A Survey on the Workload-Driven Evolution of Datacenter Transport

Junhyeok Lee, Mingyu Park and Jeongyeup Paek
Department of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea

***Abstract*—As modern applications demand large-scale computation, storage, and data movement, datacenters have scaled out with more servers and larger network fabrics. At the same time, the changing demands of these applications have also reshaped the goals and constraints of datacenter transport. In particular, recent AI/HPC workloads make it difficult to treat transport as a standalone network-stack component because collective communication, accelerator utilization, and job-level performance are tightly coupled with transport behavior. This survey examines the workload-driven evolution of datacenter transport across major application eras. We organize representative transport designs into three categories: Flow-centric, RDMA-based, and AI/HPC-oriented transport, and analyze how workload requirements reshape transport objectives, congestion signals, and control points. Through this perspective, we revisit the relationship between workloads and transport design, and discuss future directions for workload-driven datacenter transport.**

I. Introduction

Datacenter networks (DCNs) have evolved into core infrastructures for cloud services, large-scale web applications, distributed storage systems, and more recently, AI training workloads [1], [2]. As datacenters have scaled to infrastructures interconnecting tens of thousands of machines, network performance became a critical factor in application latency, resource utilization, and system efficiency. Unlike wide-area network, DCNs are typically managed by a single operator, enabling joint control of topology, switch configurations, and end-host stacks. This level of control allows operators to deploy transport mechanisms that support dominant application requirements. Therefore, workload shifts have directly shaped the objectives and control scope of DCN transport.

Early datacenter transport designs were shaped by cloud services and partition-aggregate applications, as illustrated in Fig. 1, where flow completion time (FCT) was the main optimization targets [3], [4]. Later, as datacenters began to support more data-intensive workloads such as distributed storage and databases, Remote Direct Memory Access (RDMA) was adopted to reduce CPU overhead through kernel-bypass and zero-copy data movement [5]. Because RDMA over converged Ethernet (RoCE) is highly sensitive to packet loss [6], RDMA-era transport designs focused on reducing loss, controlling queue buildup, and mitigating congestion spreading induced by Priority Flow Control (PFC) [7]. More recently,

This work was supported by the NRF of Korea grant funded by the Korea government (MSIT) (No. RS-2024-00359450), and also by Institute of Information & Communications Technology Planning & Evaluation (IITP)-ITRC (Information Technology Research Center) grant funded by the Korea government (MSIT) (IITP-2026-RS-2022-00156353, 50%).

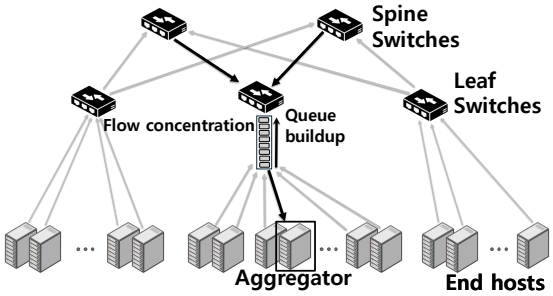

Fig. 1: Queue buildup in partition-aggregate workloads.

AI training and high-performance computing (HPC) workloads have expanded transport requirements beyond individual flows. Because these workloads rely on synchronized collective communication, the slowest path can delay an entire iteration or job. As a result, transport design has moved toward collective- and job-level optimization [2], [8].

Given this progression, understanding datacenter transport requires connecting transport designs with the workload requirements that shaped them. Prior surveys cover specific parts of this evolution, such as fast flow transmission in early DCNs [9] and RDMA transports in DCNs [5]. These surveys clarify the design challenges within their respective domains, but a unified view is still needed to explain how workload shifts expanded transport objectives, control scope, and optimization boundaries. Accordingly, this survey classifies DCN transport into three workload-driven eras, each shaped by the requirements of dominant applications.

- *Flow-centric*: Early cloud workloads, such as web services and remote procedure call (RPC) based applications, generate a diverse mix of short and long flows.
- *RDMA-based*: Data-serving workloads, such as cloud storage and in-memory databases, require fast data movement with minimal CPU involvement.
- *AI/HPC-oriented*: Distributed training and computing workloads use collective or synchronization operations across many GPUs or compute nodes.

