Toward Deterministic Path Placement in AI Backends: A Practical SRv6-Based Architecture

Clarence Filsfils*, Pablo Camarillo*, Ahmed Abdelsalam*, Arianna Quinci†‡, Angelo Tulumello‡,
Andrea Mayer†§, Pierpaolo Loreti†‡, Lorenzo Bracciale†‡, Stefano Salsano†‡,
*Cisco Systems, USA; †University of Rome Tor Vergata, Italy; ‡CNIT, Italy; §COMMON NET, Italy

Submitted version

***Abstract*—Distributed training of artificial intelligence models, such as Large Language Models (LLMs), generates highly structured and intense traffic patterns between GPUs, with synchronous and repetitive flows that can easily cause congestion and bottlenecks in data center networks. In this context, currently adopted protocols, such as RoCEv2, show significant limitations in the presence of bursty traffic and low entropy, compromising overall system efficiency.**

Segment Routing over IPv6 (SRv6) offers a programmable mechanism to steer AI workload traffic along explicitly chosen paths, enabling precise and congestion-aware routing under dynamic conditions. Lightweight monitoring modules can detect congestion conditions in real time and report them to the orchestrator or NICs, enabling dynamic rerouting decisions without requiring control-plane signaling or state in the fabric. SRv6 micro-segment (uSID) encoding allows the NIC to steer traffic along alternate, congestion-free paths simply by updating the IPv6 destination address, preserving RoCEv2 semantics while ensuring rapid adaptability. This work provides a practical implementation and experimental validation of the recent IETF Internet-Draft "SRv6 for Deterministic Path Placement in AI Backends", demonstrating its feasibility and performance benefits in RoCEv2-based infrastructures. The results highlight the potential of SRv6 as a practical and vendor-agnostic solution to enhance networking efficiency in modern AI datacenters.

***Index Terms*—SRv6, AI workloads, RoCEv2, congestion control, programmable networks, datacenter fabrics, traffic engineering.**

I. Introduction

Distributed training of large-scale artificial intelligence models, such as Large Language Models (LLMs), requires a network capable of handling extremely heavy traffic between GPUs. During model synchronization phases, as in AllReduce-like collective operations, massive, synchronized data flows, known as elephant flows, are generated. These predictable but high-intensity flows can easily saturate links and compromise overall system performance, causing high latencies and slowing down training convergence time [1]. Current technologies, such as RoCEv2 [2], are designed to offer low latency and high throughput, but they rely on lossless networks achieved through mechanisms such as Priority Flow Control (PFC) [3]. Such mechanisms, while effective in preventing packet loss, can generate harmful side effects such as head-of-line blocking and congestion propagation [4]. In addition, RoCEv2 generates a limited number of flows (due to mapping to Queue Pairs), which leads to low entropy and prevents efficient traffic distribution via Equal-Cost Multi-Path (ECMP) [5].

A recent Internet-Draft [6] proposed an architecture for SRv6-based deterministic path placement tailored to distributed AI workloads. Our work builds directly on that proposal by implementing its key mechanisms in a testbed with RoCEv2 traffic, validating its feasibility in real environments. We demonstrate that the draft's envisioned architecture can be realized with commodity hardware and without changes to existing RoCEv2 semantics.

The proposed approach is well suited for AI workloads that generate high-volume, repetitive, and synchronized traffic [1], where even brief congestion events can significantly affect performance. The programmability offered by SRv6 allows rapid path adjustments in response to congestion, enhancing network resilience and ensuring predictable behavior [7], [8].

Within this context, we propose a lightweight monitoring and control architecture that combines real-time congestion detection with SRv6-based programmable routing to improve RoCEv2 traffic handling in distributed AI training environments. By leveraging existing infrastructure metrics such as throughput drops, ECN markings, and latency variations, the system enables rapid and informed rerouting decisions without requiring fabric-wide signaling or control-plane interactions. Preliminary experiments under RoCEv2 traffic demonstrate that dynamic SRv6-based rerouting can restore throughput stability and improve fabric utilization during congestion, providing a practical and incrementally deployable solution for modern AI datacenters [4], [9].

