SOCIAL AND MANAGEMENT SCIENCES RESEARCH JOURNAL

www.smsrj.com

p-ISSN: 2791-0415 **e-ISSN:** 2791-0423

Volume:4 Issue: 02(2026)

Reinforcement Learning for RDMA Traffic Scheduling in Distributed Data Centers

[1]Christopher Poon, [2]Rebecca Tse

[1]Department of Electrical and Electronic Engineering, Faculty of Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China

[2]School of Data Science, Lingnan University, Hong Kong 999077, China

Abstract:

Modern distributed data centers increasingly rely on Remote Direct Memory Access technologies to achieve high throughput and microsecond scale latency, particularly for demanding workloads such as large language model training and distributed storage operations. Conventional congestion control and traffic scheduling mechanisms, largely reliant on heuristic thresholds and reactive priority based flow control, struggle to handle the highly dynamic, bursty traffic patterns inherent in scale out architectures. These traditional systems frequently experience structural inefficiencies, including priority based flow control deadlocks, congestion spreading, and suboptimal link utilization. To address these limitations, this paper proposes a novel multi-agent reinforcement learning framework for dynamic Remote Direct Memory Access traffic scheduling in spine leaf data center networks. By formulating the network scheduling challenge as a decentralized Markov Decision Process, the proposed system enables intelligent edge switches and network interface cards to autonomously optimize rate limiting and priority marking based on real time telemetry data. The methodology leverages a multi-agent proximal policy optimization algorithm, which stabilizes training in highly concurrent environments while maximizing global network utility. Extensive evaluations conducted in packet level simulators demonstrate that the reinforcement learning based approach significantly reduces flow completion times for delay sensitive mice flows while maintaining high throughput for background elephant flows. Furthermore, the intelligent scheduling policy successfully mitigates queue buildup and eliminates the risk of cascading flow control pauses. The results underscore the potential of deploying continuous learning algorithms to manage the growing complexity of low latency data center interconnects.

Keywords: *Reinforcement Learning, Remote Direct Memory Access, Data Center Networks, Traffic Scheduling.*

1. Introduction

1.1 The Evolution of Data Center Networking

The rapid proliferation of cloud computing, edge service architectures, and high performance computing has precipitated a fundamental transformation in data center networking over the past decade. Distributed applications, including real time artificial intelligence inference, deep learning model training, and in memory distributed databases, demand network architectures capable of delivering unprecedented bandwidth and strictly bounded latency [1]. Traditional networking stacks based on the Transmission Control Protocol and the Internet Protocol inherently incur significant central processing unit overhead due to context switching, memory copying, and interrupt handling [2]. As link speeds escalate from forty gigabits per second to four hundred gigabits per second and beyond, the processing overhead of the standard software network stack becomes a primary bottleneck, preventing applications from fully utilizing the available hardware capacity [3]. This discrepancy has driven the ubiquitous adoption of Remote Direct Memory Access protocols in modern hyperscale data centers [4]. Remote Direct Memory Access bypasses the operating system kernel, allowing network adapters to transfer data directly into application memory spaces, thereby minimizing latency and CPU utilization. However, the widespread deployment of Remote Direct Memory Access over Converged Ethernet presents a unique set of traffic management challenges [5]. Unlike standard Transmission Control Protocol traffic, which robustly handles packet loss through sophisticated window adjustments and retransmission timers, Remote Direct Memory Access over Converged Ethernet is highly sensitive to network drops. To maintain high performance, it requires a virtually lossless networking environment [6]. This lossless requirement is traditionally satisfied by employing Priority based Flow Control, an Ethernet standard that allows a downstream receiver to pause upstream transmission when its buffers cross a predefined threshold [7]. While Priority based Flow Control prevents packet drops, it introduces severe systemic vulnerabilities, most notably the phenomenon of congestion spreading and deadlocks [8]. When a single switch port becomes congested, it pauses its upstream neighbor, which may in turn pause its own upstream neighbors, leading to a cascading network freeze that degrades overall data center throughput.

