

PSN-PATH: When Multipath RDMA Meets Lossy Networks

Zhexiong Li², Shugui Wei², Puyu Zhao², Feifan Wang², Yuepeng Li²,

Lin Gu¹, Deze Zeng^{1,2}, Xiaoliang Wang³, Laiping Zhao⁴

¹Huazhong University of Science and Technology

²China University of Geosciences, Wuhan

³Nanjing University ⁴Tianjin University
China

Abstract

Remote Direct Memory Access (RDMA) is widely adopted for high-performance datacenter workloads, but its strict ordering requirement conflicts with multipath transport in lossy networks. RDMA Network Interface Cards (NICs) may interpret Out-of-Order (OOO) arrivals caused by diverse path latencies as packet loss. This triggers unnecessary retransmissions, severely degrading goodput collapse and inflating Flow Completion Time (FCT). We present PSN-PATH, a novel RDMA protocol to distinguish packet loss from OOO and avoid unnecessary retransmission, improving goodput and FCT of multi-path RDMA in lossy networks. PSN-PATH maps Packet Sequence Numbers (PSN) to network paths, enabling the receiver to verify packet continuity within each mapped path and distinguish real losses from OOO arrivals. PSN-PATH only retransmits lost packets and redirects traffic away from degraded paths. We implement PSN-PATH on Xilinx Alveo U50 FPGA using Vitis HLS and validate it via extensive simulation experiments. Experimental results demonstrate that PSN-PATH significantly outperforms existing solutions, reducing FCT by 95.81%, enhancing goodput by 5.11 $\times$, and reducing retransmissions by over 99% under high-load, lossy conditions.

CCS Concepts

- **Networks** $\rightarrow$ **Network protocol design; Transport protocols;**
- **Hardware** $\rightarrow$ **Networking hardware.**

Keywords

Multipath RDMA, Lossy RDMA, Out-of-Order

ACM Reference Format:

Zhexiong Li, Shugui Wei, Puyu Zhao, Feifan Wang, Yuepeng Li, Lin Gu, Deze Zeng, Xiaoliang Wang, and Laiping Zhao. 2026. PSN-PATH: When Multipath RDMA Meets Lossy Networks. In *ACM SIGCOMM 2026 Conference (SIGCOMM '26)*, August 17–21, 2026, Denver, CO, USA. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3789240.3829128>

1 Introduction

Remote Direct Memory Access (RDMA) has become the de facto standard for high-performance datacenter networks, as evidenced by large-scale deployments at Meta [8], Google [38], Microsoft [4,

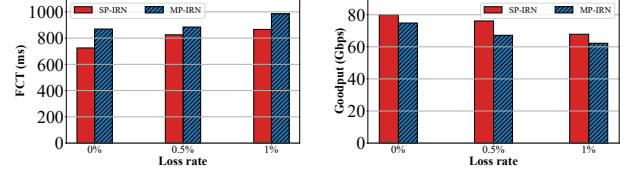


Figure 1: Performance comparison between single-path RDMA (SP-IRN) and multipath RDMA (MP-IRN).

30] and Alibaba [9, 29]. To improve network utilization, these operators increasingly combine RDMA with packet-level multipath [2, 13, 18, 34], which distributes packets of a single flow across multiple paths. Existing deployments of such fine-grained multipath RDMA rely on lossless Ethernet environments with Priority-based Flow Control (PFC) to prevent packet loss. However, lossless fabrics can experience PFC-induced head-of-line blocking, PFC storm and deadlocks [5, 7, 10, 28, 37, 38, 53], affecting cluster scalability and reliability. Consequently, lossy RDMA [20, 35, 38], which replaces PFC with selective retransmission (SR), is gaining broad adoption. It ensures end-to-end reliability and improves network utilization by checking the continuity of packet sequence number (PSN) [20, 35].

Unfortunately, multipath transmission conflicts with lossy RDMA reliability because multipath-introduced Out-Of-Order (OOO) arrivals may be misinterpreted as packet losses by recovery mechanisms. As shown in Fig. 1, we test multipath and single-path RDMA performance in a lossy spine-leaf network environment composed of 15 switches. Ideally, the performance of multipath transmission should significantly outperform that of single-path transmission. However, we observe that even when employing state-of-the-art SR (e.g., IRN [35], Flor [20]), multipath RDMA performance underperforms single-path transmission. The Flow Completion Time (FCT) increases by up to 18.85% and the goodput decreases by up to 15.86%. This is because multipath RDMA experiences more than 3 $\times$ retransmission events compared to single-path transmission (detailed in §. 2.3). Such high-frequency packet retransmission severely degrades the performance and usability of multipath RDMA in the lossy network environment.

Our further analysis shows that this performance degradation stems from PSN discontinuity caused by OOO packet arrivals in multipath transmission. The RDMA transport engine treats these discontinuities as packet losses and triggers retransmission [23]. This confusion of packet OOO with real loss increases FCT and offsets any goodput benefits from multipath RDMA [17, 32]. Prior works have used in-order delivery design to mitigate OOO arrivals. These approaches either rely on in-network reordering at advanced



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

SIGCOMM '26, Denver, CO, USA

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2467-1/26/08

<https://doi.org/10.1145/3789240.3829128>

network devices such as P4 switches [23, 40], or employ large on-chip bitmaps with timeout mechanisms at the receiver to track packet arrivals [19, 31]. However, they still operate under the constraint of a single, globally contiguous PSN. Consequently, they cannot resolve the ambiguity between OOO arrivals and real packet loss, leading to either increased deployment complexity in network devices or significantly inflated FCT due to prolonged retransmission delays.

To address this conflict, we need to equip the RNIC with the ability to distinguish between OOO arrival and packet loss. This enables the RNIC to retransmit only lost packets, thereby reducing redundant retransmissions and improving FCT and goodput. Our key insight is that while RDMA relies on a global PSN sequence for reliability, with packets expected to arrive in the order of PSNs 0, 1, 2, 3, . . . , both OOO arrivals and real losses manifest as PSN discontinuities. This inspires us to redefine the scope of PSN “continuity”. Rather than enforcing a global order across the entire network, we restrict ordering constraints to each individual transmission path. Consequently, the RNIC can distinguish that OOO arrivals across different paths are real and acceptable, whereas a PSN discontinuity within the same path indicates real packet loss. Therefore, we present PSN-PATH, a novel and hardware-efficient multipath RDMA protocol designed specifically for lossy network. PSN-PATH introduces a dynamic PSN-to-path mapping to distinguish OOO arrivals from losses, enabling fast retransmission and path control to ensure high goodput and low FCT. PSN-PATH supports both multipath and lossy operation exclusively at the hardware, requiring no modifications to existing switch infrastructure [11, 48].

To distinguish between OOO arrivals and real packet loss, PSN-PATH redefines the boundary of PSN “continuity”. Specifically, we implement a Path Mapper at the sender to establish an implicit PSN-to-path mapping, ensuring that packets sent along the same path stay in order. The mapping is implicit because it uses the existing PSN rules, changing RDMA’s original PSN semantics. To avoid the high hardware overhead of maintaining per-path state, we design a Path Recognizer at the receiver. Path Recognizer includes a bitmap and a stateless backtracking continuity check. By examining the arrival status of the logically preceding packet on the same path within the bitmap, the Path Recognizer can accurately determine whether a discontinuity indicates a real loss or just an OOO arrival. This allows PSN-PATH to trigger fast retransmissions only for confirmed losses, ensuring high performance with low hardware resource consumption.

To achieve efficient multipath transport, PSN-PATH incorporates a Path Manager that monitors path quality in real-time using a lightweight scoring metric based on ACK/NACK statistics. Based on this, we further design a Double Threshold Adaptive Path Switching (DT-Switch) algorithm. DT-Switch proactively explores alternative paths when the current path starts to degrade (exploration threshold) and isolates faulty paths when they no longer satisfy the reliability requirement (removal threshold). It redirects subsequent PSN sequences to healthy paths by updating the mapping state at the RNIC, without relying on complex control signaling.

Based on the above methods, PSN-PATH resolves the conflict between multipath RDMA and lossy network, and can be seamlessly integrated into existing infrastructures. The main contributions of this work are summarized as follows.

- We provide a comprehensive analysis of the conflict between multipath RDMA and lossy network. Through extensive experiments, we investigate the sensitivity of RDMA to packet loss and demonstrate how OOO-loss confusion inflates FCT and degrades goodput.
- To harmonize multipath RDMA with lossy networks, we propose PSN-PATH, a hardware-efficient protocol that establishes a fine-grained association between PSNs and network paths. By dynamically adjusting mapping rules on the RNIC, PSN-PATH ensures reliability and precise loss detection without requiring switch modifications.
- We prototype PSN-PATH using Xilinx Vitis HLS [12] and validate it via extensive network simulation (ns-3 [43]) experiments. Experimental results demonstrate that our approach reduces FCT by up to 95.81%, enhances goodput by 5.11×, and achieves a reduction in retransmissions of over 99% compared to existing solutions.

Open source: The source code of PSN-PATH can be accessed at <https://github.com/AINetworkLab/PSN-PATH>.

Ethics: This work does not raise any ethical issues.

2 Background and Motivation

2.1 RDMA Overview

RDMA is developed to reduce CPU overhead and latency in network communication. By leveraging kernel bypass and zero-copy techniques, RDMA achieves low latency, high bandwidth, and minimal CPU usage [6, 26, 42, 51]. Its high performance depends on a lossless network environment, where RDMA hardware (e.g., NVIDIA Mellanox ConnectX series) can use a simple Go-Back-N (GBN) retransmission strategy [16]. A key of this approach is the Packet Sequence Number (PSN), which provides a unique identity for each packet and allows parallel transmission to improve throughput. The receiver acknowledges PSNs sequentially. If a discontinuity is detected, it is treated as a packet loss event. The sender needs to retransmit all packets before the lost PSN. In lossless network environments, this mechanism is simple and efficient, as packet loss events are rare.

However, as RDMA networks scale and application workloads become more diverse, maintaining a strictly lossless fabric becomes increasingly difficult. Packet loss and OOO arrivals are therefore more likely to occur in data centers [22, 25, 27]. As a result, traditional RDMA designs that assume an underlying lossless network are less suitable for modern datacenter requirements.

