LoWAR: Enhancing RDMA over Lossy WANs with Transparent Error Correction

Tianyu Zuo[†‡], Tao Sun[§], Shuyong Zhu[‡], Wenxiao Li[‡], Lu Lu[§], Zongpeng Du[§], Yujun Zhang[†‡*]
[†]University of Chinese Academy of Sciences, [‡]Institute of Computing Technology, Chinese Academy of Sciences,
[§]China Mobile Research Institute
{zuotianyu22s, zhushuyong, liwenxiao, nrcyujun}@ict.ac.cn, {suntao, lulu, duzongpeng}@chinamobile.com

***Abstract*—As the increase of geographically distributed applications continues, the demand for high-speed, long-distance data transmission across wide area networks (WANs) has significantly increased. Remote Direct Memory Access (RDMA) is extensively deployed in data center networks (DCNs) for its high throughput, low latency, and reduced CPU utilization, and its extension to WANs is expected to fully leverage these benefits. However, existing RDMA solutions, while demonstrating superior performance in data centers, face a performance gap over WANs due to their reliance on DCNs for optimal performance and lack of optimization for WANs' high latency and loss rates. To bridge this gap, we introduce Lossy Wide-Area RDMA (LoWAR), a high-goodput, high-reliability RDMA solution for lossy WANs. LoWAR incorporates a forward error correction (FEC) shim layer to protect RDMA messages from packet loss, thus minimizing the inefficiency of retransmissions. It also fully offloads processing to RNICs with minimal computational overhead and storage burden, operating transparently on RNICs without requiring modifications to existing applications and networks. We implement a LoWAR prototype with FPGA and evaluate its performance through testbed experiments. The results demonstrate LoWAR's enhanced performance in lossy WANs: in WANs with 40ms RTT and 0.001% to 0.01% loss rates, LoWAR increases RDMA goodput by 2.05 to 5.01 times, reduces average flow completion times (FCTs) by 3.5% to 12.2%, and eliminates 99th percentile tail FCTs in most scenarios.**

***Index Terms*—Remote Direct Memory Access, Wide Area Network, Forward Error Correction**

I. Introduction

Over the past decade, Remote Direct Memory Access (RDMA) has become crucial in high-speed data center networks (DCNs) [1]. RDMA benefits from kernel bypass and hardware-offloaded protocols, leading to high throughput, low latency, and minimal CPU usage. RDMA over Converged Ethernet (RoCE) is now widely employed in large data centers [2]–[4], making itself the de-facto standard for DCNs.

Concurrently, geographically distributed applications have increased to offer large-scale online services, including high-performance computing, cloud storage, and distributed or federated machine learning [5]–[9]. As shown in Fig. 1, these applications span across multiple data centers in various regions or continents and rely on inter-DC WANs for communication. Major service providers, such as Meta [10] and Google [11], have expanded their global WANs to meet these needs, with Google offering links with bandwidth up to 100Gbps in some cases [12].

* Corresponding author.

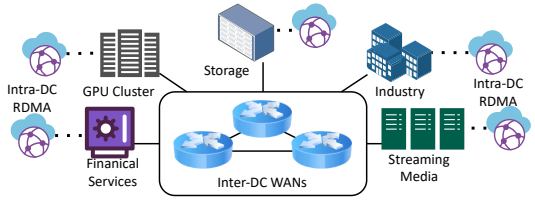


Fig. 1: Emerging scenarios for RDMA over WANs.

The simultaneous increase in both transmission demand and WAN capacity raises the question: *Can RDMA's success in DCNs be replicated in WANs?* Our position is affirmative, considering RDMA's benefit in WANs. RDMA demonstrates exceptional bandwidth utilization in WANs [13]–[15] compared to TCP, which struggles with high bandwidth-delay product (BDP) environments and requires extensive kernel tuning [16]. As RDMA already serves as the paradigmatic transport protocol in modern intra-DC applications, its adoption in inter-DC networks simplifies the deployment of geo-distributed applications by eliminating the need for different software interfaces. Moreover, RDMA's minimal CPU overhead is increasingly advantageous as inter-DC communication scales up. The conserved CPU resources can be allocated to non-communicative tasks, thereby enhancing the overall efficiency of distributed systems.

However, adapting RDMA for WANs presents significant obstacles due to the inherent high latency of WANs, which ranges from a few to tens of milliseconds, and the high rates of packet loss, typically between 0.001% and 0.1% [17]–[20]. Commercial RoCE, while optimized for lossless DCNs, shows suboptimal performance on lossy links because of its go-back-N (GBN) retransmission. This degradation is pronounced in WANs with longer round-trip times (RTTs). Although some solutions [21]–[25] have tried to support lossy RDMA by replacing GBN with selective retransmission (SR), their designs are primarily optimized for DCNs and do not translate well to high-latency lossy WANs. Therefore, there is a need for an effective RDMA solution optimized for WANs that circumvents the drawbacks of retransmission, ensuring optimal performance over lossy WANs.

Forward Error Correction (FEC) emerges as a promising approach. It offers latency-independent packet recovery and occupies less additional bandwidth than the bandwidth

979-8-3503-5012-8/24/$31.00 ©2024 IEEE

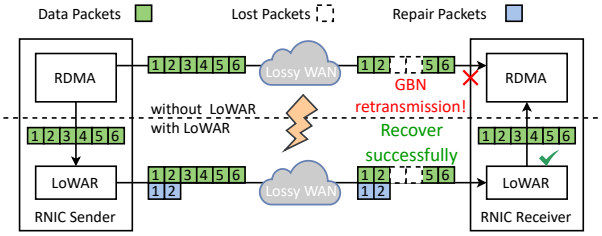


Fig. 2: Workflow of LoWAR.

wastage introduced by GBN retransmissions. End-to-end FEC requires no modifications to the WAN infrastructure, making it cost-effective and readily deployable. Moreover, FEC is expected to benefit from the hardware offloading capabilities of RDMA NICs (RNICs), thus reducing the computational burden led by decoding and encoding calculations.

However, a solution that seamlessly integrates FEC into RDMA faces several critical challenges: *(i)* **transparency.** It should avoid CPU involvement and modifications to existing upper-layer applications. *(ii)* **compatibility.** It should be compatible with RDMA's message-based transmission, providing uniform protection for messages of different sizes. *(iii)* **conciseness.** It should maintain RDMA's high speed, with optimized hardware utilization and minimal computational latency. *(iv)* **adjustability.** It should support adjustable parameters in order to accommodate various network conditions.

In this paper, we introduce Lossy Wide-Area RDMA (LoWAR), a high-goodput and high-reliability solution for lossy WANs. As shown in Fig. 2, LoWAR adopts an FEC shim layer to protect RDMA messages against packet loss. It generates **repair packets** from outgoing RDMA messages, transmits them along with the **data packets**, and then recovers corrupted messages at the receiver's end, therefore mitigating retransmissions and the associated inefficiency. LoWAR works transparently, obviating the need for modifications to host applications or the network infrastructure. It is fully hardware-offloaded and enables real-time loss recovery with minimal storage and latency overhead. Meanwhile, LoWAR introduces bidirectional negotiation for adjustable FEC parameters via header extensions.

