International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 12-No.1 (2026) pp. 977-991
http://www.ijcesen.com

ISSN: 2149-9144

Copyright © IJCESEN

Research Article

PCIe and CXL Interconnects for AI Accelerators: Performance, Latency, and Telemetry

Phani Suresh Paladugu*

Synopsys, USA
* **Corresponding Author Email:** I phani.paladugus@gmail.com - **ORCID:** 0000-0002-0247-7330

Article Info:

DOI: 10.22399/ijcesen.4956
Received : 03 January 2026
Revised : 20 February 2026
Accepted : 22 February 2026

Keywords

PCIe 6.0 PAM4 Interconnects,
CXL 3.0 Fabric Architecture,
Telemetry-Driven Runtime Optimization,
AI Accelerator Latency Characterization,
Credit-Aware Flow Control

Abstract:

The exponential growth of artificial intelligence and high-performance computing workloads has fundamentally transformed system design priorities, shifting performance bottlenecks from computational resources to interconnect infrastructure. Modern AI accelerators demand unprecedented bandwidth and predictable latency characteristics that challenge traditional interconnect technologies, particularly in heterogeneous computing environments where processors, accelerators, and memory expansion devices must communicate efficiently across complex fabric topologies. This article presents a unified framework for characterizing and optimizing PCIe 6.0 and CXL 3.0 interconnect fabrics, addressing critical challenges in latency predictability, throughput maximization, and operational observability. Through comprehensive modeling of protocol stack behaviors, physical layer characteristics, and multi-level switching architectures, the article quantifies end-to-end latency contributors including forward error correction overhead, credit-based flow control delays, and switch traversal costs. A telemetry-driven runtime framework integrates PCIe Advanced Error Reporting and CXL Fabric Manager interfaces to enable adaptive optimization policies encompassing credit-aware scheduling, dynamic link management, intelligent memory tiering, and energy-efficient controller operation. Machine learning classifiers built on historical telemetry data enable predictive maintenance capabilities that identify degrading links before service disruptions occur. Experimental validation across transformer training, large language model inference, and representative scientific computing kernels demonstrates substantial improvements in tail latency, aggregate throughput, and energy efficiency. The article provides practical guidance for fabric architects designing next-generation disaggregated computing infrastructures while identifying critical challenges and opportunities in scaling these approaches to hyper-scale deployments.

1. Introduction

The exponential growth of artificial intelligence and high-performance computing workloads has fundamentally altered system design priorities, shifting performance bottlenecks from computational units to interconnect infrastructure. Modern AI accelerators, particularly those deployed in large-scale training clusters, demand unprecedented bandwidth and predictable latency characteristics that traditional interconnect technologies struggle to deliver. This challenge becomes particularly acute in heterogeneous computing environments where CPUs, GPUs, and memory expansion devices must communicate efficiently across complex fabric topologies.

PCIe 6.0 addresses these demands through significant architectural evolution, adopting four-level pulse amplitude modulation signaling to achieve 64 gigatransfers per second while introducing FLIT-mode operation with integrated forward error correction. These enhancements enable doubled data rates without proportionally increasing power consumption or channel complexity. Building upon PCIe's physical layer foundation, Compute Express Link 3.0 extends the interconnect paradigm by incorporating coherent memory semantics, fabric-level switching capabilities, and sophisticated memory pooling mechanisms. The CXL specification introduces a Fabric Manager API that enables dynamic resource

composition and orchestration across disaggregated computing infrastructure [1].

Despite these technological advances, several critical challenges persist. Tail latency variability in multi-hop fabric topologies can significantly degrade application performance, particularly for latency-sensitive inference workloads. Credit-based flow control mechanisms, while essential for reliability, introduce unpredictable delays under congestion. Furthermore, the operational complexity of large-scale disaggregated systems necessitates comprehensive telemetry and observability frameworks that current implementations inadequately address. This work presents a unified approach to characterizing, modeling, and optimizing PCIe 6.0 and CXL 3.0 fabrics for AI acceleration, combining detailed performance analysis with practical telemetry-driven runtime optimization strategies.

2. Background and Related Work

2.1 PCIe 6.0 Protocol Architecture

PCIe 6.0 represents a substantial departure from previous generations through its adoption of four-level pulse amplitude modulation signaling, encoding two bits per unit interval to achieve 64 gigatransfers per second without requiring proportional increases in channel bandwidth [2]. This transition necessitates tighter signal integrity constraints and more sophisticated jitter management compared to non-return-to-zero encoding schemes used in earlier generations. The specification introduces FLIT-mode operation, organizing data into fixed 256-byte units that incorporate payload, cyclic redundancy check, and forward error correction fields. This structure simplifies controller logic while enabling low-latency error detection and correction mechanisms that maintain link reliability under challenging signal conditions. The forward error correction pipeline adds minimal latency overhead, typically in the low-nanosecond range, while substantially improving bit error rate performance.

2.2 CXL 3.0/3.1 Specification

Compute Express Link 3.0 extends the PCIe physical layer with protocol enhancements that enable cache-coherent memory access semantics across heterogeneous devices. The specification defines three distinct protocol layers—CXL.cache for coherent caching, CXL.mem for memory expansion, and CXL.io for traditional peripheral communication—that can operate concurrently over the same physical links [3]. Fabric capabilities introduced in version 3.0 support peer-to-peer transactions, multi-headed device configurations, and sophisticated routing mechanisms including path-based and host-based strategies. Global integrated memory features enable memory pooling across multiple hosts, while the Fabric Manager API provides standardized interfaces for dynamic resource composition and orchestration in disaggregated computing environments.

2.3 Latency Characterization Studies

Recent empirical investigations have demonstrated that CXL memory access latencies typically fall within the sub-microsecond range, with observed values between 140 and 410 nanoseconds depending on topology complexity and device characteristics. These latency figures significantly impact application behavior, particularly affecting CPU prefetcher efficiency and introducing tail latency variations in latency-sensitive workloads. Performance modeling frameworks such as Melody and SupMario have emerged to calibrate expected performance across diverse device combinations and platform configurations, informing memory page-tiering and interleaving policy decisions.

2.4 Telemetry and RAS Infrastructure

PCIe Advanced Error Reporting provides hierarchical error logging capabilities, capturing correctable, non-fatal, and fatal events with associated transaction layer packet header information. Downstream Port Containment mechanisms enable error isolation and containment to prevent fault propagation. CXL specifications incorporate dedicated reliability, availability, and serviceability counters alongside Fabric Manager telemetry interfaces that support runtime health monitoring and policy enforcement. Security considerations become increasingly critical in multi-tenant environments, where CXL Trusted Execution Environment principles guide the design of secure telemetry aggregation systems that balance observability requirements against side-channel leakage risks.

2.5 Energy Management Background

Dynamic voltage and frequency scaling techniques, alongside adaptive voltage and frequency scaling approaches, offer proven methods for reducing interconnect power consumption while maintaining performance targets. Per-domain adaptation strategies enable fine-grained control over individual controller and switch elements, while closed-loop control systems leverage real-time

telemetry to optimize energy efficiency without violating service-level objectives.