2.2 Lossy RDMA

While PFC can build a lossless network, it introduces well-documented challenges, including head-of-line blocking, deadlocks, and suboptimal network utilization [45, 52]. Therefore, recent works have explored disabling or limiting PFC in favor of a lossy RDMA design paradigm, where network reliability is managed exclusively through end-to-end retransmission mechanisms. A prominent advancement in this domain is IRN [35], which replaces the GBN protocol with a selective retransmission (SR). Building on this, Flor [20] optimizes hardware efficiency by reducing the on-chip bitmap overhead for packet tracking. Most recently, DCP [23] proposes a transport architecture that co-designs both the switch and

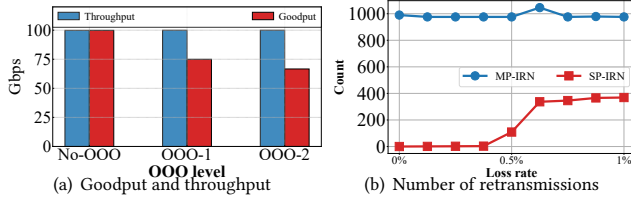


Figure 2: Performance comparison in networks with varying packet OOO levels and packet loss rates. The RTT variances of No-OOO, OOO-1, and OOO-2 are set to 0, 2, and 4, respectively.

RNICs to achieve lossless-like performance over lossy fabrics. Falcon [39] uses bitmap-based ACKs, SACK, and RACK-TLP to tolerate packet OOO and recover losses. Its bitmap feedback is carried in packet headers to help the sender identify lost packets, but this also increases packet size and adds extra processing overhead.

Nevertheless, GBN, SR, and most extant lossy RDMA solutions still rely on a reliability model that necessitates a single, globally contiguous PSN sequence. In multipath RDMA, packets from the same flow may traverse paths with different latencies, so later packets can arrive before earlier ones even when no packet is lost. Such path-induced OOO arrivals create the same PSN discontinuities as real losses, causing the receiver to trigger unnecessary recovery. Therefore, lossy RDMA mechanisms that work well under single-path transmission can suffer severe performance degradation when directly combined with packet-level multipath.

2.3 Motivation

Multipath RDMA leverages techniques such as Equal Cost Multi Path (ECMP [14, 50]) routing and switch-based rerouting [40] to distribute packets of a single flow across multiple network paths, thereby optimizing overall network utilization. Distributing a single data flow across paths with heterogeneous latency often leads to OOO packet arrivals. When these paths have different latencies, packets from the same flow may arrive out of order. As discussed above, OOO arrivals and packet losses both appear as PSN discontinuities at the receiver, so the RNIC cannot distinguish between them. This ambiguity triggers a surge of unnecessary retransmissions and degrades network performance.

As shown in Fig. 2(a), our test on Vitis HLS demonstrates that lossy RDMA schemes suffer a goodput drop of more than 25% when multipath transmission causes OOO arrivals, even though the total throughput remains high. This phenomenon is highlighted in Fig. 2(b), which shows that in the same lossy network environment, multipath RDMA generates over 3× more retransmissions than single-path transmission. Therefore, to fully unlock the potential of modern datacenter networks, it is important to resolve the conflict between multipath RDMA and lossy network, which serves as the primary focus of this work.

3 System Design

In this section, we first present a system overview of PSN-PATH and then introduce its three key modules, i.e., Path Mapper, Path Recognizer, and Path Manager.

3.1 Design Rationale

The design of PSN-PATH is rooted in the observation that global PSN is over-constrained for multipath environments. Our rationale is built upon three key principles.

Per-Path Ordering: In a multipath network, path heterogeneity often results in OOO packet arrivals at the receiver. However, within any single physical path, network switches typically maintain First-In-First-Out (FIFO) delivery. This FIFO property ensures that packet arrival remains ordered within each individual path. Consequently, we consider transforming from global sequence tracking to per-path ordering. This allows the receiver to identify any sequence discontinuity within a specific path as a confirmed packet loss rather than a potential OOO arrival, enabling immediate recovery.

Hardware Efficiency: Although maintaining a bitmap for every path could resolve PSN confusion, tracking per-path metadata increases sensitivity to workload scale and path diversity. For example, mainstream RNICs, such as ConnectX-7/8, typically provide only tens of MBs of on-chip SRAM. However, an AI training workload with 10K QPs (Queue Pairs) and 64 paths may consume about 80MB SRAM if per-path metadata is maintained for every QP, which already exceeds the 64 MB SRAM budget. PSN-PATH addresses this by employing a stateless verification logic. Instead of storing historical data for each path, we adopt a path-level backtracking continuity check on a per-QP arrival bitmap. Thus, PSN-PATH distinguishes OOO arrivals from real losses without consuming additional on-chip memory for each path individually, thereby ensuring low hardware cost.

Proactive Recovery: Traditional multipath RDMA is reactive, waiting for bitmap overflows, which inflates FCT. Our design rationale favors a proactive strategy. By establishing an implicit PSN-to-path mapping, we can trigger fast retransmission the moment a per-path discontinuity is detected. This recovery process may not be a retransmission along the original path. The PSN-PATH evaluates the health of each path. If a path always triggers packet loss, the system probes new paths and remaps subsequent sequences to healthier alternatives. This prevents the goodput or FCT collapse caused by redundant transmissions.

3.2 PSN-PATH Overview

Fig. 3 illustrates the architecture and workflow of PSN-PATH, which consists of three main components. The **Path Mapper** maintains mapping parameters and encapsulates the metadata into the protocol header of both data packets and retransmission signals. The **Path Recognizer** parses path metadata from packet headers to identify the specific transmission path. This enables the system to accurately distinguish real packet loss from OOO arrival, triggering timely retransmission signals only when necessary. The **Path Manager** records loss and OOO events to evaluate path quality and updates mapping parameters to perform path switching. All components operate exclusively within the RNIC data plane and are transparent to applications.

When the transmission starts, the RNIC invokes its DMA engine to fetch data from host memory. The data is segmented and encapsulated into individual packets. The **Path Mapper** generates specific path metadata for each packet to ensure balanced distribution across multiple paths. This metadata is then embedded into the

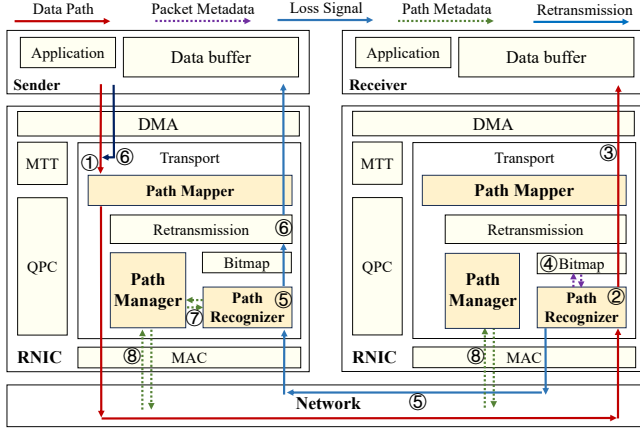


Figure 3: Overview of PSN-PATH.

RDMA protocol header ①. Meanwhile, the corresponding source port is sent to the UDP encapsulation layer for switch-level routing. After encapsulation, the packets are dispatched into the network.

When receiving a data packet, the receiver's **Path Recognizer** first parses its embedded path metadata ② and commits the payload to memory based on its PSN ③. With this metadata, **Path Recognizer** checks whether an OOO event has occurred on the corresponding path. PSN-Path adopts a classical bitmap widely used in RDMA solutions to mark the arrival status of each packet and detects OOO packets with the assistance of the bitmap ④. If the OOO arrival is confirmed as a real packet loss, the receiver immediately issues a retransmission signal to the sender ⑤. The sender-side **Path Recognizer** then localizes the loss to the faulty path and triggers SR ⑥. This loss information is also sent to the **Path Manager** ⑦, which aggregates loss events and updates the mapping parameters to reduce traffic on degraded paths ⑧.

3.3 Path Mapper

Existing multipath RDMA solutions do not establish a mapping relationship between PSNs and network paths. Consequently, the receiver cannot determine whether a PSN discontinuity is caused by OOO arrival or real packet loss. A common solution is to track OOO packets in a bitmap and trigger retransmission when the bitmap overflows. However, if a packet is lost on one path, the receiver must wait until enough later packets, which delays recovery and inflates FCT in lossy RDMA networks.

The Path Mapper provides both the sender and receiver of one QP with a shared and lightweight PSN-to-path mapping. Instead of changing the RDMA PSN semantics, Path Mapper derives a logical path index from the existing PSN and uses this index to select the packet source port. The source port is then used by the underlying load balancing mechanism, such as ECMP, to select a forwarding path. Thus, PSN-PATH controls packet distribution through standard packet headers and does not require switch modifications.

3.3.1 Path-index-based Mapping. To establish a stable PSN-to-path mapping, the Path Mapper first derives the path index from the PSN of an outgoing packet as $\ell = PSN \bmod K$, where K denotes the number of selected transmission paths maintained by one QP. PSN-PATH exposes K as a configurable parameter, allowing network

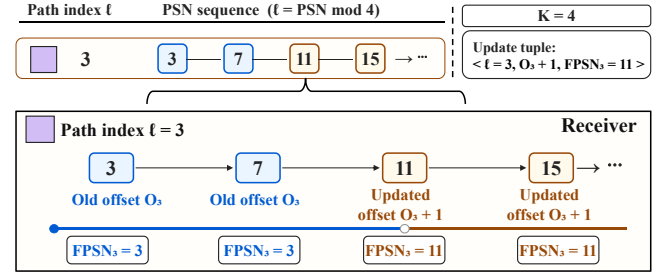


Figure 4: An example of path remapping.

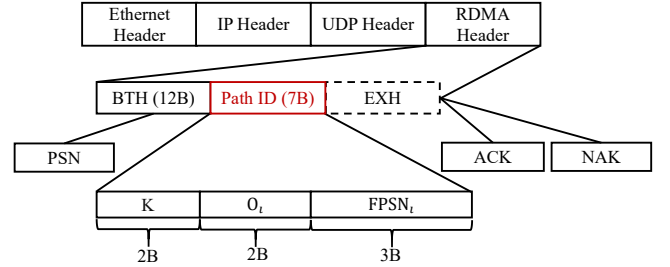


Figure 5: PSN-PATH packet header.

operators to set it according to path diversity, RNIC resource budget, and deployment requirements. This modulo mapping can create a sequence for each logical path index. For a path ℓ , packets assigned to this index shall form the PSN sequence $\ell, \ell + K, \ell + 2K, \dots$. Therefore, for a packet PSN , we can perform a path-level backtracking continuity check by verifying whether its predecessor packet with $PSN - K$ on the same path has been marked as received in the bitmap or not.