Table I summarizes the proposed taxonomy, from which this survey examines how the workloads of each era exposed limitations in existing mechanisms and how subsequent designs addressed them. We show how DCN transport has expanded in its objectives, control granularity, and control points. Finally, we discuss future directions for coordinating network resources as DCNs increasingly support heterogeneous workloads and coexisting transport mechanisms.

Era	Communication pattern	Transport objective	Scheme	Design approach
Flow-centric	Short request-response flows Fan-out/fan-in aggregation Incast-prone bursts	Short-flow FCT minimization Tail-latency reduction Throughput preservation	DCTCP [10]	ECN-based congestion control
			pFabric [3], PIAS [11]	Priority-based flow scheduling
			pHost [12], NDP [4]	Receiver-driven transmission control
RDMA-based	Remote read/write access Host-to-host transfer Bulk data movement	Low-latency access High-throughput transfer Low CPU overhead	DCQCN [7]	ECN-based RDMA congestion control
			TIMELY [13]	RTT-based RDMA congestion control
			HPCC [14]	Telemetry-based RDMA congestion ctrl
AI/HPC-oriented	Collective traffic Many-to-many synchronization Repeated iteration traffic	Collective optimization Job-level utilization Predictable iteration time	Meta RoCE [2], Aquila [15]	Workload-specialized fabric design
			SGLB [8]	Global-scale load balancing
			MCCS [16]	Runtime-level collective control

TABLE I: Workload-driven taxonomy of representative datacenter transport designs.

II. Flow-Centric Transport

This section reviews flow-centric transport designs for early cloud workloads with many short request-response flows.

A. Workload Characteristics.

Early cloud workloads, such as web search and key-value stores, often followed partition-aggregate patterns with distributed components running across multiple servers [3], [10]. A single user-facing request often fanned out to many backend servers through small request-response messages, and the final response was generated after collecting the required backend results. This created a traffic pattern with a large number of short, latency-sensitive flows, often concentrated around aggregator or storage nodes, as illustrated in Fig. 1.

Requirements. Short-flow FCT and tail latency directly affects application-level performance because a user-facing request can be delayed by the slowest backend response. These latency-sensitive flows shared the same network with longer flows from storage, data processing, and service maintenance tasks. As a result, incast congestion, queue buildup, and interference between short and long flows could increase latency even when average network throughput is high. The main objective of this era was to reduce FCT, especially short-flow and tail FCT, under latency-sensitive cloud workloads.

B. Design Approaches

Representative approaches to reduce FCT are as follows.

ECN-based congestion control. DCTCP [10] uses ECN-based congestion control, where switches mark packets when queue length exceeds a threshold and the receiver echoes this information back to the sender. The sender adjusts its congestion window based on the fraction of ECN-marked packets, allowing DCTCP to react to queue buildup before buffer overflow. This keeps queues shallow while maintaining high utilization, thereby improving short-flow FCT.

Flow scheduling and prioritization. pFabric [3] assigns priorities based the remaining flow size, and switches forward higher priority packets first. This allows short flows to bypass long flows and complete quickly, following the same intuition as serving the flow with the least remaining work first. PIAS [11] approximates this without requiring prior knowledge of flow sizes. It uses multiple priority queues and gradually demotes a flow to lower-priority queues as it sends more bytes. Thus, short flows remain in high-priority queues and complete earlier, reducing FCT.

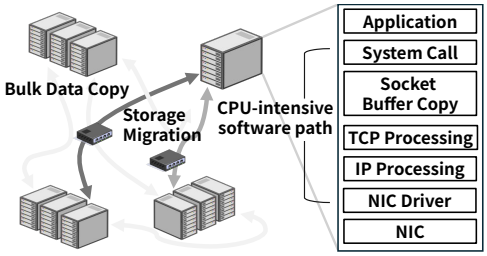

Fig. 2: Host-side TCP/IP stack overhead.

Receiver-driven transport. pHost [12] uses receiver-issued tokens or grants to decide which senders may transmit, preventing blind traffic injection during incast. NDP [4] instead lets receivers pull packets from senders, while packet spraying and trimming handle congestion. By coordinating transmission around the receiver-side bottleneck, these mechanisms reduce queue buildup and improve short-flow FCT.

C. Control Scope

Designs in this era focused on reducing flow-level latency by managing interference among short and long flows and many-to-one incast. This was mainly addressed through local control at two levels: At endpoints, transport mechanisms adjusted sending behavior, processed congestion feedback, or regulated packet admission to reduce persistent queueing, incast bursts, and receiver-side contention. At switches, data-plane mechanisms provided congestion signals, queue management, or packet priority so that latency-sensitive flows were less likely to be blocked by long flows.