II. AI Traffic Characteristics and Challenges

A. *Traffic characteristics in AI workloads*

Training modern AI models across multiple GPUs involves frequent and large-scale data exchanges that can stress datacenter networks. In particular, operations like all-reduce generate continuous high-bandwidth transfers, creating persistent, predictable flows—often referred to as elephant flows—that must be delivered reliably to prevent slowdowns in the training process [1]. Another typical behavior is synchronized bursts, which occur during pattern synchronization phases. These generate coordinated and recurrent traffic spikes, exacerbating the risk of localized congestion in the network. In addition,

This work has received funding from the Cisco University Research Program Fund and by the European Union - Next Generation EU under the Italian NRRP, Mission 4, Component 2, Investment 1.3, CUP E83C22004640001, partnership on "RESTART" program.

AI traffic tends to exhibit low entropy for ECMP, since GPU-to-GPU communication occurs through a small number of flows, usually corresponding to the Queue Pairs of RoCEv2 [5]. This limits the effectiveness of load balancing algorithms based on 5-tuple hashes, leading to uneven link utilization and hotspot formation. Finally, resilience is a key requirement. Failures or congestion in the network can result in significant delays in training time and increased operational costs. It is therefore essential that the network infrastructure be able to adapt quickly and autonomously to traffic conditions, avoiding bottlenecks and ensuring continuity of data flows between GPUs.

B. Limits of Current RoCEv2 Transport

In the context of distributed AI workloads, RoCEv2 is the de facto standard for GPU communication due to its RDMA over Ethernet capabilities. However, the protocol is tightly coupled to the assumption of a lossless network, typically enforced through Priority Flow Control (PFC) [3]. While PFC prevents packet loss, it introduces harmful side effects such as head-of-line blocking and congestion spreading [4], which degrade overall fabric performance. To mitigate these effects, RoCEv2 supports congestion control algorithms such as DCQCN, which adjust flow rates using ECN and feedback mechanisms. However, DCQCN tends to be unstable under bursty traffic, requiring careful parameter tuning to avoid oscillations or slow convergence—an issue particularly acute in AI training, where synchronous flows simultaneously compete for shared resources.

To address these limitations, several advanced congestion control schemes have been proposed. Telemetry-driven approaches like HPCC and Swift [10], [11] achieve high link utilization with low latency using real-time feedback. Enhanced DCQCN variants such as SR-DCQCN and ACC [12], [13] offer improved responsiveness through selective retransmissions and ACK-based pacing. Receiver-centric schemes like RCC [14] and PCNP [15] accelerate convergence with minimal network modifications, while RL-CC [16] leverages lightweight reinforcement learning on NICs.

At the routing level, RoCEv2 also suffers from low entropy due to the limited number of Queue Pairs, reducing ECMP effectiveness and leading to flow collisions [17]. Recent approaches such as STrack and Ethereal [18], [19] address this with programmable multipath routing, using congestion signals or application-level guidance to redistribute flows—often requiring additional fabric support or packet spraying.

A notable direction comes from the Ultra Ethernet Consortium (UEC), which is standardizing a suite of Ethernet enhancements specifically for AI workloads [20], including techniques such as packet spraying [21] and packet trimming [22]. More radical approaches such as the Network Datagram Protocol (NDP) [23] integrate packet trimming at the transport layer, offering a complete rethinking of congestion management in loss-sensitive environments.