We implement a LoWAR prototype with FPGA and evaluate its efficiency through testbed experiments. The results demonstrate LoWAR's high goodput and high reliability in lossy WANs. Specifically, in WANs with a 40ms RTT and loss rates ranging from 0.001% to 0.01%, LoWAR increases RDMA goodput by 2.05 to 5.01 times, reduces flow completion times (FCTs) by 3.5% to 12.2%, and eliminates 99th percentile tail FCTs in most scenarios.

This paper makes the following major contributions:

- We explore enhancing RDMA over lossy WANs using end-to-end FEC, detailing challenges and proposing integration strategies (§III).
- We propose LoWAR, a high-goodput, high-reliability solution that seamlessly integrates FEC into RDMA, thereby optimizing RDMA performance in lossy WANs (§IV).
- We implement a LoWAR prototype with FPGA and evaluate it with testbed experiments. The results confirm LoWAR's enhanced efficiency in lossy WANs (§V and §VI).

II. Background and Motivation

A. RDMA Preliminary

RDMA is a hardware protocol enabling direct memory access on remote hosts without CPU intervention. Leveraging RNIC hardware to offload transport, RDMA provides higher throughput, lower latency, and reduced CPU overhead compared to traditional software transports such as TCP.

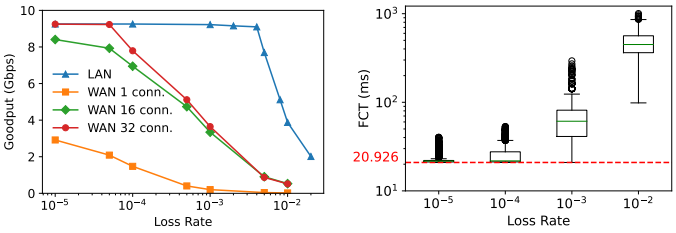
Message-based transport. RDMA operates as a message-based protocol. Message is the basic unit of RDMA transmission. A single message may contain one or several packets according to the transmission request. Users call kernel-bypass and zero-copy interfaces, known as *verbs*, to invoke transmission requests. The RNIC then directly retrieves data from the specified memory addresses in the requests via DMA, packages messages according to RoCE MTU (i.e., maximum payload per packet), and appends RDMA transport headers. Each packet contains an operation code (OpCode) that defines the message's semantics and separates messages within the same Queue Pair (QP) by indicating each message's *First*, *Middle*, *Last* or *Only* packets.

Reliable transmission. Commercial RoCE uses GBN retransmission for reliable connections. To this end, each packet contains a Packet Sequence Number (PSN), with the disrupted sequence triggering a retransmission request from the lost packet. The GBN retransmission works fine in RDMA's initial InfiniBand implementation, where underlying links ensure an extremely low loss rate and complex loss recovery could be an unnecessary expense [1] for RNIC hardware. However, this simplicity may lead to inefficiencies when the link is lossy, as any packet loss essentially leads to bandwidth wastage in the entire RTT. Therefore, commercial RoCE relies on lossless links for optimal performance.

Lossy RDMA. Lossy RDMA with selective retransmission (SR) has been studied to adapt RDMA for general lossy links. As for hardware-based SR, IRN [22] implements SR through per-connection static bitmaps, which track out-of-order packets and enable selective acknowledgment. SRNIC [24] onloads the reordering buffer and bitmap to host memory and handles selective retransmission with hardware-software co-design, thus eliminating RNIC's storage demand for SR. Recent commercial RNICs, like Mellanox's ConnectX-6 [21], have begun incorporating SR capabilities in some scenarios. On the software front, RoGUE [23] and Flor [25] introduce software overlay on commercial RNICs, achieving reliable transmission and selective retransmission based on unreliable RDMA connection.

B. RDMA in WANs

As RDMA becomes increasingly crucial in DCNs, its extension to WANs is beneficial and essential. Compared with traditional TCP, RDMA offers substantially higher bandwidth utilization and lower CPU overhead. These are especially



(a) Average Goodput (b) FCT

Fig. 3: Performance degradation of RDMA over lossy WANs.

beneficial for high-throughput applications over WANs, such as large-scale data transfers. Research integrating RDMA with GridFTP [26] shows that single-stream RDMA on a 10Gbps link achieves a 20% increase in throughput over TCP at a latency of 120ms, highlighting its superior efficiency. Further study [27] on a 40Gbps link within the ESnet testbed [28] indicate that RDMA can fully utilize the WAN's capacity, whereas the throughput of TCP struggles to exceed half of RDMA. Additionally, while TCP's CPU usage peaks at 100% at the sender and receiver's end, RDMA's remains remarkably low at 1-2%, as RDMA eliminates unnecessary data copying and context switching. Other studies [15], [29]–[31] have also demonstrated RDMA's high efficiency in WANs with extensive solid experiments. Prevailing high-speed intra-DC applications opt for RDMA as their transport protocol. Extending RDMA in WANs helps to efficiently deploy these applications in WANs, maintaining interface consistency and avoiding the extra costs of switching between TCP-based and RDMA-based communication modes.

C. RDMA in Lossy WANs

In addition to high latency, real-world wide area networks face significant non-congestion loss due to various factors, such as random bit errors, optical link degradation [32], and software or hardware malfunctions. Research [18] indicates packet loss rates between 0.01% and 0.07% in cloud connections across data centers operated by leading corporations, with random loss accounting for up to 65% of these losses. Experience [20] based on real-world inter-DC workloads also reveals a packet loss rate close to 0.01%. Previous research [17] has also shown that non-congestion loss rates in long-distance optical links range from 0.01% to 0.1%, in singleton and burst forms.

Commercial RoCE relies on lossless links for optimal performance. However, in lossy WANs, GBN retransmission is frequently triggered due to high packet loss rates, and it brings pronounced bandwidth wastage due to WANs' high RTT. We test the performance of ConnectX-5 RNICs in lossy WANs, including goodput and FCTs, on a 10Gbps link with a 40ms RTT (see §VI for testbed's detail). Our experiments indicate a notable degradation in the performance of commercial RoCE:

Goodput reduction. As illustrated in Fig. 3a, in the WAN setting, although CX-5 can saturate the link with a single connection at a goodput of 9.21Gbps, the goodput plummets to 2.92Gbps with just a 0.001% loss, and declines to nearly zero at losses exceeding 0.1%. In contrast, in the LAN setting, a significant reduction in throughput is observed only after losses surpass 0.5%. Applying parallel connections dilutes the data transmitted per connection within a single RTT, thereby reducing retransmission costs. Nevertheless, its benefit plateaus (e.g., no added gain from 16 to 32 parallel connections), and not all applications are compatible with parallel connections.

FCTs increase. Fig. 3b test the flow completion times of common 1MB messages in the WAN setting. As the loss rate escalates, FCTs increase accordingly, diverging from that observed in lossless WANs. Packet loss also introduces a significant long-tail FCT, as lost retransmission packets further trigger retransmissions.