III. RDMA-Based Transport

This section turns to RDMA-based transport, driven by frequent host-to-host data movement in datacenter workloads.

A. Workload Characteristics.

In this era, data-serving workloads such as cloud/distributed storage, caching, and databases became major sources of traffic. Because these services store, replicate, and serve data across many machines, remote data access and host-to-host data movement became common traffic patterns. This exposed host-side communication overhead as a major bottleneck. As illustrated in Fig. 2, TCP/IP communication often incurs kernel-stack and memory-copy overheads, consuming CPU cycles and increasing latency.

Requirements. Data-intensive workloads require low-latency and high-throughput data movement with reduced host-side overhead. Datacenters therefore adopted RDMA, where the NIC transfers data directly between registered memory regions on different hosts with kernel bypass, zero-copy transfer, and minimal CPU involvement. RoCE extends these benefits to commodity Ethernet, but packet loss remains costly because recovery depends on limited NIC-side resources. Therefore, RoCE deployments often rely on (near-)lossless Ethernet with PFC, which can cause head-of-line blocking, pause propagation, and congestion spreading. Thus, RDMA transport had to avoid costly packet loss while reducing PFC side effects.

B. Design Approaches

Several RDMA congestion-control designs have been proposed to avoid costly packet loss and excessive PFC pauses.

ECN-based. DCQCN [7] extends the ECN-based congestion control of DCTCP [10] to RoCE networks. While DCTCP adjusts the TCP CWND, DCQCN uses ECN-triggered congestion notifications to reduce RDMA NIC sending rates. By reacting to queue buildup before loss occurs, DCQCN helps RoCE traffic maintain near-lossless operation.

RTT-based. TIMELY [13] measures RTT at the sender and uses its gradient to infer whether queues along the path are building up or draining. Based on this gradient, it updates the target rate of each flow when completion events provide new RTT samples. The updated rate is then used by the control engine to pace subsequent segments, allowing it to react to queueing trends before loss occurs.

Telemetry-based. HPCC [14] uses in-network telemetry to obtain fine-grained path information from switches. Packets carry information such as link utilization and queue occupancy, and the sender uses it to estimate available bandwidth along the path. Based on this explicit network-state feedback, HPCC updates the sending rate more precisely than schemes that rely only on congestion marks or end-to-end delay signals.

C. Control scope

RDMA-based transport expanded the control scope by integrating RDMA-specific hardware and lossless Ethernet behavior. For RDMA traffic, congestion control was implemented as sender-side rate control rather than window adjustment, with sending rates enforced by RNIC-level rate-limiting or pacing mechanisms. At the network side, switches had to operate not only as sources of congestion signals or packet scheduling decisions, but also as part of a lossless Ethernet fabric that interacts with mechanisms such as PFC.

IV. AI/HPC-Oriented Transport

In the final era, AI/HPC workloads shift transport design toward collective and job-level performance.

A. Workload Characteristics.

AI workloads have become increasingly prominent in datacenters, with distributed training repeatedly exchanging gradients, parameters, or intermediate states across many GPUs.

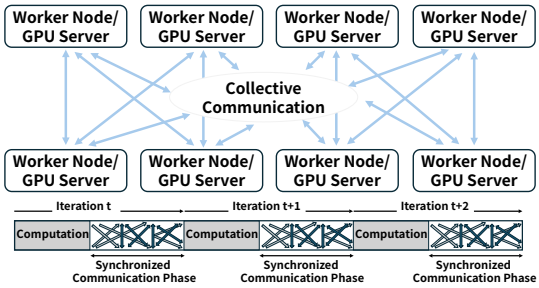

Fig. 3: Synchronized collective communication among parallel AI/HPC workers across training iterations.

HPC workloads were traditionally associated with supercomputing systems, but are increasingly served through cloud-based datacenter environments to support scientific and engineering applications [17]. Although AI and HPC differ in computation models and domains, both commonly generate structured communication among many parallel workers, often through collective or synchronization operations as illustrated in Fig. 3. As a result, traffic in this era is generated not only by independent flows, but also by repeated, synchronized, and job-structured communication patterns.