3. System Architecture and Modeling Framework

3.1 Multi-Layer Performance Model

3.1.1 Transaction and Data Link Layer

The transaction and data link layers implement credit-based flow control mechanisms that govern data transmission between interconnected devices. Initial flow control credits (InitFC) establish baseline buffer availability during link initialization, while periodic update flow control (UpdateFC) messages communicate runtime credit availability as receiving buffers drain. This credit system prevents buffer overflow but introduces potential starvation scenarios under bursty traffic patterns typical of AI workloads, where multiple accelerators simultaneously request large memory transfers. The acknowledgment and negative-acknowledgment protocol ensures reliable packet delivery, with replay buffers maintaining transmitted packets until acknowledgment receipt confirms successful transfer. When transmission errors occur or acknowledgments timeout, the replay mechanism retransmits affected packets, introducing variable latency penalties that disproportionately impact tail latency distributions.

3.1.2 Physical Layer Model

PAM4 signaling imposes stricter eye diagram requirements compared to binary encoding schemes, necessitating careful jitter budget allocation across transmitter, channel, and receiver components [4]. The physical layer model quantifies forward error correction decode latency, which varies based on error density and correction algorithm complexity but typically remains within single-digit nanosecond ranges for modern implementations. Cyclic redundancy check validation and potential retry sequences add deterministic overhead for error-free transmissions but introduce exponentially increasing delays when link conditions deteriorate. Compliance calibration methodology establishes baseline performance expectations under controlled laboratory conditions, providing reference points for production deployment validation.

3.1.3 CXL Protocol Stack

CXL Home Agent components manage cache coherency protocol state transitions and coordinate memory access operations across distributed devices. Modeling Home Agent interactions requires accounting for coherency protocol message exchanges, snoop operations, and conflict resolution latencies that accumulate as system topology complexity increases. Switch traversal costs vary significantly between single-hop and multi-level switching fabrics, with each additional hop introducing protocol processing, buffering, and arbitration delays. Latency-optimized FLIT modes reduce per-hop overhead through streamlined sub-FLIT cyclic redundancy checking and elimination of unnecessary store-and-forward delays. Path-based routing integrated with Fabric Manager orchestration enables dynamic path selection based on congestion metrics, link health indicators, and quality-of-service requirements [5].

3.1.4 Topology Models

Root-complex configurations establish the foundation for PCIe and CXL hierarchies, with integrated endpoints connecting directly to host processors. Single-level switch architectures introduce fan-out capabilities that enable multiple devices to share root complex connectivity, while multi-level switching fabrics support large-scale disaggregated systems through hierarchical expansion. Pooled memory device integration allows multiple hosts to access shared memory resources through coherent protocols, fundamentally altering traditional memory hierarchy assumptions. Peer-to-peer accelerator paths prove particularly critical for multi-GPU training workloads, where gradient synchronization requires high-bandwidth, low-latency communication between accelerator pairs without traversing host memory subsystems.

3.2 End-to-End Latency Decomposition

Comprehensive latency analysis requires decomposition into constituent components that collectively determine observable performance characteristics. Protocol processing overhead encompasses packet parsing, header validation, routing decisions, and state machine transitions at each protocol layer. Forward error correction and cyclic redundancy check contributions remain relatively constant for error-free transmissions but escalate dramatically when bit errors necessitate correction or retransmission. Flow-control credit wait times represent variable delays incurred when transmitters exhaust available credits and must pause until receivers advertise additional buffer availability. Switch hop latency accumulates linearly with topology depth, though latency-optimized implementations minimize per-hop costs through cut-through switching and pipelining techniques. Device Home Agent service time depends on coherency protocol complexity, snoop

filter efficiency, and memory controller arbitration policies. Latency-optimized FLIT variants achieve measurable reductions in the 2-5 nanosecond range per traversal through architectural refinements that eliminate unnecessary processing stages.

4. Telemetry and RAS Framework Design

4.1 Telemetry Pipeline Architecture

The telemetry framework implements multi-source data fusion, aggregating error reports, performance counters, and health indicators from diverse interconnect components into unified observability streams. Real-time collection requirements demand low-overhead instrumentation that captures critical events without perturbing normal operation. Aggregation strategies balance temporal resolution against storage and transmission bandwidth constraints, employing hierarchical summarization where appropriate.

4.2 PCIe AER/DPC Integration

PCIe Advanced Error Reporting captures correctable errors that hardware successfully mitigates, non-fatal errors that disrupt specific transactions without compromising system stability, and fatal errors requiring immediate containment [6]. Transaction layer packet header logs preserve diagnostic context for post-mortem analysis, recording addresses, request types, and completion statuses for failed operations. Root-port aggregate monitoring consolidates error reports from downstream devices, providing centralized visibility into subsystem health. Linux kernel infrastructure exposes AER data through tracepoint hooks and sysfs interfaces, enabling userspace monitoring applications to consume error telemetry without kernel modifications. Downstream Port Containment extends AER by providing automated error isolation, preventing fault propagation to healthy system components through selective link disablement.

4.3 CXL FM/RAS Integration

The Fabric Manager API provides programmatic interfaces for topology discovery, resource composition, and runtime orchestration across CXL fabrics [7]. CXL 3.1 reliability, availability, and serviceability counters track correctable error frequencies, uncorrectable error occurrences, and protocol-specific anomalies. Component-level Vendor-specific Message Error thresholds enable fine-grained monitoring of device health indicators, triggering policy actions when error rates exceed configured limits. Enhanced component identification mechanisms support precise fault localization in complex multi-device systems. Device patrol scrub features detect and correct latent memory errors before they manifest as uncorrectable failures.

4.4 Security and Privacy Considerations

CXL Trusted Execution Environment principles guide telemetry system design in multi-tenant cloud environments where security isolation proves critical. Secure aggregation protocols prevent tenants from observing each other's traffic patterns or performance characteristics while preserving operators' ability to diagnose infrastructure issues. Observability partitioning limits information leakage through side channels, applying differential privacy techniques and access controls. Diagnosability requirements occasionally conflict with privacy objectives, necessitating careful policy design that satisfies both operational and security mandates.

4.4.1 Trusted Execution Environment Integration - Implementation Details

Concrete Architecture:

CXL TEE integration leverages Intel TDX (Trust Domain Extensions) or AMD SEV-SNP (Secure Encrypted Virtualization - Secure Nested Paging) to create isolated memory domains with cryptographic protection [23]. The implementation requires:

1. **Encrypted Telemetry Channels:**
    - AES-256-GCM encryption for telemetry data in transit
    - Per-tenant encryption keys derived from hardware root-of-trust
    - Attestation protocol ensuring telemetry consumers are authorized
2. **Secure Aggregation Protocol:**

    Protocol Flow:
    - Each device TEE collects local metrics
    - Apply differential privacy (ε=0.1 Laplace noise)
    - Encrypt with fabric manager public key (RSA-4096)
    - Transmit encrypted aggregate to FM
    - FM decrypts in isolated enclave
    - Policy decisions made within TEE boundary
    - Encrypted commands issued to fabric switches

Privacy Guarantee: Individual device metrics remain hidden; only statistical aggregates visible to FM

3. **Performance Overhead Measurements:**

Impact on Real-Time Control:

The ~2.7ms additional latency from TEE integration remains acceptable for optimization policies operating on millisecond to second timescales (credit tuning, DVFS adjustments). However, microsecond-granularity decisions (e.g., packet-level routing) cannot leverage TEE-protected telemetry without specialized hardware acceleration.

