granularity on a switch without impacting innocent flows is also challenging. Directly applying queue-level PFC in THEMIS would clearly interfere with the performance of innocent flows sharing the same queue, which cannot apply different throttling strategies to flows in the same queue either. In addition, separating inter-DC and intra-DC traffic into different queues would double the number of queues required on switches. BiCC uses VOQs on the DCI-switches without imposing any limit on their queue lengths. Under severe congestion, this can lead to significant buffer overhead, placing high demands on

[2]Note that sQPN and dQPN are identifiers independently maintained on the host side.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.

switch resources. To ensure the deployability, THEMIS should avoid introducing advanced hardware features on a typical switch and keep buffer usage within controllable bounds.

IV. PROACTIVE NOTIFICATION POINT

The PNP introduces an additional control loop for inter-DC flows. In the case of near-source unfairness, the sender of inter-DC flows can receive congestion feedback within the propagation delay of a single DC. This significantly reduces the time gap between intra/inter-DC senders when near-source bottlenecks occur. To achieve this, we transmit the sQPN within the data packets without modifying the RDMA protocol or hardware. We also design a mechanism to align with DCQCN.

Generating CNPs. Switch is designed to be a packet forwarding device, not a packet generation device, thus cannot generate CNP without ground. Inspired by recent works on programmable switches [38], [39], PNP uses the replicator in the traffic manager to duplicate ECN-marked packets, and employs the editor in the egress pipeline to modify them into CNPs. Next, we will explain how we obtain the sQPN.

A straightforward solution is to modify the RDMA protocol to let data packets carry the sQPN, or to revise the CNP format so that the receiver can infer the sQPN based on other fields. However, due to the hardware-based nature of RDMA, these approaches require changes to RNIC hardware or protocol support. Although recent studies [40], [41] introduce limited support to extend flexibility for RNIC, these approaches are only research prototypes and have not been widely adopted, and relying on them reduces deployability. Another solution is to obtain the relevant information during the establishment of the RDMA connection. Since RoCEv2 uses TCP sockets or Communication Manager (CM) — which is also based on TCP socket — to establish connections between the sender and the receiver, a native solution is to obtain the sQPN from the connection establishment packets. However, this solution is ineffective because the DCI-switches cannot identify which TCP packet is used to establish the RDMA connection. Even if the host actively detects RDMA connections and transmits sQPN (e.g., by deploying eBPF to monitor connection-related kernel functions), the switch still needs to store the mapping between sQPNs and five-tuples of flows, which could consume a substantial amount of memory resources. Therefore, an ideal sQPN transmission mechanism should be able to embed sQPN information within data packets without requiring any modification to the RDMA protocol or hardware. In the generic RDMA driver of the Linux kernel [42], the maximum number of Reliable Connection QPs is set to 128K. Therefore, the transmission of sQPN requires finding 17 bits of available space within the data packet.

Fortunately, we observe that in RoCEv2, the UDP source port field is primarily used to provide entropy for load balancing (e.g., ECMP) and does not affect the RDMA protocol semantics.[3] Therefore, it can be safely repurposed without modifying the protocol. The rdma-core library provides the *mlx5dv_modify_qp_udp_sport* function, which allows users to modify the UDP source port associated with a QP. Therefore, we only need to call this function once for each QP to embed the sQPN, after which all subsequent packets can carry the sQPN in the UDP source port field. Since the UDP source port field contains only 14 bits, we embed the remaining 3 bits of the QPN into a hash of the dIP. Upon receiving a packet, the DCI-switch computes the hash of the dIP and concatenates it with the UDP source port to reconstruct the correct sQPN. Moreover, the DCI-switch sets the UDP source port of every data packet to the hash of the other four elements of the five-tuple, preventing potential information leakage over the long-haul links. Through this userspace verbs API, PNP can convey the sQPN through software-layer modifications without requiring changes to the RDMA protocol stack in the RNIC hardware.

However, a remaining challenge is that RDMA's QP creation function does not allow explicit assignment of QPNs, preventing us from ensuring that the lower 3 bits correspond to the desired hash value. In modern RDMA systems [44], [45], [46], QP pooling has become a common optimization to amortize connection setup costs and enhance scalability. Inspired by this, we create a QP pool and initialize it with QPs whose lower 3 bits of the QPN cover all 8 possible values, ensuring the availability of QPs with distinct suffixes. We record the count of QPs for each 3-bit QPN suffix in the pool. During connection establishment, a QP is selected based on the dIP's hash, and the pool is updated to ensure all eight possible suffix values remain represented. When a connection is terminated, the corresponding QP is returned to the pool, and redundant QPs are checked and destroyed. Consequently, by maintaining only a limited QP pool on the host instead of storing and querying sQPNs on the DCI-switch, our design guarantees accurate sQPN transmission with minimal overhead. We further discuss potential concerns regarding scalability and security in §IX.

Aligning with DCQCN behavior. To avoid interfering with DCQCN, we need to accomplish the following two tasks. First, as the sender has already reduced the rate upon receiving the CNP from DCI-switch, additional CNPs from the receiver will cause the inter-DC flow to be throttled even more severely. Therefore, the PNP clears data packets' ECN marks to ensure stability. Second, the frequency of CNP generation at DCI-switch should match the behavior of the end host. To achieve that, we leverage a register array in the data plane to record the timestamp of the last received ECN mark for each flow. Upon receiving an ECN-marked packet, PNP compares the timestamp in the packet with the one stored in the register. A CNP is generated and sent by the DCI-switch only if the difference between the two timestamps exceeds the *cnp_interval*, ensuring consistency with the interval used by DCQCN.

PNP overview. Figure 7 shows the full design of PNP. During QP creation, the dIP is hashed, and PNP selects a QP from the QP pool whose QPN suffix (the lower 3 bits) corresponds to the computed hash value. Subsequently, PNP calls the *mlx5dv_modify_qp_udp_sport* function to configure the UDP source port with the upper 14 bits of the QPN. When

[3]Note that if vendors prefer not to modify this field in practical deployments, they can instead adopt our previous design via eBPF [43].

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.

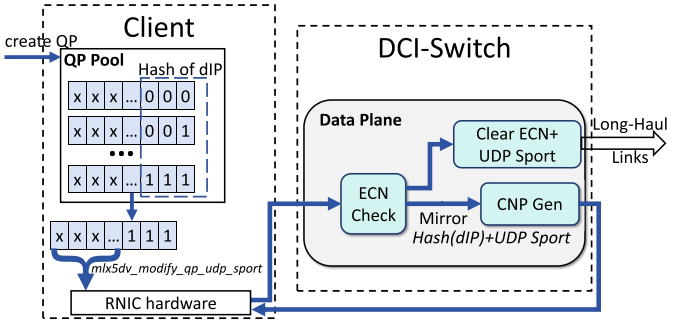


Fig. 7. Overview of PNP.

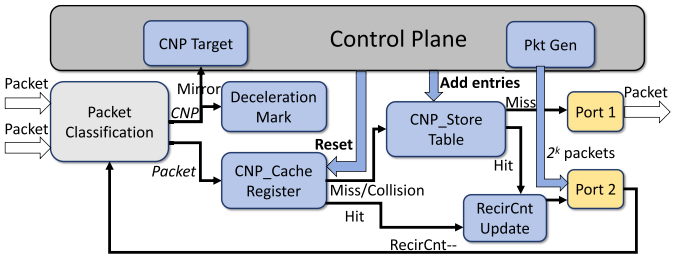


Fig. 8. Overview of TRP.

an ECN-marked data packet arrives at DCI-switch, it checks the timestamp of the most recent CNP sent for the flow. If no CNP has been sent within the past *cnp_interval*, it mirrors the data packet, concatenates the hash of the dIP with the UDP source port to get sQPN, modifies the necessary fields to form a CNP, and sends it back to the sender. DCI-switch resets ECN-mark and UDP source port, and forwards the data packet ordinarily.

