Coordinating CCL Reoptimization and WAN Reconfiguration for Cross-Regional AI Acceleration

Jiaheng Xiong[*], Hao Zhou[*], Qiaolun Zhang[*†], Paolo Medagliani[ς], Michele Ferrero[ς], Xiaomin Liu[*], Meng Lian[*]
Nicola Di Cicco[*], Mëmëdhe Ibrahimi[*], Francesco Musumeci[*], Massimo Tornatore[*]
[*]Politecnico di Milano, Italy [ς]Huawei Technologies Ltd., France

***Abstract*—Many emerging Artificial Intelligence (AI) workloads rely on cross-regional distributed training, where Collective Communication Libraries (CCLs), such as `AllGather`, `AllReduce`, and `AlltoAll`, operate across multiple datacenters (DCs) interconnected by wide-area networks (WANs). However, while WAN routing can adapt to traffic dynamics at sub-second to second timescales, CCL logical–topology reoptimization is typically applied far more slowly, often every tens or hundreds of collectives. Due to this intrinsic timescale mismatch, CCL performance (e.g., CCL "completion time") is affected by bandwidth variability in the WAN, motivating us to investigate the individual influence on performance of WAN reconfiguration and CCL reoptimization. In this work, we study the effect of applying fast WAN path reconfiguration and slow CCL logical-topology reconfiguration in cross-regional AI training. Rather than proposing a new coordination policy, we provide an initial evaluation of how these existing mechanisms behave when used individually or jointly under dynamic WAN conditions. By analyzing their combined operation, our study also offers practical insights for rapidly deployable cross-regional training systems, since it builds directly on mechanisms already supported by modern CCLs and WAN controllers. Using a simulated inter-DC environment, we observe that combined WAN reconfiguration and CCL re-optimization can significantly reduce `AllGather` completion time compared to applying only WAN or only CCL adjustments. These results highlight the tight coupling between CCL behavior and WAN dynamics and motivate the development of more principled coordination mechanisms for practical deployment of cross-regional CCLs.**

***Index Terms*—GPU, Collective Communication, Wide-Area Networks**

I. Introduction

The rapid scaling of Large Language Models (LLMs), together with practical constraints on datacenter power provisioning [1], [2], region-specific data-sovereignty requirements, and the increasingly fragmented availability of Graphics Processing Units (GPU) across cloud regions [3], has made it difficult to accommodate AI training workloads within a single GPU cluster or even a single datacenter. As a result, large-scale AI models are increasingly trained and operated over multiple *geo-distributed* or *cross-region* datacenters and rely on wide-area networks (WANs) for inter-region communication [4], [5]. As shown in Fig. 1, each datacenter follows a leaf-and-spine architecture and connects to the WAN via border routers [6]. A centralized DC controller monitors both intra-DC and WAN status, including path latency, background traffic, and potential congestion across the WAN core. such deployments form a unified but highly heterogeneous compute fabric [7]. Although a centralized controller may introduce a single point of failure [8], this issue is beyond the scope of this work. To synchronize model updates, distributed training relies heavily on *collective communication* [9], including operations such as `AllReduce`, `AllGather` or `AlltoAll` [10]. These operations are orchestrated by Collective Communication Libraries (CCLs), which construct logical topologies and determine routing and scheduling based on DC network conditions [11].

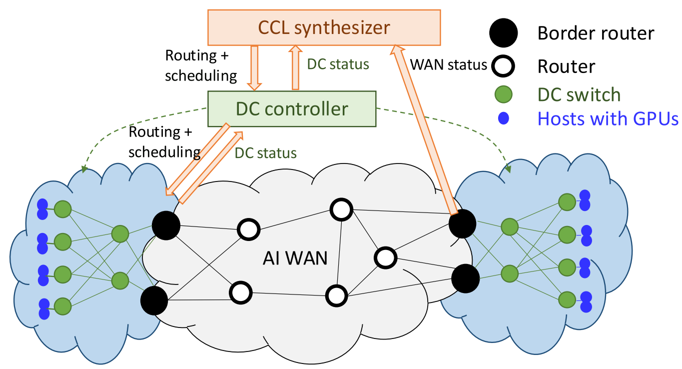


Fig. 1. System model for cross-region collective communication.