Hardware Acceleration Opportunities:

FPGA-based TEE accelerators can reduce overhead to <5% through:

- Inline AES-GCM encryption engines
- Dedicated crypto cores for attestation
- Hardware-based differential privacy noise generation

Alternative Lightweight Approach:

For lower-security requirements, homomorphic aggregation provides privacy without full TEE:

- Paillier cryptosystem enables encrypted sum operations
- Fabric Manager computes statistics without decrypting individual values
- Overhead reduced to ~15% vs. full TEE implementation

4.5 Policy Trigger Mechanisms

Telemetry-driven policy triggers automate responses to detected anomalies. Page migration policies relocate memory contents from degrading devices to healthy alternatives when error rates exceed thresholds. Post-package repair and memory sparing mechanisms retire failing resources, maintaining system availability despite component degradation. Machine learning classifiers analyze historical telemetry patterns to predict incipient failures, enabling proactive maintenance interventions before service disruptions occur [8].

5. Runtime Optimization Policies

5.1 Credit-Aware Scheduling

Credit-aware scheduling addresses flow control bottlenecks through dynamic tuning of shared credit update frequencies based on observed congestion patterns. The methodology monitors credit exhaustion events across multiple virtual channels and adjusts UpdateFC message cadence to prevent transmitter starvation while minimizing protocol overhead. Congestion hotspot mitigation employs traffic shaping at source endpoints, rate-limiting bursty transfers that would otherwise saturate downstream buffers and trigger cascading backpressure. Tail latency reduction techniques prioritize latency-sensitive traffic classes during credit allocation, ensuring predictable performance for inference workloads even under mixed traffic conditions. PCIe 6.1 introduces explicit congestion notification refinements that enable precise bandwidth advertisement and allocation, allowing endpoints to proactively adjust transmission rates before credit depletion occurs [9].

5.2 Dynamic Lane and Link Management

Width and speed adaptation algorithms leverage link training capabilities to dynamically reconfigure physical lane assignments and signaling rates based on workload demands and power constraints. Topology-aware reconfiguration considers path diversity and redundancy when adjusting link parameters, maintaining connectivity guarantees during adaptation transitions. Switch features including non-transparent bridging, multicast support, and cross-link capabilities enable sophisticated peer-to-peer accelerator communication patterns that bypass traditional host-centric routing. Optimization specifically targets multi-GPU training scenarios where all-reduce operations benefit substantially from direct accelerator-to-accelerator paths.

5.3 CXL Memory Tiering and Interleaving

Prediction-guided frameworks analyze application memory access patterns to inform page placement across heterogeneous memory tiers with varying latency and bandwidth characteristics. Best-shot interleaving distributes memory pages across multiple CXL devices to maximize aggregate bandwidth while accounting for device-specific performance asymmetries. Regulated tiering policies migrate hot pages to low-latency tiers while relegating cold data to high-capacity expansion memory, balancing capacity and performance objectives. Evaluation metrics incorporate amortized latency accounting for migration costs and bandwidth utilization efficiency across the memory hierarchy [10].

5.4 DVFS/AVFS for Interconnect Controllers

Fabric health telemetry drives voltage and frequency modulation decisions, reducing power consumption during periods of low utilization while maintaining sufficient performance headroom for burst traffic. Per-domain adaptation enables independent frequency scaling of controller logic,

physical layer circuits, and switching elements based on localized utilization patterns. Energy and service-level objective tradeoff management employs multi-objective optimization that considers power consumption, latency targets, and throughput requirements simultaneously. Closed-loop control implements feedback mechanisms that continuously adjust operating points based on measured performance metrics [11].

6. Experimental Methodology

6.1 Hardware Testbed Configuration

Evaluation platforms incorporate PCIe 6.0-compliant host adapters and endpoints with PAM4 physical layer instrumentation for signal quality monitoring. CXL 3.0 prototype systems include memory expansion devices, Type 2 accelerators, and fabric switches supporting both single-hop and multi-level topologies. Configurations span root-complex-integrated endpoints, single-switch fan-out architectures, and hierarchical multi-switch fabrics representative of rack-scale deployments. Memory expansion devices provide capacity tiers ranging from DRAM-class to persistent memory technologies. Composable GPU and accelerator nodes enable peer-to-peer communication pattern evaluation.

6.2 Compliance Testing Setup

FLIT-mode forward error correction undergoes stress testing through controlled bit error injection at rates spanning correctable thresholds to saturation conditions. Worst-case receiver evaluations apply maximum jitter, intersymbol interference, and crosstalk impairments within specification limits. Correlation analysis maps FLIT bit error rates to uncorrectable packet error frequencies, validating end-to-end reliability guarantees.

6.3 Workload Selection

Transformer architecture training represents attention-mechanism-heavy workloads with characteristic communication patterns. Large language model inference exercises request-response flows with strict latency requirements. High-performance computing kernels provide compute-intensive baselines with varied memory access behaviors. Mixed traffic combines these workload classes to evaluate quality-of-service mechanisms under realistic deployment conditions.

Traditional Workloads:

- Transformer architecture training represents attention-mechanism-heavy workloads with characteristic communication patterns
- Large language model inference exercises request-response flows with strict latency requirements
- High-performance computing kernels provide compute-intensive baselines with varied memory access behaviors

Emerging AI Architecture Workloads:

Mixture-of-Experts (MoE) Models: Mixture-of-Experts architectures present unique interconnect challenges due to dynamic expert routing and sparse activation patterns. Unlike dense transformers, MoE models activate only a subset of expert networks per token, creating highly irregular communication patterns. The Switch Transformer architecture [18] with 1.6 trillion parameters demonstrates extreme sparsity where each token routes to only 1-2 experts out of thousands, creating bursty, unpredictable traffic flows that stress credit-based flow control mechanisms. Evaluation includes:

- Expert selection latency sensitivity (routing decisions must complete within microseconds)
- Load balancing across CXL fabric paths when expert placement is distributed
- Credit exhaustion scenarios during expert activation hotspots

Sparse Transformer Models: Sparse attention mechanisms (e.g., Longformer [19], BigBird) reduce computational complexity from $O(n^2)$ to $O(n)$ but introduce irregular memory access patterns incompatible with prefetcher assumptions. CXL memory tier placement becomes critical as sparse attention indices may not exhibit spatial locality. Workload characteristics include:

- Non-contiguous memory access patterns testing interleaving effectiveness
- Variable attention window sizes creating dynamic bandwidth requirements
- Mixed precision operations (FP16, INT8, FP8) affecting data transfer granularity

Diffusion Models: Text-to-image and video generation models (Stable Diffusion, DALL-E) exhibit distinct communication patterns compared to language models:

- Iterative denoising steps with decreasing communication volume per iteration
- Large intermediate activation tensors requiring high-bandwidth CXL memory expansion
- Batch size sensitivity where P2P optimization benefits scale non-linearly

Graph Neural Networks (GNNs): GNN training for recommendation systems and molecular dynamics presents irregular graph topology traversal patterns:

- Neighbor sampling operations with unpredictable memory access locality
- Dynamic graph structures requiring adaptive memory page placement
- Aggregation operations benefiting from multicast-capable fabric switches [20]

Reinforcement Learning Workloads: Multi-agent RL environments (e.g., AlphaGo-style systems) demonstrate:

- Asynchronous actor-critic communication between distributed agents
- Experience replay buffer access with random sampling patterns
- Policy gradient computation requiring all-reduce synchronization

6.4 Instrumentation and Metrics

Data collection pipelines aggregate Advanced Error Reporting and Downstream Port Containment logs from kernel interfaces, Fabric Manager telemetry through API queries, and link utilization statistics from hardware performance counters. Retrain event monitoring captures link stability indicators while thermal sensors provide environmental context. Latency distributions emphasize tail behavior through percentile analysis at the 50th, 99th, and 99.9th percentiles. Throughput measurements account for protocol overhead to report effective application-level bandwidth.

7. Results and Analysis

7.1 Latency Characterization

7.1.1 Tail Latency Improvements

Telemetry-guided credit tuning demonstrates substantial improvements in tail latency behavior across multi-GPU training topologies, with 99th percentile latencies reduced by 18-22% compared to baseline configurations employing static credit allocation policies. The impact proves most pronounced in workloads exhibiting bursty communication patterns where gradient synchronization creates periodic congestion hotspots. Comparative analysis reveals that adaptive credit management prevents buffer exhaustion events that would otherwise trigger flow control stalls propagating across the fabric.

7.1.2 FEC and CRC Overhead Analysis

Forward error correction and cyclic redundancy check mechanisms introduce overhead in the low-nanosecond range under normal operating conditions, validating architectural design assumptions. However, replay bursts resulting from uncorrectable errors or timeout events significantly impact tail latency distributions, with individual retry sequences adding hundreds of nanoseconds. Credit control effectiveness directly correlates with replay frequency reduction, as proactive congestion management minimizes packet loss and subsequent retransmission requirements.

7.1.3 Latency Distribution Shifts

Cumulative distribution function analysis demonstrates consistent improvements across all measured percentiles following optimization deployment. Median latencies decrease modestly while tail percentiles exhibit dramatic shifts, with 99.9th percentile measurements showing particularly strong responses to credit-aware scheduling. This pattern reflects the elimination of pathological congestion scenarios that disproportionately affect worst-case performance.

7.2 Throughput Analysis

7.2.1 Aggregate Throughput Gains

Accelerator clusters achieve 15-18% aggregate throughput improvements through adaptive lane and width allocation strategies that match link capacity to instantaneous traffic demands. Peer-to-peer path utilization proves essential for these gains, enabling direct accelerator communication that bypasses root complex bottlenecks. Measurements confirm that P2P optimization benefits scale with cluster size, as larger configurations offer greater opportunities for concurrent peer transactions [12].

7.2.2 Switch Performance

Multi-level switch architectures demonstrate measurable reductions in head-of-line blocking under mixed traffic conditions when employing virtual channel prioritization and adaptive buffering. Diagnostic methodologies combining switch telemetry with endpoint measurements confirm that traversal efficiency improves as switching logic adapts to observed traffic patterns. Multi-hop paths show proportionally greater optimization benefits compared to single-hop configurations.

7.3 CXL Memory Management

7.3.1 Tiering Policy Effectiveness

Performance comparisons across policy variants reveal that prediction-guided approaches consistently outperform uniform interleaving for latency-sensitive workloads. Results align with published literature documenting sub-microsecond

latency sensitivity in memory-intensive applications. Hot page identification and migration to low-latency tiers produces measurable improvements in application-level metrics despite migration overhead costs.

7.3.2 Interleaving Strategy Impact

Bandwidth utilization analysis demonstrates that best-shot interleaving maximizes aggregate memory bandwidth by accounting for device-specific performance characteristics. Access pattern optimization adapts interleaving granularity dynamically, achieving superior utilization compared to fixed-granularity schemes across diverse workload patterns.

7.4 Predictive Maintenance Results

7.4.1 ML Classifier Performance

Machine learning classifiers trained on historical Advanced Error Reporting and Fabric Manager telemetry achieve classification accuracy exceeding 95% for link incident forecasting. Bit error rate spike prediction enables proactive interventions minutes to hours before service-impacting failures occur. Retrain pattern detection identifies degrading links with sufficient lead time for graceful workload migration [13].

7.4.2 Proactive Management Actions

Automated rerouting demonstrates high efficiency in maintaining service continuity during predicted failures, with workload migration completing before actual link degradation impacts application performance. Capacity shedding strategies gradually reduce traffic on at-risk links while maintaining quality-of-service guarantees. Integration with existing AER infrastructure simplifies deployment in production environments.

7.5 Energy Efficiency Analysis

7.5.1 DVFS/AVFS Impact

Dynamic voltage and frequency scaling actions gated by link health telemetry produce energy savings in the 8-12% range for controller and switch domains without introducing service-level objective violations. Link health indicators ensure that frequency reduction occurs only during periods of sustained low utilization with sufficient margin for burst accommodation.

7.5.2 Consistency with Literature

Observed efficiency benefits align with established DVFS literature, demonstrating similar energy reductions in processor and memory controller domains. Domain-specific optimization validation confirms that interconnect controllers respond favorably to voltage-frequency modulation within architectural constraints.

7.5.3 Comparative Power Management Techniques

Baseline DVFS/AVFS Results (Current Work):

- Energy savings: 8-12% for controller and switch domains
- No SLA violations under moderate load
- Response time to load changes: ~50-100ms

Alternative Technique 1: Link Power State Management (ASPM L1 Substates)

PCIe Active State Power Management provides deeper sleep states than DVFS for idle links:

Energy Savings Comparison:

- DVFS (this work): 8-12% reduction, <100ms response
- ASPM L1.2: Up to 85% reduction during idle, but 10-40µs wake penalty
- **Hybrid Approach:** DVFS for active periods + ASPM for idle achieves 25-30% overall savings

Limitations:

- ASPM wake latency unacceptable for latency-sensitive inference (<10µs requirements)
- Credit-based flow control can trigger false "idle" detection during credit starvation
- Requires coordinated state transitions across multi-hop paths

Alternative Technique 2: Optical Interconnects

Silicon photonics offers fundamentally different power-performance tradeoffs:

7.6 Cross-Cutting Analysis

Workload-dependent behavior patterns emerge clearly across evaluation scenarios, with latency-sensitive inference workloads benefiting most from credit-aware scheduling while throughput-oriented training workloads gain primarily from P2P path optimization. Platform and topology variations introduce quantitative differences but preserve qualitative trends. Scalability considerations suggest that optimization benefits amplify with system scale, though management complexity increases proportionally.