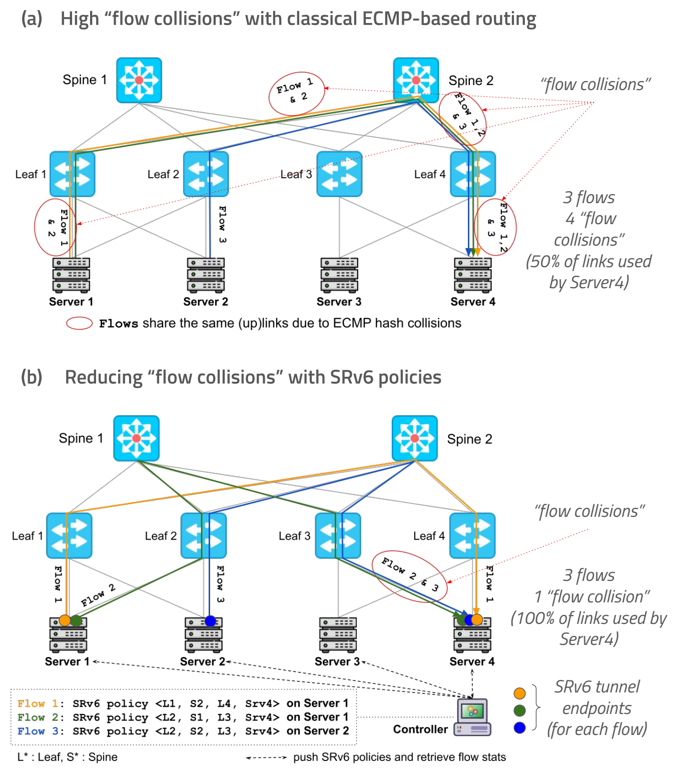
(a) High "flow collisions" with classical ECMP-based routing
Spine 1
Spine 2
Flow 1 & 2
Flow 1,2 & 3
"flow collisions"
Leaf 1
Leaf 2
Leaf 3
Leaf 4
Flow 1 & 2
Flow 3
Flow 1,2 & 3
3 flows
4 "flow collisions"
(50% of links used by Server4)
Server 1
Server 2
Server 3
Server 4
Flows share the same (up)links due to ECMP hash collisions
(b) Reducing "flow collisions" with SRv6 policies
Spine 1
Spine 2
"flow collisions"
Leaf 1
Leaf 2
Leaf 3
Leaf 4
Flow 1
Flow 2
Flow 3
Flow 2 & 3
Flow 1
3 flows
1 "flow collision"
(100% of links used by Server4)
Server 1
Server 2
Server 3
Server 4
Flow 1: SRv6 policy <L1, S2, L4, Srv4> on Server 1
Flow 2: SRv6 policy <L2, S1, L3, Srv4> on Server 1
Flow 3: SRv6 policy <L2, S2, L3, Srv4> on Server 2
Controller
SRv6 tunnel endpoints (for each flow)
L* : Leaf, S* : Spine
push SRv6 policies and retrieve flow stats

Fig. 1. Simplified leaf-spine topology. (a) ECMP causes flow collisions as multiple elephant flows are hashed to the same links. (b) SRv6 assigns distinct paths per flow, reducing contention and improving link utilization.

All these approaches aim to increase path diversity and reduce congestion impact but often introduce complexity at both the data and control planes.

In contrast, our work follows the SRv6-based architecture proposed in [6], which achieves adaptive traffic engineering using a simpler, source-driven approach. By leveraging SRv6 for deterministic path placement together with lightweight congestion monitoring, we enable fast and transparent rerouting of RoCEv2 flows without requiring network-wide signaling or fabric modifications. This lightweight programmability makes SRv6 a compelling alternative for congestion mitigation in AI backends.

III. SRv6 for Deterministic Path Placement

This section describes how SRv6 can be used to achieve fine-grained path control of AI traffic within a data center network, enabling dynamic re-routing in response to congested conditions. The ability to encode source-side explicit paths enables deterministic routing of flows between GPUs, without the need for state in intermediate devices or network signaling.

A. SRv6 in the AI Context

SRv6 (Segment Routing over IPv6) allows an explicit path to be defined for each packet directly in the IPv6 header, via a sequence of micro-segments (uSID). In distributed AI training scenarios, characterized by synchronous and repetitive flows, this technique allows each flow to be assigned a custom path, avoiding congested links and better balancing the load on the network. Unlike traditional ECMP-based mechanisms, which rely on the entropy of fields in packets, SRv6 offers total source-side control. The sender machine's NIC (or DPU) directly encodes the desired path in the Destination Address field of the IPv6 packet, using micro-segment coding. No interaction with the network control plane or state update on intermediate routers is required.

When a congestion condition is detected along a path (e.g., through ECN signals or throughput drop), the source can change the flow path on the fly by simply updating the destination address in the packet. The re-routing occurs exclusively on the source side, with latencies on the order of microseconds, and is completely transparent to the network. This capability is critical for AI workloads, which do not tolerate interruptions or slow re-routings.