While cross-region training enables unprecedented scalability, it also exposes collectives to the variability of WANs. Unlike intra-DC networks, WAN paths frequently experience fluctuating background traffic, routing shifts, and asymmetric capacity or delay [12], [13]. Most existing CCLs, however, implicitly assume uniform and stable high-bandwidth environments [14]–[17], leaving them unaware of WAN dynamics and unable to adapt their logical topologies accordingly. This mismatch creates a new performance bottleneck, as WAN variation might dominate collective completion time.

Several key challenges arise when attempting to make collective communication WAN-aware. **(1) Collective reconfiguration.** Logical CCL topologies (e.g., rings or trees) can be remapped to different physical paths, but the design space for such reconfiguration and its interaction with fluctuating WAN conditions remains poorly understood [18]. **(2) Interaction across timescales.** WAN routing can reconfigure at sub-second to second timescales [19], whereas CCL logical-topology updates typically occur far more slowly, often every tens or hundreds of collectives [20]. Understanding how these two control processes influence each other under dynamic WAN

Corresponding authors: Qiaolun Zhang (qiaolun.zhang@polimi.it).

conditions remains largely unexplored.

In this work, we provide an initial study of *jointly applying* WAN routing and CCL logical-topology reconfigurations during cross-region training. Our focus is not on designing a new collective communication algorithm, but on understanding how existing WAN and CCL-level adaptations behave when enabled together in a dynamic multi-region environment. Using `AllGather` as a representative collective, we evaluate these mechanisms in a simulated inter-DC setting and observe that combined adaptation can significantly reduce completion time relative to applying either mechanism alone. These results highlight the tight coupling between collective communication and WAN dynamics, motivating the need for more principled, WAN-aware approaches to CCL optimization in future cross-region training systems.

II. Related Work

Optimizing collective communication has been extensively studied in tightly-coupled intra-DC GPU clusters, where the network is typically assumed to be high-bandwidth, low-jitter, and largely static. Algorithmic advances such as collective synthesis [11] and topology-aware execution frameworks [14]–[17] can exploit detailed knowledge of intra-DC fabrics; however, these techniques presuppose stable connectivity and do not account for the variability introduced by cross-regional deployments. In practice, even intra-DC links may experience transient contention or background-traffic fluctuations, which become more pronounced once collective traffic extends across WANs with heterogeneous latency and congestion [21]. As a result, their performance degrades when collective traffic traverses WAN paths whose delays, asymmetry, and queueing evolve over time.

In the networking community, WAN traffic engineering and routing adaptation have been studied in great detail. To respond to frequent shifts experienced by WAN paths, due to, e.g., load balancing or rapid fluctuations in background traffic [12], [13], several (e.g., SDN-based) WAN re-routing mechanisms have been investigated, demonstrating that routing reconfiguration can occur at sub-second to second timescales, significantly faster than the timescale at which distributed training systems typically adjust CCL topologies [22], [23]. However, these efforts are generally developed independently from collective communication: CCL research typically assumes a stable network [18], [24], [25], whereas WAN research focuses on routing without considering collective semantics or timescale mismatches.

Recent work on geo-distributed LLM training, such as GeoPipe [26], demonstrates cross-DC execution over a lossless RDMA-enabled optical transport network. GeoPipe assumes a stable and congestion-free WAN substrate, and therefore does not model WAN dynamics or their interaction with collective-communication behavior. Complementary efforts such as NETSTORM [18] incorporate WAN variability into geo-distributed learning, but operate at the DC-overlay level and do not coordinate WAN dynamics with GPU-level CCL logical-topology reconfiguration. These two lines of work, respectively, ignore WAN dynamics or treat CCL structure as fixed, leaving their joint influence on collective communication unaddressed.

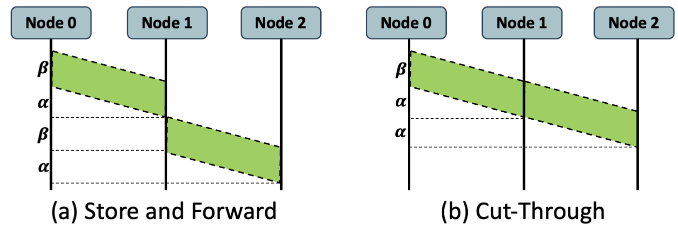


Fig. 2. Illustration of forwarding models

To date, the interaction between CCL logical-topology and dynamic WAN routing reconfigurations remains largely unexplored. To the best of our knowledge, no prior work examines how CCL and WAN reconfigurations behave when activated concurrently in cross-regional training. Our work provides an initial step toward understanding this cross-layer coupling under WAN variability.