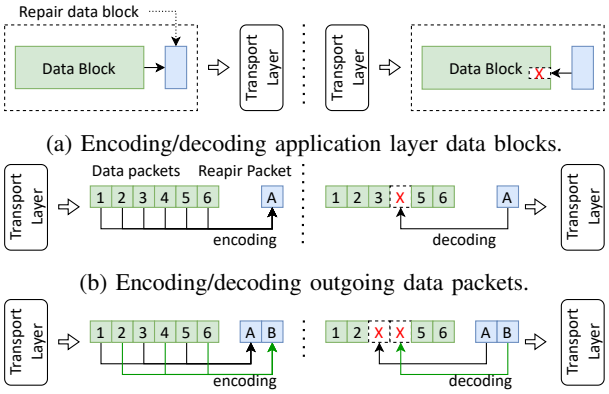
Lossy RDMA solutions eliminate commercial RoCE's dependence on lossless links through selective retransmission, achieving improved performance in general ethernet. However, these solutions are designed and optimized for DCNs, which have relatively low latency and loss rates and are less well-suited to high-latency lossy WANs. On the one hand, their on-chip memory and CPU demands escalate with increased RTT and loss rates, thereby raising deployment costs; on the other hand, the long-tail effect of retransmission delay is inadequately mitigated, posing potential challenges in applications that are sensitive to delays or depend on synchronization.

Motivated by the inconvenience caused by retransmissions in lossy WANs, we propose leveraging FEC to circumvent the need for retransmission. We systematically analyze the considerations for FEC's integration and achieve high-goodput, high-reliability RDMA in lossy WANs with the derived strategies.

III. Analysis

FEC has been a staple in network communications for its ability to compensate for data loss by utilizing additional bandwidth. FEC enables in-flight loss recovery by encoding redundant data at the sender's end and recovering lost data with this redundancy at the receiver's end. It enhances transmission reliability, especially when retransmission is impractical or too costly. However, integrating FEC into RDMA is non-trivial due to RDMA's hardware-offloaded and high-speed nature. These challenges, as we have outlined in §I, necessitate meticulous design considerations.

***Consideration I*: the placement of FEC in the protocol stack.** FEC operates at different positions within the protocol stack to protect different layers. Designs like RTP [33], rQUIC [34], and Nem [35] focus on ensuring application layer data integrity and operate above the transport layer. They encode and decode data blocks without concerning their encapsulation or transmission, as shown in Fig. 4a. In contrast, methods such as Maelstrom [17] aim to protect data packets that have been encapsulated by the transport layer. They directly encode and decode the data packets, as shown in Fig. 4b. For RDMA, it's better to perform FEC as a shim layer after transport layer encapsulation. Since encoding is compute-intensive, processing data blocks before they are managed by RNIC fails



(a) Encoding/decoding application layer data blocks.

(b) Encoding/decoding outgoing data packets.

(c) Encoding/decoding outgoing data packets, with interleaving.

Fig. 4: Common implementations of FEC.

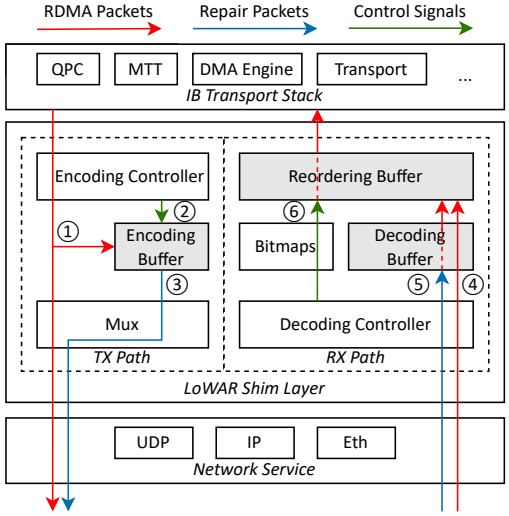


Fig. 5: LoWAR architecture.

to utilize RNIC's offloading capabilities. Additionally, this method circumvents the need for modifications to the RDMA transport layer and upper-layer applications.

***Consideration II*: the selection of the error correction code.** At the core of FEC lies the selection of error correction codes. Among these codes, we advocate the use of XOR-based code. Although more sophisticated coding algorithms, such as Reed-Solomon code, convolutional code, or fountain code, can provide better loss tolerance, their packet-level implementation may strain RNICs' resources, due to their extensive storage demands for packet buffering or hardware complexity for coding calculation. Compared with these codes, XOR-based code stores only the intermediate value in computation due to its associative nature, and its calculation is hardware-friendly. Additionally, by scheduling packet encoding and decoding at a certain interleaving depth, as illustrated in Fig. 4c, the interleaved XOR-based code can create multiple repair packets from one single coding block, showing loss tolerance for bursty losses as well.

***Consideration III*: the impact of message size variability**. Packet-level FEC is usually parameterized by (r, c) tuple, where r denotes the size of the coding block (i.e., number of data packets) and c is the number of repair packets generated per coding block. While dividing data packet flow into such coding blocks is crucial in coding calculation, the boundaries of different RDMA messages are also important. A coding block that spans multiple messages can result in delayed arrival of repair packets if these messages belong to the coding block but are not transmitted sequentially. Such timing mismatch complicates the decision to wait for repair packets or to trigger retransmission. As a remedy, each new message should initiate its own encoding and decoding cycle, ensuring the seamless transmission of data packets and their corresponding repair packets to start loss recovery immediately at the receiver's end.

***Consideration IV*: the adjustability of FEC parameters.** WANs usually experience fluctuating link conditions. Insufficient redundancy rates or interleaving depths may inadequately recover packets, whereas excessive redundancy leads to increased bandwidth consumption. Meanwhile, Some connections, like dedicated Data Center Interconnects or intra-DC links, might not need FEC. Therefore, it's necessary to support adjustable parameters at the hardware level and establish a negotiation channel between hardware endpoints. Such a channel enables the receiver to relay the actual link condition, facilitating proactive adjustments to FEC parameters based on real-time feedback.

***Consideration V*: the utilization of RDMA's features.** The hardware-offloaded acceleration capabilities of RNICs, alongside RDMA's structured protocol, lay a robust foundation for FEC integration into RDMA. Firstly, the offloading capabilities of RNICs enable hardware-accelerated XOR computations. Secondly, message identification is easy by examining the QP Number (QPN) and OpCode in the BTH field. Thirdly, the sequential PSN of data packets aids in locating lost packets within RDMA messages. Lastly, the uniform length of RDMA packets, consistent with the RoCE MTU, simplifies the structuring of repair packets.

IV. LoWAR Design

A. Architecture Overview

Guided by these considerations, we introduce Lossy Wide-Area RDMA (LoWAR), a high-goodput, high-reliability RDMA solution optimized for lossy WANs. As shown in Fig. 5, LoWAR works by adopting a hardware-offloaded FEC shim layer in hardware RDMA protocol stacks.

LoWAR is structured around two data pathways: the sending (TX) path and the receiving (RX) path. In the TX path (in this paper, mainly referring to *WRITE* and *READ Response* semantics), the process unfolds as follows: **The Encoding Controller** identifies messages and routes them to the **Encoding Buffer** for FEC calculation (①). It also monitors outgoing messages for each Queue Pair (QP), generating control signals to manage the encoding operations in the Encoding Buffer (②). Upon encoding completion, repair packets are generated.

These repair packets, together with their corresponding data packets, are then prepared and sent over underlying links (③).