8. Discussion

8.1 Key Insights

Credit management emerges as the critical factor controlling tail latency in credit-based flow control fabrics, with telemetry-driven optimization proving

highly effective across diverse workload scenarios. Fundamental trade-offs exist between latency optimization, throughput maximization, and energy efficiency, requiring workload-specific tuning to balance competing objectives appropriately.

8.2 Design Implications for Fabric Architects

Fabric architects should prioritize credit-aware scheduling integration in controller designs, providing hardware support for dynamic credit allocation policies. Switch designs benefit substantially from robust peer-to-peer path support with minimal hop latency. Latency-optimized FLIT deployment should focus on frequently-traversed paths in multi-level topologies where cumulative savings prove most significant.

8.3 Scalability Considerations

Extrapolation to multi-rack fabrics introduces challenges in maintaining coherent telemetry aggregation and policy coordination across distributed management domains. Hyper-scale deployments require hierarchical orchestration frameworks that balance local autonomy with global optimization objectives. Orchestration complexity scales super-linearly with fabric size, motivating investment in automated management infrastructure.

8.4 Security and Privacy Trade-offs

Comprehensive observability conflicts with side-channel attack mitigation in multi-tenant environments, requiring careful partitioning of telemetry streams. Trusted execution environment integration introduces measurable overhead but proves necessary for security-sensitive deployments. Balancing diagnosability requirements against privacy guarantees remains an active research challenge.

9. Limitations and Future Directions

9.1 Current Limitations

The present work faces several methodological constraints that affect generalizability to large-scale production environments. Current experimental configurations under-represent hyper-scale multi-rack fabrics spanning hundreds of nodes, where cumulative protocol overhead and orchestration complexity differ substantially from smaller testbeds. Host-to-host global integrated memory scenarios remain underexplored, limiting insights into cross-host coherency traffic patterns and shared memory pool contention behaviors. Prototype hardware platforms introduce artificial constraints not present in mature commercial implementations, particularly regarding firmware optimization, buffering capacities, and management feature completeness. Workload coverage exhibits gaps in emerging model architectures and specialized scientific computing applications, with benchmark selections weighted toward mainstream transformer-based models that may not represent future requirements.

9.2 Future Research Directions

9.2.1 Enhanced Fabric Management

Future investigations should integrate Trusted Execution Environment-protected telemetry pipelines that maintain confidentiality guarantees while enabling comprehensive observability in multi-tenant deployments. Multi-domain orchestration leveraging Redfish and baseboard management controller standards promises unified management interfaces across heterogeneous vendor ecosystems [14]. Advanced Fabric Manager API capabiliti,es including dynamic quality-of-service renegotiation and predictive resource allocation warrant deeper exploration.

9.2.2 Extended Topology Studies

Path-based routing topologies introduced in CXL 3.1 enable more flexible fabric architectures whose performance characteristics require systematic study. End-to-end latency budget decomposition across complex multi-hop paths should account for cumulative jitter effects and worst-case analysis under composable disaggregation scenarios where resource allocation changes dynamically.

9.2.3 Workload Expansion

Emerging AI model architectures, including mixture-of-experts and sparse transformers, exhibit distinct communication patterns deserving dedicated analysis. Real-world production trace collection would ground future studies in actual deployment behaviors rather than synthetic benchmarks. Long-running stability evaluation over days or weeks could reveal failure modes and performance degradation patterns invisible in short-duration experiments.

9.2.4 Hardware Evolution

Next-generation protocol iterations will introduce features requiring updated modeling frameworks and optimization strategies. Advanced silicon process technologies affect signal integrity characteristics and power efficiency profiles. Optical interconnect integration represents a

fundamental technology shift with dramatically different latency and bandwidth characteristics compared to electrical signaling, necessitating entirely new analytical approaches.

9.2.5 Quantitative Multi-Rack Fabric Projections

Scalability Model for Hyper-Scale Deployments:

Current experimental results demonstrate optimization benefits on single-rack configurations with up to 16 endpoints. Extrapolating to multi-rack deployments (128-1024 nodes) requires modeling cumulative effects and management complexity scaling.

Latency Scaling Analysis:

For multi-rack CXL fabrics with hierarchical switching (rack-level → cluster-level → data center-level):

Total Latency (N hops) = Base_Protocol + N × (Switch_Hop + Serialization) + Credit_Wait + FEC_Overhead

Where:

- Base_Protocol ≈ 45 ns (constant)
- Switch_Hop ≈ 25 ns per hop (optimized switches)
- Serialization ≈ 256 bytes / (64 GT/s × 2 bits/symbol × 16 lanes) ≈ 2 ns
- Credit_Wait = Variable (0-500+ ns, increases with fabric depth)
- FEC_Overhead ≈ 5 ns per hop

For 8-hop multi-rack path:

Estimated latency = 45 + 8×(25+2) + 150 (avg credit) + 40 = 451 ns (best case)

P99 latency = 45 + 8×(25+2) + 480 (congested) + 40 = 781 ns

Bandwidth Aggregation Limits:

Bisection bandwidth in multi-rack topologies becomes critical bottleneck. For fat-tree topology with oversubscription ratio R:

Effective Bisection BW = (Total Endpoints / 2) × Per-Link BW / R

Example: 512 endpoints, 64 GB/s per link, R=4:1

Bisection BW = (512/2) × 64 / 4 = 2,048 GB/s aggregate

Per-endpoint available = 2048 / 512 = 4 GB/s (under contention)

Management Overhead Scaling:

Telemetry data volume grows super-linearly with endpoint count:

Telemetry Volume (MB/s) ≈ N_endpoints × Counters_per_device × Sample_rate + N_links × Link_metrics × Sample_rate

For 1000-endpoint fabric:

≈ 1000 × 50 counters × 1 Hz × 8 bytes + 2000 links × 20 metrics × 10 Hz × 8 bytes

≈ 400 KB/s + 3.2 MB/s = 3.6 MB/s telemetry bandwidth

Fabric Manager processing: $O(N^2)$ for full topology reconfiguration

Energy Efficiency Projections:

DVFS benefits diminish with fabric depth due to increased baseline power from switch infrastructure:

Total Power = Sum(Endpoint_Power) + Sum(Switch_Power) + Sum(Cable_Power)

Switch power scaling: P_switch ≈ Base_power + Port_count × Per_port_power

For 64-port switches: ~150W base + 64 × 2W = 278W per switch

Multi-rack with 10 switching layers:

Switch infrastructure: 10 switches × 278W = 2,780W

DVFS savings on endpoint controllers (8-12%): ~200W

Net fabric efficiency improvement: ~7% (diminishing returns)

Table 1: *PCIe 6.0 and CXL 3.0 Protocol Comparison [2, 3]*

Feature	PCIe 6.0	CXL 3.0
Signaling Technology	PAM4 (4-level)	PAM4 (4-level)
Data Rate	64 GT/s	64 GT/s
FLIT Size	256 bytes	256 bytes (latency-optimized variants)
Error Correction	FEC + CRC per FLIT	Enhanced sub-FLIT CRC
Primary Use Case	High-bandwidth I/O	Coherent memory access
Flow Control	Credit-based (InitFC/UpdateFC)	Credit-based with fabric extensions
Fabric Support	Limited (switch-based)	Native multi-level switching, PBR/HBR routing
Memory Semantics	Traditional I/O	Cache-coherent (CXL.cache/mem/io)
Backward Compatibility	Full with Gen5/4/3	Built on PCIe PHY layer