III. System Model and Problem Statement

We model the intra-DC network as a directed graph $G(V, E)$, where the node set V contains GPUs and border routers (BRs), and the edge set E represents logical communication links. To represent cross-DC communication, the WAN is abstracted as direct logical connections between BRs, capturing the dominant inter-DC paths shaped by routing policies or traffic-engineering systems, similar to B4 [23] and SWAN [27]. We consider distributed training workloads in which every GPU initially stores a *chunk* of size L, corresponding to its local data portion to be exchanged. During an `AllGather` operation[1], each GPU must obtain the other $n-1$ chunks, possibly traversing multi-hop routes over GPUs and BRs. This makes `AllGather` a representative many-to-many collective used widely in large-scale training [24], [28]. Each link $e \in E$ is associated with a propagation delay α_e and a bandwidth b_e. Transmitting a chunk of size L over link e incurs a transmission delay $\beta_e = L/b_e$.

We model GPU nodes with *store-and-forward* forwarding (as in Ref. [16]), meaning that a node must receive an entire chunk before forwarding. For a path $P = (e_1, \dots, e_h)$, where h is the number of hops, with per-link propagation delays α_{e_i} and bandwidths b_{e_i}, sending a chunk of size L over link e_i takes $\beta_{e_i}(L) = L/b_{e_i}$. The end-to-end completion time under store-and-forward is $T_{\mathsf{SF}}(L; P) = \sum_{i=1}^{h} \big(\alpha_{e_i} + \beta_{e_i}(L)\big)$. For homogeneous links ($\alpha_{e_i} = \alpha$, $b_{e_i} = b$), this simplifies to $T_{\mathsf{SF}} = h(\alpha + \beta)$, where $\beta = L/b$, as shown in Fig. 2(a).

Border routers (BRs) use *cut-through forwarding*: forwarding begins as soon as data arrives [29]. Since WAN packets are typically small (e.g., smaller than 1.5 kB) [23], a chunk effectively travels as a stream of packets pipelined across hops. With sufficient buffering to absorb rate mismatches, the completion time along P is $T_{\mathsf{CT}}(L; P) = \sum_{i=1}^{h} \alpha_{e_i} + \max_{1 \le i \le h} \beta_{e_i}(L)$, that is, propagation delays accumulate while the transmission delay is determined by the bottleneck

[1] Although our evaluation focuses on `AllGather`, the formulation and coordinated framework naturally extend to other collectives.

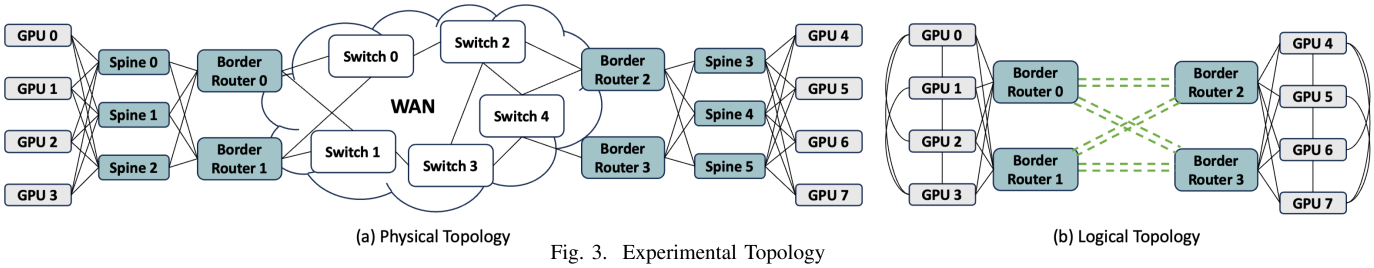


Fig. 3. Experimental Topology

link. In the homogeneous case, this reduces to $T_{\mathrm{CT}} = h\alpha + \beta$, as illustrated in Fig. 2(b).