V. Temporary Reaction Point

The TRP temporarily reduces the rate of the certain inter-DC flow upon receiving a CNP, effectively acting on behalf of the original sender until the sender receives the CNP and adjusts its rate. This ensures that inter/intra-DC flows respond to the near-destination bottleneck nearly simultaneously, significantly alleviating unfairness and congestion. To achieve this, TRP uses a combination of MAT and register arrays to respond to incoming CNPs in a timely, precise, and memory-efficient manner. It enables flow granularity control, applying differentiated rate-limiting strategies for individual flows.

A. Strawman Solution

There are two fundamental objectives in implementing TRP. First, although the number of inter-DC flows passing through the DCI-switches is massive, we observe in early experiments that only a small fraction contributes to the near-destination bottleneck. Therefore, it is important to ensure that rate reduction does not negatively impact the large number of normal flows. Second, different inter-DC flows contribute to congestion in varying degrees, necessitating differentiated rate reduction strategies. However, existing switch mechanisms like PFC operate at the granularity of ports for flow control, and can only pause traffic indiscriminately.

A straightforward solution is to separate inter-DC and intra-DC traffic into different queues during congestion. To distinguish between inter/intra-DC flows, DCI-switch can tag all passing packets with a dedicated 1-bit identifier, indicating whether the packet belongs to an inter-DC flow. Furthermore, when the switches detect that the queue depth exceeds the ECN threshold, subsequent inter-DC traffic should be placed in a lower-priority queue, thereby preserving the performance of intra-DC traffic. However, this approach presents two practical challenges. First, different network services already have distinct priority levels. Isolating inter-DC and intra-DC traffic would double the required number of queues. Second, intra-DC traffic typically dominates in volume [47], [48]. Assigning inter-DC traffic to a lower-priority queue risks starving inter-DC flows due to a lack of forwarding opportunities.

B. Flow Deceleration

In contrast, our approach does not introduce any additional queues on switches and allows for adaptive rate adjustments for inter-DC traffic, ensuring that different service requirements are satisfied. Our inspiration comes from the recirculation functionality of the switch pipeline. When a data packet is detected as requiring rate reduction, TRP redirects it to the designated recirculation ports. These ports serve as a temporary holding path for packets that require throttling, allowing each packet to loop a controlled number of times before being forwarded to its output port. However, because only a small number of packets enter the recirculation ports, the resulting queuing is insufficient, and packets traverse the recirculation path too quickly, undermining the intended rate reduction. Therefore, to achieve persistent queuing at the recirculation ports, TRP first pre-injects 2^k dummy packets into them via the control plane, which are continuously recirculated to ensure that the recirculation ports consistently maintain sufficient occupancy. This design not only achieves the desired flow deceleration, but also is isolated from the normal forwarding ports physically, thereby avoiding performance impacts on innocents. However, there remain two problems: (1) How can the DCI-switches respond quickly to CNPs and accurately identify the target inter-DC flows? (2) How do DCI-switches calculate the number of loops each data packet of the target flow should undergo?

To resolve the problem (1), for a received CNP, DCI-switch can store a hash of its IP and QPN in the register array, which will be used to judge whether the data packets from the opposite direction should experience a rate reduction. However, as the number of inter-DC flows is extremely large, the size of the register array should be large enough to reduce the hash collision, which would result in significant switch memory overhead. We observe that although the number of inter-DC flows is large, only a small number of them contribute to the near-destination congestion and should undergo a rate reduction. Therefore, MAT, which supports discrete matching, seems more promising. However, to insert an entry into the MAT, the data plane should forward CNPs to the control plane CPU, causing unacceptable delays. Therefore, we decide to use MAT as a fallback strategy for the register array to avoid the trade-off between updates and queries as shown in Figure 8.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.

Specifically, the data plane writes immediately to the register array — named CNP_Cache — upon receiving a CNP. At the same time, the CNP is sent to the switch CPU, where flow table inserts are performed. When a table entry is inserted, the control plane resets the register, making expired register entries available for reuse. Therefore, the CNP_Cache register array only needs to store CNPs within the delay of inserting a CNP_Store table entry, reducing its memory requirement. During runtime, each data packet is hashed using its *dstIp* and *dQPN* to check for a hit in the CNP_Cache register array. If there is a miss, the CNP_Store table is applied. If there is still no hit, it indicates the flow is not a target for decelerating.

Moreover, we need to design a throttling algorithm to address the problem (2). DCI-switch can determine the severity of congestion based on the frequency of CNP for the target flow. When the DCI-switch receives a CNP, it records the CNP reception time as cnp_time for the corresponding flow and increments its cnp_num counter, which will be used to compute the $loop_num$ parameter. Concretely, to achieve rapid rate reduction, the first CNP increases the $loop_num$ and changes the $flow_status$ to *throttled*, indicating that data packets of this flow need to be redirected to address congestion. Subsequently, every α CNP triggers an increment in $loop_num$. Meanwhile, the switch maintains a register at the recirculation port to record the number of packets entering the loop port, named $recycle_packet_count$, including the initially pre-injected 2^k dummy packets. Each packet reduces its $loop_num$ by one *step* during each recirculation. When $loop_num$ becomes less than or equal to zero, it indicates that the required throttling has been completed, and the packet is forwarded to the normal output port. The value of *step* is calculated as follows:

$$step = recycle_packet_count \gg k \tag{1}$$

In order to prevent inter-DC flows from being blocked due to continuously receiving CNPs, α itself is progressively augmented to ensure that the increment rate of $loop_num$ gradually slows with time. Meanwhile, to ensure the buffer occupancy of the recirculation ports remains within a controllable range, an upper limit is imposed on the number of the $loop_num$. To ensure that the upper bound of recirculation buffer occupancy remains near 2^{limit} packets, we dynamically adjust the maximum value of $loop_num$ according to the pressure on the recirculation port to ensure:

$$loop_num \le 2^{\gamma} - (recycle_packet_count - 1 \ll k) \gg shift \tag{2}$$

$$shift = limit - \gamma \tag{3}$$

where γ is pre-defined to guarantee that the maximum value of $loop_num$ is 2^{γ}. With this mechanism, different subflows derive varying $loop_num$ values, amplifying their time gap and thereby achieving rate reduction at the subflow granularity, while ensuring that the switch buffer remains within safe limits. If no CNP is received for a continuous period of β, we consider that the congestion has been alleviated. At this point, we change the $flow_status$ to *recover*, indicating that the flow has entered the rate recovery phase, and reset cnp_num to 0. At the same time, we apply a multiplicative decrease strategy to α to ensure that if the flow experiences subsequent congestion, it will undergo aggressive throttling. When a data packet is forwarded to its correct port, TRP removes its $loop_num$ header field. This algorithm ensures that, in most cases, buffer occupancy does not exceed $2^{limit} \times packet_size$. Under a large-scale and extreme congestion scenario, the buffer upper bound is capped by the limit enforced by PFC. At such congestion levels, end-to-end CC, due to its delayed reaction, would in fact trigger PFC pauses even earlier. A detailed description of the algorithm is seen in Algorithm 1.