B. SRv6 architecture and routing mechanism

The SRv6 architecture adopted in AI data centers allows packet routing to be programmed by encoding a sequence of micro-segments (uSIDs) within the Destination Address field of the IPv6 header. Each segment represents a node in the network, such as a leaf or spine switch, and is consumed progressively by devices along the path. This compact encoding allows completely stateless forwarding and does not require the use of additional headers such as the Segment Routing Header (SRH), simplifying packet processing at intermediate nodes. The route is defined by a central component, typically the AI scheduler, which has visibility into both the network topology and active AI jobs. During the provisioning phase, the controller assigns SRv6 blocks to each node in the network. These segments are associated with behaviors such as End with NEXT-CSID, PSP & USD (uN), which tell the router to read the next segment and forward the packet accordingly. Once the network is configured, the AI scheduler can construct arbitrary paths without further interaction with the control-plane or updates in the routers.

The NIC, once it receives or computes the desired route, encapsulates the RoCEv2 packet within an IPv6 packet. The Destination Address field of this packet encodes the entire sequence of uSID segments required to traverse the network.The encoding can include up to six micro-segments directly in the IPv6 address, a sufficient number to represent a complete path within a three-tier Clos topology [24], [25], even in the presence of super-threads. This solution provides scalability and operational simplicity, as routers do not have to maintain state for flows and the path can be updated instantly by changing only the IPv6 address at the source level. In addition, the system is designed to integrate a congestion-based feedback loop: the NIC can detect signals such as ECN markings, latency changes or packet loss, and react in real time by updating the traffic path to avoid congested links.All this is done without introducing new signaling mechanisms or control plane dependencies.

C. Reducing Flow Collisions

One of the most common issues in networks supporting distributed AI training is congestion caused by flow collisions. The so-called elephant flows generated by GPU collective operations exhibit low-entropy and predictable patterns. This reduces the effectiveness of ECMP, which relies on 5-tuple hashing for flow distribution. Similar hash results lead to path collisions, causing link saturation and exacerbated congestion, as shown in Figure 1.a. As depicted, three flows originating from Server 1 and Server 2 are poorly distributed by ECMP, resulting in four flow collisions over six possible uplinks to Server 4. This inefficiency stems directly from the lack of entropy in the flow headers and the randomness of ECMP hashing.

On the other hand, the implementation of SRv6 has the advantage of facilitating the enforcement of optimal flow paths within the network, thereby circumventing links prone to collisions and ensuring the distribution of elephant flows across disparate paths, as illustrated in Figure 1.b that shows how explicit SRv6 policies can effectively overcome this limitation. Each flow is associated with a deterministic SRv6 path, encoded via uSID in the Destination Address field of the IPv6 header. In this scenario, the three flows are routed along fully disjoint paths, reducing the number of collisions to just one. These SRv6 policies are computed by a central controller based on the current network topology and congestion state, and are installed on the source servers.

D. Lightweight Monitoring Modules and Data Handling

Lightweight monitoring modules play a key role in managing flow-level telemetry within the AI training infrastructure. These modules operate at the server or NIC level, collecting essential statistics such as packet and byte counts, throughput variations, and congestion indicators like ECN markings and latency spikes. By focusing on a minimal yet informative set of metrics, the monitoring system enables comprehensive analysis of GPU-to-GPU flows with low overhead.

Collected data can be stored in lightweight, time-series or in-memory data structures, ensuring rapid access to recent telemetry for the orchestrator or NIC control logic. To optimize performance, these monitoring modules can employ batching strategies, aggregating updates over short time windows before making them available to the decision logic. This reduces processing overhead while maintaining near real-time visibility into network conditions.

This monitoring capability enables the orchestrator or NIC to promptly detect congestion events and dynamically update SRv6 micro-segment (uSID) paths, steering traffic along alternate, congestion-free routes without requiring fabric-wide signaling. By integrating these lightweight telemetry modules with SRv6-based programmable path control, the architecture ensures rapid adaptability while maintaining operational simplicity within large-scale distributed AI workloads.