Requirements. In synchronized AI/HPC workloads, computation often waits for communication or synchronization among workers to complete. Thus, a slow communication flow, path, or participant can delay the entire parallel job, even when average network throughput is high. These workloads therefore require not only high bandwidth and low latency, but also predictable performance across many paths and participants. Network imbalance, congestion, link failures, or bandwidth asymmetry can create communication stragglers and reduce GPU, accelerator, or compute-node utilization. The main objective of this era was therefore to improve collective performance, job completion time, and accelerator utilization under synchronized AI/HPC workloads.

B. Design Approaches

The following approaches aim to improve collective- and job-level performance in AI/HPC-oriented workloads.

Workload-specialized fabric design. Meta deployed RoCE for large-scale AI training by separating the backend GPU training fabric from the frontend network [2]. Within this backend fabric, Meta adapts routing and traffic placement to the low-entropy and repetitive communication patterns of distributed training. It extends multipath forwarding with destination queue-pair information and uses centralized traffic engineering when more explicit path control is needed.

For HPC-style low-latency communication, Aquila [15] changes how traffic is carried inside the datacenter fabric. It introduces a layer-2 fabric protocol that breaks traffic into small cells and forwards them through integrated switching hardware, reducing latency variation compared with conventional packet forwarding.

Global load balancing. SGLB [8] controls how collective traffic is distributed across the network fabric using global

path information. A global controller tracks path health, congestion, failures, and bandwidth limitations, while switches use this information instead of relying only on local hash-based forwarding. By steering traffic away from overloaded or degraded paths, SGLB reduces persistent path imbalance that can delay collective operations.

Runtime- and service-level control. MCCS [16] provides collective communication as a cloud-network service, rather than leaving it entirely inside application-level libraries. The service observes the shared cloud network and coordinates collective operations from multiple jobs, so that their communication does not blindly compete inside the fabric. This moves collective scheduling and placement above conventional transport mechanisms and into the transport control scope.

C. Control Scope

In this era, transport control extends beyond endpoints and switches. At the fabric and global levels, networks may be separated, tuned, or redesigned around AI/HPC communication patterns, while path selection may use fabric-wide information about congestion, failures, and bandwidth asymmetry. Meanwhile, runtime or cloud-level services may coordinate collective execution across multiple jobs. Thus, AI/HPC-oriented transport shows a broader control scope than earlier eras, reflecting the need to optimize communication across the workload as a whole.

V. Discussion and Open Challenges

Overall, workload shifts have expanded the control scope of transport design, creating a challenge for heterogeneous workload coexistence. Even when those workloads share goals such as low latency and high throughput, optimizing for one workload may require mechanisms that interfere with another. For example, protecting RDMA traffic from packet loss may increase queueing, hurting latency-sensitive RPC traffic.

One possible direction is that future datacenters may become less like a single uniform fabric and more like a heterogeneous networking system composed of multiple workload-specialized subnetworks or control domains. In this model, different workload classes could be assigned to different transport domains, fabrics, or control policies, while a higher-level management plane coordinates resource sharing across them. This resembles a smaller-scale version of a wide-area networking problem, where multiple network domains coexist under a broader operational framework.

Another possible direction is, layering may emerge inside the transport itself. Instead of relying on one monolithic transport layer, future designs may separate common transport functions from workload-specific control functions. A lower layer could provide shared mechanisms such as congestion signaling, isolation, and basic reliability, while upper layers adapt routing, rate control, traffic placement, or collective scheduling to each workload class. Furthermore, layer-2 transmission scheduling mechanisms in the TSN standards can be employed to provide deterministic service for highly latency-sensitive flows [18]. The open challenge is how to provide this specialization without losing the efficiency, fairness, and manageability of a unified datacenter network.

VI. Conclusion

We analyzed DCN transport studies from a workload-driven perspective. DCN transport has evolved by expanding its objectives and roles in response to changes in dominant workloads. These shifts suggest that future DCNs will require resource coordination that can balance heterogeneous workload demands with coexisting transport mechanisms. Although the next dominant workload remains uncertain, this survey provides a historical foundation for understanding how datacenter transport can adapt to future workload shifts.

References

[1] A. Singh, J. Ong, A. Agarwal, G. Anderson, A. Armistead, R. Bannon, S. Boving, G. Desai, B. Felderman, P. Germano *et al.*, "Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network," in *ACM SIGCOMM*, 2015, pp. 183–197.