Algorithm 1 Throttling Algorithm

Input: pkt, α, β, γ, k, $recycle_packet_count$
Output: pkt_loop_num, $flow_status$

```
if pkt is cnp then
    cnp_num ← cnp_num + 1
    flow_status ← throttled
    cnp_time ← now
    if (cnp_num mod α == 1) and (loop_num ≤ 2^γ − (recycle_packet_count − 1 ≪ k) ≫ shift) then
        loop_num ← loop_num + 1
        cnp_num ← cnp_num − α
        α ← α + 1
    end if
else
    if flow_status = throttled or flow_status = recover then
        pkt_loop_num ← loop_num
        redirect pkt
    else
        send pkt
    end if
end if
if now − cnp_time > β then
    flow_status ← recover
    α ← α/2
    cnp_num ← 0
end if
return:
```

In addition to throttling flows, TRP is also responsible for rate recovery. A naive solution is to set $loop_num$ to 0 once throttling is no longer required, allowing all data packets of the flow to be forwarded directly to the correct port. However, direct recovery can cause packet reordering. Consider the scenario shown in Figure 9: packets 1-3 are redirected to pre-filled ports and loop several times (Figure 9(a)(b)). If TRP then determines that the flow should resume its original rate and forwards packets 1-3 to the correct port, packets 4-6 and 1-3 may converge at the correct port, resulting in packet reordering in the queue (Figure 9(c)(d)). Since some NICs do not support out-of-order packet delivery, we specifically design a rate recovery mechanism to ensure the general applicability of THEMIS. To achieve rate recovery without causing packet reordering, we allow subsequent packets (e.g., packets 4-6 in the above diagram) to loop only once at the pre-filled port before being forwarded to the correct port. This ensures that subsequent packets will always arrive at the correct port later

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.

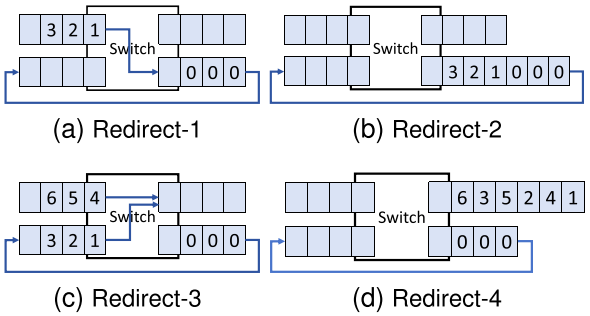


Fig. 9. Out of order scenario.

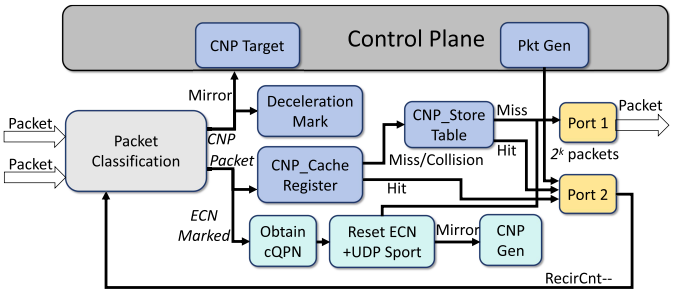


Fig. 10. Component layout of THEMIS.

than the earlier ones, and the unnecessary delay introduced to the overall inter-DC flow is minimal, limiting the delay to just one loop. Moreover, we use registers at the ingress to store the sequence number of data packets that have already been redirected to the correct port. If the sequence number of the current packet in the pre-filled port loop immediately follows the stored value, then there is no need to loop that packet, and TRP can forward it to the correct port.

VI. PUTTING IT TOGETHER

Cooperation of THEMIS. When an inter-DC flow triggers near-source congestion, PNP reduces its rate, which can prevent some congestion that would otherwise occur near the destination. Since PNP clears ECN marks, the flow will not activate TRP or the receiver's CNP generation if it no longer experiences congestion. Only when a rate-reduced inter-DC flow encounters congestion again in the destination DC will TRP be triggered by CNPs. As the flow has already been throttled, PNP helps mitigate buffer occupancy at the DCI-switch.

Cooperation with short-haul CC. Due to limited on-chip resources and computational constraints, THEMIS does not fully replicate the end-host functionality of CC, such as maintaining precise state or executing fine-grained rate updates. THEMIS achieves approximate fairness with short-haul DCQCN, as PNP implements a CNP generation that closely approximates the receiver-side feedback behavior of DCQCN, and TRP achieves rate reduction based on the frequency of CNPs. This approximation is sufficient as it operates at a much faster timescale in long-haul scenario, as shown in VIII-B.

Cooperation with long-haul CC. On one hand, PNP clears ECN marks, preventing long-haul DCQCN from being triggered. On the other hand, TRP provides a fast throttling mechanism, while long-haul DCQCN remains a slower, RTT-level end-to-end control loop; thus, TRP and long-haul DCQCN do not throttle concurrently. Consider a bottleneck link with capacity C. Suppose a packet experiences L recirculations at the receiver-side DCI-switch; then a packet consumes L forwarding opportunities. In large-scale DCs, since multiple flows traverse the DCI-switch simultaneously, each flow is allocated forwarding opportunities roughly proportional to its sending rate. Thus, the real rate of a flow in the throttling period becomes

$$R \approx \frac{R_0}{L} \tag{4}$$

where R_0 is the original sending rate. Assume n flows share the bottleneck, each initially sending at rate R_0. The aggregate input rate is $R_{\text{in}} = nR_0$, and under congestion ($nR_0 > C$), the total effective rate after L recirculations is

$$R_{\text{tot}} \approx n\frac{R_0}{L} \tag{5}$$

Congestion is relieved once

$$R_{\text{tot}} < C \tag{6}$$

which yields the required recirculation level

$$L^* > \frac{nR_0}{C} \tag{7}$$

In TRP, the increase of L is driven by congestion signals (i.e., CNPs). Let the CNP arrival rate be λ, and let the initial α in Algorithm 1 be α_0. The total number of CNPs is

$$N = \sum_{k=0}^{L-1}(\alpha_0 + k) \tag{8}$$

resulting in time to address this congestion

$$T = \frac{1}{\lambda}\left(L\alpha_0 + \frac{L(L-1)}{2}\right) \tag{9}$$

Substituting $L = L^*$, the time for TRP to suppress congestion can be approximated as

$$T \approx \frac{(nR_0)^2}{2\lambda C^2} \tag{10}$$

Importantly, the aggregate input rate nR_0 is upper-bounded by the link capacity, making T on the order of tens of microseconds. This is significantly shorter than the start of the sender's CC (typically > ms). Therefore, TRP effectively addresses congestion before long-haul CC enters its control phase. Moreover, most flows complete before long-haul CC significantly reduces their rates.

VII. IMPLEMENTATION

We implement a prototype of THEMIS on Tofino2 switch, and make our code publicly available [20]. Figure 10 illustrates the component layout of THEMIS on the data plane switching ASICs and the control plane switch CPUs.

The data plane part is implemented with ~600 lines of P4-16 code for the Intel Tofino2 ASIC. In THEMIS, since the number of concurrent flows causing near-source unfairness within a DC is small, we set the size of the CNP_Store table

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.

TABLE I
THEMIS PARAMETERS

Parameter	Description	Default Value
k	TRP pre-injects 2^k packets.	10
$limit$	Each recirculation port can accommodate up to 2^{limit} packets simultaneously.	16
α	Every α CNP triggers an increment in *loop_num*.	5
β	After no CNP is received for a duration of β, the flow enters the rate recovery phase.	500 microseconds
γ	2^γ is the upper bound of *loop_num*.	8

and the CNP_Cache register array to 1024. One practical issue here is that the ICRC checksum of CNP packets needs to be recalculated since DCI-switch changes its *dQPN* field. Prior studies have shown that calculating ICRC fits within the switch pipeline [18], [49]. THEMIS directly adopts their approach and calculates ICRC for CNP packets on the DCI-switch.

The control plane part is written in ~2K lines of C code. It is responsible for initializing the data plane, receiving daemon packets, establishing a mapping for inter-DC flow identification, updating MAT entries/registers according to the received CNP from the data plane, and injecting dummy packets.

The values of k, limit, and γ are determined by the buffer resources allocated to THEMIS. In our configuration, we set k = 10, limit = 16, and γ = 8. Each packet is 1 KB in size; thus, we pre-inject 1,024 dummy packets, each recirculation port consumes approximately 16 MB of buffer, and the maximum *loop_num* is set to 256. We implement α as 5 and β as 500 microseconds, and conduct a parameter sensitivity analysis in §VIII-B. We summarize all parameters of THEMIS and their configurations in Table I.