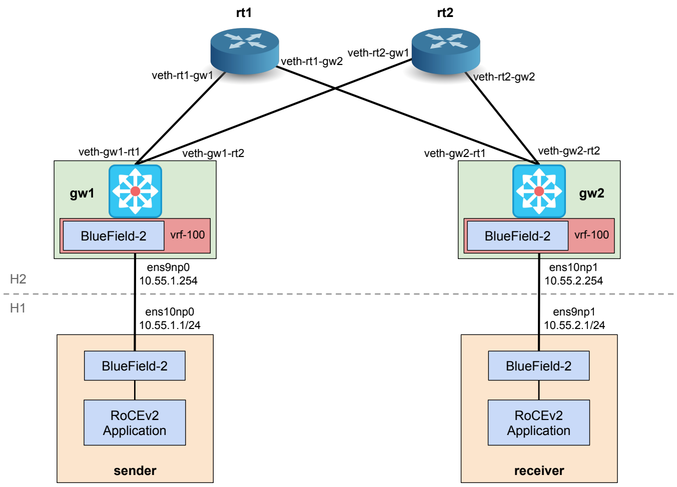

Fig. 2. Experimental testbed comprising RDMA VMs with hardware DCQCN and SRv6-based rerouting.

IV. Validating SRv6-Based Path Steering for RoCEv2 Traffic

To evaluate the practical viability of integrating SRv6 with GPU-to-GPU communication over RoCEv2, we developed a controlled demonstration environment aimed at validating the key components of our approach: deterministic source-side path control and seamless RDMA operation during re-routing. The main goal of the demonstration is to show that SRv6 can be used to steer traffic between GPUs without disrupting RoCEv2 semantics, and that this mechanism can be activated dynamically in response to congestion. The experiment validates that even under path changes, RoCEv2 flows remain stable, congestion is avoided, and performance is restored. We designed a testbed with two servers equipped with RDMA-enabled DPUs, connected via a programmable SRv6 fabric. Each server runs RoCEv2-enabled workloads that simulate collective operations. Congestion is introduced intentionally on one of the paths, and the system responds by switching flows to an alternate SRv6 path configured at the source. We monitor the end-to-end throughput and latency, as well as the impact on flow continuity and RDMA performance.

A. Experimental Setup

The proposed solution has been implemented in a virtual lab environment composed of two bare-metal servers equipped with dual-port 100 Gbps RoCEv2 NICs, as depicted in Fig. 2. One server is configured to run RDMA workloads through VMs acting as senders and receivers, with congestion control handled via hardware-offloaded DCQCN. The second server emulates a small leaf-spine datacenter network that performs SRv6 encapsulation and decapsulation.

Because DCQCN is offloaded to hardware and cannot be modified in software within the VMs, SRv6 encapsulation and decapsulation are handled externally and logically integrated with the sender and receiver workflows. Additionally, the second server hosts lightweight monitoring modules that collect essential congestion indicators, such as throughput drops and ECN markings, enabling rapid congestion detection during experiments.

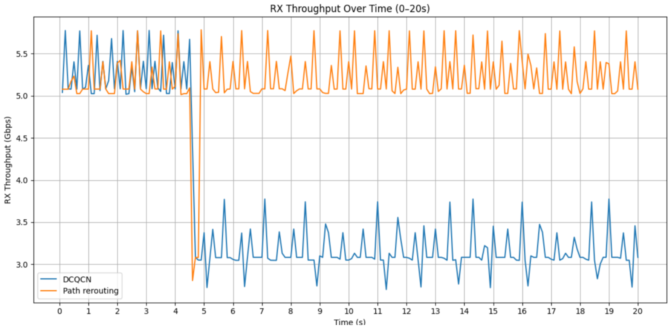

Fig. 3. Throughput over time under congestion. The rerouted flow (orange) recovers rapidly, while the DCQCN-only flow (blue) suffers degradation.

Based on the collected telemetry, a user-space SRv6 controller dynamically updates the micro-segment (uSID) paths to steer RoCEv2 flows along alternate routes in response to congestion. This setup simulates a multi-path datacenter fabric and enables detailed functional evaluation of the SRv6-based rerouting mechanism under controlled conditions, preserving RDMA semantics while demonstrating the effectiveness of source-driven programmable path control.

