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

Zihan Niu[1], Menghao Zhang[1], Jue Zhang[1], Renjie Xie[2], Yuan Yang[3], Xiaohe Hu[4]
[1]Beihang University [2]Guilin University of Electronic Technology [3]Tsinghua University [4]Infrawaves

***Abstract*—RDMA is promising for enhancing the performance of cross-datacenter (DC) services. However, deploying RDMA over wide-area 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 testbed and large-scale simulations. Compared to DCQCN, Annulus and BiCC, THEMIS reduces the intra-DC FCT by up to 79.2%, 63.6% and 55.6%, and decreases overall FCT by up to 61.2%, 31.9% and 59.5% respectively.**

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) [8], [9], [3], [39], [16], [52], [45], [20], [44]. 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 [47], [6], [2], [33], [36]. 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 wide-area 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 [54] 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.

979-8-3315-0376-5/25/$31.00 ©2025 IEEE

While recent efforts have been devoted to extending RDMA to cross-DC services [4], [7], [46], [35], [40], they do not fully address the unfairness issue. Swing [7] and Bifrost [46] 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. Closely related to the unfairness issue, Annulus [35] 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) [40] 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 [54],

[21], 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 external switches (ESW) connecting the DC to the Wide Area Networks (WAN), 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 as near-source unfairness), THEMIS at the ESW 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 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 client Queue Pair Number (QPN), a field essential for generating congestion feedback, i.e., CNP. This makes it difficult for the switch to notify the sender correctly. Moreover, as the number of flows in DCs is extremely large, THEMIS should be resource-efficient to avoid overwhelming switch resources. 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 ESW. For PNP, by leveraging eBPF techniques on the client side (i.e., the sender) to capture the client QPN (cQPN) and asynchronously transmitting this information to the ESW, it can promptly generate a CNP to the sender. In addition, PNP also incorporates event-driven flow state maintenance and TCP handshake monitor to reduce resource overhead at ESW. 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.

We implement a prototype of THEMIS in both NS-3 simulation and hardware testbed, and make its source code publicly available at GitHub [38]. We compare THEMIS with DCQCN, Annulus and BiCC under various workloads. Compared to DCQCN, THEMIS reduces the intra-DC FCT by 47.2%-79.2% and decreases the PFC triggers by 93%-96%. Compared to Annulus and BiCC, THEMIS reduces intra-DC FCT by 30.6%-63.6% and 25.1%-55.6% with better deployability. Although THEMIS slightly reduces the rate of inter-DC flows, it improves fairness and stability, thereby decreasing the overall FCT by up to 61.2%, 31.9%, and 59.5%, 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 65.0% and 67.4%. And one of our optimizations, e.g., alignment with DCQCN, achieves a 37.5% 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, §V).
- We implement an open-source prototype of THEMIS and evaluate THEMIS via testbed experiments and NS-3 large-scale simulations (§VI, §VII).

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

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 wide-area 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 [4], [14], [11]. 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 [24]. 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 [36]. 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 [13]. 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 [26]. Unlike TCP, RDMA handles data transmission directly on network hardware, bypassing the kernel network stack and achieving high performance. As a result, adopting RDMA among cross-DC services is promising for meeting 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

[1]Note that ConnectX-5 and newer NICs support hardware retransmission, and ConnectX-6 Dx supports a proprietary selective repeat protocol [31].

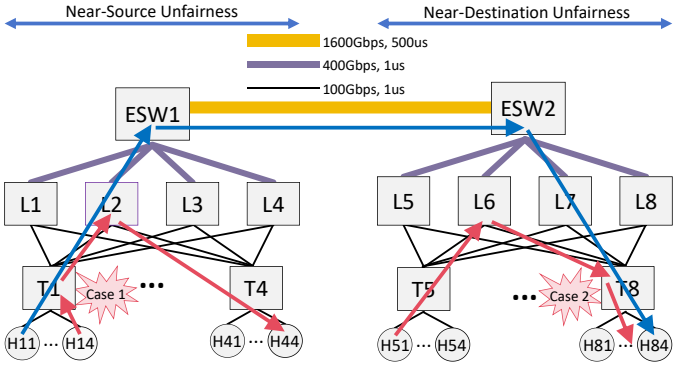


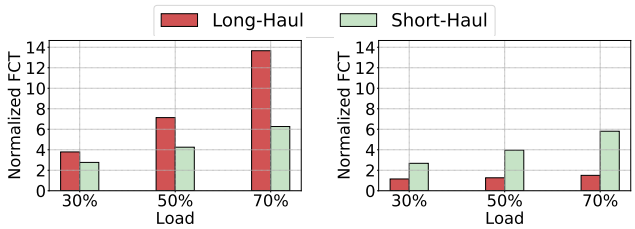
Figure 1: The topology for motivation experiments.

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 excessively deep switch queues. However, practical deployments reveal that PFC has significant drawbacks [54], [42]. 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 [54] 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 [21] relies on switches adding in-network telemetry (INT) headers to carry detailed network information, and TIMELY [28], along with its improved version Swift [18], focuses on feeding round-trip time (RTT) information back to the sender.

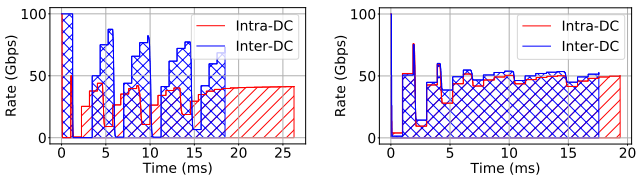
C. Motivating Experiments

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 experimental setup in §VII-A. To illustrate the impact of long haul on RDMA transmission, we compare the FCT of DCQCN [54] 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 118% while reducing the FCT of inter-DC flows by up to 74%. As network load increases, congestion becomes more frequent, exacerbating this unfairness issue.



(a) intra flows FCT (b) inter flows FCT

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



(a) Throughput in long-haul scenario. (b) Throughput in short-haul scenario.