The Path Mapper dynamically associates each logical path ℓ with an actual physical path by specifying the UDP source port according to the path statuses. For all the packets satisfying $PSN \bmod K \equiv \ell$, Path Mapper further translates their path index ℓ into a UDP source port $P_{source} = P_{base} + \ell + O_\ell$, where P_{base} is the base source port and O_ℓ is the offset to control the source port for path ℓ . The source port P_{source} is used by the UDP/IP encapsulation stage and ECMP hashing to select a forwarding path. Upon any change in path status, PSN-PATH may update O_ℓ to modify the UDP source port and remap packets to a new physical path, i.e., path remapping, as detailed in §. 3.5.2.

3.3.2 Remapping Boundary. During path remapping, packets generated before and after the remapping may coexist in the network. If the receiver treats them as one continuous sequence and still performs backtracking continuity check, it may mistakenly treat an in-flight packet from the old path as lost. To avoid this ambiguity, PSN-PATH defines a remapping boundary $FPSN_\ell$, as the PSN of the first packet sent along the physical path mapped to ℓ , to distinguish packets from different path mappings.

Each packet sent in path ℓ carries the current $FPSN_\ell$, and the receiver also maintains a local $FPSN_\ell^*$ for path ℓ . When the sender selects a new path by updating O_ℓ , it also sets a new $FPSN_\ell$ to all following packets. If a received packet carries an $FPSN_\ell$ larger

than the receiver's local $FPSN_\ell^*$, path remapping has happened. In this case, this packet is regarded as the first arrival packet in the new path. Accordingly, the receiver updates its local $FPSN_\ell^*$ to the value of $FPSN_\ell$, and does not perform backtracking continuity check. Otherwise, if the carried $FPSN_\ell$ is smaller, the packet belongs to the old path, and $PSN - K$ backtracking continuity check is applied normally. Hence, PSN-PATH can detect path remapping by comparing $FPSN_\ell$ and $FPSN_\ell^*$, and only provides continuity check within the same path. Therefore, OOO arrivals during remapping will not be mistakenly identified as packet losses.

Edge case analysis. To demonstrate the correctness of remapping, we further provide edge case analysis. For a packet with PSN P and path index ℓ , let $S(P)$ denote its $FPSN_\ell$, and $R(P)$ as the local $FPSN_\ell^*$ maintained at the receiver. Since PSN increases monotonically, $P \geq S(P)$ is always satisfied for any packet. Then, there are four possible edge cases, including (1) new-path packets, (2) normal packets, (3) packets lost across a remapping boundary, and (4) OOO packets from the old path.

In case (1) (i.e., $S(P) > R(P)$ and $P = S(P)$), this packet is the first packet arriving via the new path, the receiver accepts it without continuity checking, and any in-flight old-path packet will not be falsely reported as lost. In case (2) (i.e., $S(P) \geq R(P)$ and $P > R(P)$), the packet is regarded as a normal one, and the receiver performs continuity check normally. In case (3) (i.e., $P = R(P) - K$), the last packet before path remapping is lost, it will be recovered by the normal timeout-based retransmission mechanism. In case (4) (i.e., $S(P) < R(P)$), a packet along the old path arrives after remapping. The receiver accepts it as an old-path packet and performs continuity check normally by checking whether packet $P - K$ is marked as received in the bitmap or not.

Fig. 4 shows an example of case (4) when a packet from the old path arrives after remapping, with $K = 4$, $\ell = 3$ and local $FPSN_3^* = 3$. The PSNs of the packets to be sent are 3, 7, 11, and 15. The sender switches paths while sending packet 11, by updating O_3 to $O_3 + 1$ and all the following packets will carry $FPSN_3 = 11$. Upon receiving packet 11, the receiver will know that packet 11 is the first packet of the new path, since $FPSN_3$ is larger than the local $FPSN_3^*$. It will update the local $FPSN_3^*$ from 3 to 11 and skip checking whether $11 - 4 = 7$ has arrived. In this way, the in-flight packet 7 will not be mistakenly treated as lost, and it will be reliably transmitted via mechanisms such as timeout-based retransmission. The remapping boundary covers all edge cases and avoids confusing in-flight old-path packets with new-path packets, thus ensuring safe remapping in PSN-PATH.

3.3.3 Path Metadata Encoding. The RNIC maintains path metadata as "PATH ID" to manage outgoing traffic. For each data packet, the Path Mapper retrieves its PSN, computes the path index $\ell = PSN \bmod K$, and calculates the source port. The source port is then passed to the UDP encapsulation stage to influence ECMP forwarding.

To allow the receiver to interpret the same mapping, PSN-PATH inserts a compact path control field into the RDMA extension header. Such header extensions are common in high-performance protocol design [23, 40]. As illustrated in Fig. 5, each PSN-PATH packet carries a compact path control field *Path ID* in the RDMA extension header, including the number of selected paths K , the port offset O_ℓ ,

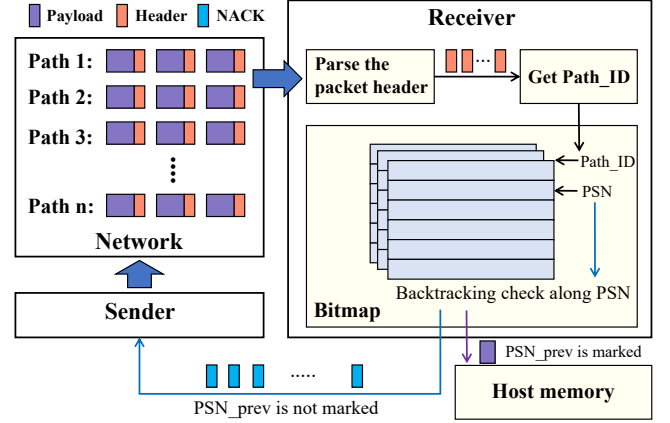


Figure 6: Stateless OOO detection.

and the remapping boundary $FPSN_\ell$. The source port, computed as $P = P_{base} + \ell + O_\ell$, is written into the UDP header and used by the network's ECMP mechanism to select the forwarding path. This design embeds all mapping-relevant information directly in the packet header, keeping PSN-PATH transparent to switches while allowing the receiver to distinguish packets from different mapping paths.

3.4 Path Recognizer

Based on the Path Mapper, we further introduce the Path Recognizer for OOO tolerance and loss detection in this section. This mechanism treats OOO arrivals from different paths as real, whereas a sequence discontinuity event detected within the same path will immediately trigger retransmission signals. This approach features two key innovations: Stateless OOO Detection and Fast Retransmission.

3.4.1 Stateless OOO Detection. To detect OOO packets, existing lossy RDMA schemes maintain an EPSN at the receiver RNIC. If $PSN = EPSN$, the packet is considered in order, and the EPSN is increased. If $PSN > EPSN$, the packet is regarded as OOO. However, with multipath RDMA, RNICs should maintain separate states for each path, which consumes non-negligible hardware resources. To improve the hardware efficiency, we propose a Stateless OOO Detection approach that logically distinguishes packets from different paths without maintaining per-path state on the RNIC.

As detailed in Fig. 6, PSN-PATH uses a bitmap to record the arrival status of packets. Upon receiving a data packet, the receiver first parses the path control field from the packet header to determine the declared path (i.e., *Path ID*) of the packet. Instead of relying solely on a single EPSN, the system performs backtracking check along the PSN sequence and $FPSN_\ell$ specific to that path. That is, the Path Recognizer queries the arrival status in the bitmap for the logically preceding packet on the same path based on boundary $FPSN_\ell$ (i.e., $PSN_{prev} = PSN_{current} - K$). If PSN_{prev} is marked as received, this indicates that the sequence of packets on the current path is continuous, and the current packet is classified as OOO and accepted. Conversely, if PSN_{prev} is not marked, it is immediately determined that a real packet loss has occurred on the current

path. In this case, the system generates a NACK signal to request retransmission of PSN_{prev} . This entire process is based on information contained in the packet and the current state of the bitmap, obviating the need to maintain any historical data for each path.

3.4.2 Fast Retransmission. When packet loss is detected, the system should trigger SR. Existing multipath RDMA solutions (e.g., MP-RDMA [31], ORNIC [32], LEFT [17]) use a bitmap overflow detection mechanism to prevent the misclassification of OOO packets as losses. Retransmissions are only passively triggered when the bitmap used for tracking OOO arrivals becomes full. This approach inherently suffers from high latency problem. First, lost packets recovery begins only after the bitmap overflows, which significantly increases FCT. Second, since the bitmap only records successfully received OOO packets, retransmission requests often span a PSN range that includes delayed packets, resulting in redundant retransmissions and bandwidth wastage.

PSN-PATH implements a Fast Retransmission mechanism based on backtracking check. The key innovation of this mechanism lies in its ability to immediately trigger retransmissions upon detecting a sequence discontinuity on any individual path. It does not rely on timer-based delays or buffer overflow thresholds. As described in §. 3.4.1, the receiver immediately determines that a packet loss has occurred on path $Path ID$ if $PSN_{current} - K$ is not marked in the bitmap during the backtracking check. In response, the receiver generates a specialized NACK signal that identifies the affected path and the missing PSN range. The signal carries a control tuple $\langle Path ID, K, Start_PSN, End_PSN \rangle$ in the reserved header fields, where $Start_PSN$ and End_PSN specify the first and last missing PSNs to be retransmitted.

Upon receiving the NACK, the sender's Path Mapper parses the $Path ID$. The Path Recognizer then records the loss event in a failure counter, which serves as a metric for potential path switching decisions (as detailed in §. 3.5). Meanwhile, the Path Recognizer immediately retrieves the target packet from the retransmission buffer and resends it. The retransmitted packet keeps its original PSN and $FPSN_\ell$. Therefore, even if the sender has already remapped the corresponding path index to a new path, the receiver can still interpret this retransmitted packet correctly, as discussed in case (3) in §. 3.3.2. This allows retransmitted packets to be sent through a different, healthy path rather than the original faulty one. After receiving a retransmitted packet, the receiver's Path Recognizer decodes its path and checks continuity in the same manner as for regular packets. The bitmap tracks all successfully received packets, and hence any duplicate retransmissions are safely discarded, ensuring data consistency. This design guarantees fast retransmission and reduces the impact of packet loss on FCT.

