

Path-Aware Networking for Cross-Datacenter Disaggregated LLM Inference

1 Introduction

Large language model (LLM) inference is increasingly moving toward disaggregated architectures that separate computation with different resource requirements. A prominent example is prefill-decode (PD) disaggregation: prefill is primarily compute-intensive, whereas decode is primarily memory-bandwidth-intensive. Separating these phases allows each to run on hardware better suited to its workload and enables their resources to be provisioned independently.

Most existing disaggregated inference systems assume that the participating workers remain within the same datacenter or high-bandwidth network domain. This assumption is beginning to change. Recent model architectures substantially reduce the size of the key-value (KV) cache, making it increasingly practical to place prefill and decode in different clusters or datacenters. Recent work such as Prefill-as-a-Service (PrfaaS) demonstrates that selectively offloading prefill and transferring the resulting KV cache over commodity inter-datacenter networks can enable heterogeneous cross-datacenter serving [1].

The network, however, becomes a central bottleneck once disaggregation crosses datacenter boundaries. After prefill, the generated KV cache must reach the decode worker before decoding can proceed. Compared with intra-datacenter interconnects, wide-area paths generally provide lower bandwidth and higher latency. Recent measurements show that cross-datacenter KV transfer can account for a substantial fraction of end-to-end latency, particularly for agentic workloads with long contexts and short outputs [2]. Existing systems address this problem through mechanisms such as selective offloading, cache-aware placement, KV-cache reduction, and overlapping communication with computation [1, 2].

These developments suggest a broader shift in how networks should be treated by LLM inference systems. Rather than viewing the network as a passive transport mechanism, recent work argues that bandwidth, latency, and transfer cost should directly influence where KV state is stored, transferred, or recomputed [3]. However, existing cross-datacenter disaggregation systems largely reason about the available inter-datacenter network as a given communication channel.

This project investigates a complementary direction: making cross-datacenter disaggregated inference explicitly *path-aware*. In realistic deployments, two clusters may be connected through multiple paths, providers, tunnels, or network domains with different bandwidth, latency, congestion, and reliability. At the same time, KV transfers differ substantially in size and urgency: some

transfers lie directly on the TTFT critical path, while others can be partially cached, recomputed, prefetched, or overlapped with useful computation.

The goal of this project is to determine how these inference-level semantics can be combined with network measurements to make better communication decisions. Rather than simply deciding *where* inference executes, a network-aware system can jointly determine *what to transfer*, *when to transfer it*, and *over which path*, with the objective of minimizing inference-visible communication overhead under heterogeneous wide-area network conditions.

2 Project Description

The project will study path-aware communication for cross-datacenter disaggregated LLM inference. It will first characterize the network traffic and critical-path dependencies introduced by disaggregation, then design mechanisms for selecting and scheduling network paths according to both inference state and network conditions. The project is organized into the following tasks.

Task 1: Characterizing Cross-Datacenter Disaggregation

Build an experimental environment for disaggregated LLM inference and characterize the communication generated between geographically separated inference workers. Prefill-decode disaggregation will serve as the initial case study, where completion of prefill produces KV state that must be made available to a remote decode worker.

Key questions include:

- How much data must cross the datacenter boundary for different models, context lengths, attention mechanisms, and prefix-cache hit rates?
- When does KV-cache transfer become the bottleneck for TTFT or overall inference throughput?
- How do bandwidth, latency, jitter, packet loss, and transient congestion affect cross-datacenter KV transfer?
- How much communication can be hidden through caching, prefetching, recomputation, or overlap with useful inference computation?
- Which transfers lie on the inference critical path, and how does this change across requests and disaggregation architectures?

The resulting measurements should identify the network regimes in which cross-datacenter disaggregation is beneficial and establish the communication requirements that a path-aware network layer must satisfy.

Task 2: Characterizing Path Heterogeneity

Extend the experimental environment to expose multiple network paths between inference clusters. These paths may differ in bandwidth, propagation latency, congestion, loss, or available capacity, and their relative performance may change over time.

The project will characterize how these properties affect KV-cache transfers of different sizes and determine whether the path that provides the highest bandwidth is necessarily the path that minimizes inference-visible transfer time.

This task should investigate:

- How transfer completion time changes with path bandwidth, latency, congestion, and loss.
- How the preferred path changes with KV-transfer size and request characteristics.
- How concurrent KV transfers interact when sharing the same bottleneck.
- Whether path rankings remain stable across workload and network regimes.
- How accurately lightweight network measurements can predict the completion time of future KV transfers.