Figure 3: Throughput of DCQCN in long/short-haul scenarios.

Other CC algorithms [21], [28], [18] also suffer from the similar unfairness issue, thus perform poorly, as detailed in Appendix X-A.

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 inter-DC flow senders. Consequently, intra-DC flows must shoulder the entire responsibility for congestion before inter-DC flow senders 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 3 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 3(a)), 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 3(b)), both intra/inter-DC flows receive CNPs almost simultaneously, resulting in nearly synchronized sending rates.

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 [46] and Swing [7] 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 (one-hop 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 [46] 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 [7] adds PFC-Relay devices to share the 2BDP space requirement across multiple devices. They successfully reduce the total PFC buffer space requirement for DCI switches. However, neither of them notices the unfairness issue in long-haul RDMA networks.

To the best of our knowledge, Annulus [35] is the first representative work highly related to the unfairness issue. It relies on L3-routed Quantized Congestion Notification (QCN) [12] 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) [12] format modification, it needs changes in the network protocol and thus the DC infrastructure (including switch and RNIC hardware). Recently BiCC [40] 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 cQPN 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 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 [10] 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 [55] 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 [49] 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 [25] do not support window-based CC.

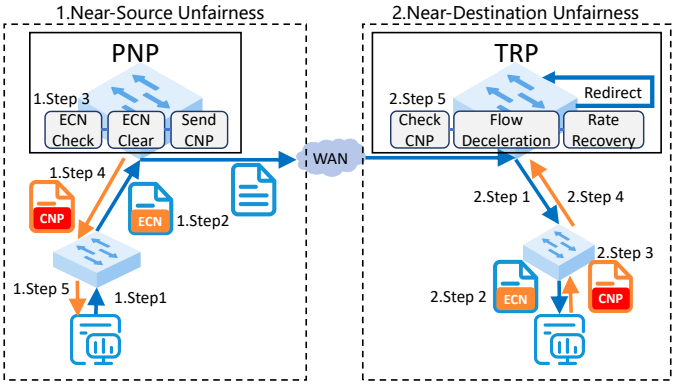


Figure 4: Workflow overview of THEMIS.

III. 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 ESW 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 ESW can receive this information within the propagation delay of a single DC. As a result, ESW can serve as a sweet point to address both near-source and near-destination unfairness.

Workflows. Figure 4 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 has started to slow down.

B. Challenges and Requirements

However, implementing THEMIS at the ESW presents two main challenges. First, RDMA packets only carry the server-side QPN (sQPN) [43]. However, if we want the ESW to send a CNP, we must ensure that the ESW can obtain the cQPN[2] as soon as the first data packet arrives, to prevent a situation where the ESW wants to send a CNP but lacks the cQPN. In BiCC, since the ESW cannot obtain the cQPN, the sender is unable to determine which flow the CNP sent from the ESW 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 ESW.

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 ESW without imposing any limit on their queue lengths. Under severe congestion, this can lead to significant buffer overhead, placing high demands on switch resources. To ensure the deployability, THEMIS should avoid introducing advanced hardware features on a typical switch.

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, the cQPN information is asynchronously sent to ESW via a sender-side daemon process using eBPF. We also design mechanisms to align with DCQCN and reduce switch memory overhead.

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 [19], [50], 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 cQPN.

A straightforward solution is to modify the RDMA protocol to let data packets carry the cQPN, or to revise the CNP format so that the receiver can infer the cQPN based on other fields. However, due to the hardware-based nature of RDMA, these approaches require redesigning the RNIC hardware, which 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 cQPN and sQPN mapping from the connection establishment packets. However, this solution is ineffective because the ESW cannot identify which TCP packet is used to establish the RDMA connection.

THEMIS leverages daemon process on the sender to asynchronously send information to the ESW control plane. In the RoCEv2 protocol, before the sender and the receiver perform a RDMA communication, the *ibv_modify_qp* function is used to set the QP state sequentially to INIT, Ready to Receive (RTR), and Ready to Send (RTS). THEMIS uses eBPF to monitor this function and capture cQPN at the host side, and mark daemon packets with lossless DSCP value to ensure delivery. Subsequently, the corresponding entries are inserted into the QP_field table. Since eBPF can obtain and transmit the cQPN during the INIT stage, the control plane can insert the table entry before the first RDMA data packet arrives at ESW. We conduct relevant experiments to validate it further (§VII-D).

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 ESW, additional CNPs from the receiver will cause the inter-DC flow to be throttled even more severely. Therefore, the ESW clears ECN marks to ensure stability. Second, the frequency of CNP generation at ESW 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, the system compares the timestamp in the packet with the one stored in the register. A CNP is generated and sent by the ESW only if the difference between the two timestamps exceeds the *cnp_interval*, ensuring consistency with the interval used by DCQCN on the host.

Event-driven flow state maintenance. THEMIS leverages eBPF to extract the cQPN, sQPN and dstIP parameters from the *ibv_modify_qp* function, and sends them asynchronously to the ESW control plane. The srcIP in the daemon packet's header as well as the dstIP and sQPN in the payload form a unique match field, with cQPN as an action parameter. A packet replicated from an ECN-marked packet will pass through this QP_field table, where its BTH header's *destination QPN* field is modified, converting it into a CNP directed to the original sender. In addition, when the host invokes the *ibv_destroy_qp* function, the ESW control plane is also notified to delete the corresponding QP_field entry and reset the register storing its CNP timestamp.

Reducing switch memory overhead. Due to the existence of VM/container migrations, the eBPF program at the host side cannot simply determine whether an established RDMA connection is intra/inter-DC based on the destination IP. Simply storing all the daemon packets into the ESW indiscriminately would waste the scarce data plane resources. A naive solution

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

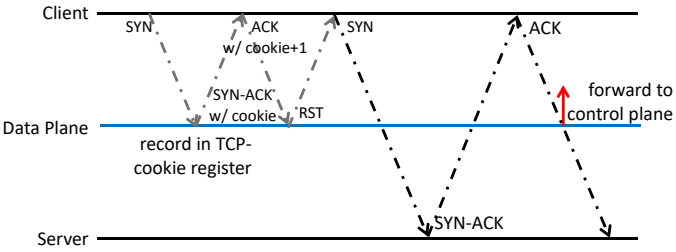


Figure 5: TCP handshake monitor.