3.5 Path Manager

When packet loss occurs, the system should avoid degraded paths to prevent them from hurting overall performance. However, monitoring all possible paths would introduce additional state overhead and is often unnecessary. Using too many paths can also exacerbate packet OOO arrivals. PSN-PATH therefore needs to balance path utilization and system complexity. To this end, we introduce a Path Manager that maintains a set of K transmission paths. When a transmission path degrades, the Path Manager seamlessly redirects

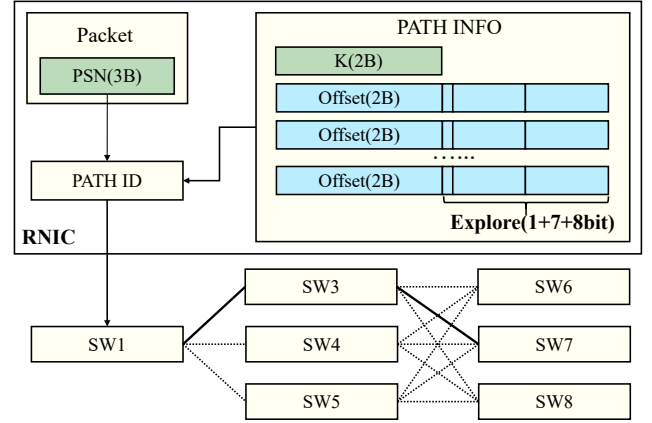


Figure 7: Based on the packet and the maintained path information, a Path ID is generated, and the switch forwards it along the corresponding path.

its traffic to a healthier alternative or incorporates a newly discovered high-quality path. It also ensures strict mapping, allowing the receiver to distinguish historical and new path allocations without additional signaling.

3.5.1 Path Evaluation. To evaluate path quality, PSN-PATH uses a lightweight ACK/NACK-based metric for each path. The ACK counter records successfully acknowledged packets, indicating how many packets have been delivered without retransmission, while the NACK counter records confirmed packet losses that trigger fast retransmission.

By periodically reading and resetting these counters, the sender is able to compute the effective delivery rate of each path within the current epoch and calculate the path score as $Score = ACK_count - \beta \times NACK_count$, where β is a penalty coefficient. This design is resource-efficient and easy to implement in RNIC hardware.

3.5.2 Path Switch. Based on the above metric, we further propose a Double Threshold Adaptive Path Switching Algorithm (DT-Switch). This algorithm sets two thresholds for each transmission path: (1) an exploration threshold ($T_{explore}$) and (2) a removal threshold (T_{remove}).

If the score of path i falls below T_{remove} , DT-Switch treats it as severely degraded and removes it from the transmission path set. For the path index ℓ mapped to path i , the Path Manager increments its offset (i.e., $O_\ell \leftarrow O_\ell + 1$) to search for an available source port until a healthy path is found. The Path Mapper then updates the packet header with the new offset and the corresponding $FPSN_\ell$ defined in §. 3.3.2. In this way, traffic is redirected to an available path by changing only the offset in the source-port calculation.

If the score of path i lies between T_{remove} and $T_{explore}$, DT-Switch regards it as degraded but not failed. Instead of removing it immediately, the Path Manager probes candidate paths and evaluates them using the same ACK/NACK-based metric. If a candidate path achieves a score no lower than S_i , DT-Switch replaces path i with this candidate, and advances O_ℓ until the selected source port maps to the new path.

After confirming a new path, DT-Switch gradually shifts traffic to the new path instead of moving all packets at once, avoiding abrupt OOO bursts. This process is controlled by the *PATH ID* metadata in Fig. 7, where the Explore field records the switching status, step size f , and candidate offset. During switching, the status bit is set to 1, indicating that the path index is in the switching phase. Then, for every f packets sent along the current path, one packet will be forwarded to the new path using the candidate offset. Each time f ACKs are received from the new path (default $f = 10$), f is decremented by 1, gradually increasing the traffic to the new path. If a NACK is received from the candidate path, DT-Switch terminates the probing phase and retains the original mapping. Note that during the gradual path switch process, old-path and new-path packets coexist and this does not affect the correctness of the packet loss identification, as analyzed in §. 3.3.2.

The parameters T_{explore} , T_{remove} , and β are configured according to network conditions. Based on the experimental results, RDMA performance begins to degrade at a packet loss rate of 0.4% and drops sharply at 1%. Consequently, we set the default value of β to 0.5 and collect *ACK_count* and *NACK_count* statistics for each path over one RTT. The two thresholds are then calculated based on the loss rates of 0.4% ($T_{\text{explore}} = 0.95$) and 1% ($T_{\text{remove}} = 0.85$).

4 Implementation

To implement PSN-PATH, we adopt a high-level synthesis (HLS)-based hardware design methodology. All core modules are described at the register-transfer level in C++ and synthesized using the Xilinx Vitis HLS tool, targeting the Xilinx Alveo U50 FPGA platform. Our hardware implementation is based on the RoCEv2 [10, 24] protocol and focuses on three key modules (i.e., the Path Mapper, Path Recognizer, and Path Manager). These modules are designed as event-driven, pipelined processes.

4.1 Implementation of Path Mapper

As a component of the sender data path, the Path Mapper is invoked after the construction of the RoCEv2 transport layer header and prior to UDP/IP encapsulation. Path Mapper maintains a set of control status registers that track the path parameters, including the number of transmission paths K , offset O_t , and additional path exploration metadata. We use the `#pragma HLS PIPELINE II=1` directive to ensure the throughput of each clock cycle. The values of K and O_t , along with an $FPSN_t$ flag that marks path switch boundary, are jointly encoded into a compact path control word. This control word is inserted directly after the RoCEv2 Base Transport Header in the packet's header fields. Such header extensions are widely adopted in high-performance RDMA protocol designs and do not incur performance degradation [23, 40]. The updated source port and encapsulated path control word then proceed with the packet into the UDP/IP encapsulation pipeline, ensuring accurate and efficient path management throughout the transmission process.

4.2 Implementation of Path Recognizer

The Path Recognizer is implemented at the RoCEv2 protocol parsing stage for stateless and low-latency loss detection. For each

incoming packet, it extracts the path control word from the RoCEv2 header, decodes K , O_t and $FPSN_t$. This decoding process is also fully pipelined. For per-QP loss tracking, the Path Recognizer maintains a bitmap for each QP in the FPGA Block RAM. Upon a packet with PSN arrival, it marks the corresponding bit of PSN as received, and performs continuity check by verifying whether $PSN - K$ has been marked as received or not. If so, the current path sequence is interpreted as continuous, and the packet is accepted and recorded in the bitmap. Otherwise, the hardware immediately detects a loss event and generates a NACK containing the missing PSN and the corresponding *Path ID* for fast retransmission.

To ensure pipeline efficiency, the backtracking and update are assigned to different BRAM banks and executed in parallel. We further use `#pragma HLS ARRAY_PARTITION` to allow loss detection and bitmap maintenance to complete within a single clock cycle. This implementation enables the receiver to provide timely loss feedback while sustaining high-speed RDMA packet processing.

4.3 Implementation of Path Manager

The Path Manager is implemented as a lightweight, event-driven finite state machine (FSM). It is triggered by the “packet processing complete” signal from the Path Recognizer, which carries the *Path ID* and ACK/NACK event type. Upon receiving the signal, the FSM indexes the corresponding counter pair (ACK_{cnt} , $NACK_{cnt}$), updates the counters, computes the path quality score, and compares it with T_{explore} and T_{remove} . If a switching condition is met, the FSM generates new mapping parameters in the next clock cycle. The update is propagated through a synchronized handshake protocol to ensure that the sender's Path Mapper and the relevant receiver contexts are updated atomically.

5 Performance Evaluation

5.1 Experiment Settings

Real system testbed. We develop an end-to-end hardware testbed by implementing three logically independent, yet tightly integrated subsystems across Xilinx Alveo U50 FPGAs. In this testbed, we conduct a small-scale leaf-spine network environment with 15 switches (consisting of 10 leaf switches and 5 spine switches) on the Xilinx Vitis HLS logic and develop a multi-port switch model capable of line-rate forwarding at 100Gbps. This switch model supports emulation of ECMP hashing, variable queue delays (ranging from 1μs to 500μs to represent path diversity), and random packet loss rates (from 0% to 1%).

Large-scale simulation environment. To evaluate PSN-PATH at datacenter scales, we employ the network simulator (ns-3) to create a large-scale multipath RDMA environment. In our simulation, we build a leaf-spine data center topology comprising 128 – 1024 hosts, organized as 16 leaf switches. The topology also includes 16 spine switches, with each spine switch fully interconnected to all leaf switches to enable multipath connectivity. To simulate industry-standard RDMA deployments, all network links are configured with a default bandwidth of 100Gbps, a default packet loss rate of 0.4% and a delay of 1μs, based on the data from NVIDIA Mellanox ConnectX-series NICs [23].

Performance metrics. As primary performance metrics, we use the 99-percentile flow completion time (FCT), representing the

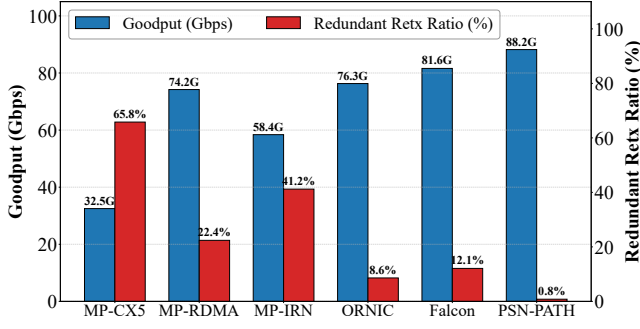


Figure 8: Goodput and redundant retransmissions under OOO arrivals.

time required to complete 99% of the traffic [46]. To account for the invalid throughput caused by packet loss and OOO arrival, we also utilize the average goodput of each flow as a key metric and quantify the number of retransmissions by recording the total count of NACK signals throughout the entire data transmission process.

5.2 Baselines and Benchmarks

Baselines. To assess the advantages of PSN-PATH, we compare our system with the following baselines:

- MP-CX5 [15]: The traditional lossless RoCEv2 implementation based on NVIDIA Mellanox ConnectX series NIC employs the GBN retransmission mechanism. We implement multipath support for it by modifying the port numbers and switch configurations.
- MP-IRN [20, 35, 39]: We implement the lossy RDMA approach in multipath environment. We refer to this configuration as MP-IRN, utilizing IRN as the representative classic lossy RDMA implementation upon which subsequent protocols are based.
- MP-RDMA [31]: A widely adopted multipath RDMA method that achieves load balancing across multiple paths via ECMP, but employs a bitmap overflow detection mechanism at the receiver for loss identification, resulting in retransmission delay.
- Falcon [39]: Falcon is a state-of-the-art schemes with OOO support for loss recovery. Falcon carries the receiver-side bitmap in ACK packets for precise retransmission and uses RACK-TLP to distinguish losses from reordering.
- ORNIC [32]: ORNIC tolerates reordering within an OOO Tolerance Distance (OTD), and triggers retransmission when the PSN distance exceeds the configured OTD.