VIII. EVALUATION

In this section, we evaluate THEMIS via real-world P4 testbed experiment and large-scale NS-3 [50] simulation to answer the following questions:

- (1) How does THEMIS perform overall (§VIII-B)?
- (2) How necessary are the optimizations in THEMIS (§VIII-C)?
- (3) Can THEMIS be efficiently implemented on current switching hardware (§VIII-D)?

A. Experimental Setup

THEMIS setup. The topology in our setup uses the same ToR-Leaf structure as in Figure 1, where each DC has 4 Leaf and 4 ToR switches, with totaling 32 servers in the network. We set two inter-DC latency values as 500 microseconds (corresponding to a distance of 100km) and 1 microsecond (acting as a comparison with short distance, and referred to as "short" in the experiments). Therefore, the total maximum RTT within a DC is 8 microseconds, while the RTT between DCs is 1012 microseconds. As illustrated in HPCC [19], to maintain packet lossless for PFC, each ingress port requires a buffer of at least 2 × rate × delay, i.e., 25KB for 100Gbps links, 100KB for 400Gbps links, and 2GB for long-haul 1600Gbps links. We activate shared buffer across ports, and configure 32MB of buffer for intra-DC switches and approximately 3GB for DCI-switches. We configure a dynamic PFC threshold to trigger PFC when an ingress queue consumes more than 11% of the available buffer. We adopt the same ECN marking scheme as DCQCN [13]. For egress ports, we configure $(K_{\min}, K_{\max}, P_{\max})$ as (400KB, 1600KB, 0.2), (800KB, 6400KB, 0.2), and (1600KB, 25600KB, 0.2) for 100Gbps, 400Gbps, and 1600Gbps links. We use ECMP as our routing scheme.

Baselines. We compare THEMIS to DCQCN, DCQCN_short, Annulus and BiCC. We normalize the FCT by flow's performance when using the network exclusively. To emulate complex traffic patterns, we use a composite workload consisting of three types of traffic: WebSearch traffic trace [19], micro-burst traffic, and large cross-DC pipeline parallelism flows. Under load levels of 30%, 50%, and 70%, the three traffic types occupy the link bandwidth in proportions of (10%, 10%, 10%), (20%, 20%, 10%), and (30%, 30%, 10%). Primary metrics include the normalized average FCT, the normalized P99 FCT and the occurrence of PFC triggers. For native DCQCN, we use the parameters from HPCC [19]. Additionally, since Annulus has not been open-sourced, we simulate its characteristics in NS-3 by replacing its L3-routed QCN mechanism at ToR switches with a mechanism that returns CNPs at the ToR switches. Moreover, for BiCC, since the CNPs it sends do not carry the sQPN field, we enhance BiCC to enable the sender to respond correctly. Additionally, as BiCC does not discuss the buffer limitations of VOQ, we allow the VOQ to grow indefinitely in our implementation.

B. Overall Effectiveness

THEMIS improves intra-DC traffic performance. Figure 11 shows the average FCT for intra-DC and inter-DC traffic achieved by each scheme as the fabric load varies from 30% to 70%. We break down the results for small [0, 10KB], medium (10KB, 100KB], and large (>100KB) flows. For intra-DC flows, THEMIS achieves an FCT reduction of at least 21.65% and up to 81.19% compared to DCQCN across three load levels and three flow size categories. As expected, since small flows are more affected by unfairness issue, THEMIS provides the most protection for small flows, resulting in the highest FCT reduction. Additionally, as network load increases and congestion worsens, unfairness becomes more frequent under DCQCN, causing the normalized average FCT ratio between intra-DC and inter-DC flows to grow. In contrast, THEMIS ensures that inter-DC flows take their fair share of rate reduction, significantly alleviating unfairness. Moreover, compared to Annulus, THEMIS achieves up to a 66.97% reduction in FCT for intra-DC flows smaller than 10KB. This improvement is primarily due to THEMIS's TRP, which effectively alleviates near-destination unfairness, a problem Annulus cannot address. Because BiCC constrains the inflight bytes between the receiver-side DCI switch and the receiver to the intra-DC BDP and does not impose an interval between consecutive CNP generations, inter-DC large flows struggle to

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.

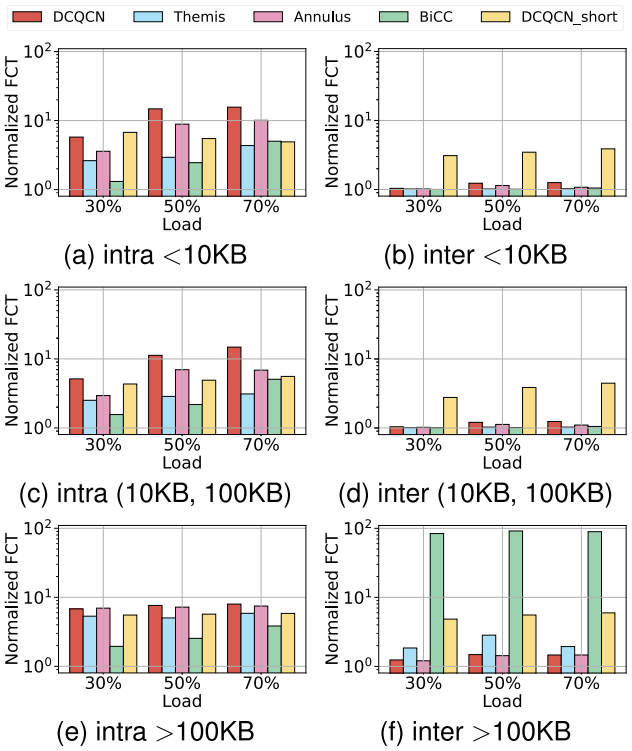


Fig. 11. Average FCT for different flow size (log scale).

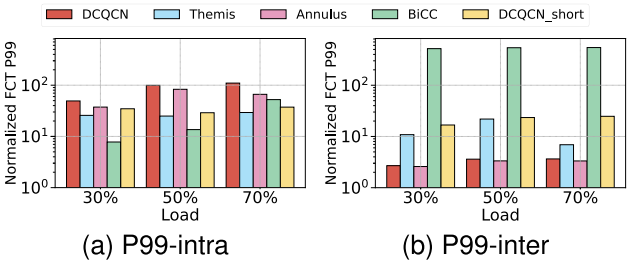


Fig. 12. The 99th percentile of FCT (log scale).

fairly acquire bandwidth under congestion. Although this strategy protects short flows and enables their rapid completion, it severely limits the rate of large inter-DC flows and makes them nearly unable to transmit. Furthermore, we observe that under THEMIS, the FCT of intra-DC flows is even better than that under DCQCN_short. This benefit mainly comes from the combined effect of THEMIS's two modules. The PNP enables faster CNP feedback through the DCI-switch, which is even a few microseconds faster than the short-haul scenarios. Similarly, the TRP responds to CNPs with rate reduction sooner than it does in short-haul cases.

THEMIS improves intra-DC P99 FCT. The benefits of THEMIS are also apparent in tail latencies. Figure 12 shows the 99 percentile (P99) FCTs. Due to the complexity of traffic patterns, severe congestion can occur even under 30% and 50% load, leading to high P99 FCTs. Compared to DCQCN, THEMIS effectively protects intra-DC flows from excessive rate reduction, reducing P99 FCT by 47.67%, 74.87%, and 73.36% at 30%, 50%, and 70% load, respectively. Due to the buffer limitation imposed on the recirculation ports under

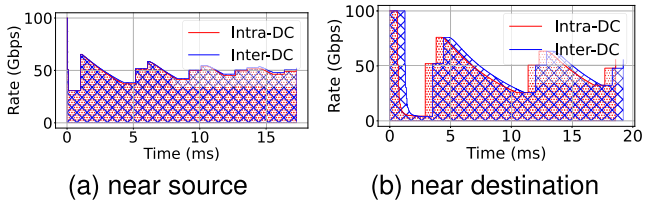


Fig. 13. Sending rate of THEMIS.