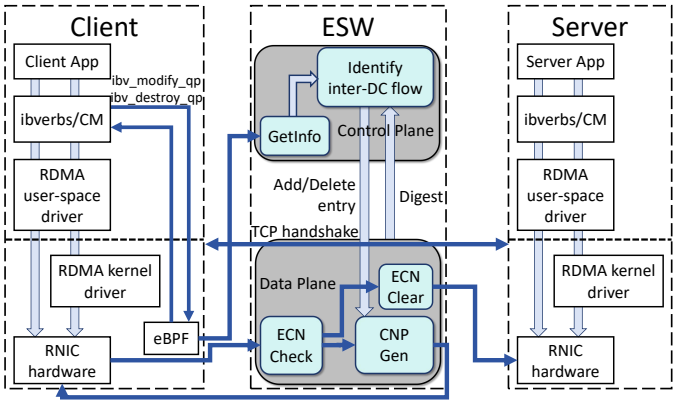


Figure 6: Overview of PNP.

is to rely on RTT measurements, however, this approach has two significant issues. First, setting a rough RTT threshold to differentiate intra/inter-DC flows can result in misclassification, causing the ESW to miss obtaining the cQPN for specific inter-DC flows. Second, RTT is derived from ACKs received after partial RDMA traffic has already been received. By the time the ESW obtains the cQPN, it is already too late.

Fortunately, we observe that the ESW naturally does not receive any intra-DC traffic due to its topological position[3]. Therefore, the data plane of the ESW can monitor the TCP connection establishment procedure to collect information about inter-DC flows. To prevent SYN-flood attacks [41] from consuming ESW resources, we design an SYN-Cookie-based mechanism in the data plane. When ESW receives an SYN packet, it replies an SYN-ACK packet with a cookie, instead of maintaining a state of TCP flow in the data plane. If ESW receives an ACK packet with the correct sequence number, it replies with a TCP RST packet to re-establish transmission and record the flow in the TCP-cookie register array. Subsequently, the TCP connection can re-establish the handshake procedure. Since SYN-cookie validation only involves retransmissions within DC, it introduces merely tens of microseconds additional delay, which is negligible compared to the milliseconds-level inter-DC TCP handshake delay. The data plane only forwards flow information to the control plane upon detecting an ACK packet, i.e., during the third handshake, as shown in Figure 5. The QP_field table entries are inserted after receiving daemon packets from the host side only if the control plane recognizes the flow as an inter-DC flow.

[3]Note that in a few topologies, the ESW may receive a portion of intra-DC traffic, but this only adds resource usage without compromising correctness.

PNP overview. Figure 6 shows the full design of PNP. The data plane of ESW monitors TCP handshake and forwards the IP of inter-DC flows to the control plane to make sure the QP_field table only contains entries for inter-DC flows. When ESW receives daemon packets for inter-DC flows, it records its QP information in the QP_field table. When an ECN-marked data packet arrives at ESW, ESW 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, modifies the necessary fields to form a CNP, and sends it to the sender. ESW clears ECN-mark and forwards the data packet ordinarily. The eBPF process monitors *ibv_destroy_qp* function and informs the ESW control plane to delete the corresponding entry in the QP_field table.

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 ESW is massive, 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, ESW 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 [5], [34]. 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

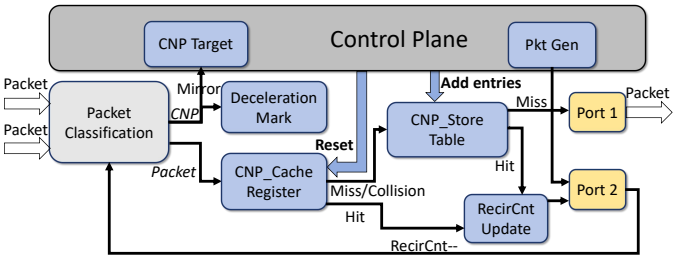


Figure 7: Overview of TRP.

for inter-DC traffic, ensuring that different service requirements are satisfied. Our inspiration comes from the recirculation functionality of the switch pipeline. TRP first pre-injects a few dummy packets into designated ports via the control plane, which will be circulated continuously. When a data packet is detected as requiring rate reduction, it is redirected to these pre-filled ports, where it loops a predefined number of times before being forwarded to its proper port. Therefore, these looped packets not only obtain the desired flow deceleration, but also are isolated from the normal forwarding ports physically, thereby avoiding performance impacts on innocents. However, there remain two problems: (1) How can the ESW respond quickly to CNPs and accurately identify the target inter-DC flows? (2) How does ESW calculate the number of loops each data packet of the target flow should undergo?

To resolve the problem (1), for a received CNP, ESW can store a hash of its *srcIp* and *cQPN* 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 7. 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 *sQPN* 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). ESW can determine the severity of congestion based on the frequency of CNP for the target flow. When the ESW 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*. 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. With this mechanism, different subflows derive varying *loop_num* values, amplifying their time gap and thereby achieving rate reduction at the subflow granularity. If no CNP is received for a continuous period of β, we consider 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. A detailed description of the algorithm is seen in Algorithm 1 in Appendix X-B.

In addition to throttling flows, TRP is also responsible for rate recovery. However, direct rate recovery can cause packet reordering. On one hand, for RNICs that support out-of-order delivery [32], TRP can perform direct recovery without impacting performance, as discussed in Appendix X-E. On the other hand, for RNICs without such functionality, we design a rate recovery mechanism that avoids packet reordering, as detailed in Appendix X-D and Appendix X-E.