[2] A. Gangidi, R. Miao, S. Zheng, S. J. Bondu, G. Goes, H. Morsy, R. Puri, M. Riftadi, A. J. Shetty, J. Yang *et al.*, "RDMA over Ethernet for Distributed AI Training at Meta Scale," in *ACM SIGCOMM*, 2024.

[3] M. Alizadeh, S. Yang, M. Sharif, S. Katti, N. McKeown, B. Prabhakar, and S. Shenker, "pFabric: Minimal Near-Optimal Datacenter Transport," in *Proceedings of the ACM SIGCOMM Conference*, 2013, pp. 435–446.

[4] M. Handley, C. Raiciu, A. Agache, A. Voinescu, A. W. Moore, G. Antichi, and M. Wójcik, "Re-architecting datacenter networks and stacks for low latency and high performance," in *ACM SIGCOMM*, 2017.

[5] J. Hu, H. Shen, X. Liu, and J. Wang, "RDMA Transports in Datacenter Networks: Survey," *IEEE Network*, pp. 380–387, 2024.

[6] R. Mittal, A. Shpiner, A. Panda, E. Zahavi, A. Krishnamurthy, S. Ratnasamy, and S. Shenker, "Revisiting Network Support for RDMA," in *ACM SIGCOMM*, 2018, pp. 313–326.

[7] Y. Zhu, H. Eran, D. Firestone, C. Guo, M. Lipshteyn, Y. Liron, J. Padhye, S. Raindel, M. H. Yahia, and M. Zhang, "Congestion Control for Large-Scale RDMA Deployments," in *ACM SIGCOMM*, 2015, pp. 523–536.

[8] C. Qi, W. Wu, Y. Wang, K. He, Y.-H. Kao, Z. He, C.-Y. Yen, Z. Jiang, F. Luo, S. Anubolu *et al.*, "SGLB: Scalable and Robust Global Load Balancing in Commodity AI Clusters," in *ACM SIGCOMM*, 2025.

[9] R. Rojas-Cessa, Y. Kaymak, and Z. Dong, "Schemes for Fast Transmission of Flows in Data Center Networks," *IEEE Communications Surveys & Tutorials*, pp. 1391–1422, 2015.

[10] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data Center TCP (DCTCP)," in *ACM SIGCOMM*, 2010, pp. 63–74.

[11] W. Bai, L. Chen, K. Chen, D. Han, C. Tian, and H. Wang, "Information-Agnostic Flow Scheduling for Commodity Data Centers," in *USENIX NSDI*, 2015, pp. 455–468.

[12] P. X. Gao, A. Narayan, G. Kumar, R. Agarwal, S. Ratnasamy, and S. Shenker, "pHost: Distributed Near-Optimal Datacenter Transport Over Commodity Network Fabric," in *ACM CoNEXT*, 2015, pp. 1–12.

[13] R. Mittal, V. T. Lam, N. Dukkipati, E. Blem, H. Wassel, M. Ghobadi, A. Vahdat, Y. Wang, D. Wetherall, and D. Zats, "TIMELY: RTT-Based Congestion Control for the Datacenter," in *ACM SIGCOMM*, 2015.

[14] 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 *ACM SIGCOMM*, 2019, pp. 44–58.

[15] D. Gibson, H. Hariharan, E. Lance, M. McLaren, B. Montazeri, A. Singh, S. Wang, H. M. Wassel, Z. Wu, S. Yoo *et al.*, "Aquila: A unified, low-latency fabric for datacenter networks," in *USENIX NSDI*, 2022, pp. 1249–1266.

[16] Y. Wu, Y. Xu, J. Chen, Z. Wang, Y. Zhang, M. Lentz, and D. Zhuo, "Mccs: A Service-based Approach to Collective Communication for Multi-Tenant Cloud," in *ACM SIGCOMM*, 2024, pp. 679–690.

[17] M. A. Netto, R. N. Calheiros, E. R. Rodrigues, R. L. Cunha, and R. Buyya, "HPC Cloud for Scientific and Business Applications: Taxonomy, Vision, and Research Challenges," *ACM Computing Surveys*, 2018.

[18] Y. Seol, D. Hyeon, J. Min, M. Kim, and J. Paek, "Timely Survey of Time-Sensitive Networking: Past and Future Directions," *IEEE Access*, vol. 9, pp. 142 506–142 527, Oct 2021.