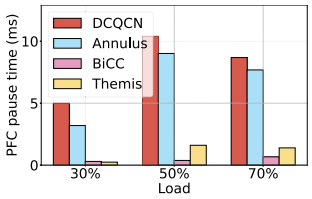


Fig. 14. PFC activation.

severe congestion, some inter-DC flows cannot achieve the intended rate reduction under high workload. This results in a decrease in inter-DC P99 FCT and partial performance degradation for intra-DC flows at 70% workload. Nevertheless, THEMIS still outperforms other schemes under high workload. Since BiCC constrains inflight bytes between the receiver-side DCI switch and the receiver to the BDP of intra-DC, it severely impacts the rate of long inter-DC flows during congestion, leading to a significant increase in the P99 of inter-DC flows.

THEMIS addresses unfairness. THEMIS addresses near-source and near-destination unfairness effectively. Figure 13 shows the sender rate in near-source and near-destination scenarios after deploying THEMIS. THEMIS achieves Jains Fairness Index values of 0.99 and 0.96, compared with 0.87 for DCQCN, as shown in Figure 4 (a). Compared to Figure 4, THEMIS ensures fairness while quickly mitigating congestion. Since TRP throttles traffic at the DCI-switch, the representation of inter-DC flow rate reduction in the near-destination scenario experiences some delay.

THEMIS decreases the occurrence of PFC triggers. Figure 14 shows the PFC pause time under different schemes. DCQCN triggers large-scale PFC pauses in long-haul scenarios, while Annulus and BiCC slightly reduce the PFC pause time. THEMIS achieves fast congestion avoidance across all three load levels. Compared to DCQCN and Annulus, THEMIS reduces the PFC pause time by 84%-95% and 81%-92%. BiCC reduces PFC by severely limiting large inter-DC flows; however, this strategy significantly slows down their transmission. By significantly reducing the occurrence of PFC, THEMIS improves network throughput and enhances network stability.

THEMIS improves overall network performance. Essentially, THEMIS protects intra-DC flows by ensuring that inter-DC flows bear their fair share of the congestion responsibility. Despite this, THEMIS still improves overall network performance. Figure 15 shows the normalized average FCT for all intra/inter-DC flows. As can be seen, THEMIS outperforms DCQCN, Annulus and BiCC across all three load levels. By alleviating fairness issues and rapidly mitigating congestion, THEMIS increases overall network throughput. BiCC performs

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.

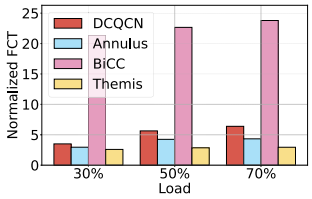


Fig. 15. Overall average FCT.

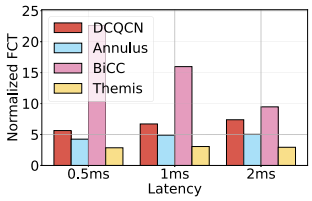


Fig. 16. Overall average FCT under different latencies.

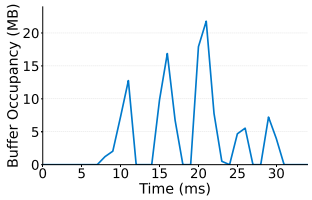


Fig. 17. Buffer utilization of recirculation in TRP.

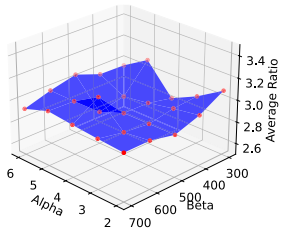


Fig. 18. Influence of α and β.

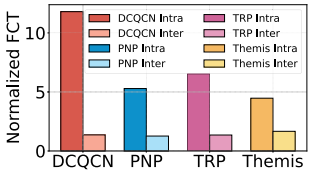


Fig. 19. Effectiveness of TRP and PNP under 70% workload.

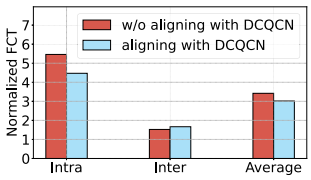


Fig. 20. Aligning with DCQCN behavior.

worse compared to Annulus due to its inability to generate effective CNPs. We record the locations of packets marked with ECN due to congestion and find that 33.9%–38.3% of congestion occurs on Leaf switches. Annulus cannot address congestion on Leaf switches and near-destination unfairness, leading to worse performance than THEMIS. As the workload increases, the fairness issue becomes more severe, further amplifying the advantages of THEMIS.

Buffer utilization of TRP. We set only one port and one queue for TRP to recirculate packets, and we record the buffer utilization of the TRP's recirculation port under 70% workload in NS-3. As in Figure 17, recirculation consumes ~20MB DCI-switch buffer even under severe congestion. Considering DCI-switches typically have a deep buffer [35], this buffer utilization is acceptable for current DCI-switch hardware. TRP imposes limited overhead and is capable of scaling to high-load scenarios.

Sensitivity to long-haul latency. To evaluate the robustness of our design under different propagation delays, we conduct experiments with long-haul RTTs of 500μs, 1ms, and 2ms, corresponding to link distances of approximately 100km, 200km, and 400km, respectively. Figure 16 shows that the performance advantage of THEMIS remains consistent across all latency scales. In DCQCN, the feedback delay increases as the long-haul latency grows, leading to performance degradation as the long-haul distance increases. In contrast, THEMIS decouples reaction time from RTT, therefore exhibits nearly identical performance across different RTTs.

Parameters tuning. The parameters k, limit, and γ are determined by the buffer resources allocated to THEMIS. The other two parameters, α and β, are set to 5 and 500 microseconds, respectively. We conduct a sensitivity analysis on these two parameters. As shown in Figure 18, different parameter settings have limited impact on THEMIS, with the maximum difference in average FCT being less than 9%. Additionally, when congestion becomes severe, it is recommended to reduce the value of α and increase the value of β to reduce the rate of target flows more aggressively.

C. Efficiency Breakdown

We first conduct an ablation study to evaluate the effectiveness of PNP and TRP. As shown in Figure 19, both are essential to alleviate the unfairness issue. When using only TRP, under severe congestion, all inter-DC flows requiring throttling are redirected to the same port, resulting in a serious slowdown. When traffic triggers near-source congestion, PNP sends back CNP and clears ECN. Only if it subsequently triggers near-destination congestion, it will be throttled at TRP. The combination of both modules achieves a more effective overall performance. Here, we further conduct several experiments to demonstrate the necessity of the designs we incorporate into PNP and TRP.

Aligning with DCQCN behavior. To align with DCQCN behavior, we clear the ECN markings carried in inter-DC packets and use the same *cnp_interval* as DCQCN. As shown in Figure 20, our efforts to align with DCQCN minimize the sacrifice of inter-DC flows as much as possible to alleviate unfairness. Without clearing the ECN markings, the receiver continues to send CNPs after THEMIS generates CNPs, causing inter-DC traffic to experience a slowdown again. Without using *cnp_interval*, the inter-DC traffic is severely suppressed due to receiving more CNPs. These lead to rate instability of inter-DC traffic, which significantly increases the FCT of inter-DC traffic and causes persistent congestion.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.

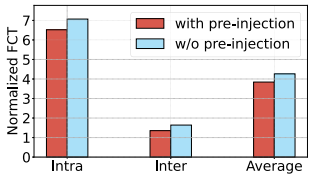


Fig. 21. Effect of packets pre-injection.

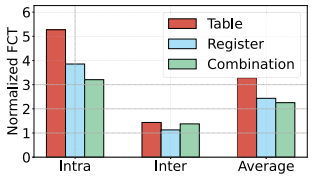


Fig. 22. Flow deceleration.

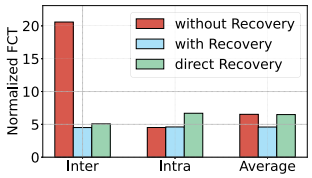


Fig. 23. Rate recovery under GBN.