VI. Implementation

We implement a prototype of THEMIS on Tofino switch, and make our code publicly available [38]. Appendix X-C 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 ∼500 lines of P4-16 code for the Intel Tofino ASIC. In THEMIS, since the number of concurrent flows causing near-source unfairness within a DC is small, we set the size of the setCNP table, the CNP_Store table, 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 ESW changes its *dstQPN* field. However, ICRC fields are redundant for RoCEv2 packets as Ethernet frames already have checksums [43]. This feature was inherited from the Infiniband version of RDMA, and can be disabled in RoCEv2 settings. Our setup disables ICRC to resolve this issue. We implement α as 5 and β as 500 microseconds, and conduct a parameter sensitivity analysis in Appendix X-F.

The control plane part is written in ∼1K lines of C code. It is responsible for initializing the data plane, receiving daemon packets, establishing mapping for inter-DC flow identification, updating MAT entries/registers according to the received CNP from the data plane, and injecting dummy packets.

VII. Evaluation

In this section, we evaluate THEMIS via real-world P4 testbed experiment and large-scale NS-3 [30] simulation to answer the following questions: (1) How does THEMIS perform overall (§VII-B)? (2) How necessary is the optimizations in THEMIS (§VII-C)? (3) Can THEMIS be efficiently implemented on current switching hardware (§VII-D)?

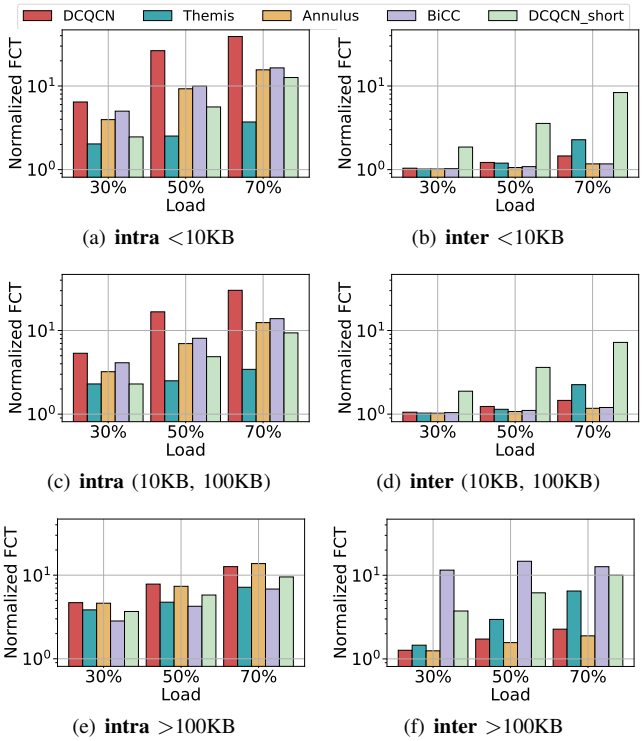
A. Experimental Setup

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. All switches in the topology have a buffer capacity of 16MB. Additionally, we configure the dynamic PFC threshold to trigger PFC when an ingress queue consumes more than 11% of the available buffer as illustrated in HPCC [21]. 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). The maximum RTT within a DC is 8 microseconds, while the maximum RTT between DCs is 1012 microseconds. We use ECMP as our routing scheme.

Baselines. We compare THEMIS to DCQCN, DCQCN_short, Annulus and BiCC under WebSearch traffic trace [21]. We normalize the FCT by flow's performance when using the network exclusively. 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 [21]. 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 sent by BiCC do not carry the cQPN field, the sender cannot 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 8 shows the average FCT for intra-DC and inter-DC traffic achieved by each scheme as the fabric load varies from 30% to 70% for WebSearch workload. 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 18.16% and up to 90.47% compared to DCQCN across three load levels and three flow size categories. As expected, since small flows are more affected by switch queuing delays, 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 76.21% reduction in FCT for intra-DC flows smaller

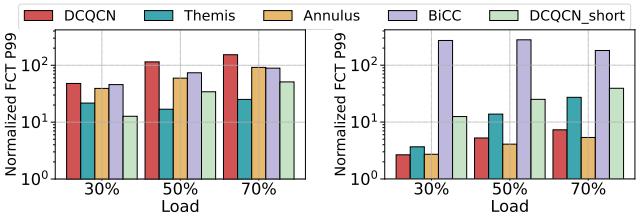


(a) **intra** <10KB (b) **inter** <10KB

(c) **intra** (10KB, 100KB] (d) **inter** (10KB, 100KB]

(e) **intra** >100KB (f) **inter** >100KB

Figure 8: Average FCT for different flow size (log scale).

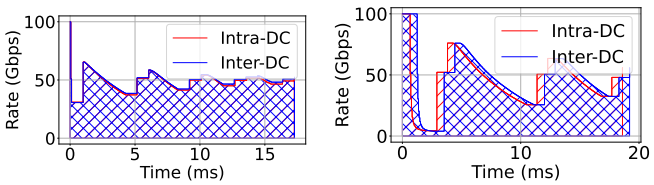


(a) **P99-intra** (b) **P99-inter**

Figure 9: The 99 percentile of FCT (log scale).

than 10KB. This improvement is primarily due to THEMIS' TRP, which effectively alleviates near-destination unfairness, a problem Annulus cannot address. Since the sender cannot correctly process the CNPs sent by BiCC's Near Source Loop, its Near Destination Loop has to take full responsibility for mitigating the unfairness issue. This causes the Near Destination Loop to impose restrictions on a large number of inter-DC flows, significantly increasing their FCT. 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' two modules. The PNP enables faster CNP feedback through the ESW, 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 9 shows the 99 percentile FCTs. Compared to DCQCN, THEMIS effectively protects intra-DC flows from excessive rate reduction, reducing P99 FCT by 54.84%, 85.30%, and 83.61% at 30%, 50%, and 70% load, respectively. Since BiCC constrains



(a) **near source** (b) **near destination**

Figure 10: Throughput of THEMIS.