Conversely, the RX path involves the **Decoding Controller** distinguishing between data and repair packets post-UDP decapsulation. It directs all incoming data packets to the **Decoding Buffer** to perform XOR decoding. Out-of-order data packets are temporarily saved to the **Reordering Buffer** (ROB) (④). Upon receiving repair packets, the Decoding Controller identifies them and recovers data packets with decoding intermediate values in the Decoding Buffer and these repair packets. These recovered packets are then moved to the Reordering Buffer (⑤). Once the **Bitmaps** confirm repair completion, packets in the ROB are seamlessly fed into the upper RDMA stack, maintaining the sequence as if there were no losses occurred (⑥).

Building upon this architecture, LoWAR embeds two core designs: **Message-oriented Real-time Coding**, which enables loss recovery for RDMA messages with minimal storage burden and computational latency (§IV-B), and **Repair Header Extension** (§IV-C), which achieves accurate repair packets identification and establishes a bidirectional negotiation channel for LoWAR's parameters.

B. Message-oriented Real-time Coding

LoWAR introduces a message-oriented real-time mechanism for FEC encoding and decoding. LoWAR regards RDMA messages as the target of protection, instead of the traditional datagrams or byte streams, to avoid the inefficiency of cross-message coding. Based on this insight, LoWAR adopts *Packets Alignment Coding Model* to perform message-oriented FEC coding. Meanwhile, RDMA requires high-speed transmission. Therefore, LoWAR uses *Buffer-based Update-style Calculation* for real-time encoding and decoding of repair packets. This approach avoids the heavy need for caching the entire coding block in loss recovery, enabling scalable connections with limited on-chip memory.

1) Packets Alignment Coding Model: LoWAR adapts coding blocks for RDMA messages. A coding block contains a certain number of packets. For LoWAR(r, c), r denotes the size of the coding block and c stands for the interleaving depth. LoWAR aligns the packets within each message with the predefined coding block size for each connection, ensuring each message begins its own encoding and decoding cycle to prevent cross-message coding. Data packets in the same coding block perform actual calculations of the interleaved XOR-based code at the packet level to generate repair packets.

This coding model results in variable coding patterns for messages with different lengths. As illustrated in Fig. 6, for LoWAR$(8, 2)$ encoding three RDMA messages containing 1, 5, and 61 packets, respectively:

Message ①: If a message has packets equal to or fewer than the interleaving depth, LoWAR first guarantees interleaving and generates repair packets (e.g., repair packet *A*) as duplicates of original packets (i.e., data packet *0*). This duplication is essentially XOR operations with zero.

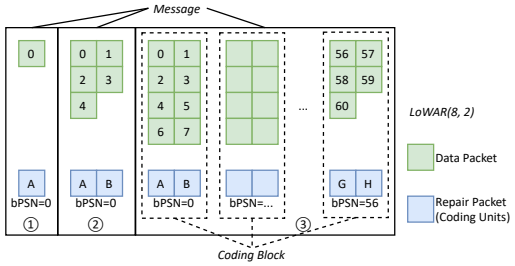


Fig. 6: The packets alignment coding model.

Message ②: For messages with packets equal to or fewer than the size of the coding block (8 here), LoWAR performs XOR operations on these packets with an interval of interleaving depth (2 here), encoding data packets (e.g., data packets *1*, *3*) to corresponding repair packet (i.e., repair packet *B*).

Message ③: When a message exceeds a single coding block, it is divided based on the block size. The encoding of this message is performed successively in these coding blocks. This process makes the coding process of long messages a combination of the first two cases.

LoWAR introduces two extra parameters to link repair packets with coding blocks: *base PSN* (bPSN) and an *Interleaving Sequence Number* (ISN). The bPSN is the PSN of the first packet in a coding block to specify which coding block a repair packet belongs to after QPN already distinguishes different connections. ISN orders repair packets within the same coding block.

2) Buffer-based Update-style Calculation: The XOR-based code effectively reduces on-chip storage by storing only the intermediate values of encoding and decoding calculation. LoWAR pre-allocates **coding units** in Encoding Buffer and Decoding Buffer, with each of which storing one such intermediate value. These coding units are essentially the precursor of repair packets or recovered packets, with their size matching the maximum potential length of RoCE transport packets.

LoWAR dynamically assigns these units to different connections. For instance, in the encoding process shown in Fig. 7, coding units 0∼6 are assigned to connections with QPNs 2∼4, to store intermediate values with different interleaving depth. The introduction of coding units allows shared and reusable storage across multiple connections, minimizing LoWAR's memory demand. It also allows adjustable interleaving depth by "allocating" new coding units or releasing them (our appendix shows how allocation and release work).

Real-time coding. LoWAR enables real-time encoding and decoding. As the encoding process demonstrated in Fig. 7 and Alg. 1, for any outgoing data packet, the Encoding Controller selects its coding unit based on its QPN and PSN (①). Then, the outgoing data packet performs XOR calculation with this coding unit and updates it with the result (②). If a packet ends its corresponding message or coding block, all encoded coding units of this coding block are then encapsulated and sent as repair packets (③). RNIC's high-bandwidth, low-latency SRAM and hardware's pipelined design ensure such calcu-

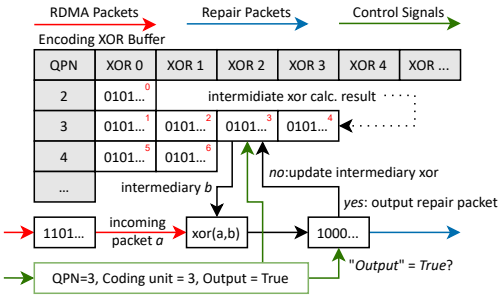


Fig. 7: The Encoding Buffer. Decoding Buffer works likewise.

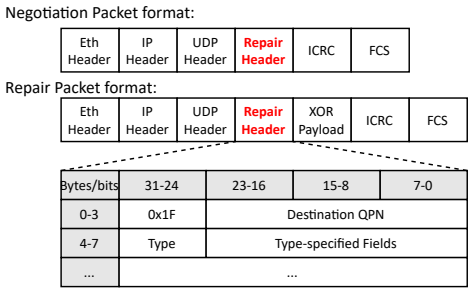


Fig. 8: Packet and Repair Header format of LoWAR.

Algorithm 1 Real-time encoding in LoWAR

Require: data packets of a message $M = \{d_1, d_2, ..., d_n\}$, coding block size r, interleaving depth c, coding units $\{e_1, e_2, ..., e_c\}$
Ensure: outgoing packets sequence P

```
for each packet d_i ∈ M do
    Select coding unit index j = i mod c {①}
    Perform bit-wise XOR operation: e_j ← e_j ⊕ d_i {②}
    Send data packet: P ← d_i
    if i mod c = 0 or i = n then
        Encapsulate and send encoded coding units: P ← {e_1, ..., e_k}, k ≤ c {③}
        Reset coding units {e_1, ..., e_k} ← 0
    end if
end for
```

lation works in parallel with normal traffic, without hindering their efficiency.

Packet decoding. Although our discussion has primarily focused on encoding, decoding in LoWAR is virtually its mirror. Every incoming data packet will be XORed with its decoding unit in the Decoding Buffer. By doing constant decoding calculations upon receiving data packets, LoWAR obviates the need for caching entire coding blocks for loss recovery. When a repair packet is received, and only one data packets that encode the repair packet is lost, XORing the repair packet with its corresponding decoding units will recover the lost packet. Another notable fact is that the absence of the first packet in a coding block won't obstruct the entire decoding, as the decoding units used for recovering lost packets are specified by ISN and bPSN recorded in repair packets, instead of the order of these units.