To compare how different mechanisms handle OOO arrivals in lossy multipath RDMA, our implementation of Falcon and ORNIC focuses on their OOO-aware loss recovery mechanisms rather than their complete systems. Specifically, we implement Falcon’s Packet Delivery Layer (PDL) loss recovery logic, including bitmap-based feedback and RACK-TLP-based retransmission, and ORNIC’s OTD-based loss recovery logic, where retransmission is triggered when the PSN distance exceeds the configured OTD.

Benchmarks. To evaluate the adaptiveness of PSN-PATH, we conduct experiments under two widely used traffic benchmarks in data centers.

Table 1: Hardware Resource Utilization Comparison

Architecture	LUTs (Count)	FFs (Count)	BRAM (36Kb)
MP-RDMA [31]	18,240	26,450	86
PSN-PATH (Total)	8,142	12,725	14
– Path Mapper	1,245	2,108	0
– Path Recognizer	4,562	6,842	12
– Path Manager	2,335	3,775	2

Table 2: Hardware Resource Usage under Different K

K	LUTs (Count)	FFs (Count)	BRAM (36Kb)
4	8,142	12,725	14
8	9,014	13,165	14
16	9,205	15,670	14
32	11,587	17,758	15

- WebSearch [1, 3]: This benchmark simulates a backend datacenter workload, where 60% of flows are $\leq 200KB$, 37% range from 200KB to 10MB, and 3% exceed 10MB.
- AI Training [23, 47]: This benchmark simulates an all-to-all communication pattern, with each host sending a 10MB flow to every other host in the group.

5.3 Real Testbed Evaluation

5.3.1 Hardware Performance. To evaluate the hardware efficiency and verify the accuracy of loss detection under OOO arrivals, we implement PSN-PATH and conduct an experiment in our real system testbed. As illustrated in Fig. 8, legacy protocols suffer from severe performance degradation under OOO arrivals. MP-CX5 drops to 32.5Gbps due to GBN-based redundant retransmissions (65.8%), while MP-RDMA and MP-IRN still incur 22.4% and 41% redundant retransmissions, respectively. The SOTA Falcon further improves goodput to 81.6Gbps, but still introduces 12.1% redundant retransmissions. ORNIC reduces redundant retransmissions to 8.6%, but its delay-tolerant mechanism puts it at a disadvantage in goodput. In contrast, our PSN-PATH attains a 88.2Gbps goodput with nearly zero redundant traffic (0.8%), resolving the OOO-loss ambiguity.

The superior performance of PSN-PATH is attributed to its resource-efficient hardware design. We use MP-RDMA as the hardware comparison baseline because it is a classic and lightweight multipath RDMA design. As summarized in Table. 1, compared to MP-RDMA, which also employs bitmaps, PSN-PATH significantly lowers hardware resource overhead, reducing Look-Up Tables (LUTs), Flip-Flops (FF), and BRAM consumption by 55.3%, 51%, and 83.7%, respectively. Notably, by employing the Stateless OOO Detection, we significantly reduce the BRAM consumption compared to MP-RDMA. To handle OOO arrival, MP-RDMA relies heavily on large on-chip buffers (BRAM) and requires 86 BRAM blocks. However, PSN-PATH reduces BRAM usage by 83.7%. This is because our Path Recognizer replaces the need for deep, wide-window bitmaps with a stateless backtracking check. Instead of tracking the entire range of potential OOO packets, we only require a small BRAM to maintain path mapping parameters and a compact, per-QP arrival status for predecessor verification.

5.3.2 Hardware Scalability with Transmission Paths. We further analyze the hardware scalability of PSN-PATH by increasing the number of transmission paths K . Following common lossy RDMA

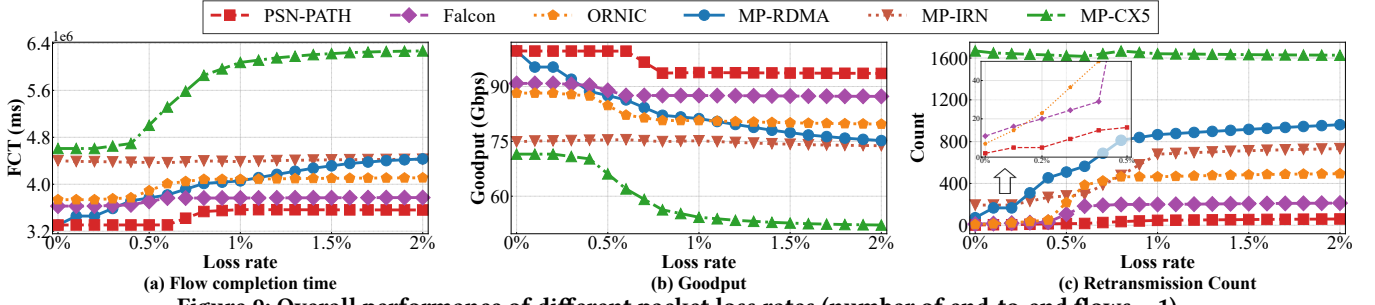


Figure 9: Overall performance of different packet loss rates (number of end-to-end flows = 1).

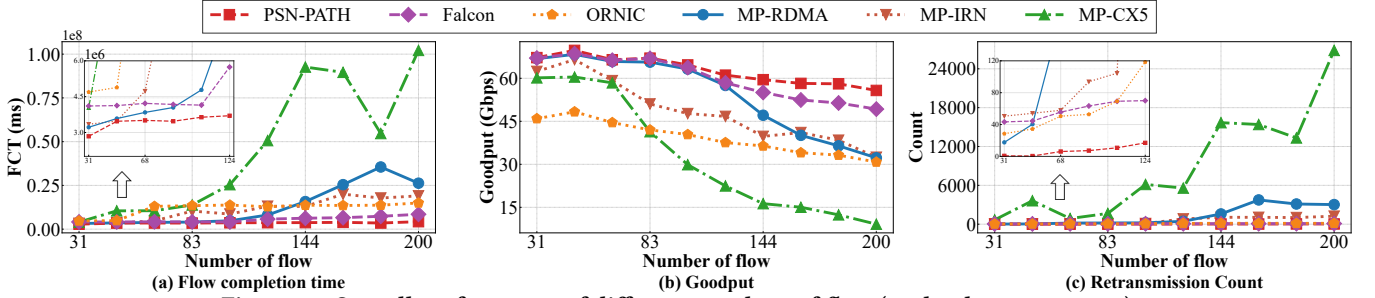


Figure 10: Overall performance of different numbers of flow (packet loss rate = 0.4%).

designs, each QP maintains a 32-bit bitmap in memory, which is sufficient for loss detection [39]. Hence, the memory cost is dominated by the number of QPs, and will not be effected by K . For example, consider a datacenter with 1000 servers. Even if each QP maintains $K = 32$ paths, PSN-PATH only needs 32 entries of per-index metadata encoded with 2 bytes, which can represent up to 2^{16} path indices per QP, sufficient for large-scale multipath deployment. We further synthesize PSN-PATH in HLS with different number of transmission paths. Table 2 reports the measured and estimated resource usage. BRAM remains at 14 blocks from $K = 4$ to $K = 16$ and increases slightly to 15 blocks when $K = 32$, mainly due to BRAM block granularity and metadata packing. When K increases from 4 to 32, the LUT usage increases from 8,142 to 11,587, and the FF usage increases from 12,725 to 17,758. This verifies that the hardware overhead increases only slightly with K and remains low and stable. This further confirms that PSN-PATH can scale to large networks with low hardware overhead.

5.4 Large Scale Simulations

To evaluate PSN-PATH under realistic large-scale settings, we run benchmark-driven simulations across different network conditions, system scales, and protocol configurations.

5.4.1 Overall Performance. Impact of loss rate. We first evaluate performance by varying the network packet loss rate from 0% to 2%. As shown in Fig. 9, PSN-PATH can always achieve the best performance, and in comparison with MP-CX5, MP-IRN, and MP-RDMA, it reduces FCT by over 40%, 23%, and 10%, and improves goodput by over 1.67 $\times$, 1.32 $\times$, and 1.11 $\times$, respectively. It also reduces retransmissions by up to 99% compared to other baselines. Even when compared to the OOO-aware baselines (i.e., ORNIC and Falcon), PSN-PATH still reduces FCT by over 18% and improves

goodput by more than 1.25 $\times$. We observe that the performance of MP-RDMA is next only to PSN-PATH when the packet loss rate is lower than 0.4% as shown in Fig. 9. This is because MP-RDMA triggers retransmission when the bitmap overflow occurs. Such passive design inherently provides a period for OOO packets to arrive, and reduces unnecessary retransmissions. Falcon achieves accurate packet loss detection by proactively querying the bitmap carried in packets. However, the associated processing overhead makes it inefficient in low packet loss conditions. Nevertheless, once the packet loss rate exceeds 0.4%, Falcon's proactive strategy can retransmit the lost packet immediately without waiting delay. That is why Falcon can work well in high-loss scenarios. By leveraging the logical PSN-to-path mapping mechanism, our PSN-PATH can proactively distinguish OOO arrivals from real packet loss but with lower overhead, thereby achieving the highest performance in all scenarios.

Impact of the number of flows. To evaluate the impact of network load on system performance, we vary the number of flows from 31 to 200 in this experiment. As shown in Fig. 10, PSN-PATH demonstrates significant advantages under high load (i.e., the number of flows exceeds 144). Compared to other solutions, our approach reduces the FCT by up to 95.81% and enhances goodput by 5.11 $\times$. This advantage stems from the efficient path management of PSN-PATH. By dynamically sending packets to high-quality paths, PSN-PATH is able to maintain lower FCT and higher goodput. It is interesting that MP-CX5 exhibits significant performance degradation and FCT fluctuations under high-load conditions, as shown in Fig. 10(a). This stems from its use of the GBN retransmission mechanism, where the volume of retransmitted data varies drastically depending on the loss location. For example, losing a packet at the start of the PSN sequence means resending all the packets behind it, but losing a packet at the end of the PSN sequence only needs

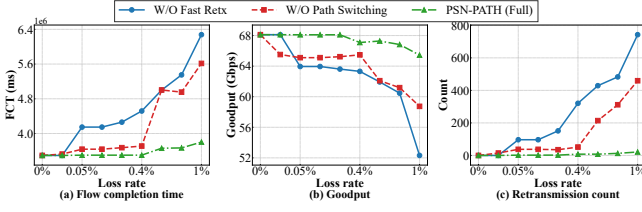


Figure 11: Ablation experiments under different packet loss rates (number of end-to-end flows = 1).