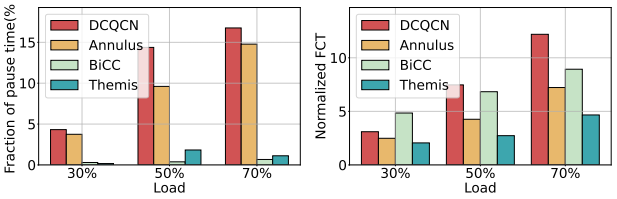


Figure 11: PFC activation. Figure 12: Overall average FCT.

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 10 shows the throughput in near-source and near-destination scenarios after deploying THEMIS. Compared to Figure 3, THEMIS ensures fairness while quickly mitigating congestion. Since TRP throttles traffic at the ESW, the representation of inter-DC flow rate reduction in the near-destination scenario experiences some delay.

THEMIS decreases the occurrence of PFC triggers. Figure 11 shows the PFC pause time under different schemes. DCQCN triggers large-scale PFC pauses in long-haul scenarios, while Annulus slightly reduces 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 93%-96% and 92%-95% respectively. BiCC's Near Destination Loop does not rely on feedback and limits the inflight bytes of inter-DC flows in advance, also significantly reducing PFC pause time. 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 12 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 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

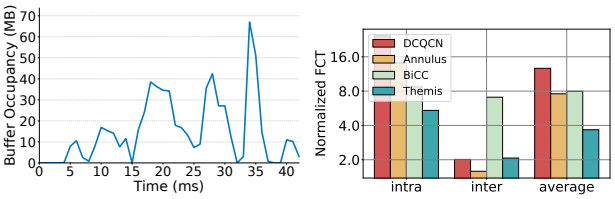


Figure 13: Buffer utilization of recirculation in TRP. Figure 14: Performance under complex traffic patterns.

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 13, recirculation consumes less than 70MB ESW buffer even under severe congestion. Considering ESWs typically have deep buffer [49], this buffer utilization is acceptable for current ESW hardware. TRP imposes limited overhead and is capable of scaling to high-load scenarios.

Performance under complex traffic patterns. To evaluate THEMIS under complex traffic patterns, we use a traffic mix where WebSearch traffic utilizes 30% of the link capacity, micro-burst traffic utilizes 30%, and cross-DC pipeline parallelism traffic accounts for 20%. As shown in Figure 14, compared with DCQCN, Annulus, and BiCC, THEMIS alleviates the unfairness problem more effectively and reduces overall average FCT by 71.1%, 51.6%, and 54.0%, respectively.

C. Efficiency Breakdown

We first conduct an ablation study to evaluate the effectiveness of PNP and TRP. As shown in Figure 15, 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 16, 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.

Conserving switch resources. We design a TCP-handshake-based monitor and an event-driven flow state management mechanism to reduce the switch resources used by PNP. We

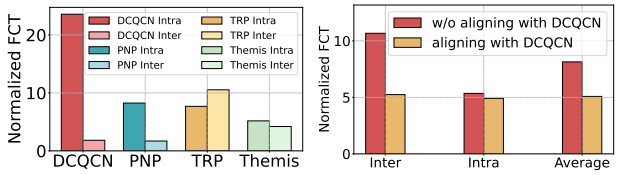


Figure 15: Effectiveness of TRP and PNP under 70% DCQCN workload.

Figure 16: Aligning with DCQCN behavior.

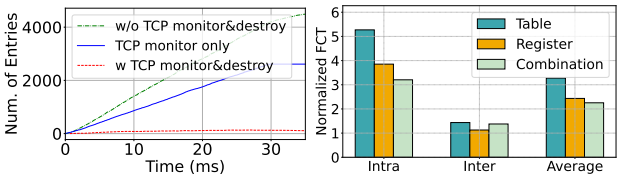


Figure 17: Effect of conserving resource overhead.

Figure 18: Flow deceleration.

record the number and arrival times of flows. As shown in Figure 17, comparing the resource overhead with and without these two mechanisms, our approach significantly reduces the switch's memory consumption by eliminating resource usage from intra-DC flows and proactively removing expired entries.

Flow Deceleration. We set the latency between the data plane and control plane to 1 millisecond [48] and configure the size of CNP_Cache register array to 1024. As shown in Figure 18, 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.

D. Hardware Testbed

Resource overhead. We evaluate the resource overhead of THEMIS on the Tofino switch. Table I in Appendix X-G shows THEMIS 's hardware resource utilization. THEMIS consumes minimal computational and memory resources, leaving sufficient capacity for concurrent execution of traditional forwarding behaviors [27]. Besides, THEMIS does not modify network protocols, and data packets do not carry any additional fields.

Race condition. We run THEMIS on the switch and modify its code to enable the switch to respond a CNP to every RDMA data packet. Since a CNP is generated by mirroring the corresponding data packet, it carries the same sequence number as the original packet. Through our several concurrent testings, we observe that the first CNP received by the sender always has a sequence number of 0. This confirms that the control plane has successfully inserted the forwarding rule before the arrival of the first RDMA data packet, allowing the switch to generate CNPs correctly and promptly.

VIII. DISCUSSION

Extension to other CC algorithms. THEMIS can be easily extended to support other CC algorithms. In terms of PNP, for INT-based HPCC [21], 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 [18], 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.

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 utilizes MATs to modify them into CNPs. TRP detects CNPs, redirects target packets to the ingress ports of ESW, 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.

IX. RELATED WORK

Besides the most relevant work we have discussed in §II, 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 [27], [17], [37], network monitoring [53], [19], attack defense [51], [23], [1], [43], and key-value stores[15], [22]. 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.

X. CONCLUSION

Deploying RDMA over wide-area 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 ESW. THEMIS consists of two key modules: PNP and TRP. When ESW 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 NS-3 large-scale simulations, we demonstrate the effectiveness and deployability of THEMIS.

ACKNOWLEDGMENT

We thank our shepherd and the anonymous ICNP reviewers for their valuable comments. This work is supported in part by the National Natural Science Foundation of China (No. 62402025, 623B2062), the Fundamental Research Funds for the Central Universities and Huawei-BUAA Joint Lab. Menghao Zhang is the corresponding author.