OOO packets tracks. LoWAR uses a per-connection bitmap and a shared ring reordering buffer (ROB) with self-clocking for tracking out-of-order (OOO) packets, which are smaller than what selective retransmission requires since it only maps packets within a coding block, rather than the entire BDP. If a coding block fails to repair or triggers a timeout in ROB, LoWAR will pass it to RDMA stacks to activate GBN retransmission and stop tracking the connection until its retransmission arrives.

C. Repair Header Extension

LoWAR introduces a Repair Header extension to achieve bidirectional negotiation for FEC parameters without altering upper-layer applications. These negotiations fall into three aspects: *(i)* When packet loss worsens or alleviates, the receiver prompts the sender (requester in WRITE semantic and responder in READ semantic) for more proper redundancy rates or interleaving depths. *(ii)* The sender synchronizes the redundancy rate and interleaving depth to the receiver to adjust coding units for decoding. *(iii)* When coding blocks contain the first, last, or only packet of a message, the sender should send extra metadata to keep the decoding accurate. In LoWAR, the repair header is embedded in repair packets and can be sent alone as a negotiation packet. We keep outgoing RDMA data packets unchanged, to keep LoWAR's transparency.

Repair Header format. Fig. 8 illustrates the formats of LoWAR packets, including repair packet and negotiation packet, and Repair Header. Repair Header replaces the field where OpCode lies within RDMA BTH header with a unique value (0x1F), enabling receivers to differentiate between data packets and LoWAR packets easily by checking the same field. It also includes a target QPN field to map LoWAR packets to their connections. Depending on the Type field shown in the figure, the Repair Header carries different information. In the UDP encapsulation after the Repair Header encapsulation, LoWAR packets multiplex the UDP port of their connections.

Repair packets. As shown in Fig. 8, the repair packet substitutes the RDMA transport header and payload with the Repair Header and redundant payload (i.e., encoding results in coding units). In a repair packet, the Repair Header contains decoding metadata. These metadata, in addition to coding block's (r, c) tuple, includes: ***(i)*** *bPSN* and *ISN* for associating repair packets to the correct coding block as discussed in §IV-B1. ***(ii)*** parameters for edge conditions, including *has_First*, *has_Last* and *Last_Length*. These edge case parameters ensure precise trimming of recovered packets, especially when these packets lie in the first or last of a message, incorporating extra RDMA transport, like RETH and AETH, or have a unique length.

Negotiation packets. Negotiation packets that include only the Repair Header serve dual purposes. On the one hand, the sender notifies the receiver of the (r, c) tuple and the packet number in the upcoming coding block to proactively adjust

coding units for decoding in the receiver; on the other hand, packets from the receiver to the sender facilitate adaptive (r, c) adjustment for network fluctuation or improvement. LoWAR itself does not mandate specific adaptive algorithms, although we have included the adaption and exploration of effective adaptive algorithms in our timetable. In LoWAR's architecture, negotiation packets are not deemed essential for decoding: decoding will proceed even if no leading negotiation packet is received, but only if the (r, c) parameters used for decoding calculation align with that included in repair packets will the recovered data packets be accepted.

D. Design Summary

Optimized for RDMA over lossy WANs, LoWAR seamlessly integrates FEC into RDMA, addressing the challenges mentioned in §I. LoWAR's overall architecture maintains transparency to both networks and applications, thus addressing ***challenge (i)***. LoWAR comprehensively takes account of the variability and independence between different RDMA messages, addressing ***challenge (ii)***. Buffer-based update-style coding calculation minimizes FEC's storage burden and computational latency, thus addressing ***challenge (iii)***. Meanwhile, the Repair Header enables bidirectional parameter negotiation and allows for the adjustment of encoding parameters, therefore addressing ***challenge (iv)***.

V. Implementation

We build a LoWAR prototype using Xilinx Alveo U200 FPGA board.

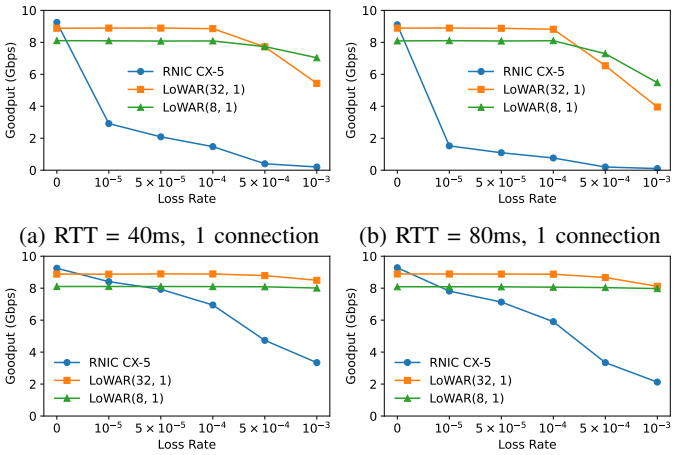
RDMA protocol stacks. LoWAR functions as a shim layer within RDMA stacks, so we integrate LoWAR into existing open-source RoCEv2 implementation [36] with Vitis HLS [37], ensuring it operates effectively between the RDMA transport layer and the UDP/IP stack. We also simplify the RoCEv2 stack to some extent to ensure timing closure.

RNIC parameters. Some parameters may affect the performance of RDMA itself in WANs, including TX queue length and RoCE MTU size. We set the hardware queue size to 512 to ensure in-flight messages to saturate the link. Although LoWAR supports jumbo frames, we've limited the RoCE MTU to 1024 bytes to ensure compatibility with typical Ethernet environments where the MTU is 1500 bytes.

Resource usage. The additional resource usage in the FPGA and the memory breakdown are measured in Table I. The *Encoding Buffer* and *Decoding Buffer* consume each 1.1MB on-chip SRAM, which is the 1024 coding units we set. The shared *Reordering Buffer* is configured to accommodate up to 2048 out-of-order packets, consuming 2.4MB on-chip SRAM. The State Table includes extra states for tracking messages and the bitmap, consuming 0.3MB SRAM in our setting of 512 max QPs. Although LoWAR seemingly consumes huge SRAM for intermediate encoding and decoding values storage, its hardware logic complexity for FEC controlling remains minimal, so we believe that with expanding on-chip storage, LoWAR exhibits promising scalability for connections.

TABLE I: Extra resource usage of LoWAR prototype.

Extra FPGA Resource Usage			
LUT	Register	BRAM	URAM
21995	7302	762	120
Extra Memory Breakdown (MB)			
State Table	Encoding Buffer	Decoding Buffer	Reordering Buffer
0.3	1.1	1.1	2.4



(a) RTT = 40ms, 1 connection (b) RTT = 80ms, 1 connection

(c) RTT = 40ms, 16 connections (d) RTT = 80ms, 16 connections

Fig. 9: Goodput.

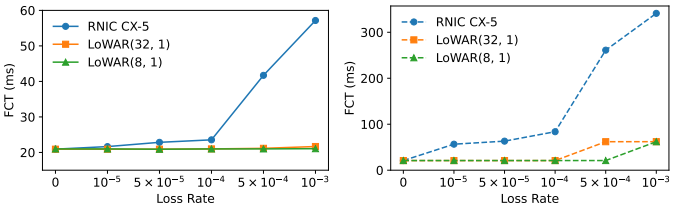
VI. Evaluation

We evaluate LoWAR with a dumbbell testbed consisting of two host servers and a Spirent SNE-X [38] network emulator. Each server is equipped with a Mellanox ConnectX-5 RNIC and a Xilinx Alveo U200 FPGA card, both connected to the network emulator.