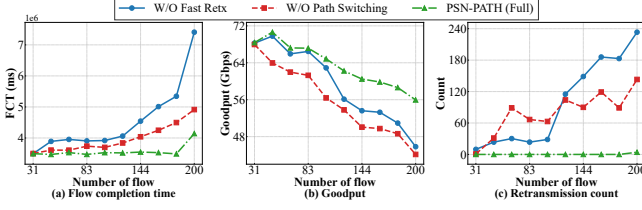


Figure 12: Ablation experiments under different numbers of flows (packet loss rate = 0.4%).

to resend itself. As a result, the retransmission count of MP-CX5 varies also fluctuates with the loss position, as shown in Fig. 10(c), which further leads to severe FCT variations as the number of flows increases.

5.4.2 Ablation Experiments. In addition to the PSN-to-path mapping mechanism, fast retransmission is also a key optimization in PSN-PATH. To evaluate their contributions, we conduct ablation experiments by disabling each mechanism separately and evaluate the performance under different packet loss rates and flow counts. As illustrated in Figs. 11 and 12, disabling the fast retransmission mechanism results in an average 22.56% increase in FCT and an average 7.32% decrease in goodput. This is because it can achieve quicker packet retransmission compared to the traditional passive timeout-based recovery mechanism. Similarly, without the path switch mechanism, the system also suffers a significant performance degradation, with FCT increasing by an average of 13.77% and goodput decreasing by 13.88%. We attribute this to the fact that the path switch mechanism redirects subsequent packets away from paths with persistent loss, which helps maintain stable goodput under dynamic network conditions. These results demonstrate the effectiveness of these mechanisms, and they can work in synergy to achieve high performance.

5.4.3 Performance on Different K . We next evaluate the impact of the transmission path number K by varying K from 1 to 32. The performance evaluation results in Fig. 13 show that the performance of all mechanisms increases when the K increases from 1 to 4. This is because the increasing K allows the packets to be transmitted on more different paths to exploit available network bandwidth. However, as K continues to increase, their performance shows a non-decreasing trend. Especially when $K > 16$, the performance of most mechanisms is even worse than the single-path case ($K = 1$). We attribute this phenomenon to the fact that transmitting packets across too many paths incurs excessive packet loss and OOO arrivals. This leads to frequent retransmissions, degrading overall

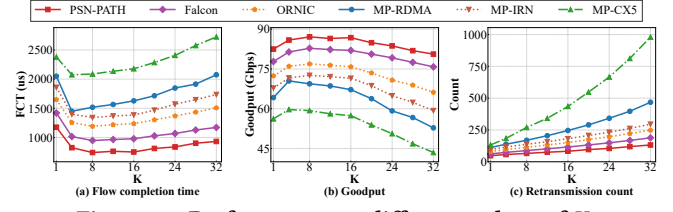


Figure 13: Performance on different values of K .

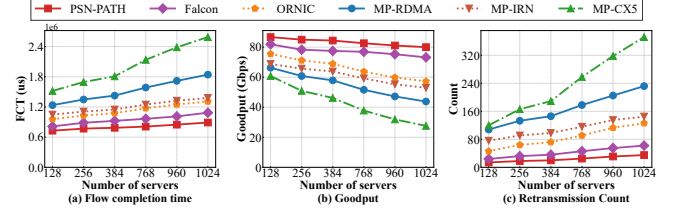


Figure 14: Performance of different system scales.

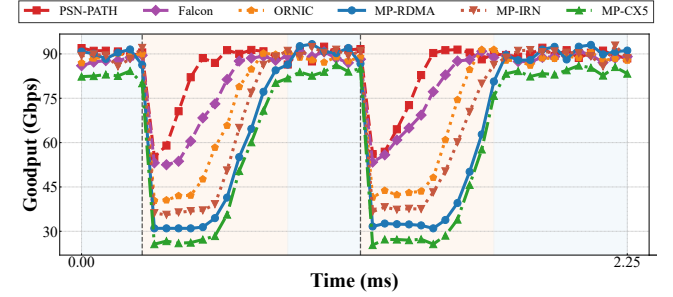


Figure 15: Performance of Robustness.

performance and even offsetting the benefit of multipath transmission. Therefore, we use $K = 4$ as the default setting, which provides a good balance between performance and hardware efficiency.

5.4.4 Performance of Scalability and Robustness. We also evaluate PSN-PATH under varying system scales and network dynamics. First, we evaluate the scalability by varying the number of servers from 128 to 1024, and report the results in Fig. 14. We can see that the performance of either mechanism degrades with the system scale. With the increasing of system scale, more flows compete for the limited network capacity, incurring more lost and OOO events. Nevertheless, PSN-PATH can still deliver comparatively steady and the highest performance at any system scale. For example, PSN-PATH shows the minimal FCT growth (17%) from 128 to 1024 servers. We attribute such an advantage to the path switching mechanism of PSN-PATH, which dynamically selects high-quality paths and hence safeguards performance against significant degradation.

Next, we evaluate robustness under burst network noise. In this experiment, we divide the measurement period into multiple time segments and inject noisy at specific intervals (i.e., burst losses). As shown in Fig. 15, compared to other baselines, PSN-PATH not only performs well but also can handle burst network noise fast. It recovers the goodput in about 0.31ms, which is 1.87 $\times$ faster than other baselines. Such fast performance recovery and high robustness advantage also come from the capability of PSN-PATH's path switch capability by redirecting packets to better paths.

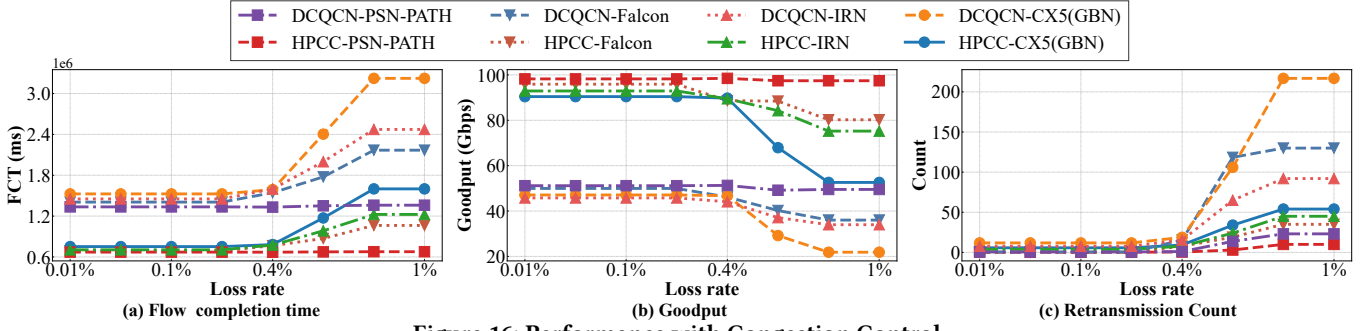


Figure 16: Performance with Congestion Control.

5.4.5 Performance with Congestion Control. PSN-PATH is orthogonal to congestion control: DCQCN and HPCC regulate the sending rate based on network conditions, while PSN-PATH resolves OOO-loss ambiguity and redirects traffic away from faulty paths without changing their control logic. We evaluate the performance of PSN-PATH under different packet loss rates with both DCQCN and HPCC.

As shown in Fig. 16, with either congestion control mechanism, PSN-PATH can always achieve the best performance in comparison with other baselines. In addition, a comparative analysis of the results in Figs. 16 and 9 further demonstrates that PSN-PATH aims to guarantee accurate loss identification, which can get benefit from congestion control. Compared to only PSN-PATH, they reduce the FCT by 63.2%, improve the goodput by 16.7%, and reduce the retransmission counts by 11.3%. This confirms that congestion control effectively helps PSN-PATH utilize network resources for improving the performance of multipath RDMA in the lossy network.

6 Related Work

RDMA has emerged as a dominant technology in modern datacenters, attracting extensive research interest [16, 21, 33, 49, 54]. Particularly, the advent of multipath RDMA addresses the limitations of single-path transmission and attracts substantial attention. Many existing works have focused on enhancing the performance of multipath RDMA transmission or ensuring RDMA reliability in lossy network environments. For example, MP-RDMA [31] achieves reliable multipath RDMA by using bitmap-overflow triggers for recovery. IRN [35] and Flor [20] replace the inefficient GBN protocol with SR to significantly improve the packet loss recovery efficiency. LIBRA [44] integrates segment routing with packet-level spraying to achieve deterministic path assignment, effectively improving tail performance for large message transfers. MP-CreditINT [36] uses a credit-based flow control with In-band Network Telemetry for fine-grained multipath congestion management to improve the system performance. However, these studies ignore a fundamental challenge. That is, the heterogeneous path latencies lead to OOO packet arrival, creating the ambiguity between OOO arrival and packet loss. Consequently, the system incurs substantial unnecessary retransmission overhead.

To tackle such problems, many solutions have been proposed. For instance, Maestro [41] leverages software-defined path selection and load balancing mechanisms to mitigate OOO arrivals. Falcon [39] employs the bitmap carried in ACK/SACK to distinguish

OOO arrivals and packet loss, thereby achieving accurate loss detection and retransmission. Besides, some recent studies try to improve OOO tolerance through more efficient buffer management. For example, ORNIC [32] and LEFT [17] do not distinguish OOO from loss but try to improve OOO tolerance using optimized bitmap management to reduce memory overhead.

Although existing studies have attempted to address OOO issues, they fail to distinguish between OOO arrival and actual packet loss with low overhead. This motivates us to propose PSN-PATH, a multipath RDMA mechanism that fundamentally eliminates the OOO-loss ambiguity at the RNIC level, offering a deployable solution that supports both multipath and lossy operations with lower additional overhead.