The WAN-aware CCL optimization problem is defined as follows. **Given** a physical network topology with per-link propagation and transmission delays, link bandwidths, the collective operation (e.g., `AllGather`, `AllReduce`, `AllToAll`, etc.), and the chosen chunk size, **determine** the routing paths and transmission schedule of all chunks. The solution must be designed **subject to** three constraints: link capacity must not be oversubscribed, forwarding causality must be respected (a node can only forward data after receiving it), and delivery completeness must be ensured (all nodes must eventually obtain all chunks). Finally, **the objective is to minimize the completion time**, defined as the time until all GPUs receive all required chunks.

IV. Coordinated Control Framework

Cross-region deployments typically suffer from heterogeneous WAN latency and background traffic, with routing and logical CCL topology updates managed by separate control-plane mechanisms (Fig. 1). The system naturally decomposes into two distinct structures: the intra-DC topology, which governs GPU–to–GPU connectivity and CCL logical mappings, and the WAN topology, which determines inter-DC routing between border routers. Each operates under its own control mechanism and reconfiguration cadence, leading to two adaptation processes that interact tightly during cross-region collective communication. WAN routing can shift paths within seconds in response to congestion or latency changes [19], enabling rapid adaptation to short-lived network conditions. Reconfiguring the logical CCL topology, on the other hand, is substantially more expensive: computing an optimal CCL topology corresponds to a combinatorial graph-embedding problem that is NP-hard [20]. As a result, practical CCL systems update their logical topologies only infrequently, placing CCL adaptation on a much slower rhythm than WAN routing. Given these distinct temporal constraints, *we do not propose a new coordination algorithm; instead, we characterize how fast WAN reactions and slow CCL reconfigurations interact when both are active in a dynamic multi-region environment*. WAN routing determines the instantaneous path characteristics experienced by a fixed CCL topology, while CCL reconfiguration reshapes the distribution of collective traffic across WAN links and thereby influences future routing behavior. Consequently, their combined dynamics emerge from coupling through network state rather than from explicit joint optimization.

TABLE I
Four CCL–WAN reconfiguration modes.

Reconf. cadence	WAN fixed	WAN per-`AllGather`
CCL fixed	Base CCL	Reconfigurable WAN
CCL periodic	Reconfigurable CCL	Reconfigurable CCL+WAN

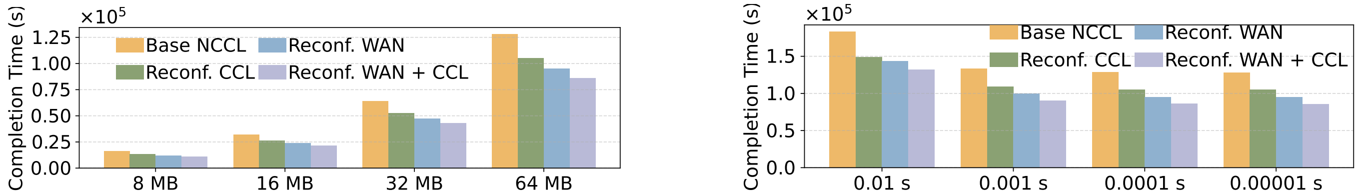
To analyze this interaction in a controlled manner, we compare four execution modes that isolate the contribution of each mechanism, as shown in Table I. In the *Base CCL* mode, both the logical CCL topology and the corresponding WAN routes are fixed at the start of training and remain unchanged throughout execution. In the *Reconfigurable WAN* mode, the CCL topology is held constant, but WAN routes are recomputed for every `AllGather` based on current network conditions. In the *Reconfigurable CCL* mode, the CCL topology is periodically rebuilt and synchronized across GPUs, while WAN routing follows its standard behavior without triggering per-collective reconfiguration; that is, the WAN may evolve naturally with background traffic but does not explicitly adapt to each collective. Finally, the *Reconfigurable CCL+WAN* mode enables both mechanisms: CCL reconfiguration occurs at scheduled intervals, and every `AllGather` triggers a fresh WAN routing decision. Together, these modes allow us to observe how WAN- and CCL-level reconfigurations influence each other and how their combined use affects collective performance under WAN variability.