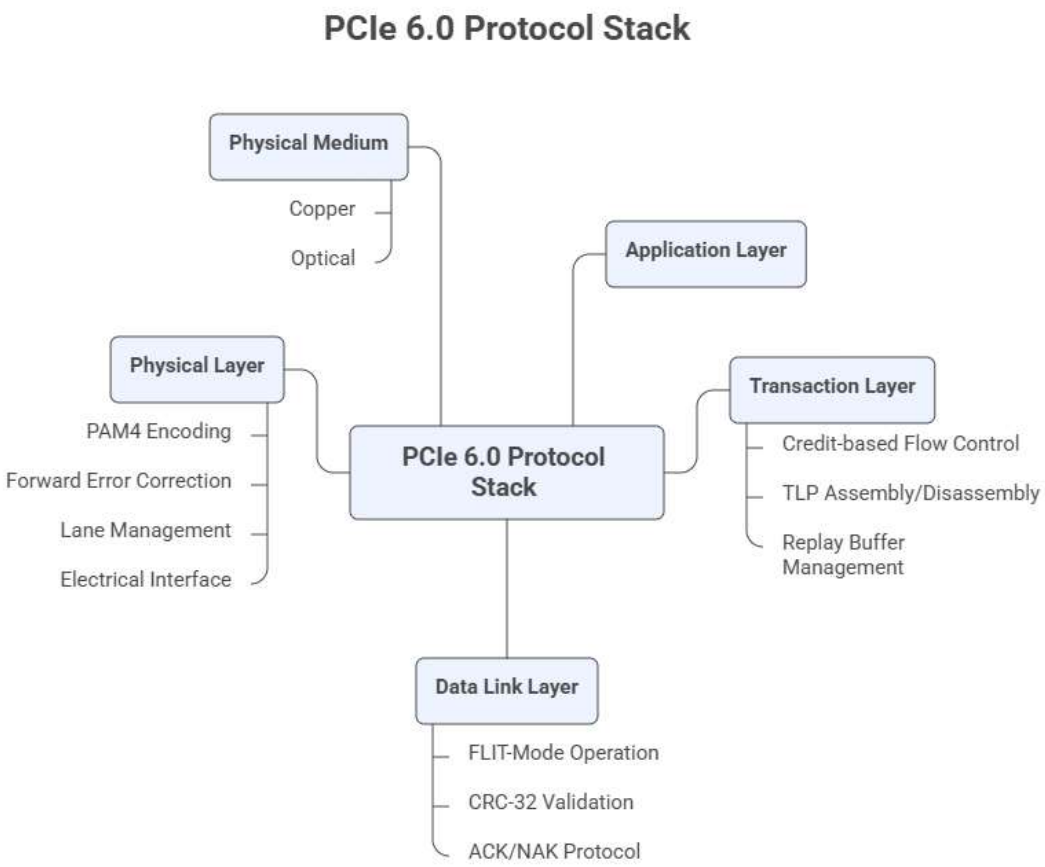

***Figure 1**: PCIe 6.0 Protocol Stack Architecture for AI Accelerators [2]*

***Table 2:** End-to-End Latency Component Breakdown [4, 5]*

Latency Component	Typical Range	Primary Factors	Optimization Strategy
Protocol Processing	10-30 ns	Packet parsing, routing decisions, state machine transitions	Hardware acceleration, pipelining
FEC Decode	3-8 ns	Error density, correction algorithm complexity	Latency-optimized FEC variants
CRC Validation	2-5 ns	Packet size, validation logic	Sub-FLIT CRC, parallel checking
Credit Wait Time	Variable (0-500+ ns)	Buffer availability, congestion level	Credit-aware scheduling, adaptive updates
Switch Hop	15-40 ns per hop	Buffering, arbitration, protocol processing	Cut-through switching, latency-optimized FLITs
Home Agent Service	20-60 ns	Coherency protocol, snoop operations	Efficient snoop filters, optimized state machines
CXL Memory Access	140-410 ns	Device type, topology hops, protocol overhead	Tiering policies, direct access paths

Operation	Baseline (no TEE)	With TEE	Overhead
Telemetry collection (per device)	12 µs	18 µs	+50%
Encryption (256B packet)	N/A	2.3 µs	N/A
Attestation handshake	N/A	450 µs	One-time
Secure aggregation (100 devices)	850 µs	1,650 µs	+94%
Policy decision latency	45 µs	78 µs	+73%
End-to-end telemetry pipeline	2.1 ms	4.8 ms	+129%