7 Conclusion

In this paper, we investigate how to improve the performance of multipath RDMA in lossy networks. We first find that the confusion of packet OOO with real loss is the key bottleneck that prevents multipath RDMA from operating efficiently in lossy networks. To this end, we propose PSN-PATH, which redefines ordering as a per-path constraint and establishes a lightweight PSN-to-path mapping relationship within the RNIC. By introducing a novel path-level backtracking continuity check, PSN-PATH eliminates cross-path OOO ambiguity and achieves fast, accurate loss confirmation and retransmission triggering. Furthermore, PSN-PATH measures and evaluates path quality at runtime, and dynamically adjusts packet transmission paths as needed to strengthen system robustness. Real-world trace-driven hardware simulations and large-scale ns-3 experiments demonstrate that PSN-PATH significantly improves multipath RDMA performance. It reduces FCT by up to 95.81%, improves goodput by 5.11 $\times$, and eliminates retransmissions by over 99%, in comparison with state-of-the-art solutions.

Acknowledgement

This research was supported by the NSF of China (No. 62432015, 62372200, 62232011, 62532010), China Scholarship Council (No. 202606410022), and Beijing Natural Science Foundation (L2603007). Deze Zeng is the corresponding author (e-mail: dzzeng@hust.edu.cn).

References

- [1] Vamsi Addanki, Wei Bai, Stefan Schmid, and Maria Apostolaki. 2024. Reverie: Low Pass Filter-Based Switch Buffer Sharing for Datacenters with RDMA and TCP Traffic. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 651–668.
- [2] Mohammad Al-Fares, Sivasankar Radhakrishnan, Barath Raghavan, Nelson Huang, and Amin Vahdat. 2010. Hedera: Dynamic Flow Scheduling for Data Center Networks. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Vol. 10. San Jose, USA, 89–92.
- [3] Mohammad Alizadeh, Albert Greenberg, David A Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. 2010. Data center TCP (DCTCP). In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM)*. 63–74.
- [4] Wei Bai, Shanim Sainul Abdeen, Ankit Agrawal, Krishan Kumar Attre, Paramvir Bahl, Ameya Bhagat, Govri Bhaskara, Tanya Brokhman, Lei Cao, Ahmad Cheema, Rebecca Chow, Jeff Cohen, Mahmoud Elhaddad, Vivek Ette, Igal Figlin, Daniel Firestone, Mathew George, Ilya German, Lakshmeet Ghai, Eric Green, Albert Greenberg, Manish Gupta, Randy Haagens, Matthew Hendel, Ridwan Howlader, Neetha John, Julia Johnstone, Tom Jolly, Greg Kramer, David Kruse, Ankit Kumar, Erica Lan, Ivan Lee, Avi Levy, Marina Lipshteyn, Xin Liu, Chen Liu, Guohan Lu, Yuesun Lu, Xiakun Lu, Vadim Makhervaks, Ulad Malashanka, David A. Maltz, Ilias Marinos, Rohan Mehta, Sharda Murthi, Anup Namdhari, Aaron Ogus, Jitendra Padhye, Madhav Pandya, Douglas Phillips, Adrian Power, Suraj Puri, Shachar Raindel, Jordan Rhee, Anthony Russo, Maneesh Sah, Ali Sheriff, Chris Sparacino, Ashutosh Srivastava, Weixiang Sun, Nick Swanson, Fuhou Tian, Lukasz Tomczyk, Vamsi Vadlamuri, Alec Wolman, Ying Xie, Joyce Yom, Lihua Yuan, Yanzhao Zhang, and Brian Zill. 2023. Empowering Azure Storage with RDMA. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 49–67.
- [5] Wenxue Cheng, Kun Qian, Wanchun Jiang, Tong Zhang, and Fengyuan Ren. 2020. Re-architecting Congestion Management in Lossless Ethernet. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 19–36.
- [6] Mingyu Choi, Sugi Lee, and Younghoon Kim. 2022. UD-assisted Multi-path Transport in RDMA. In *Proceedings of International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 127–129.
- [7] Zhenguo Cui and Steven Y Rim. 2020. G-PFC: A Packet-Priority Aware PFC Scheme for the Datacenter. In *Proceedings of the Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 385–388.
- [8] Adithya Gangidi, Rui Miao, Shengbao Zheng, Sai Jayesh Bondu, Guilherme Goes, Hany Morsy, Rohit Puri, Mohammad Riftadi, Ashmitha Jeevaraj Shetty, Jingyi Yang, Shuqiang Zhang, Mikel Jimenez Fernandez, Shashidhar Gandham, and Hongyi Zeng. 2024. RDMA over Ethernet for Distributed Training at Meta Scale. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM)*. 57–70.
- [9] Yixiao Gao, Qiang Li, Lingbo Tang, Yongqing Xi, Pengcheng Zhang, Wenwen Peng, Bo Li, Yaohui Wu, Shaozong Liu, Lei Yan, Fei Feng, Yan Zhuang, Liu Fan, Liu Pan, Liu Xingkui, Wu Zhongjie, Junping Wu, Zheng Cao, Chen Tian, Jinbo Wu, Jiayi Zhu, Haiyong Wang, Cai Dennis, and Jiesheng Wu. 2021. When Cloud Storage Meets RDMA. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 519–533.
- [10] Chuanxiong Guo, Haitao Wu, Zhong Deng, Gaurav Soni, Jianxi Ye, Jitu Padhye, and Marina Lipshteyn. 2016. RDMA over Commodity Ethernet at Scale. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM)*. 202–215.
- [11] Zerui Guo, Jiaxin Lin, Yuebin Bai, Daehyeok Kim, Michael Swift, Aditya Akella, and Ming Liu. 2023. LogNIC: A High-Level Performance Model for SmartNICs. In *Proceedings of the Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 916–929.
- [12] Maximilian J Heer, Benjamin Ramhorst, Jonas Dann, and Gustavo Alonso. 2025. RoCE BALBOA: Towards FPGA-enhanced RDMA. In *Proceedings of the European Conference on Computer Systems (EuroSys)*, Vol. 25.
- [13] Jinbin Hu, Yi He, Wangqing Luo, Jiawei Huang, and Jin Wang. 2024. Enhancing Load Balancing With In-Network Recirculation to Prevent Packet Reordering in Lossless Data Centers. *IEEE/ACM Transactions on Networking (TON)* 32, 5 (2024), 4114–4127.
- [14] Jinbin Hu, Wenxue Li, Xiangzhou Liu, Junfeng Wang, Bowen Liu, Ping Yin, Jianxin Wang, Jiawei Huang, and Kai Chen. 2025. FLB: Fine-grained Load Balancing for Lossless Datacenter Networks. In *Proceedings of the USENIX Annual Technical Conference (ATC)*. 365–380.
- [15] Hongjing Huang, Jie Zhang, Xuzheng Chen, Ziyu Song, Jiajun Qin, and Zeke Wang. 2025. SwCC: Software-Programmable and Per-Packet Congestion Control in RDMA Engine. In *Proceedings of USENIX Annual Technical Conference (ATC)*. 1243–1260.
- [16] Peihao Huang, Guo Chen, Xin Zhang, Can Liu, Hongyu Wang, Huijun Shen, Ying Bian, Yuanwei Lu, Zhenyuan Ruan, Bojie Li, Jiansong Zhang, Yongfeng Liu, and Zhigang Chen. 2025. Fast and Scalable Selective Retransmission for RDMA. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 1–10.
- [17] Peihao Huang, Xin Zhang, Zhigang Chen, Can Liu, and Guo Chen. 2024. LEFT: Lightweight and Fast packet Reordering for RDMA. In *Proceedings of Asia-Pacific Workshop on Networking (APNet)*. 67–73.
- [18] Jialong Li, Haotian Gong, Federico De Marchi, Aoyu Gong, Yiming Lei, Wei Bai, and Yiting Xia. 2024. Uniform-Cost Multi-Path Routing for Reconfigurable Data Center Networks. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM)*. 433–448.
- [19] Mingyan Li, Shuo Wang, Tao Huang, and Yunjie Liu. 2024. MPTD: Optimizing Multi-Path Transport with Dynamic Target Delay in Datacenters. *Cluster Computing* 27, 8 (2024), 11455–11469.
- [20] Qiang Li, Yixiao Gao, Xiaoliang Wang, Haonan Qiu, Yanfang Le, Derui Liu, Qiao Xiang, Fei Feng, Peng Zhang, Bo Li, Jianbo Dong, Lingbo Tang, Hongqiang Harry Liu, Shaozong Liu, Weijie Li, Rui Miao, Yaohui Wu, Zhiwu Wu, Chao Han, Lei Yan, Zheng Cao, Zhongjie Wu, Chen Tian, Guihai Chen, Dennis Cai, Jinbo Wu, Jiayi Zhu, Jiesheng Wu, and Jiwei Shu. 2023. Flor: An open high Performance RDMA Framework Over Heterogeneous RNICs. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. 931–948.
- [21] Shibao Li, Longfei Li, Wei Dou, Yunwu Zhang, Chengzhi Wang, Xuerong Cui, and Lianghai Li. 2025. SDLoRe: A loss recovery algorithm based on segment detection in lossy RDMA networks. *Computer Networks (CN)* 258 (2025), 111019.
- [22] Shibao Li, Chengzhi Wang, Yunwu Zhang, Chenxu Ma, Longfei Li, Xuerong Cui, and Jianhang Liu. 2023. FG-PFC: A Fine-Grained PFC Mechanism for Lossless RDMA. In *Journal of Physics: Conference Series (JPCS)*, Vol. 2575. IOP Publishing, 012008.
- [23] Wenxue Li, Xiangzhou Liu, Yunxuan Zhang, Zihao Wang, Wei Gu, Tao Qian, Gaoxiong Zeng, Shoushou Ren, Xinyang Huang, Zhenghang Ren, Bowen Liu, Junxue Zhang, Kai Chen, and Bingyang Liu. 2025. Revisiting RDMA Reliability for Lossy Fabrics. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM)*. 85–98.
- [24] Wenxue Li, Junyi Zhang, Yufei Liu, Gaoxiong Zeng, Zilong Wang, Chaoliang Zeng, Pengpeng Zhou, Qiaoling Wang, and Kai Chen. 2024. Cepheus: Accelerating Datacenter Applications with High-Performance RoCE-Capable Multicast. In *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 908–921.
- [25] Xinyi Li, Minglin Li, Xin Ai, Yongbo Gao, Jiang Shao, Zixuan Chen, Sen Liu, and Yang Xu. 2024. R-PFC: Enhancing RDMA Network With Restricted And Fine-grained PFC. In *Proceedings of IEEE/ACM International Symposium on Quality of Service (IWQoS)*. IEEE, 1–10.
- [26] Zhaoyi Li, Jiawei Huang, Shiqi Wang, and Jianxin Wang. 2024. Achieving Low Latency for Multipath Transmission in RDMA Based Data Center Network. *IEEE Transactions on Cloud Computing (TCC)* 12, 1 (2024), 337–346.
- [27] Hwijoon Lim, Wei Bai, Yibo Zhu, Youngmok Jung, and Dongsu Han. 2021. Towards Timeout-less Transport in Commodity Datacenter Networks. In *Proceedings of European Conference on Computer Systems (EuroSys)*. 33–48.
- [28] Qian Liu, Robert D Russell, Fabrice Mizero, Malathi Veeraraghavan, John Dennis, and Benjamin Jamroz. 2015. Implementation of PFC and RCM for RoCEv2 Simulation in OMNeT++. *arXiv preprint arXiv:1509.03559* (2015).
- [29] Jie Lu, Jiaqi Gao, Fei Feng, Zhiqiang He, Menglei Zheng, Kun Liu, Jun He, Binbin Liao, Suwei Xu, Ke Sun, Yongjia Mo, Qinghua Peng, Jilie Luo, Qingxu Li, Gang Lu, Zishu Wang, Jianbo Dong, Kunling He, Sheng Cheng, Jiamin Cao, Hairong Jiao, Pengcheng Zhang, Shu Ma, Lingjun Zhu, Chao Shi, Yangming Zhang, Yiquan Chen, Wei Wang, Shuhong Zhu, Xingru Li, Qiang Wang, Jiang Liu, Chao Wang, Wei Lin, Ennan Zhai, Jiesheng Wu, Qiang Liu, Binzhang Fu, and Dennis Cai. 2025. Alibaba Stellar: A New Generation RDMA Network for Cloud AI. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM)*. 453–466.
- [30] Xiaoyi Lu, Dipti Shankar, Haiyang Shi, and Dhabaleswar K Panda. 2018. SparkuDAPL: Cost-Saving Big Data Analytics on Microsoft Azure Cloud with RDMA Networks. In *Proceedings of IEEE International Conference on Big Data (IEEE BigData)*. IEEE, 321–326.
- [31] Yuanwei Lu, Guo Chen, Bojie Li, Kun Tan, Yongqiang Xiong, Peng Cheng, Jiansong Zhang, Enhong Chen, and Thomas Moscibroda. 2018. Multi-Path Transport for RDMA in Datacenters. In *Proceedings of the USENIX symposium on networked systems design and implementation (NSDI)*. 357–371.
- [32] Jiandong Ma, Zhichuan Guo, Yipeng Pan, Mengting Zhang, Zhixiang Zhao, Zezheng Sun, and Yiwei Chang. 2024. ORNIC: A High-Performance RDMA NIC with Out-of-Order Packet Direct Write Method for Multipath Transmission. *Electronics* 14, 1 (2024), 88.
- [33] Qingkai Meng, Yiran Zhang, Shan Zhang, Zhiyuan Wang, Tong Zhang, Hongbin Luo, and Fengyuan Ren. 2023. Switch-Assistant Loss Recovery for RDMA Transport Control. *IEEE/ACM Transactions on Networking (TON)* 32, 3 (2023), 2069–2084.
- [34] Rui Miao, Lingjun Zhu, Shu Ma, Kun Qian, Shujun Zhuang, Bo Li, Shuguang Cheng, Jiaqi Gao, Yan Zhuang, Pengcheng Zhang, Rong Liu, Chao Shi, Binzhang Fu, Jiayi Zhu, Jiesheng Wu, Dennis Cai, and Hongqiang Harry Liu. 2022. From Luna to Solar: The Evolutions of the Compute-to-Storage Networks in Alibaba Cloud. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM)*. 753–766.
- [35] Radhika Mittal, Alexander Shipner, Aurojit Panda, Eitan Zahavi, Arvind Krishnamurthy, Sylvia Ratnasamy, and Scott Shenker. 2018. Revisiting Network