We replicate lossy WAN conditions through bidirectional lossy long-haul links with various RTTs and loss rates created using the network emulator. Restricted by the limited performance of open-source RoCE implementation at a low RoCE MTU setting (1024 bytes), we capped the bandwidth of the emulated long-haul links at 10Gbps. This bandwidth cap ensured that open-source RoCE could fully utilize the link under lossless conditions. Observationally, the standalone LoWAR shim layer does not limit the achievable bandwidth. For testing, we employ OFED *perftest* [39] for CX-5 and application based on Xilinx XDMA driver for LoWAR.

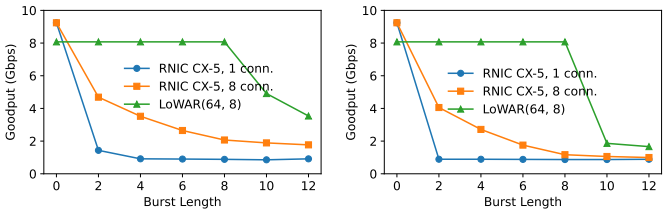
A. Performance Improvements

Goodput. We compared the goodput of LoWAR and CX-5 on lossy long-haul links with RTTs of 40ms and 80ms, respectively, and loss rate spanning 0.001% to 0.1% to represent typical WANs conditions. For commercial RDMA, applying parallel connections on the same link can reduce the bandwidth wastage due to GBN, thereby improving the overall goodput, so we measure both LoWAR and CX-5 under a single connection and 16 parallel connections to fully illustrate



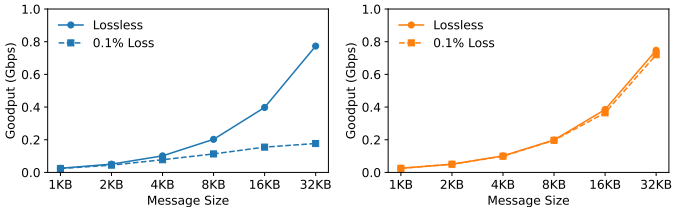
(a) Average, RTT = 40ms (b) 99*th* tail, RTT = 40ms

Fig. 10: Flow completion times.



(a) $p_{\text{high}} = 0.3$ (b) $p_{\text{high}} = 0.7$

Fig. 11: Burst tolerance.



(a) RNIC CX-5 (b) LoWAR$(32, 1)$

Fig. 12: Small message support.

LoWAR's high goodput. For LoWAR, we fixed the coding block size and interleaving depth (denoted as (r, c)) to assess its maximum goodput potential under varying conditions.

Fig. 9 delineates the results. For single connection conditions, the goodput of CX-5 drops rapidly when losses occur and RTTs increase, whereas LoWAR maintains its high goodput until the loss rate exceeds 0.01%. In WANs with 40ms RTT and 0.001% to 0.01% loss rates, compared to CX-5, LoWAR$(32, 1)$ increases RDMA goodput by 2.05 to 5.01 times. Meanwhile, LoWAR's enhancement in RDMA goodput becomes more noticeable as RTT lengthens, with LoWAR$(32, 1)$ exhibiting 11.55 to 19.07 times goodput as much as that of CX-5 at 80ms RTTs.

In scenarios with 16 parallel, CX-5 shows a higher loss tolerance than those with a single connection. However, LoWAR still surpasses it, as LoWAR also benefits from the diluted retransmission wastage. At a 0.01% loss rate, LoWAR$(32, 1)$ achieves goodput 1.28 and 1.50 times higher than CX-5, and remarkably, maintains 87.29% of its goodput even as packet loss rate ascends to 0.1%. Comparing different parameters, LoWAR$(8, 1)$ exhibits greater loss tolerance than LoWAR$(32, 1)$ due to high redundancy, though at the cost of increased redundant bandwidth usage, indicating the need and importance for adjustable parameters.

Flow completion times. We further compare LoWAR with CX-5 regarding flow completion times (FCTs). For our evaluation, we opt for messages of 1MB size, as smaller messages lead to lower bandwidth utilization in high-latency links due to issues related to queue, while larger message triggers retransmission too frequently and lead to distorted high FCTs. Given that FCT is theoretically proportional to the RTT of the link at stable packet loss rates, we limit our experiments to a 40ms RTT scenario. As shown in Fig. 10a, CX-5's average FCT significantly increases as the loss rate increases, while LoWAR's average FCT stays relatively stable. At a 0.01% packet loss rate, the average FCT of CX-5 is 12% higher than that of LoWAR, and this difference rapidly increases to 97% at a 0.05% loss rate. Fig. 10b reveals that CX-5 experiences significant 99th tail FCT whenever loss exists, while LoWAR eliminates the prolonged FCTs in most scenarios, and shows a long tail only when the loss rate exceeds 0.1%.

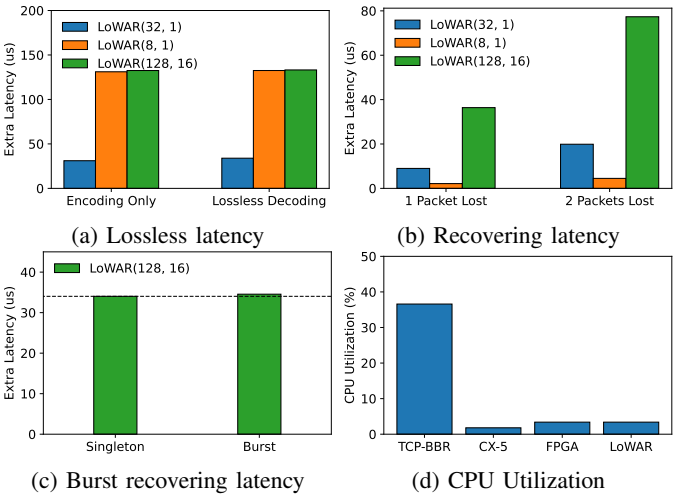
Burst tolerance.We assessed LoWAR's resilience to bursty losses under their two-state Markov model [40]. We do two experiments with the high loss rate $p_{high} = 0.3$ and $p_{high} = 0.7$ respectively. The low loss rate is set to 0, and the transition probability is 0.01% from the low loss state to the high loss state. We use LoWAR$(64, 8)$ in our experiments, which produces 8 repair packets in one coding block. By controlling the burst length from 2 to 12, we compare LoWAR with CX-5. The results, depicted in Fig. 11, show a decline in CX-5's goodput as burst length increases. Conversely, LoWAR$(64, 8)$ demonstrated stable goodput of approximately 8Gbps until burst lengths surpassed 8; when the burst length exceeds 8, LoWAR maintains the resistance to bursty losses, as the bursty losses may fall into different coding blocks, thereby not effecting the repair.