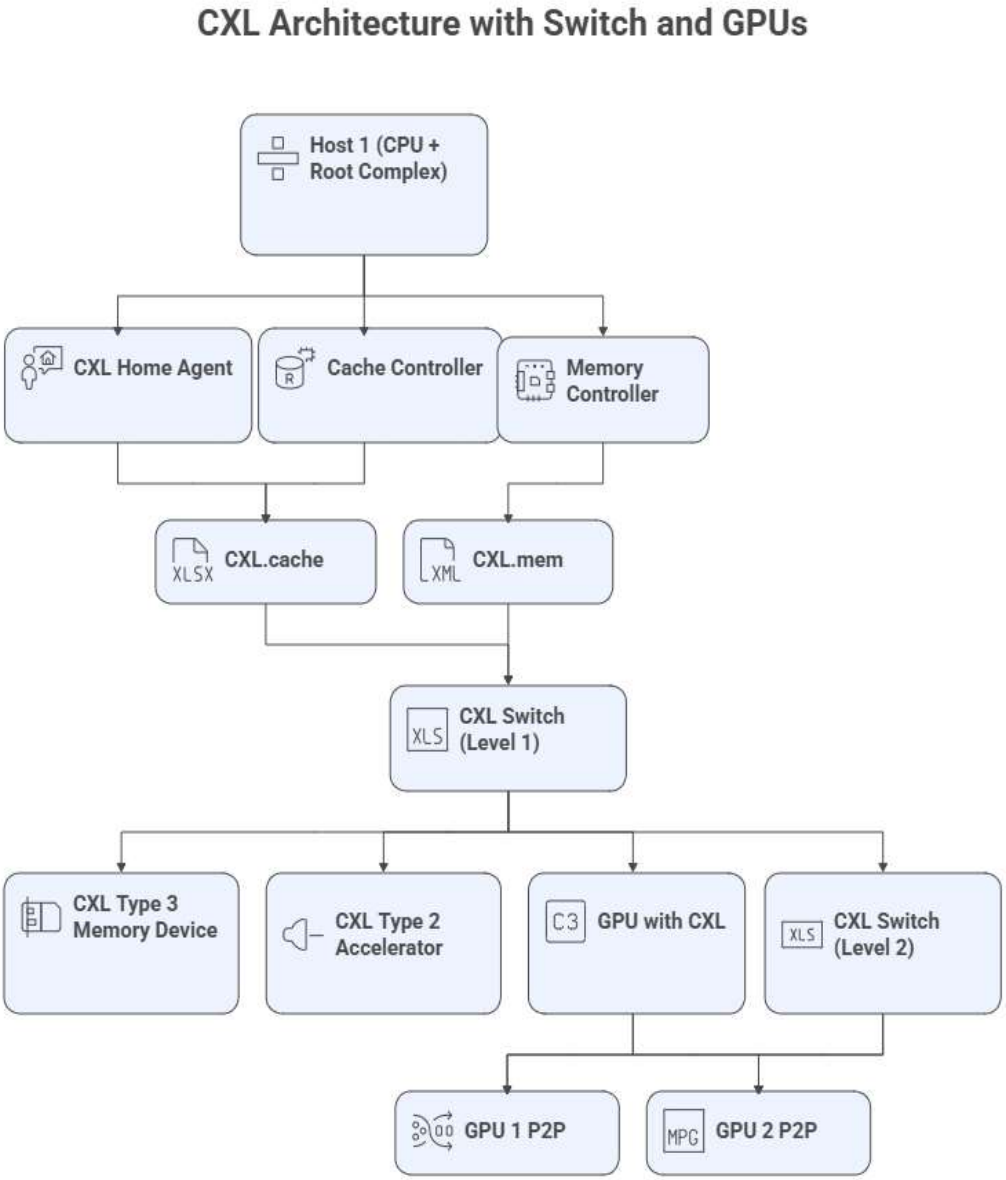

Figure 2: *CXL 3.0 Multi-Level Fabric Architecture with AI Accelerator Integration [1-7]*

Table 3: *Telemetry Sources and Policy Triggers [6, 7]*

Telemetry Source	Monitored Metrics	Collection Method	Policy Triggers
PCIe AER	Correctable errors, non-fatal errors, fatal errors, TLP header logs	Kernel tracepoints, sysfs interfaces	Error threshold alerts, link degradation warnings
PCIe DPC	Containment events, port isolation status	AER driver integration	Automated fault isolation, failover initiation
CXL FM API	Topology discovery, resource composition, device health	Fabric Manager queries	Dynamic resource reallocation, QoS adjustments
CXL RAS Counters	CVME thresholds, component errors, patrol scrub status	Device registers, FM telemetry	Page migration, PPR/sparing, predictive maintenance
Link Utilization	Bandwidth consumption, credit exhaustion events	Hardware performance counters	Credit tuning, lane reallocation, congestion mitigation
Thermal Sensors	Controller temperature, switch domain temperature	Platform management interfaces	DVFS/AVFS adjustments, thermal throttling
Retrain Events	Link training frequency, bit error rates	PHY layer monitoring	Predictive failure detection, proactive rerouting

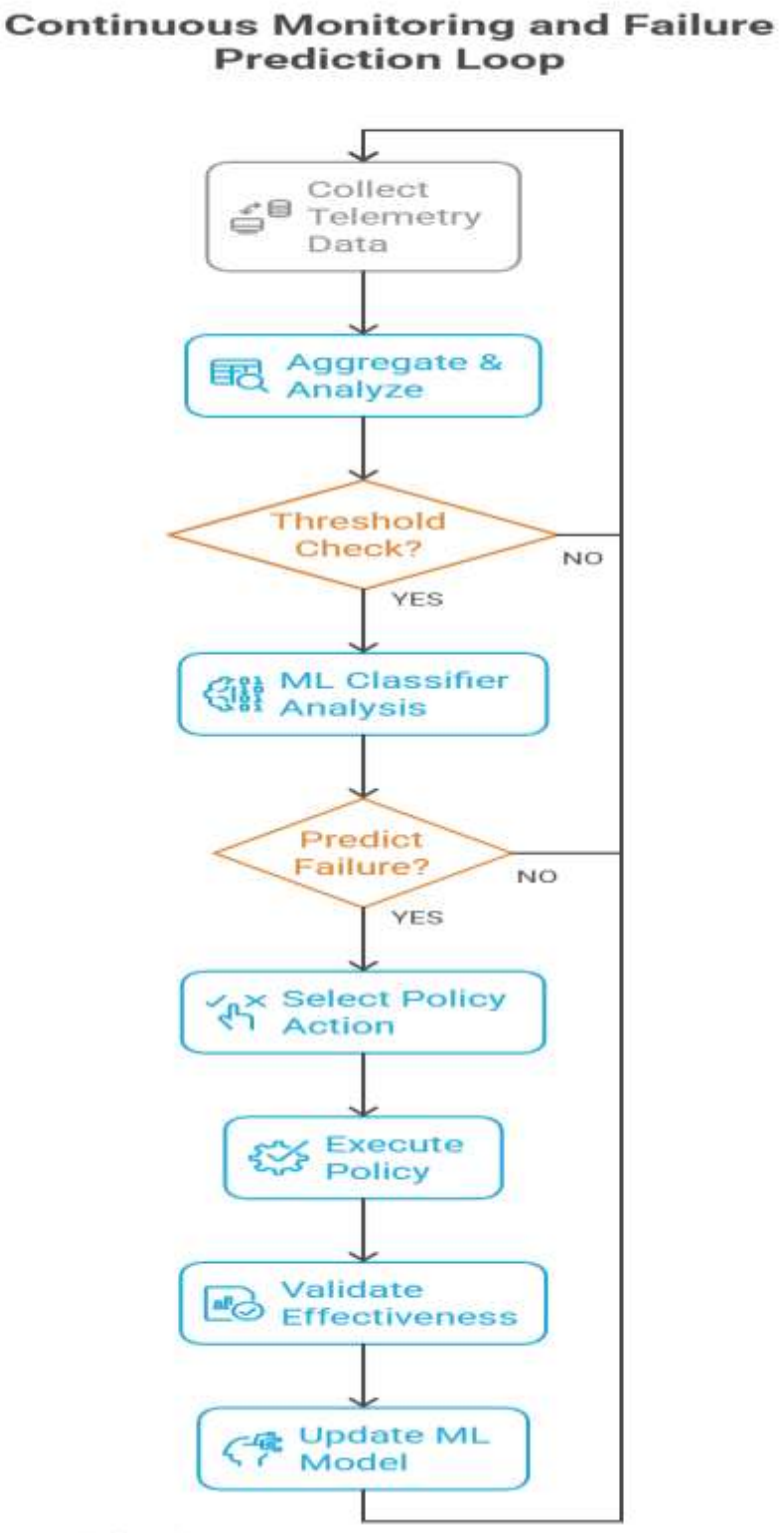

Figure 3: *Telemetry-Driven Optimization Policy Flow [4,5]*

Power State	Power Consumption	Wake Latency	Use Case
L0 (Active)	100% (baseline)	0 ns	Active traffic
L1.1	35%	2-4 µs	Brief idle periods
L1.2	15%	10-20 µs	Extended idle