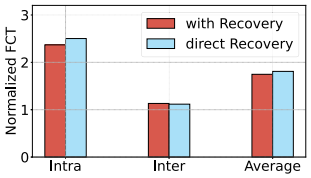


Fig. 24. Rate recovery under IRN.

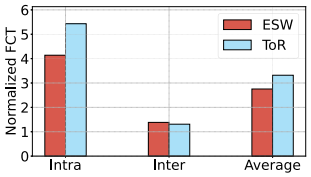


Fig. 25. Comparison of implementing THEMIS at DCI-switch or ToR.

TABLE II
RESOURCE OVERHEAD OF THEMIS ON SWITCH

Computational		Memory	
ALUs	Gateways	SRAM	TCAM
34.38%	14.84%	5.00%	0.00%

Packets pre-injection. We pre-inject dummy packets in TRP to ensure that each recirculation achieves the intended throttling effect. We compare the performance of TRP with and without pre-injecting packets. Without packets pre-injection, the recirculation ports cannot maintain stable delays during the initial congestion phase, which prevents TRP from working effectively and degrades overall network performance by 10% as shown in Figure 21.

Flow deceleration. We set the latency between the data plane and control plane to 1 millisecond [51] and configure the size of CNP_Cache register array to 1024. As shown in Figure 22, using only the CNP_Store results in high latency, leading to insufficient rate reduction for inter-DC flows. Using only the CNP_Cache leads to frequent hash collisions, resulting in incorrect rate reductions for inter-DC flows without effectively mitigating congestion. In contrast, our combination of CNP_Store and CNP_Cache can decelerate flows timely and correctly, achieving the lowest FCT.

Rate recovery. We perform several long inter-DC flows and short intra-DC flows that cause congestion to evaluate the performance of inter-DC flows after the intra-DC flows have finished. As in Figure 23, compared to no recovery and direct recovery — the naive solution mentioned in §V — the Rate Recovery module we designed achieves fast recovery after congestion is alleviated and reduces the overall average FCT by 29.7%. Without Rate Recovery, the inter-DC flows remain throttled for an extended period even after congestion is resolved, significantly increasing their FCT. Direct recovery causes packet reordering, leading to retransmissions under the GBN mechanism and reducing the overall network throughput.

Furthermore, we implement TRP on an NS-3 simulation framework [52] built upon IRN (Improved RoCE NIC) [53]. We evaluate the performance of direct recovery on IRN. Since IRN supports out-of-order packet delivery, direct recovery does not cause retransmissions, thus achieving performance comparable to rate recovery as shown in Figure 24. Therefore, in the case of IRN-compatible NICs, TRP employs direct recovery without the need for an additional rate recovery mechanism.

Deploying THEMIS at different locations. We explore an alternative design by deploying THEMIS at different locations. To achieve this setup, ToR switches have to maintain a global mapping from servers to their corresponding DCs, enabling them to distinguish between inter/intra-DC data packets and CNPs. Even deployed, the ToR-based solution still exhibits on average 20% lower performance compared to the original THEMIS, as illustrated in Figure 25. This is primarily because ToR switches lack timely visibility into congestion occurring at the upper layers. Considering both deployability and performance, deploying THEMIS at DCI-switches is a more effective choice.

D. Hardware Testbed

Resource overhead. We evaluate the resource overhead of THEMIS on the Tofino2 switch, and open-source the resource consumption source files [20]. Table II shows THEMIS's hardware resource utilization. THEMIS consumes minimal computational and memory resources, with its computational overhead primarily stemming from ICRC checksum calculation, leaving sufficient capacity for concurrent execution of traditional forwarding behaviors [54]. Besides, THEMIS does not modify network protocols, and data packets do not carry any additional fields.

Stress test. To further demonstrate that Tofino2 can satisfy the resource requirements of THEMIS under high workload, we conduct a stress test at 100Gbps by modifying the

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.

implementation so that 50% of packets undergo recirculation while the remaining 50% trigger CNP generation [20]. We additionally modify the CNP format to prevent sender-side rate reduction. Experimental results show that Tofino2 can still sustain this functionality at the full 100Gbps line rate. In real long-haul DCs, although the aggregate link bandwidth is much higher (Tbps scale), switches only need to process a small fraction of packets rather than all packets as in this stress test. Therefore, the processing requirements of THEMIS remain well within the capabilities of modern switches.

IX. DISCUSSION

Extension to other CC algorithms. THEMIS can be easily extended to support other CC algorithms. In terms of PNP, for INT-based HPCC [19], PNP can parse this INT information to get the load of the egress ports and decide whether to mirror the packets back as feedback. For delay-based CC algorithms, e.g., TIMELY [28] and Swift [29], PNP can return feedback to the sender, enabling the end host to compute the intra-DC RTT and quickly react to the congestion within the near-source DC. In terms of TRP, to support DCQCN, we use CNP frequency to detect congestion. When extended to other CC algorithms, TRP can utilize other signals (e.g., INT, RTT) to determine the appropriate degree of rate reduction for inter-DC flow.

Security considerations for sQPN transmission. PNP embeds the sQPN within the UDP source port, which may raise concerns that the QPN could be exposed to potential attackers and exploited for malicious actions. Actually, during RDMA connection establishment, the source and destination already exchange QPNs and related metadata. If an attacker can monitor packets on the link, they could capture this information; therefore, PNP does not increase the attack surface of the QPNs. In addition, the near-source DCI-switch resets the UDP source port for all data packets, preventing potential risks outside DC.

Scalability of QP Pool. The QP Pool maintains a small number of pre-allocated QPs to ensure correct sQPN assignment under the current hardware and driver limitation of 128K QPs. Prior work [55], [56], [57] shows that a large number of concurrent QPs on existing RNICs (e.g., Mellanox ConnectX-5 and ConnectX-6) can lead to severe performance degradation. As hardware capabilities and architectures continue to improve, future RNICs may support more than 128K QPs. PNP can be easily scaled by increasing the number of hash bits. Although this increases the number of QPs maintained in the QP Pool, the overhead remains acceptable on RNICs that support ultra-large QPs.

THEMIS on commodity switches. Although our THEMIS prototype is implemented on programmable switches, our approach is not tied to it and can be extended to commodity switches. PNP detects ECN-marked packets, uses the packet replicator engine to mirror them, and modify them into CNPs. TRP detects CNPs, redirects target packets to the ingress ports of DCI-switch, as well as records *loop_num* in their header. These are capable and common functions in modern commodity switches. As switch hardware is becoming more capable and programmable, we believe THEMIS can feasibly be applied to next-generation commodity switches.

X. RELATED WORK

Besides the most relevant work we have discussed in §II, our work is also inspired by the following topics.

QP management. Scalable QP management is a critical research topic in large-scale RDMA systems. KRCore [45] employs QP pooling to enable QP virtualization and sharing, amortizing the per-connection overhead. SMART [46] and SwiftRDMA [58] identify contention among QPs and optimize QP scheduling. SRNIC [55] introduces a cache-free QP scheduler and increases the number of on-chip QPs to enhance scalability. SRC [57] decouples network connections from QPs, significantly improving RDMA scalability. Justitia [59] and PeRF [60] further optimize QP management in multi-tenant environments. By maintaining a lightweight QP Pool, THEMIS enables obtaining specific QPNs, thereby enabling the transport of sQPN.

Programmable switches. Our work is also inspired by the recent trends using programmable switches in diverse applications in networking and distributed systems. Representative aspects include load balancing [52], [54], [61], network monitoring [38], [62], attack defense [37], [63], [64], and key-value stores [65], [66]. Inspired by these works, THEMIS is designed to implement the fairness maintenance mechanisms on programmable switches to alleviate unfairness issues in long-haul RDMA networks, which has distinct optimization mechanisms and implementation details.