Small message support. Although small messages are less efficient for transmission on high-latency links, they are an indispensable part of RDMA traffic. LoWAR provides protection for messages with a size smaller than the coding block, thereby providing uniform protection for RDMA messages. We set both CX-5 and LoWAR's TX queue to 128 to align their baseline throughput and then test their throughput for small messages on the link with 40ms RTT and 0.1% loss rate. Fig. 12 reveals that LoWAR achieves goodput nearly equivalent to lossless conditions for these small messages in lossy WANs, in contrast to the reductions seen with CX-5. LoWAR doesn't show a good decline due to redundant bandwidth, as the total throughput has not reached a bottleneck.

B. Overhead Evaluation

We evaluated the overhead introduced by LoWAR, as shown in Fig. 13. The extra latency can be categorized into two components: lossless latency, arising from the encoding, decoding, and transmission of repair packets in the absence of packet loss, and recovering latency, which is incurred only upon the detection of packet loss. We test these two types of latency in a 10Gbps LAN with 1MB messages as the test object:

Lossless latency. Fig. 13a illustrates the lossless latency incurred by LoWAR. When LoWAR is limited to the encoding



(a) Lossless latency (b) Recovering latency

(c) Burst recovering latency (d) CPU Utilization

Fig. 13: Overhead of LoWAR.

and transmission of repair packets, without engaging in decoding, the additional latency at the receiver's end aligns closely with the transmission latency of these packets, exhibiting a direct correlation with the redundancy rate. The introduction of real-time decoding at the receiver does not significantly increase this latency, indicating that computational efforts minimally contribute to the overall lossless latency.

Recovering latency. Fig. 13b shows the recovering latency, using lossless decoding as a reference point. The recovering latency is roughly proportional to the size of the coding block, i.e., the r parameter. We believe the primary source of recovering latency stems from the reordering and draining of out-of-order packets, other than the computation for recovery. On the one hand, the recovering latency nearly doubles when two losses span across two coding blocks, as compared in the figure; on the other hand, Fig. 13c compares the average recovering latency of LoWAR$(128, 16)$ for a single loss against 16 bursty losses, both in one coding block. Despite involving different packet recovery computations, they demonstrate similar recovering latency. Although recovery latency increases with coding block size, it shows only when loss occurs, thus having a marginal overall impact.

CPU utilization. Fig. 13d shows the CPU utilization of LoWAR. TCP-BBR, while efficient in WANs experiencing non-congestion packet loss, incurs high CPU utilization, in contrast to RDMA solutions. Compared to CX-5, the open-source FPGA RoCE has higher CPU utilization, likely due to its less engineering optimization. LoWAR, not requiring software modifications, duplicates the CPU utilization of open-source RoCE.

VII. Discussion

LoWAR shows significant performance improvements, yet its journey towards optimization is far from complete.

Reordering optimization. LoWAR employs a shared ring ROB for out-of-order packets. Although efficient in existing settings, this approach leads to storage demands and recovering latency. To eliminate the need for ROB, recent SRNIC [24] has advocated in-place reordering by directing out-of-order packets into user memory. This design inspires us to envision more optimized reordering in our future work to better fit RNICs' precious memory and high speed.

Adaptive FEC. Adaptive parameters are essential for optimizing FEC systems in fluctuating networks. LoWAR allows adjustable coding block size and interleaving depth and provides a negotiation channel through repair header extension, thus laying the groundwork for adaptive FEC. Yet, creating LoWAR-optimized adaptive algorithms requires careful consideration and remains our future step.

VIII. Related Works

Long-haul RDMA. Recent years have seen an upsurge in interest towards long-haul RDMA. Mellanox provides long-haul RDMA [41] based on InfiniBand for cross-regional data centers. Swing [42] extends Priority Flow Control to Data Center Interconnect (DCI) by adding extra PFC relay devices, achieving long-haul lossless RoCE. Bifrost [43] eliminates PFC and performs downstream-driven lossless flow control for long-haul RDMA. BiCC [44] alleviates hybrid congestion through bilateral DCI switch modification in long-haul RoCE deployment. These works realize optimized RDMA performance across broader network scales by extending lossless InfiniBand or Ethernet. Unlike these developments, LoWAR focuses on innovations within RDMA protocol and explores the efficient combination of FEC and RDMA to achieve lossy WAN compatibility.

iWARP. iWARP [45] facilitates RDMA via TCP offload engine. It leverages TCP's reliability for general link applicability. However, it introduces greater hardware complexity and costs without clear benefits over RoCE [22]. Its limited deployment [46] also constraints its further adoption in WANs.

IX. Conclusion

This study investigates RDMA over lossy WANs and presents LoWAR as a high-goodput, high-reliability solution. LoWAR incorporates a fully hardware-offloaded FEC shim layer into RNICs, which generates repair packets from RDMA messages. By recovering lost packets with repair packets, LoWAR protects RDMA against packet loss and significantly mitigates the retransmission inefficiency with minimal storage overhead and computational latency. It works transparently and requires no modifications on both applications and networks. We implement LoWAR with FPGA and evaluate it through testbed experiments. The results demonstrate that LoWAR significantly outperforms commercial RoCE in goodput and FCTs within lossy WANs. We believe LoWAR will help bridge the gap in RDMA deployments in lossy WANs.

Acknowledgment

This work was supported in whole or in part, by National Science Foundation of China (U2333201, 62372429), and the Institute of Computing Technology, Chinese Academy of Sciences-China Mobile Communications Group Co.,Ltd. Joint Institute. Corresponding Author: Yujun Zhang.

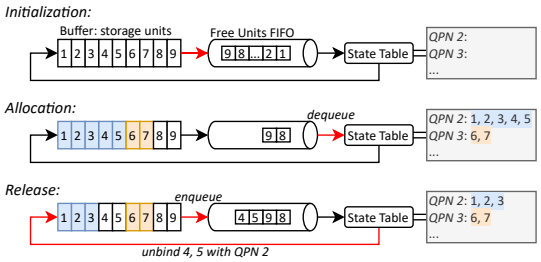


Fig. 14: Dynamic storage units allocation.

Appendix
Coding Units Allocation

LoWAR allocates coding units to connections based on a FIFO queue, as illustrated in Fig. 14. In the initialization stage, all coding units are valid, and their "addresses" are enqueued into *free units queue*. In the allocation stage, free units are dequeued and binded to connections. In the release stage, they are re-enqueued into the queue with binds dissolved.

References

[1] T. Hoefler, D. Roweth, K. Underwood, R. Alverson, M. Griswold, V. Tabatabaee *et al.*, "Data center ethernet and remote direct memory access: Issues at hyperscale," *Computer*, 2023.

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

[3] Y. Gao, Q. Li, L. Tang, Y. Xi, P. Zhang, W. Peng *et al.*, "When cloud storage meets rdma," in *NSDI*, 2021.

[4] C. Guo, H. Wu, Z. Deng, G. Soni, J. Ye, J. Padhye *et al.*, "Rdma over commodity ethernet at scale," in *SIGCOMM*, 2016.

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

[6] K. Hsieh, A. Harlap, N. Vijaykumar, D. Konomis, G. R. Ganger, P. B. Gibbons *et al.*, "Gaia: Geo-distributed machine learning approaching lan speeds," in *NSDI*, 2017.

[7] X. Jin, Y. Li, D. Wei, S. Li, J. Gao, L. Xu *et al.*, "Optimizing bulk transfers with software-defined optical wan," in *SIGCOMM*, 2016.