V. Experiment and Numerical Result

We now evaluate the proposed framework to quantify how WAN routing reconfiguration and CCL reoptimization interact under cross-regional training. Experiments focus on the four configurations introduced in Sec. IV, comparing their performance under varying chunk sizes, WAN propagation delays, and GPU cluster scales.

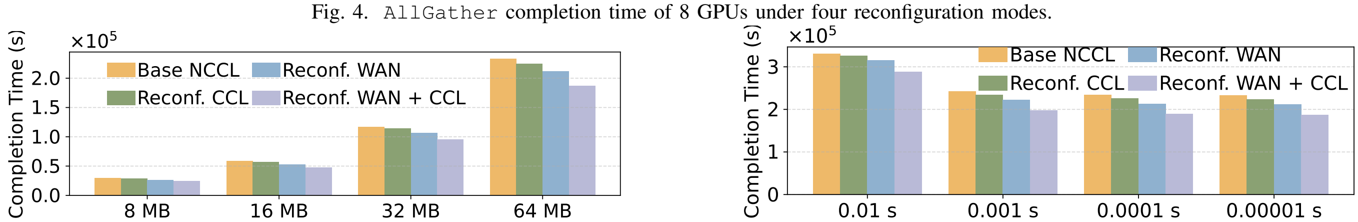
A. Experimental Setup

We consider a simulated cross-regional environment with two DCs, each hosting a configurable number of GPUs connected through local border routers (BRs). The physical topology, organized as a leaf-and-spine fabric, is shown in Fig. 3(a). Both the intra-DC fabric and the WAN carry background traffic that induces time-varying link rates and queueing. To capture this variability, at each evaluation step we randomly sample every link's available bandwidth within its range, 20–200 Gbit/s for intra-DC links and 10–100 Gbit/s for WAN links. We adopt NCCL [30] as the underlying collective communication library to model baseline collective behavior and apply reconfiguration policies on top of it.



(a) **Completion Time vs. Chunk Size** (b) **Completion Time vs. WAN Propagation Delay**

Fig. 4. `AllGather` completion time of 8 GPUs under four reconfiguration modes.



(a) **Completion Time vs. Chunk Size** (b) **Completion Time vs. WAN Propagation Delay**

Fig. 5. `AllGather` completion time of 16 GPUs under four reconfiguration modes.

We compute the total completion time by simulating 100,000 consecutive collective operations (in this work, `AllGather`) and summing their durations. Between consecutive collectives, link bandwidths are resampled to emulate temporal variations in background traffic. This approach captures both steady-state and transient effects of WAN and CCL reconfiguration dynamics, and the large number of iterations is sufficient to smooth out randomness-induced fluctuations.

To analyze scalability, we test multiple GPU configurations ranging from 8 to 16 devices distributed evenly across the two DCs. We examine two regimes: (i) fixed WAN propagation delays $\{\alpha_e\}$ while sweeping the chunk size L; and (ii) fixed L while varying $\{\alpha_e\}$ to emulate geography- or path-induced latency differences. Unless otherwise stated, GPU nodes use store-and-forward forwarding, and BRs use cut-through forwarding. In configurations involving CCL reoptimization, the logical topology is rebuilt every ten collective operations to approximate a realistic adaptation cadence. All four reconfiguration modes from Sec. IV are evaluated.

B. Numerical Result

a) Impact of chunk size: Fig. 4a reports the completion time of `AllGather` across chunk sizes from 8 MB to 64 MB in a two-DC topology, where each DC hosts four GPUs (eight GPUs in total). Completion time increases nearly linearly with L for all configurations, reflecting a bandwidth-dominated regime with negligible fixed overhead. Among the four modes, *Reconf. WAN+CCL* consistently achieves the lowest completion time, while *Base NCCL* remains the slowest. *Reconf. WAN* outperforms *Reconf. CCL* across all sizes, indicating that fast per-collective WAN path adaptation better tracks variations in WAN background traffic than infrequent CCL re-optimizations. Combining the two (*Reconf. WAN+CCL*) further improves performance by roughly 33% over the baseline, confirming that WAN- and CCL-level reconfigurations provide complementary rather than redundant benefits. In the 8-GPU configuration (16 GPUs total) as shown in Fig. 5a, the relative ordering among the four modes remains consistent across chunk sizes, indicating that the performance advantages of WAN and CCL reconfigurations persist under varying communication volumes. The near-linear scaling of completion time with L further confirms that the system operates in a bandwidth-dominated regime rather than being limited by fixed per-collective overheads.