XI. CONCLUSION

Deploying RDMA over long-haul networks can lead to severe congestion-induced unfairness and performance degradation. To address this issue, we propose THEMIS, a fairness maintenance patch for long-haul RDMA networks that requires modifications solely at DCI-switches. THEMIS consists of two key modules: PNP and TRP. When DCI-switch detects inter-DC flow packets marked with ECN, PNP rapidly generates CNPs and sends them to the sender. This significantly reduces the time gap in receiving congestion feedback between inter-DC and intra-DC flow senders, addressing near-source unfairness. Upon receiving a CNP from the receiver, TRP temporarily rate-limits the target inter-DC flow before the sender reacts, mitigating near-destination unfairness. Through P4 testbed experiments and large-scale NS-3 simulations, we demonstrate the effectiveness and deployability of THEMIS.

REFERENCES

[1] A. Dragojevi, D. Narayanan, M. Castro, and O. Hodson, "FaRM: Fast remote memory," in *Proc. 11th USENIX Symp. Networked Syst. Design Implement.*, 2014, pp. 401–414.

[2] A. Dragojević et al., "No compromises: Distributed transactions with consistency, availability, and performance," in *Proc. 25th Symp. Operating Syst. Princ.*, Oct. 2015, pp. 54–70.

[3] T. E. Anderson et al., "Assise: Performance and availability via client-local NVM in a distributed file system," in *Proc. OSDI*, 2020, pp. 1011–1027.

[4] S.-Y. Tsai, Y. Shan, and Y. Zhang, "Disaggregating persistent memory and controlling them remotely: An exploration of passive disaggregated key-value stores," in *Proc. USENIX ATC*, 2020, pp. 33–48.

[5] A. Kalia, M. Kaminsky, and D. G. Andersen, "FASST: Fast, scalable and simple distributed transactions with two-sided (RDMA) datagram RPCS," in *Proc. OSDI*, 2016, pp. 185–201.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.

[6] M. Zhang, Y. Hua, P. Zuo, and L. Liu, "Ford: Fast one-sided RDMA-based distributed transactions for disaggregated persistent memory," in *Proc. FAST*, 2022, pp. 51–68.

[7] J. Yang, J. Izraelevitz, and S. Swanson, "Orion: A distributed file system for non-volatile main memory and RDMA-capable networks," in *Proc. FAST*, 2019, pp. 221–234.

[8] P. Li, Y. Hua, P. Zuo, Z. Chen, and J. Sheng, "ROLEX: A scalable RDMA-oriented learned key-value store for disaggregated memory systems," in *Proc. FAST*, 2023, pp. 99–114.

[9] J. Xue, Y. Miao, C. Chen, M. Wu, L. Zhang, and L. Zhou, "Fast distributed deep learning over RDMA," in *Proc. 14th EuroSys Conf.*, Mar. 2019, pp. 1–14.

[10] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proc. HotCloud*, 2010, pp. 1–7.

[11] OpenAI.(Nov. 2024). *ChatGpt*. [Online]. Available: https://chatgpt.com/

[12] SemiAnalysis.(Sep. 2024). *Multi-Datacenter Training: Openai's Ambitious Plan to Beat Google's Infrastructure*. [Online]. Available: https://semianalysis.com/2024/09/04/multi-datacenter-training-openais/

[13] Y. Zhu et al., "Congestion control for large-scale RDMA deployments," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 523–536, Sep. 2015.

[14] W. Bai et al., "Empowering Azure storage with RDMA," in *Proc. NSDI*, 2023, pp. 49–67.

[15] Y. Chen et al., "Swing: Providing long-range lossless RDMA via PFC-relay," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 1, pp. 63–75, Jan. 2023.

[16] P. Yu et al., "Bifrost: Extending RoCE for long distance inter-DC links," in *Proc. IEEE 31st Int. Conf. Netw. Protocols (ICNP)*, Oct. 2023, pp. 1–12.

[17] A. Saeed et al., "Annulus: A dual congestion control loop for datacenter and WAN traffic aggregates," in *Proc. Annu. Conf. ACM Special Interest Group Data Commun. Appl., Technol., Architectures, Protocols Comput. Commun.*, Jul. 2020, pp. 735–749.

[18] Z. Wan et al., "BiCC: Bilateral congestion control in cross-datacenter RDMA networks," in *Proc. IEEE Conf. Comput. Commun.*, May 2024, pp. 1381–1390.

[19] Y. Li et al., "HPCC: High precision congestion control," in *Proc. ACM Special Interest Group Data Commun.*, Aug. 2019, pp. 44–58.

[20] (2025). *Themis*. [Online]. Available: https://github.com/Networked-System-and-Security-Group/Themis/tree/TON

[21] S. Jain et al., "B4: Experience with a globally-deployed software defined wan," *SIGCOMM*, vol. 43, no. 4, pp. 3–14, 2013.

[22] Facebook Engineering.(2017). *Building Express Backbone: Facebook's New Long-Haul Network*. [Online]. Available: https://engineering.fb.com/2017/05/01/data-center-engineering/building-express-backbone-facebook-s-new-long-haul-network/

[23] McKinsey & Company.(2023). *Ai Power: Expanding Data Center Capacity To Meet Growing Demand*. [Online]. Available: https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/ai-power-expanding-data-center-capacity-to-meet-growing-demand

[24] Y. Gao et al., "When cloud storage meets RDMA," in *Proc. NSDI*, 2021, pp. 519–533.

[25] Mertech.(2023). *Cloud Outages: Causes & Risks (and How To Handle Them)*. [Online]. Available: https://www.mertech.com/blog/cloud-outages-handling-guide

[26] NVIDIA Corporation.(2023). *Ethernet Network*. [Online]. Available: https://docs.nvidia.com/networking/display/winof2v320/ethernet

[27] S. Wang et al., "LoRDMA: A new low-rate DoS attack in RDMA networks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2024, pp. 1–18.

[28] R. Mittal et al., "TIMELY: RTT-based congestion control for the datacenter," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 537–550, Sep. 2015.

[29] G. Kumar et al., "Swift: Delay is simple and effective for congestion control in the datacenter," in *Proc. ACM SIGCOMM Conf. Appl. Technol. Archit. Protocols Comput. Commun. (SIGCOMM)*, 2020, pp. 514–528.

[30] P. Taheri, D. Menikkumbura, E. Vanini, S. Fahmy, P. Eugster, and T. Edsall, "RoCC: Robust congestion control for RDMA," in *Proc. 16th Int. Conf. Emerg. Netw. Exp. Technol.*, Nov. 2020, pp. 17–30.

[31] X. Zhong, J. Zhang, Y. Zhang, Z. Guan, and Z. Wan, "PACC: Proactive and accurate congestion feedback for RDMA congestion control," in *Proc. IEEE Conf. Comput. Commun.*, May 2022, pp. 2228–2237.

[32] N. Finn. (2010). IEEE802.1QAU Congestion Notification. [Online]. Available: https://1.ieee802.org/dcb/802-1qau/

[33] (2024). *Prague Congestion Control*. [Online]. Available: https://www.ietf.org/archive/id/draft-briscoe-iccrg-prague-congestion-control-04.html

[34] S. Zou, J. Huang, J. Liu, T. Zhang, N. Jiang, and J. Wang, "GTCP: Hybrid congestion control for cross-datacenter networks," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2021, pp. 932–942.

[35] G. Zeng et al., "Congestion control for cross-datacenter networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 5, pp. 2074–2089, Oct. 2022.

[36] N. Mellanox. (2024). *Connectx-6*. [Online]. Available: https://www.nvidia.com/en-sg/networking/ethernet/connectx-6/

[37] J. Xing, K.-F. Hsu, Y. Qiu, Z. Yang, H. Liu, and A. Chen, "Bedrock: Programmable network support for secure RDMA systems," in *Proc. 31st USENIX Security Symp.*, 2022, pp. 2585–2600.