An important outcome of this task is a completion-time model that maps a KV transfer and the current network state to its expected cost on each candidate path. Depending on project progress, this may range from an empirical model to an analytical model that explicitly captures bandwidth- and latency-sensitive transfer regimes.

Task 3: Path-Aware KV Transfer

Based on the characterization above, design a path-aware communication layer for cross-datacenter disaggregated inference. Rather than relying on a fixed WAN path or conventional load balancing, the system should explicitly assign KV transfers to paths according to their expected impact on inference.

Possible mechanisms include:

- Selecting a path for each KV transfer based on predicted transfer completion time.
- Steering different requests or KV-cache fragments over different paths.
- Reserving low-latency paths for transfers on the TTFT critical path while using high-capacity paths for larger background transfers.
- Spreading transfers across multiple paths when the additional aggregate capacity outweighs the cost of path heterogeneity.
- Dynamically adapting path selection as available bandwidth and congestion change.

The objective is not necessarily to minimize the completion time of every individual network transfer. Instead, path selection should minimize *inference-visible* communication time and avoid network contention that delays critical requests.

Task 4: Joint Inference and Network Scheduling

Investigate whether inference semantics can be combined with network measurements to make better scheduling decisions than network-only path selection.

A scheduler may jointly consider:

- KV-cache size and the amount of state already available at the destination.
- Whether the transfer blocks decode and therefore contributes directly to TTFT.
- Prefix-cache availability and expected cache reuse.
- The cost of recomputing versus transferring missing KV state.
- Opportunities to overlap transfer with local or remote computation.
- Current and predicted bandwidth, latency, and load of candidate paths.
- Contention between concurrent KV transfers.

Based on these signals, the scheduler will determine which transfers should be prioritized and how network capacity should be allocated among them. For example, a small transfer that immediately unblocks decode may be placed on a low-latency path, while a large transfer that can be overlapped with computation may use a higher-bandwidth but higher-latency path.

Depending on project progress, the project may further explore joint decisions across execution placement and networking. In this setting, the system chooses not only the network path but also whether remote disaggregation is worthwhile for a request given its expected compute benefit and communication cost. This turns the scheduling problem into a joint decision over *where to execute*, *what to transfer*, *when to transfer it*, and *over which path*.

Task 5: Experimental Evaluation

Evaluate the proposed mechanisms under controlled and realistic network and inference conditions. Experiments should vary both workload and network characteristics, including model architecture, context length, cache hit rate, request concurrency, KV-cache size, path bandwidth, latency, path heterogeneity, and transient network degradation.

The evaluation should compare against conventional cross-datacenter disaggregation using a fixed path or network-oblivious load balancing, as well as network-aware baselines where appropriate. Metrics should include:

- Time-to-first-token and end-to-end request latency.
- Tail latency under concurrent inference workloads.
- Inference throughput or SLA-compliant goodput.
- KV-transfer completion time and communication time exposed on the inference critical path.
- Network utilization and transferred bytes.
- Robustness to heterogeneous and time-varying network conditions.
- Sensitivity to path selection, scheduling, caching, and workload parameters.

An important goal is to distinguish the benefit of intelligent path selection from simply providing additional network capacity. The evaluation should therefore include controlled experiments with identical aggregate bandwidth but different path heterogeneity, as well as comparisons between path-aware scheduling and bandwidth-oblivious traffic spreading.

Background and Skills

The project lies at the intersection of **computer networks**, **distributed systems**, and **LLM inference systems**. A successful candidate will benefit from familiarity with:

- Computer networks, including bandwidth, latency, congestion, transport protocols, and multipath communication.
- Distributed systems and scheduling.
- LLM inference and common serving architectures.
- Python and machine-learning frameworks such as PyTorch.
- Experimental methodology and performance analysis.

Prior experience with GPU communication, RDMA, LLM serving frameworks, wide-area networking, or network emulation is advantageous but not required.

References

- [1] R. Qin, W. He, Y. Wang, Z. Li, X. Xu, Y. Wu, W. Zheng, and M. Zhang, “Prefill-as-a-service: Kvcache of next-generation models could go cross-datacenter,” 2026.
- [2] Y. Wang, X. Li, J. Ma, G. Sun, Y. Xu, B. Han, J. Ng, Y. Luo, K. Hong, G. Dai, B. Li, and Y. Wang, “Pdd: Unleashing economical and flexible heterogeneous llm inference via cross-datacenter prefill-decode disaggregation,” 2026.
- [3] S. Ray, N. Feamster, and J. Jiang, “An internet for the kv cache: Rethinking classical infrastructure boundaries in the llm inference age,” 2026.