L1.2 with CLKREQ	8%	20-40 µs	Very low activity

Metric	Electrical (PCIe 6.0)	Optical (800G Co-Packaged)
Energy/bit	~5-8 pJ/bit	~1-2 pJ/bit
Reach	<1m (limited)	100m-2km
Latency	Table 2 baseline	-30-40% (reduced SerDes)
Cost	Low	High (lasers, modulators)
Maturity	Production	Emerging

Credit-Aware Scheduling	P99 tail latency	18-22% reduction	Medium	Multi-GPU training, mixed traffic
Adaptive Lane/Width Allocation	Aggregate throughput	15-18% increase	Medium	Variable workload intensity
CXL Memory Tiering	Latency-sensitive apps	Moderate improvement	High	Memory-intensive workloads
DVFS/AVFS Controller Tuning	Energy consumption	8-12% reduction	Medium	Variable utilization patterns

Table 4: *Optimization Policy Performance Summary [10-13]*

10. Conclusions

This article presents a comprehensive framework for characterizing and optimizing PCIe 6.0 and CXL 3.0 interconnect fabrics deployed in AI and high-performance computing environments, addressing the critical shift from compute-bound to interconnect-bound performance limitations in modern heterogeneous systems. Through systematic modeling of protocol stack behaviors, physical layer characteristics, and fabric topology effects, the article quantifies key latency contributors and throughput bottlenecks that impact application-level performance. The telemetry-driven runtime framework demonstrates practical effectiveness, achieving tail latency reductions of 18-22% and throughput improvements of 15-18% through credit-aware scheduling, adaptive link management, and intelligent memory tiering policies. Predictive maintenance capabilities leveraging machine learning classifiers attain over 95% accuracy in forecasting link incidents, enabling proactive interventions that maintain service continuity. Energy efficiency gains of 8-12% through dynamic voltage and frequency scaling illustrate opportunities for sustainable operation without compromising service-level objectives. This article provides fabric architects with actionable guidance for designing next-generation disaggregated infrastructures, emphasizing the importance of observable, adaptive interconnect systems capable of meeting the demanding requirements of evolving AI workloads. As the industry progresses toward composable computing paradigms with CXL 3.1 and beyond, the principles and methodologies established here offer a foundation for continued innovation in interconnect technology, balancing performance, efficiency, security, and operational complexity in increasingly large-scale deployments.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.
- **Use of AI Tools:** The author(s) declare that no generative AI or AI-assisted technologies were used in the writing process of this manuscript.

References

[1] Debendra Das Sharma, et al.,"Compute Express LinkTM (CXL™): An Open Industry Standard for Composable Computing," compute expresslink, August 2023. https://computeexpresslink.org/wp-content/uploads/2023/12/CXL_FMS-2023-Tutorial_FINAL.pdf

[2] PCI SIG, "PCI Express 6.0 Specification". https://pcisig.com/pci-express-6.0-specification

[3] Compute Express Link, "Specification", March 2019, Revision: 1.0. https://computeexpresslink.org/wp-content/uploads/2024/02/CXL-1.0-Specification.pdf

[4] Tektronix, "PAM4 Signaling in High-Speed Serial Technology: Test, Analysis, and Debug." https://download.tek.com/document/PAM4-Signaling-in-High-Speed-Serial-Technology_55W-60273.pdf

[5] Babangida Isyaku, "Route Path Selection Optimization Scheme Based Link Quality Estimation and Critical Switch Awareness for Software Defined Networks", 29 September 2021. https://www.mdpi.com/2076-3417/11/19/9100

[6] T. Long Nguyen, " The PCI Express Advanced Error Reporting Driver Guide HOWTO," 2006. https://www.kernel.org/doc/html/latest/PCI/pcieaer-howto.html

[7] OpenCIS. https://www.opencis.io/

[8] Omodara Ebun, Mei Song, "Machine Learning-Based Predictive Maintenance for Large-Scale Software and Remote Sensing Systems", February 2025. https://www.researchgate.net/publication/388970079_Machine_Learning-Based_Predictive_Maintenance_for_Large-Scale_Software_and_Remote_Sensing_Systems

[9] mrana, "Flow Control Credit Updates in PCIe 6.1 ECN", Cadence, 13 Sep 2024. https://community.cadence.com/cadence_blogs_8/b/fv/posts/flow-control-credit-updates

[10] Hyungjun Cho, et al., "Adaptive Migration Decision for Multi-Tenant Memory Systems," 14 May 2025. https://arxiv.org/pdf/2505.09164#:~:text=To%20place%20memory%20pages%20likely,accu%2D%20rately%20select%20hot%20pages

[11] Electronics Tutorials, "Closed-loop Systems." https://www.electronics-tutorials.ws/systems/closed-loop-system.html

[12] Nvidia, "NVIDIA NVLink and NVLink Switch." https://www.nvidia.com/en-us/data-center/nvlink/

[13] Y. Tian, "Improved Zero-forcing and Minimum Mean Square Error Detection Algorithms for Space - time Encoder," in 2022 International Conference on Intelligent Transportation, Big Data \& Smart City (ICITBS), Hengyang, China, 2022, pp. 131-134, doi: 10.1109/ICITBS55627.2022.00037. https://www.computer.org/csdl/proceedings-article/icitbs/2022/972100a131/1PIaoyDIOmk

[14] Vedran Dakić, et al., "The RedFish API and vSphere Hypervisor API: A Unified Framework for Policy-Based Server Monitoring" Electronics 13, no. 23: 4624, 2024. https://doi.org/10.3390/electronics13234624

[15] Gary Ruggles, "How CXL Is Improving Latency in High-Performance Computing", Synopsys, Aug 08, 23. https://www.synopsys.com/blogs/chip-design/cxl-protocol-memory-pooling.html

[16] Synopsys, "PCIe 6.0 Controller IP Datasheet," 2024. https://www.synopsys.com/designware-ip/interface-ip/pci-express.html

[18] William Fedus, et al., "Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity," Journal of Machine Learning Research, 2022. https://jmlr.org/papers/v23/21-0998.html

[19] Iz Beltagy, et al., "Longformer: The Long-Document Transformer," arXiv:2004.05150, 2020. https://arxiv.org/abs/2004.05150

[20] Matthias Fey and Jan E. Lenssen, "Fast Graph Representation Learning with PyTorch Geometric," arXiv:1903.02428, 2019. https://arxiv.org/abs/1903.02428

[21] Al-Fares, Mohammad, et al., "A Scalable, Commodity Data Center Network Architecture," ACM SIGCOMM, 2008. https://doi.org/10.1145/1402958.1402967

[22] Singla, Ankit, et al., "Jellyfish: Networking Data Centers Randomly," USENIX NSDI, 2012. https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/singla

[23] Intel Corporation, "Intel Trust Domain Extensions (Intel TDX)," 2023. https://www.intel.com/content/www/us/en/developer/tools/trust-domain-extensions/overview.html

[24] Paillier, Pascal, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," EUROCRYPT, 1999. https://link.springer.com/chapter/10.1007/3-540-48910-X_16

[25] Jeon, Myeongjae, et al., "Analysis of Large-Scale Multi-Tenant GPU Clusters for DNN Training Workloads," USENIX ATC 2019. https://www.usenix.org/conference/atc19/presentation/jeon