[8] Q. Pu, G. Ananthanarayanan, P. Bodik, S. Kandula, A. Akella, P. Bahl *et al.*, "Low latency geo-distributed data analytics," *ACM SIGCOMM Computer Communication Review*, 2015.

[9] Ceph, "Multi-site - ceph documentation," December 2023. [Online]. Available: https://docs.ceph.com/en/lastest/radosgw/multisite/

[10] J. Meza, T. Xu, K. Veeraraghavan, and O. Mutlu, "A large scale study of data center network reliability," in *IMC*, 2018.

[11] C. Y. Hong, S. Mandal, M. Al Fares, M. Zhu, R. Alimi, C. Bhagat *et al.*, "B4 and after: managing hierarchy, partitioning, and asymmetry for availability and scale in google's software-defined wan," in *SIGCOMM*, 2018.

[12] Google, "Cross-cloud interconnect overview." [Online]. Available: https://cloud.google.com/network-connectivity/docs/interconnect/concepts/cci-overview

[13] Y. Ren, T. Li, D. Yu, S. Jin, T. Robertazzi, B. L. Tierney *et al.*, "Protocols for wide-area data-intensive applications: Design and performance issues," in *SC*, 2012.

[14] Y. Ren, T. Li, D. Yu, S. Jin, and T. Robertazzi, "Design and performance evaluation of numa-aware rdma-based end-to-end data transfer systems," in *SC*, 2013.

[15] M. F. Aktas, J. Diaz-Montes, I. Rodero, and M. Parashar, "Wa-dataspaces: Exploring the data staging abstractions for wide-area distributed scientific workflows," in *ICPP*, 2017.

[16] ESnet, "Test/measurement host tuning." [Online]. Available: https://fasterdata.es.net/host-tuning/test-measurement-host-tuning/

[17] M. Balakrishnan, T. Marian, K. Birman, H. Weatherspoon, and E. Vollset, "Maelstrom: Transparent error correction for lambda networks." in *NSDI*, 2008.

[18] O. Haq, M. Raja, and F. R. Dogar, "Measuring and improving the reliability of wide-area cloud paths," in *WWW*, 2017.

[19] F. Y. Yan, J. Ma, G. D. Hill, D. Raghavan, R. S. Wahby, P. Levis *et al.*, "Pantheon: the training ground for internet congestion-control research," in *ATC*, 2018.

[20] G. Zeng, W. Bai, G. Chen, K. Chen, D. Han, Y. Zhu *et al.*, "Congestion control for cross-datacenter networks," in *ICNP*, 2019.

[21] NVIDIA, "Nvidia connectx-6 dx network adapters." [Online]. Available: https://www.nvidia.com/en-us/networking/ethernet/connectx-6-dx/

[22] R. Mittal, A. Shpiner, A. Panda, E. Zahavi, A. Krishnamurthy, S. Ratnasamy *et al.*, "Revisiting network support for rdma," in *SIGCOMM*, 2018.

[23] Y. Le, B. Stephens, A. Singhvi, A. Akella, and M. M. Swift, "Rogue: Rdma over generic unconverged ethernet," in *SoCC*, 2018.

[24] Z. Wang, L. Luo, Q. Ning, C. Zeng, W. Li, X. Wan *et al.*, "Srnic: A scalable architecture for rdma nics," in *NSDI*, 2023.

[25] Q. Li, Y. Gao, X. Wang, H. Qiu, Y. Le, D. Liu *et al.*, "Flor: An open high performance rdma framework over heterogeneous rnics," in *OSDI*, 2023.

[26] E. Kissel and M. Swany, "Evaluating high performance data transfer with rdma-based protocols in wide-area networks," in *HPCC*, 2012.

[27] E. Kissel, M. Swany, B. Tierney, and E. Pouyoul, "Efficient wide area data transfer protocols for 100 gbps networks and beyond," in *Proceedings of the Third International Workshop on Network-Aware Data Management*, 2013.

[28] ESnet, "Experimental network testbeds and test circuit services." [Online]. Available: https://www.es.net/network-r-and-d/experimental-network-testbeds/

[29] W. Yu, N. S. Rao, and J. S. Vetter, "Experimental analysis of infiniband transport services on wan," in *International Conference on Networking, Architecture, and Storage*, 2008.

[30] P. Lai, H. Subramoni, S. Narravula, A. Mamidala, and D. K. Panda, "Designing efficient ftp mechanisms for high performance data-transfer over infiniband," in *ICPP*, 2009.

[31] Y. Kim, S. Atchley, G. R. Vallée, and G. M. Shipman, "Lads: Optimizing data transfers using layout-aware data scheduling," in *FAST*, 2015.

[32] Z. Dong, F. N. Khan, Q. Sui, K. Zhong, C. Lu, and A. P. T. Lau, "Optical performance monitoring: A review of current and future technologies," *Journal of Lightwave Technology*, 2015.

[33] A. C. Begen, "RTP Payload Format for 1-D Interleaved Parity Forward Error Correction (FEC)," RFC 6015, Oct. 2010. [Online]. Available: https://www.rfc-editor.org/info/rfc6015

[34] P. Garrido, I. Sanchez, S. Ferlin, R. Aguero, and O. Alay, "rquic: Integrating fec with quic for robust wireless communications," in *GLOBECOM*, 2019.

[35] X. Wang, C.-T. Nguyen, B. Ye, Z. Qian, B. Tang, W. Li *et al.*, "Nem: Toward fine-grained load balancing through rnic ec offloading," in *HPSR*, 2018.

[36] ETH Zürich System Group, "Scalable network stack for fpgas." [Online]. Available: https://github.com/fpgasystems/fpga-network-stack

[37] Xilinx, "Vitis hls." [Online]. Available: https://www.xilinx.com/products/design-tools/vitis/vitis-hls.html

[38] Spirent, "Network emulator-x datasheet - spirent." [Online]. Available: https://www.spirent.com/assets/u/datasheet-network-emulator-x-sne-x

[39] OFED, "Infiniband verbs performance tests." [Online]. Available: https://github.com/linux-rdma/perftest

[40] M. Zorzi and R. R. Rao, "Lateness probability of a retransmission scheme for error control on a two-state markov channel," *IEEE ToC*, 1999.

[41] NVIDIA, "Infiniband long-haul systems." [Online]. Available: https://www.nvidia.com/en-us/networking/infiniband-long-haul-systems/

[42] Y. Chen, C. Tian, J. Dong, S. Feng, X. Zhang, C. Liu *et al.*, "Swing: Providing long-range lossless rdma via pfc-relay," *IEEE TPDS*, 2022.

[43] P. Yu, F. Xue, C. Tian, X. Wang, Y. Chen, T. Wu *et al.*, "Bifrost: Extending roce for long distance inter-dc links," in *ICNP*, 2023.

[44] Z. Wan, J. Zhang, M. Yu, J. Liu, J. Yao, X. Zhao *et al.*, "Bicc: Bilateral congestion control in cross-datacenter rdma networks," in *INFOCOM*, 2024.

[45] R. J. Recio, P. R. Culley, D. Garcia, B. Metzler, and J. Hilland, "A Remote Direct Memory Access Protocol Specification," RFC 5040, Oct. 2007. [Online]. Available: https://www.rfc-editor.org/info/rfc5040

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