REFERENCES

[1] A. G. Alcoz, M. Strohmeier, V. Lenders, and L. Vanbever, "Aggregate-based congestion control for pulse-wave ddos defense," in *Proceedings of the ACM SIGCOMM 2022 Conference*, 2022, pp. 693–706.

[2] D. P. Anderson, "Boinc: A system for public-resource computing and storage," in *Fifth IEEE/ACM international workshop on grid computing*. IEEE, 2004, pp. 4–10.

[3] T. E. Anderson, M. Canini, J. Kim, D. Kostić, Y. Kwon, S. Peter, W. Reda, H. N. Schuh, and E. Witchel, "Assise: Performance and availability via client-local nvm in a distributed file system," in *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, 2020, pp. 1011–1027.

[4] W. Bai, S. S. Abdeen, A. Agrawal, K. K. Attre, P. Bahl, A. Bhagat, G. Bhaskara, T. Brokhman, L. Cao, A. Cheema *et al.*, "Empowering azure storage with rdma," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, 2023, pp. 49–67.

[5] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, 2010, pp. 267–280.

[6] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, and K. Tzoumas, "Apache flink: Stream and batch processing in a single engine," *The Bulletin of the Technical Committee on Data Engineering*, vol. 38, no. 4, 2015.

[7] Y. Chen, C. Tian, J. Dong, S. Feng, X. Zhang, C. Liu, P. Yu, N. Xia, W. Dou, and G. Chen, "Swing: Providing long-range lossless rdma via pfc-relay," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 1, pp. 63–75, 2022.

[8] A. Dragojević, D. Narayanan, M. Castro, and O. Hodson, "Farm: Fast remote memory," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, 2014, pp. 401–414.

[9] A. Dragojević, D. Narayanan, E. B. Nightingale, M. Renzelmann, A. Shamis, A. Badam, and M. Castro, "No compromises: distributed transactions with consistency, availability, and performance," in *Proceedings of the 25th symposium on operating systems principles*, 2015, pp. 54–70.

[10] K. D. S. et al., "Prague congestion control," https://www.ietf.org/archive/id/draft-briscoe-iccrg-prague-congestion-control-04.html, 2024.

[11] Facebook Engineering, "Building express backbone: Facebook's new long-haul network," https://engineering.fb.com/2017/05/01/data-center-engineering/building-express-backbone-facebook-s-new-long-haul-network/, 2017.

[12] N. Finn, "Ieee802.1qau congestion notification," https://1.ieee802.org/dcb/802-1qau/, 2010.

[13] Y. Gao, Q. Li, L. Tang, Y. Xi, P. Zhang, W. Peng, B. Li, Y. Wu, S. Liu, L. Yan *et al.*, "When cloud storage meets rdma," in *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, 2021, pp. 519–533.

[14] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu *et al.*, "B4: Experience with a globally-deployed software defined wan," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3–14, 2013.

[15] X. Jin, X. Li, H. Zhang, R. Soulé, J. Lee, N. Foster, C. Kim, and I. Stoica, "Netcache: Balancing key-value stores with fast in-network caching," in *Proceedings of the 26th symposium on operating systems principles*, 2017, pp. 121–136.

[16] A. Kalia, M. Kaminsky, and D. G. Andersen, "Fasst: Fast, scalable and simple distributed transactions with two-sided (rdma) datagram rpcs," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 185–201.

[17] N. Katta, M. Hira, C. Kim, A. Sivaraman, and J. Rexford, "Hula: Scalable load balancing using programmable data planes," in *Proceedings of the Symposium on SDN Research*, 2016, pp. 1–12.

[18] G. Kumar, N. Dukkipati, K. Jang, H. M. Wassel, X. Wu, B. Montazeri, Y. Wang, K. Springborn, C. Alfeld, M. Ryan *et al.*, "Swift: Delay is simple and effective for congestion control in the datacenter," in *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, 2020, pp. 514–528.

[19] G. Li, M. Zhang, C. Guo, H. Bao, M. Xu, H. Hu, and F. Li, "Imap: Fast and scalable in-network scanning with programmable switches," in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, 2022, pp. 667–681.

[20] 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 *21st USENIX Conference on File and Storage Technologies (FAST 23)*, 2023, pp. 99–114.

[21] Y. Li, R. Miao, H. H. Liu, Y. Zhuang, F. Feng, L. Tang, Z. Cao, M. Zhang, F. Kelly, M. Alizadeh *et al.*, "Hpcc: High precision congestion control," in *Proceedings of the ACM special interest group on data communication*, 2019, pp. 44–58.

[22] M. Liu, L. Luo, J. Nelson, L. Ceze, A. Krishnamurthy, and K. Atreya, "Incbricks: Toward in-network computation with an in-network cache," in *Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems*, 2017, pp. 795–809.

[23] Z. Liu, H. Namkung, G. Nikolaidis, J. Lee, C. Kim, X. Jin, V. Braverman, M. Yu, and V. Sekar, "Jaqen: A high-performance switch-native approach for detecting and mitigating volumetric ddos attacks with programmable switches," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 3829–3846.

[24] McKinsey & Company, "Ai power: Expanding data center capacity to meet growing demand," https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/ai-power-expanding-data-center-capacity-to-meet-growing-demand, 2023.

[25] N. Mellanox., "Connectx-6," https://www.nvidia.com/en-sg/networking/ethernet/connectx-6/, 2024.

[26] Mertech, "Cloud outages: Causes & risks (and how to handle them)," https://www.mertech.com/blog/cloud-outages-handling-guide, 2023.

[27] 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 *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, 2017, pp. 15–28.

[28] R. Mittal, V. T. Lam, N. Dukkipati, E. Blem, H. Wassel, M. Ghobadi, A. Vahdat, Y. Wang, D. Wetherall, and D. Zats, "Timely: Rtt-based congestion control for the datacenter," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 537–550, 2015.