B. Experimental Results

To evaluate the system, we conducted experiments that simulate congestion by throttling a virtual link between two routers (rt1 and gw2). Initial RDMA communication proceeds along a designated ECMP path under normal conditions, achieving a target data rate. Throughput is continuously monitored at the receiving gateway (GW1), with sampling intervals of 100 milliseconds. Upon introducing congestion, we observe a marked drop in throughput under DCQCN-only conditions, with total data delivered falling to 69.81 Gb over a 30-second period. When telemetry-informed SRv6 rerouting is activated, the system detects congestion and promptly updates the encapsulation headers to steer traffic onto the alternate path. This redirection restores throughput to 98.00 Gb over the same time period, representing a 28.7% improvement. Notably, the path switch occurs within approximately 200 milliseconds of the congestion event, indicating a rapid convergence time that effectively mitigates the performance impact, especially when considered against the much larger timescale of end-to-end training duration. Figure 3 illustrates the RX throughput over time for both the baseline (DCQCN-only) and the rerouting case. After congestion is introduced, the rerouted flow quickly stabilizes, while the DCQCN-only flow continues to experience reduced throughput.

V. Conclusion

This work demonstrates the feasibility of integrating SRv6-based path steering with RoCEv2 to enhance congestion management in distributed AI training environments. By combining source-driven, deterministic routing with lightweight congestion monitoring, it is possible to dynamically reroute RDMA traffic in response to congestion without requiring fabric-wide signaling or disrupting flow stability.

Preliminary experiments in a controlled lab environment show that this approach can restore throughput under congestion, offering improved fabric utilization while preserving RoCEv2 semantics. Although the evaluation is conducted at a small scale, it demonstrates the feasibility of the proposed design. The architecture is conceived to scale to datacenter fabrics supporting multiple concurrent jobs, with full-scale deployments expected to rely on more advanced hardware platforms.

To this end, future work will explore integration with programmable SmartNICs and DPUs, such as FPGA-based SmartNICs, to enable fully in-hardware telemetry collection and path control, further reducing latency and overhead. Additionally, predictive congestion detection using lightweight telemetry could enable proactive rerouting, enhancing network efficiency during large-scale AI training.

Overall, this work validates the architectural model proposed in the recent IETF draft on SRv6 for AI backends [6], validating their applicability in real-world GPU-to-GPU communication scenarios and reinforcing SRv6 as a practical tool for congestion-aware AI traffic steering. The results show that SRv6, combined with lightweight monitoring, offers a practical and incrementally deployable solution for improving network performance in distributed AI workloads without requiring changes to the underlying fabric.

References

[1] L. Liu, P. Zhou, G. Sun, X. Chen, T. Wu, H. Yu, and M. Guizani, "Topologies in distributed machine learning: Comprehensive survey, recommendations and future directions," *Neurocomputing*, vol. 567, p. 127009, 2024.

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

[3] "Ieee standard for local and metropolitan area networks–media access control (mac) bridges and virtual bridged local area networks–amendment 17: Priority-based flow control," *IEEE Std 802.1Qbb-2011 (Amendment to IEEE Std 802.1Q-2011 as amended by IEEE Std 802.1Qbe-2011 and IEEE Std 802.1Qbc-2011)*, pp. 1–40, 2011.

[4] Y. Zhu, H. Eran, D. Firestone, C. Guo, M. Lipshteyn, Y. Liron, J. Padhye, S. Raindel, M. H. Yahia, M. Zhang, "Congestion control for large-scale rdma deployments," in *Proceedings of SIGCOMM'15*. London, United Kingdom: ACM, aug 2015, pp. 523–536.

[5] Cisco Systems, *Cisco IOS XE MPLS Layer 3 VPNs Configuration Guide, Release 17.1 for Cisco ASR 900 Series Routers*, 2020.

[6] C. Filsfils, P. Camarillo, and A. Abdelsalam, "SRv6 for Deterministic Path Placement in AI Backends," Internet-Draft, IETF, April 2025, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-filsfils-spring-srv6-ai-backend/

[7] D. O. Awduche, L. Berger, D.-H. Gan, T. Li, D. V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels," RFC 3209, Dec. 2001. [Online]. Available: https://www.rfc-editor.org/info/rfc3209

[8] P. L. Ventre, S. Salsano, M. Polverini, A. Cianfrani, A. Abdelsalam, C. Filsfils, P. Camarillo, and F. Clad, "Segment routing: A comprehensive survey of research activities, standardization efforts, and implementation results," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 182–221, 2020.