- Support for RDMA. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM)*. 313–326.
- [36] Yi Pan and Jiali You. 2025. MP-CreditINT: Enhancing multi-path RDMA transport with credit-based congestion control and in-band network telemetry. *Computer Networks (CN)* (2025), 111665.
- [37] Jiang Shao, Xinyi Li, Minglin Li, Sen Liu, and Yang Xu. 2023. S-PFC: Enabling Semi-Lossless RDMA Network with Selective Response to PFC. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 1135–1141.
- [38] Arjun Singhvi, Aditya Akella, Dan Gibson, Thomas F. Wenisch, Monica Wong-Chan, Sean Clark, Milo M. K. Martin, Moray McLaren, Prashant Chandra, Rob Cauble, Hassan M. G. Wassef, Behnam Montazeri, Simon L. Sabato, Joel Scherpelz, and Amin Vahdat. 2020. 1RMA: Re-envisioning Remote Memory Access for Multitenant Datacenters. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM)*. 708–721.
- [39] Arjun Singhvi, Nandita Dukkkipati, Prashant Chandra, Hassan M. G. Wassef, Naveen Kr. Sharma, Anthony Rebello, Henry Schuh, Praveen Kumar, Behnam Montazeri, Neelesh Bansod, Sarin Thomas, Inho Cho, Hyejeong Lee Seibert, Baijun Wu, Rui Yang, Yuliang Li, Kai Huang, Qianwen Yin, Abhishek Agarwal, Srinivas Vaduvatha, Weihuang Wang, Masoud Moshref, Tao Ji, David Wetherall, and Amin Vahdat. 2025. Falcon: A Reliable, Low Latency Hardware Transport. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM)*. 248–263.
- [40] Cha Hwan Song, Xin Zhe Khoi, Raj Joshi, Inho Choi, Jialin Li, and Mun Choon Chan. 2023. Network Load Balancing with In-network Reordering Support for RDMA. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM)*. 816–831.
- [41] Feng Tian, Yang Zhang, Wei Ye, Cheng Jin, Ziyang Wu, and Zhi-Li Zhang. 2021. Accelerating Distributed Deep Learning using Multi-Path RDMA in Data Center Networks. In *Proceedings of the ACM SIGCOMM Symposium on SDN Research (SOSR)*. 88–100.
- [42] Junliang Wang, Baohong Lin, Jiao Zhang, Mengyu Sun, and Yongchen Pan. 2025. An Optimized RDMA QP Communication Mechanism for Hyperscale AI Infrastructure. *Cluster Computing* 28, 1 (2025), 1–18.
- [43] Shicheng Wang, Menghao Zhang, Xiao Li, Qiyang Peng, Haoyuan Yu, Zhiliang Wang, Mingwei Xu, Xiaohe Hu, Jiahai Yang, and Xingang Shi. 2025. Hawk-eye: Diagnosing RDMA Network Performance Anomalies with PFC Provenance. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM)*. Association for Computing Machinery, New York, NY, USA, 481–495.
- [44] Xinsuo Wang, Baihua Chen, and Lei Liu. 2025. LIBRA: Multi-Path Transport in Distributed Training Data Center Network. In *Proceedings of IEEE International Conference on Network Protocols (ICNP)*. IEEE, 1–12.
- [45] Yibo Wang, Wei Wang, Qiaojun Hu, Yiyang Li, Yajie Li, Yongli Zhao, Xiaoyu Wang, and Jie Zhang. 2025. Ltfc: Loss-Tolerant Flow Control with RDMA Network for Machine Learning Clusters. In *Proceedings of IEEE International Conference on Communications (ICC)*. IEEE, 5343–5348.
- [46] Zilong Wang, Layong Luo, Qingsong Ning, Chaoliang Zeng, Wenxue Li, Xinchun Wan, Peng Xie, Tao Feng, Ke Cheng, Xiongfei Geng, Tianhao Wang, Weicheng Ling, Kejia Huo, Pingbo An, Kui Ji, Shideng Zhang, Bin Xu, Ruiqing Feng, Tao Ding, Kai Chen, and Chuanxiong Guo. 2023. SRNIC: A Scalable Architecture for RDMA NICs. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 1–14.
- [47] Ertza Warraich, Ali Imran, Annus Zulfiqar, Shay Vargaftik, Sonia Fahmy, and Muhammad Shahbaz. 2025. Reimagining RDMA Through the Lens of ML. *IEEE Computer Architecture Letters* (2025).
- [48] Xingda Wei, Rongxin Cheng, Yuhang Yang, Rong Chen, and Haibo Chen. 2023. Characterizing Off-path SmartNIC for Accelerating Distributed Systems. In *Proceedings of USENIX the Symposium on Operating Systems Design and Implementation (OSDI)*. 987–1004.
- [49] Zhiyi Yang, Zhexiong Li, Deze Zeng, and Lin Gu. 2025. DCTS-RDMA: Adaptive FEC via Dynamic Coding for Efficient RDMA over Lossy Networks. In *Proceedings of IFIP International Conference on Network and Parallel Computing (NPC)*. Springer, 235–246.
- [50] Jin-Li Ye, Chien Chen, and Yu Huang Chu. 2018. A Weighted ECMP Load Balancing Scheme for Data Centers Using P4 Switches. In *Proceedings of IEEE International Conference on Cloud Networking (CloudNet)*. IEEE, 1–4.
- [51] Ling Zhang, Xuefei Yang, Zhenlong Wan, Hang Liu, Wei Gu, Pingjing Liu, Qilin Dai, Shanwei Ye, and Yingcheng Lin. 2024. A High-Performance RDMA NIC With Ultra-Highly Scalable Connections. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)* (2024).
- [52] Chenxingyu Zhao, Jaehong Min, Ming Liu, and Arvind Krishnamurthy. 2025. White-Boxing RDMA with Packet-Granular Software Control. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 427–449.
- [53] Sirui Zhou, Yizhou Gong, Zhefan Fan, Yunfang Chen, Wei Zhang, Wanting Tian, and Yunqu Liu. 2022. SR-DCQCN: Combining SACK and ECN for RDMA Congestion Control. In *Proceedings of the IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 788–794.
- [54] Tianyu Zuo, Tao Sun, Shuyong Zhu, Wenxiao Li, Lu Lu, Zongpeng Du, and Yujun Zhang. 2024. LoWAR: Enhancing RDMA over Lossy WANs with Transparent Error Correction. In *Proceedings of IEEE/ACM International Symposium on Quality of Service (IWQoS)*. IEEE, 1–10.