b) Impact of WAN propagation delay: Fig. 4b shows the completion time of `AllGather` under varying WAN propagation delays from 10 ms to 10 μs, again using the 8-GPU configuration (Fig. 5b). Completion time decreases with lower propagation delay across all configurations, highlighting the dominant impact of inter-DC latency on collective duration. The relative ranking remains consistent, with *Reconf. WAN+CCL* achieving the best performance and *Base NCCL* the worst. *Reconf. WAN* maintains a clear advantage over *Reconf. CCL*, while their combination yields an additional 30–35% improvement over the baseline. This consistency across delay scales suggests that the observed relationship between WAN and CCL reconfigurations holds robustly across different inter-DC distances.

VI. Conclusion and Future Work

This work evaluated the combined use of fast WAN path reconfiguration and slower CCL logical–topology reconfiguration for cross-regional training. In a simulated inter-DC environment, the joint mode consistently reduced `AllGather` completion time relative to using only WAN or only CCL adjustments, and the relative gains remained stable across chunk sizes and WAN delay scales. These results indicate a tight coupling between CCL behavior and WAN dynamics and suggest that leveraging already-available WAN controllers and CCL mechanisms is a practical path to near-term deployment. Future work will (i) move from simulation to a prototype that interfaces NCCL with SDN/WAN controllers to validate end-to-end gains; (ii) study policy design for when and how often to trigger each reconfiguration, accounting for overheads and stability; (iii) extend beyond `AllGather` to other collectives and larger multi-DC topologies with richer traffic mixtures and fault events.

References

[1] "Grid engineering and distributed energy resource integration training," https://www.energy.gov/eere/solar/grid-engineering-and-distributed-energy-resource-integration-training, 2022.

[2] T. Mehboob, L. Guo, N. Tallent, M. Zink, and D. Irwin, "Powertrip: Exploiting federated heterogeneous datacenter power for distributed ml training," *arXiv preprint arXiv:2507.17904*, 2025.

[3] F. Strati, P. Elvinger, T. Kerimoglu, and A. Klimovic, "Ml training with cloud gpu shortages: Is cross-region the answer?" in *Proceedings of the 4th Workshop on Machine Learning and Systems*, 2024, pp. 107–116.

[4] M. Leon, "The escalating ai's energy demands and the imperative need for sustainable solutions," *WSEAS Transactions on Systems*, vol. 23, pp. 444–457, 2024.

[5] H. Dong, J. Jiang, R. Lu, J. Luo, J. Song, B. Li, Y. Shen, and Z. Wang, "Beyond a single ai cluster: A survey of decentralized llm training," *arXiv preprint arXiv:2503.11023*, 2025.

[6] V. Sharma and R. Mishra, "A comprehensive survey on data center network architectures," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2020, pp. 222–228.

[7] B. Wu, K. Qian, B. Li, Y. Ma, Q. Zhang, Z. Jiang, J. Zhao, D. Cai, E. Zhai, X. Liu *et al.*, "Xron: A hybrid elastic cloud overlay network for video conferencing at planetary scale," in *Proceedings of the ACM SIGCOMM 2023 Conference*, 2023, pp. 696–709.

[8] A. Hilt, J. Zemán, and G. Járó, "Dimensioning challenges of telecommunication network elements migrating onto the cloud," in *2024 International Conference on Broadband Communications for Next Generation Networks and Multimedia Applications (CoBCom)*. IEEE, 2024, pp. 1–10.

[9] U. Wickramasinghe and A. Lumsdaine, "A survey of methods for collective communication optimization and tuning," *arXiv preprint arXiv:1611.06334*, 2016.

[10] M. A. Soytürk, P. Akhtar, E. Tezcan, and D. Unat, "Monitoring collective communication among gpus," in *European Conference on Parallel Processing*. Springer, 2021, pp. 41–52.

[11] Z. Cai, Z. Liu, S. Maleki, M. Musuvathi, T. Mytkowicz, J. Nelson, and O. Saarikivi, "Synthesizing optimal collective algorithms," in *Proceedings of the 26th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 2021, pp. 62–75.