[9] S. Dash, I. R. Lyngaas, J. Yin, X. Wang, R. Egele, J. A. Ellis, M. Maiterth, G. Cong, F. Wang, and P. Balaprakash, "Optimizing distributed training on frontier for large language models," in *ISC High Performance 2024 Research Paper Proceedings (39th International Conference)*. Prometeus GmbH, 2024, pp. 1–11.

[10] Y. Li, R. Miao, H. H. Liu, Y. Zhuang, F. Feng, L. Tang, Z. Cao, M. Zhang, F. Kelly, M. Alizadeh, and M. Yu, "Hpcc: high precision congestion control," in *Proceedings of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM '19. Association for Computing Machinery, 2019, p. 44–58.

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

[12] S. Zhou, Y. Gong, Z. Fan, Y. Chen, W. Zhang, W. Tian, and Y. Liu, "Sr-dcqcn: Combining sack and ecn for rdma congestion control," in *2022 IEEE 8th International Conference on Computer and Communications (ICCC)*, 2022, pp. 788–794.

[13] Y. Zhang, Q. Meng, C. Hu, and F. Ren, "Revisiting congestion control for lossless ethernet," in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. Santa Clara, CA: USENIX Association, Apr. 2024, pp. 131–148. [Online]. Available: https://www.usenix.org/conference/nsdi24/presentation/zhang-yiran

[14] J. Zhang, J. Shi, X. Zhong, Z. Wan, Y. Tian, T. Pan, and T. Huang, "Receiver-driven rdma congestion control by differentiating congestion types in datacenter networks," in *2021 IEEE 29th International Conference on Network Protocols (ICNP)*. IEEE, 2021, pp. 1–12.

[15] D. Yan, Y. Liu, S. Zhang, B. Fang, F. Zhao, and Z. Yang, "Pcnp: A rocev2 congestion control using precise cnp," *Computer Networks*, vol. 247, p. 110453, 2024.

[16] B. Fuhrer, Y. Shpigelman, C. Tessler, S. Mannor, G. Chechik, E. Zahavi, and G. Dalal, "Implementing reinforcement learning datacenter congestion control in nvidia nics," in *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. IEEE, 2023, pp. 331–343.

[17] D. Yan, Y. Liu, S. Zhang, Z. Yang, and Y. Wang, "A survey of rocev2 congestion control," in *International Conference on Information Science, Communication and Computing*. Springer, 2023, pp. 42–56.

[18] Y. Le, R. Pan, P. Newman, J. Blendin, A. Kabbani, V. Jain, R. Sivaramu, and F. Matus, "Strack: A reliable multipath transport for ai/ml clusters," 2024. [Online]. Available: https://arxiv.org/abs/2407.15266

[19] V. Addanki, P. Goyal, I. Marinos, and S. Schmid, "Ethereal: Divide and conquer network load balancing in large-scale distributed training," 2025. [Online]. Available: https://arxiv.org/abs/2407.00550

[20] Ultra Ethernet Consortium, "Ultra ethernet consortium progresses towards v1.0 set of specifications," https://ultraethernet.org/uec-progresses-towards-v1-0-set-of-specifications, 2024, accessed May 2025.

[21] A. Vahdat, M. Alizadeh, A. Greenberg, and D. A. Maltz, "Conga: Distributed congestion-aware load balancing for datacenters," in *Proc. SIGCOMM*, 2014, describes packet spraying techniques for path diversity in datacenters.

[22] P. Goyal, S. Schmid *et al.*, "Pint: Packet trimming for scalable high-performance rdma," in *Proc. ACM SIGCOMM*, 2022, introduces packet trimming mechanisms for adaptive path control in lossless networks.

[23] M. Handley, C. Raiciu, J. Sherry, S. Ratnasamy *et al.*, "Re-architecting datacenter networks and stacks for low latency and high performance," in *Proceedings of the ACM SIGCOMM 2017 Conference*, 2017, pp. 29–42.

[24] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, S. Sengupta, "Vl2: a scalable and flexible data center network," *Proceedings of the ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 51–62, 2009.

[25] R. B. C. Camarero, C. Martínez, "Random folded clos topologies for datacenter networks," in *Proceedings of the IEEE International Symposium on High Performance Computer Architecture (HPCA)*. Austin, TX: IEEE, 2017, pp. 193–204.