[29] R. Mittal, A. Shpiner, A. Panda, E. Zahavi, A. Krishnamurthy, S. Ratnasamy, and S. Shenker, "Revisiting network support for rdma," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 313–326.

[30] NS-3project, "Ns-3 simulator," Feb. 2025. [Online]. Available: https://www.nsnam.org/

[31] NVIDIA Corporation, "Ethernet network," https://docs.nvidia.com/networking/display/winof2v320/ethernet+network, 2023.

[32] ——, "Out-of-order (ooo)," https://docs.nvidia.com/networking/display/rdmacore50/out-of-order+(ooo), 2023.

[33] OpenAI, "Chatgpt," Nov. 2024. [Online]. Available: https://chatgpt.com/

[34] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, "Inside the social network's (datacenter) network," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 123–137.

[35] A. Saeed, V. Gupta, P. Goyal, M. Sharif, R. Pan, M. Ammar, E. Zegura, K. Jang, M. Alizadeh, A. Kabbani *et al.*, "Annulus: A dual congestion control loop for datacenter and wan traffic aggregates," in *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, 2020, pp. 735–749.

[36] SemiAnalysis, "Multi-datacenter training: Openai's ambitious plan to beat google's infrastructure," September 2024. [Online]. Available: https://semianalysis.com/2024/09/04/multi-datacenter-training-openais/

[37] 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 *Proceedings of the ACM SIGCOMM 2023 Conference*, 2023, pp. 816–831.

[38] Themis, "Themis," https://github.com/Networked-System-and-Security-Group/Themis, 2025.

[39] S.-Y. Tsai, Y. Shan, and Y. Zhang, "Disaggregating persistent memory and controlling them remotely: An exploration of passive disaggregated key-value stores," in *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, 2020, pp. 33–48.

[40] Z. Wan, J. Zhang, M. Yu, J. Liu, J. Yao, X. Zhao, and T. Huang, "Bicc: Bilateral congestion control in cross-datacenter rdma networks," in *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*. IEEE, 2024, pp. 1381–1390.

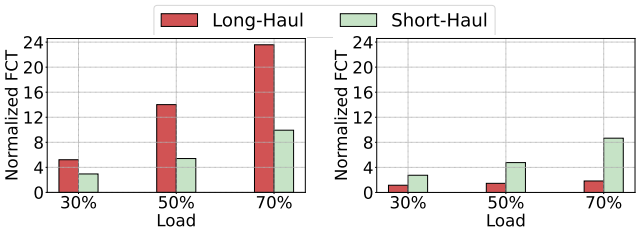
[41] H. Wang, D. Zhang, and K. G. Shin, "Detecting syn flooding attacks," in *Proceedings. Twenty-first annual joint conference of the IEEE computer and communications societies*, vol. 3. IEEE, 2002, pp. 1530–1539.
[42] S. Wang, M. Zhang, Y. Du, Z. Chen, Z. Wang, M. Xu, R. Xie, and J. Yang, "Lordma: A new low-rate dos attack in rdma networks," *in NDSS*, 2024.
[43] J. Xing, K.-F. Hsu, Y. Qiu, Z. Yang, H. Liu, and A. Chen, "Bedrock: Programmable network support for secure rdma systems," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 2585–2600.
[44] J. Xue, Y. Miao, C. Chen, M. Wu, L. Zhang, and L. Zhou, "Fast distributed deep learning over rdma," in *Proceedings of the Fourteenth EuroSys Conference 2019*, 2019, pp. 1–14.
[45] J. Yang, J. Izraelevitz, and S. Swanson, "Orion: A distributed file system for non-volatile main memory and rdma-capable networks," in *17th USENIX Conference on File and Storage Technologies (FAST 19)*, 2019, pp. 221–234.
[46] P. Yu, F. Xue, C. Tian, X. Wang, Y. Chen, T. Wu, L. Han, Z. Han, B. Wang, X. Gong *et al.*, "Bifrost: Extending roce for long distance inter-dc links," in *2023 IEEE 31st International Conference on Network Protocols (ICNP)*. IEEE, 2023, pp. 1–12.
[47] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *2nd USENIX workshop on hot topics in cloud computing (HotCloud 10)*, 2010.
[48] C. Zeng, L. Luo, T. Zhang, Z. Wang, L. Li, W. Han, N. Chen, L. Wan, L. Liu, Z. Ding *et al.*, "Tiara: A scalable and efficient hardware acceleration architecture for stateful layer-4 load balancing," in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, 2022, pp. 1345–1358.
[49] G. Zeng, W. Bai, G. Chen, K. Chen, D. Han, Y. Zhu, and L. Cui, "Congestion control for cross-datacenter networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 5, pp. 2074–2089, 2022.
[50] M. Zhang, G. Li, C. Guo, H. Bao, M. Xu, H. Hu, and F. Li, "Imap: Toward a fast, scalable and reconfigurable in-network scanner with programmable switches," *IEEE Transactions on Information Forensics and Security*, 2023.
[51] M. Zhang, G. Li, S. Wang, C. Liu, A. Chen, H. Hu, G. Gu, Q. Li, M. Xu, and J. Wu, "Poseidon: Mitigating volumetric ddos attacks with programmable switches," in *the 27th Network and Distributed System Security Symposium (NDSS 2020)*, 2020.
[52] M. Zhang, Y. Hua, P. Zuo, and L. Liu, "Ford: Fast one-sided rdma-based distributed transactions for disaggregated persistent memory," in *20th USENIX Conference on File and Storage Technologies (FAST 22)*, 2022, pp. 51–68.
[53] Y. Zhou, Z. Xi, D. Zhang, Y. Wang, J. Wang, M. Xu, and J. Wu, "Hypertester: high-performance network testing driven by programmable switches," in *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, 2019, pp. 30–43.
[54] Y. Zhu, H. Eran, D. Firestone, C. Guo, M. Lipshteyn, Y. Liron, J. Padhye, S. Raindel, M. H. Yahia, and M. Zhang, "Congestion control for large-scale rdma deployments," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 523–536, 2015.
[55] S. Zou, J. Huang, J. Liu, T. Zhang, N. Jiang, and J. Wang, "Gtcp: Hybrid congestion control for cross-datacenter networks," in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2021, pp. 932–942.