[12] J. Fei, C.-Y. Ho, A. N. Sahu, M. Canini, and A. Sapio, "Efficient sparse collective communication and its application to accelerate distributed deep learning," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 2021, pp. 676–691.

[13] L. Zhao, S. Pal, T. Chugh, W. Wang, J. Fantl, P. Basu, J. Khoury, and A. Krishnamurthy, "Efficient {Direct-Connect} topologies for collective communications," in *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*, 2025, pp. 705–737.

[14] A. Shah, V. Chidambaram, M. Cowan, S. Maleki, M. Musuvathi, T. Mytkowicz, J. Nelson, O. Saarikivi, and R. Singh, "{TACCL}: Guiding collective algorithm synthesis using communication sketches," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, 2023, pp. 593–612.

[15] H. Kim, J. Ryu, and J. Lee, "Tccl: Discovering better communication paths for pcie gpu clusters," in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, 2024, pp. 999–1015.

[16] X. Liu, B. Arzani, S. K. R. Kakarla, L. Zhao, V. Liu, M. Castro, S. Kandula, and L. Marshall, "Rethinking machine learning collective communication as a multi-commodity flow problem," in *Proceedings of the ACM SIGCOMM 2024 Conference*, 2024, pp. 16–37.

[17] J. Cao, S. Shi, J. Gao, W. Liu, Y. Yang, Y. Xu, Z. Zheng, Y. Guan, K. Qian, Y. Liu *et al.*, "Syccl: Exploiting symmetry for efficient collective communication scheduling," in *Proceedings of the ACM SIGCOMM 2025 Conference*, 2025, pp. 645–662.

[18] Z. Li, W. Feng, W. Cai, H. Yu, L. Luo, G. Sun, H. Du, and D. Niyato, "Accelerating geo-distributed machine learning with network-aware adaptive tree and auxiliary route," *IEEE/ACM Transactions on Networking*, vol. 32, no. 5, pp. 4238–4253, 2024.

[19] G. Zeng, W. Bai, G. Chen, K. Chen, D. Han, Y. Zhu, and L. Cui, "Congestion control for cross-datacenter networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 5, pp. 2074–2089, 2022.

[20] W. Wang, M. Khazraee, Z. Zhong, M. Ghobadi, Z. Jia, D. Mudigere, Y. Zhang, and A. Kewitsch, "{TopoOpt}: Co-optimizing network topology and parallelization strategy for distributed training jobs," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, 2023, pp. 739–767.

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

[22] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven wan," in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, 2013, pp. 15–26.

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

[24] A. Sergeev and M. Del Balso, "Horovod: fast and easy distributed deep learning in tensorflow," *arXiv preprint arXiv:1802.05799*, 2018.

[25] J. Xiong, Q. Zhang, P. Medagliani, M. Ferrero, X. Liu, M. Lian, N. Di Cicco, B. Zhao, M. Ibrahimi, F. Musumeci *et al.*, "Scale-ccl: A scalable collective communication library for wide-area distributed training," in *Proceedings of the 1st Workshop on Inter-networking challenges for AI*, 2025, pp. 21–27.

[26] J. Dai, X. Wang, K. Fang, Z. Yang, Y. Ji, and J. Zhang, "Geopipe: a geo-distributed llm training framework with enhanced pipeline parallelism in a lossless rdma-enabled datacenter optical transport network," *arXiv preprint arXiv:2510.12064*, 2025.

[27] W. W. Song, M. Jeon, and B.-G. Chun, "Swan: Wan-aware stream processing on geographically-distributed clusters," in *Proceedings of the 13th ACM SIGOPS Asia-Pacific Workshop on Systems*, 2022, pp. 78–84.

[28] A. T. Suresh, X. Y. Felix, S. Kumar, and H. B. McMahan, "Distributed mean estimation with limited communication," in *International conference on machine learning*. PMLR, 2017, pp. 3329–3337.

[29] C. Kim, M. Caesar, and J. Rexford, "Floodless in seattle: a scalable ethernet architecture for large enterprises," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 3–14, 2008.

[30] NVIDIA, "NVIDIA Collective Communications Library (NCCL)," https://developer.nvidia.com/nccl, 2017, online; accessed 6 December 2017.