[38] G. Li et al., "IMap: Fast and scalable in-network scanning with programmable switches," in *Proc. NSDI*, 2022, pp. 667–681.

[39] M. Zhang et al., "IMap: Toward a fast, scalable and reconfigurable in-network scanner with programmable switches," *TIFS*, vol. 19, pp. 601–615, 2023.

[40] A. Skiadopoulos et al., "High-throughput and flexible host networking for accelerated computing," in *Proc. OSDI*, 2024, pp. 405–423.

[41] C. Zhao, J. Min, M. Liu, and A. Krishnamurthy, "White-boxing RDMA with packet-granular software control," in *Proc. NSDI*, 2025, pp. 427–449.

[42] M. Technologies, N. Networking, and T. L. K. Community. (2025). MLX5core: Mellanox Connectx Ethernet and RDMA Driver for Linux Kernel. [Online]. Available: https://github.com/torvalds/linux/blob/master/drivers/net/ethernet/mellanox/mlx5/core

[43] Z. Niu, M. Zhang, J. Zhang, R. Xie, Y. Yang, and X. Hu, "THEMIS: Addressing congestion-induced unfairness in long-haul RDMA networks," in *Proc. IEEE 33rd Int. Conf. Netw. Protocols (ICNP)*, Sep. 2025, pp. 1–13.

[44] B. Rothenberger, K. Taranov, A. Perrig, and T. Hoefler, "ReDMArk: Bypassing RDMA security mechanisms," in *Proc. 30th USENIX Secur. Symp.*, 2021, pp. 4277–4292.

[45] X. Wei, F. Lu, R. Chen, and H. Chen, "KRCORE: A microsecond-scale RDMA control plane for elastic computing," in *Proc. USENIX ATC*, 2022, pp. 121–136.

[46] F. Ren, M. Zhang, K. Chen, H. Xia, Z. Chen, and Y. Wu, "Scaling up memory disaggregated applications with SMART," in *Proc. 29th ACM Int. Conf. Architectural Support Program. Lang. Operating Syst.*, Volume 1, Apr. 2024, pp. 351–367.

[47] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proc. 10th ACM SIGCOMM Conf. Internet Meas.*, Nov. 2010, pp. 267–280.

[48] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, "Inside the social network's (datacenter) network," in *Proc. ACM Conf. Special Interest Group Data Commun.*, Aug. 2015, pp. 123–137.

[49] W. Mansour, N. Janvier, and P. Fajardo, "FPGA implementation of RDMA-based data acquisition system over 100-GB Ethernet," *IEEE Trans. Nucl. Sci.*, vol. 66, no. 7, pp. 1138–1143, Jul. 2019.

[50] NS-3project. (Feb. 2025). *Ns-3 Simulator*. [Online]. Available: https://www.nsnam.org/

[51] C. Zeng et al., "Tiara: A scalable and efficient hardware acceleration architecture for stateful layer-4 load balancing," in *Proc. 19th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2022, pp. 1345–1358.

[52] C. H. Song, X. Z. Khooi, R. Joshi, I. Choi, J. Li, and M. C. Chan, "Network load balancing with in-network reordering support for RDMA," in *Proc. SIGCOMM*, 2023, pp. 816–831.

[53] R. Mittal et al., "Revisiting network support for RDMA," in *Proc. SIGCOMM*, 2018, pp. 313–326.

[54] R. Miao, H. Zeng, C. Kim, J. Lee, and M. Yu, "SilkRoad: Making stateful layer-4 load balancing fast and cheap using switching ASICs," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2017, pp. 15–28.

[55] Z. Wang et al., "SRNIC: A scalable architecture for RDMA NICs," in *Proc. NSDI*, 2023, pp. 1–14.

[56] Z. Wang et al., "Fast, scalable, and accurate rate limiter for RDMA NICs," in *Proc. ACM SIGCOMM Conf.*, Aug. 2024, pp. 568–580.

[57] Y. Zhao, R. Shu, and Y. Xiong, "SRC: A scalable reliable connection for RDMA with decoupled QPs and connections," in *Proc. 9th Asia–Pacific Workshop Netw.*, Aug. 2025, pp. 44–50.

[58] Y. Huang, Y. Qiu, Y. Xiao, A. Bhatnagar, S. Ratnasamy, and A. Chen, "Exposing RDMA NIC resources for software-defined scheduling," in *Proc. 9th Asia–Pacific Workshop Netw.*, Aug. 2025, pp. 1–8.

[59] Y. Zhang, Y. Tan, B. Stephens, and M. Chowdhury, "Justitia: Software multi-tenancy in hardware kernel-bypass networks," in *Proc. NSDI*, 2022, pp. 1307–1326.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.

[60] S. Lee, M. Choi, I. Yeom, and Y. Kim, "PeRF: Preemption-enabled RDMA framework," in *Proc. USENIX ATC*, 2024, pp. 209–225.
[61] N. Katta, M. Hira, C. Kim, A. Sivaraman, and J. Rexford, "HULA: Scalable load balancing using programmable data planes," in *Proc. Symp. SDN Res. (SOSR)*, 2016, pp. 1–12.
[62] Y. Zhou et al., "HyperTester: High-performance network testing driven by programmable switches," in *Proc. 15th Int. Conf. Emerg. Netw. Exp. Technol.*, Dec. 2019, pp. 30–43.
[63] M. Zhang et al., "Poseidon: Mitigating volumetric DDoS attacks with programmable switches," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2020, pp. 1–18.
[64] Z. Liu et al., "Jaqen: A high-performance switch-native approach for detecting and mitigating volumetric DDoS attacks with programmable switches," in *Proc. USENIX Secur. Symp.*, 2021, pp. 3829–3846.
[65] X. Jin et al., "NetCache: Balancing key-value stores with fast in-network caching," in *Proc. 26th Symp. Operating Syst. Princ.*, Oct. 2017, pp. 121–136.
[66] M. Liu, L. Luo, J. Nelson, L. Ceze, A. Krishnamurthy, and K. Atreya, "IncBricks: Toward in-network computation with an in-network cache," in *Proc. 22nd Int. Conf. Archit. Support Program. Lang. Oper. Syst.*, 2017, pp. 795–809.

Menghao Zhang received the B.S. and Ph.D. degrees in computer science from Tsinghua University, China. He is currently an Associate Professor with the School of Software, Beihang University. His research interests include programmable networks, high-performance networks, networked LLM systems, and network security.

Zihan Niu received the B.S. degree from the School of Software, Beihang University, where he is currently pursuing the master's degree. His research interests include programmable networks, RDMA networks, and transport protocols.

Jue Zhang (Graduate Student Member, IEEE) received the B.S. degree from the School of Computer Science and Engineering, Beihang University. He is currently pursuing the Ph.D. degree with the School of Software, Beihang University. His research interests include programmable networks and routing protocols.

Dongyang Wang received the B.S. and Ph.D. degrees in computer science from the University of Science and Technology of China. He is currently a Network Research and Development Expert at ByteDance. His research interests include RDMA networks, collective communication libraries, and LLM systems.

Renjie Xie received the Ph.D. degree from Tsinghua University. He is currently an Associate Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology. His current research interests include networks and machine learning security.

Yuan Yang received the B.S., M.S., and Ph.D. degrees from Tsinghua University. He was a Visiting Ph.D. student at The Hong Kong Polytechnic University. He is currently an Associate Researcher with the Department of Computer Science and Technology, Tsinghua University. His major research interests include computer network architecture and routing protocols.

Mingwei Xu received the B.S. and Ph.D. degrees from Tsinghua University. He is currently a Full Professor with the Department of Computer Science, Tsinghua University. His research interests include computer network architecture, high-speed router architecture, and network security.

Chunming Hu received the Ph.D. degree in computer science from Beihang University in 2006. He is currently a Professor with the School of Software, Beihang University. His research interests include distributed systems, system virtualization, data management, and processing systems.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on November 05,2026 at 02:14:25 UTC from IEEE Xplore. Restrictions apply.