APPENDIX

A. Other CC Algorithms' Performance in Long-haul RDMA.

We use the same experimental setup as in II-C to evaluate HPCC [21] in long-haul scenario. As shown in Figure 19, for intra-DC flows, HPCC's FCT under long-haul RDMA is up to 136% higher than under short-haul. HPCC relies on INT to measure per-RTT queue occupancy and compute the target sending rate. However, in long-haul RDMA, the INT feedback is already stale when it reaches the sender, resulting in performance similar to that of DCQCN.

TIMELY [28] is originally designed for intra-DC networks, where the baseline RTT is on the order of microseconds,



(a) intra flows FCT (b) inter flows FCT

Figure 19: Normalized FCT of HPCC in long/short-haul scenarios.

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 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.

B. The Throttling Algorithms in Flow Deceleration

We provide an explanation of the rate reduction algorithm and its design objectives in §V-B, and now present its detailed pseudocode in Algorithm 1.

Algorithm 1 Throttling Algorithm

Input: pkt, α, β
Output: pkt_loop_num, $flow\_status$

if pkt is cnp **then**
  $cnp\_num \leftarrow cnp\_num + 1$
  $flow\_status \leftarrow throttled$
  $cnp\_time \leftarrow now$
  if $cnp\_num\%\alpha == 1$ **then**
    $loop\_num \leftarrow loop\_num + 1$
    $\alpha \leftarrow \alpha + 1$
  end if
else
  if $flow\_status == throttled \mid\mid recover$ **then**
    $pkt\_loop\_num \leftarrow loop\_num$
    $redirect\ pkt$
  else
    $send\ pkt$
  end if
end if
if $now - cnp\_time > \beta$ **then**
  $flow\_status \leftarrow recover$
  $\alpha \leftarrow \alpha/2$
  $cnp\_num \leftarrow 0$
end if
return

C. Component Layout

Figure 20 shows details of how THEMIS is organized across switch data plane hardware and control plane CPUs.

D. Rate Recovery

TRP needs to recover the rate of target flows when congestion has been addressed. 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 21: packets 1-3 are

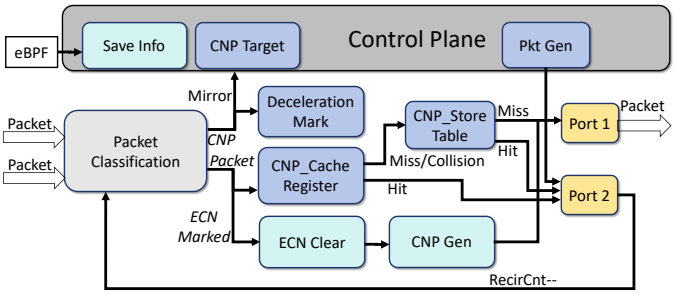
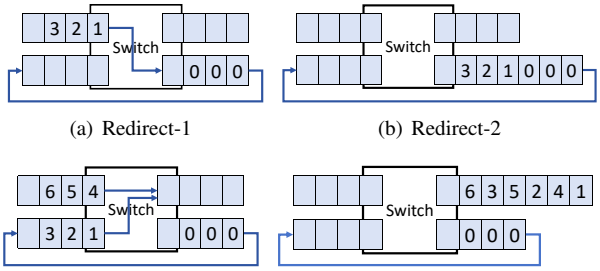


Figure 20: Component layout of THEMIS.



(a) Redirect-1 (b) Redirect-2

(c) Redirect-3 (d) Redirect-4

Figure 21: Out of order scenario.

redirected to pre-filled ports and loop several times (Figure 21(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 21(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 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 already 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.

E. Efficiency of 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 22, compared to no recovery and direct recovery — the naive solution mentioned in Appendix §X-D — 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.

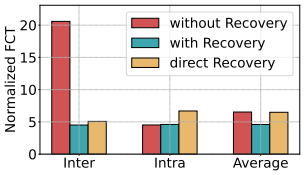


Figure 22: Rate recovery under GBN.

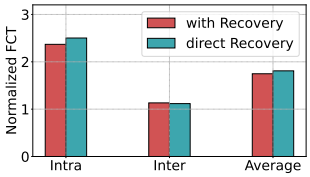


Figure 23: Rate recovery under IRN.

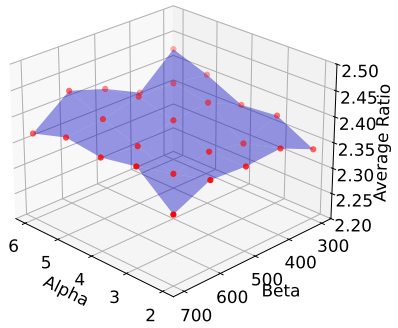


Figure 24: Influence of α and β.

Table I: Resource overhead of THEMIS on switch.

Computational		Memory	
ALUs	**Gateways**	**SRAM**	**TCAM**
6.25%	10.94%	6.56%	0.00%

Furthermore, we implement TRP on an NS-3 simulation framework [37] built upon IRN (Improved RoCE NIC) [29]. We evaluate the performance of direct recovery on IRN under WebSearch traffic trace. 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 23. Therefore, in the case of IRN-compatible NICs, TRP employs direct recovery without the need for an additional rate recovery mechanism.

F. Parameters Tuning

TRP employs two parameters in its rate reduction strategy, α and β. As shown in Figure 24, different parameter settings have limited impact on THEMIS, with the maximum difference in average FCT being less than 4%. 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.

G. Hardware Resource Overhead

We evaluate THEMIS on Tofino switch and record its hardware resource utilization, as shown in Table I.