1.2 Limitations of Current Scheduling Approaches

To mitigate the adverse effects of Priority based Flow Control, researchers and engineers have proposed various congestion control algorithms tailored for Remote Direct Memory Access environments [9]. Prominent heuristic solutions include Data Center Quantized Congestion Notification and Timely, which rely on explicit congestion notification marks and round trip time measurements, respectively, to modulate the sending rates of network interface cards. Although these algorithms provide significant improvements over basic flow control, they rely heavily on static parameter configurations and heuristic rules designed for average case scenarios [10]. The highly bursty nature of modern data center traffic, characterized by microsecond scale incast patterns and simultaneous elephant flows from storage replication, frequently overwhelms these static heuristics. Administrators are forced into a constant cycle of manual parameter tuning, attempting to balance the trade off between maximizing throughput and minimizing queue buildup [11]. Consequently, there is a pressing need for adaptive, self optimizing scheduling frameworks capable of responding to the highly dynamic state of the network in real time. Reinforcement learning has emerged as a promising paradigm for solving complex, high dimensional control problems where traditional mathematical modeling falls short [12]. By treating network devices as intelligent agents that continuously interact with the environment, reinforcement learning can discover non intuitive scheduling policies that dynamically adjust traffic rates based on comprehensive telemetry, including queue depths, link utilization, and historical congestion trends. This paper addresses the fundamental challenge of applying

reinforcement learning to Remote Direct Memory Access traffic scheduling by proposing a scalable, multi-agent architecture designed specifically for the rigorous demands of distributed data centers [13].

2. Background on RDMA and Distributed Systems

2.1 Remote Direct Memory Access Mechanisms

Remote Direct Memory Access fundamentally alters the paradigm of host to host communication. In a conventional network transfer, data must traverse multiple layers of the operating system kernel, requiring context switches between user space and kernel space, as well as multiple memory copies between application buffers and socket buffers [14]. Remote Direct Memory Access bypasses this entirely through hardware offloading on the Network Interface Card. The application registers a region of its memory with the network card, and subsequent read or write operations are executed directly by the hardware. This zero copy mechanism reduces the end to end latency to the order of a few microseconds, making it highly desirable for distributed consensus protocols and tightly coupled parallel computing tasks [15]. There are three primary implementations of Remote Direct Memory Access: InfiniBand, Internet Wide Area Remote Direct Memory Access Protocol, and Remote Direct Memory Access over Converged Ethernet. InfiniBand is a purpose built network architecture that requires specialized switches and cables, offering native lossless transmission at the link layer [16]. While InfiniBand provides exceptional performance, its lack of interoperability with standard Ethernet infrastructure limits its widespread adoption in general purpose cloud data centers. Conversely, Remote Direct Memory Access over Converged Ethernet, particularly version two, encapsulates the transport protocol within standard User Datagram Protocol and Internet Protocol headers, allowing it to traverse traditional Ethernet switches and routers [17]. This convergence reduces capital expenditure and simplifies network management, but it places the burden of ensuring lossless transmission on the underlying Ethernet configuration.

Table 1Comparison of Data Center Network Protocol Overheads

Protocol Type	Transport Layer	CPU Utilization	Memory Copying	Base Latency
In Kernel TCP	Standard TCP	High	Multiple Copies	Milliseconds
Kernel Bypass	Custom UDP	Medium	Single Copy	Microseconds
InfiniBand	Native IB	Minimal	Zero Copy	Sub Microsecond
RoCEv2	UDP Encapsulated	Minimal	Zero Copy	Microsecond

2.2 The Congestion Management Dilemma

The implementation of RoCEv2 necessitates the configuration of Data Center Bridging protocols on all intervening switches [18]. Priority based Flow Control divides the physical Ethernet link into multiple virtual lanes, each mapped to a specific priority level. When the buffer allocated to a specific virtual lane reaches a critical threshold, the switch transmits a pause frame to the upstream device, halting further transmission on that specific priority class until the buffer drains [19]. This mechanism ensures that packets are not dropped due to buffer overflow, which is critical because the standard Remote Direct Memory Access transport protocol uses a go back N retransmission strategy. A single dropped packet can trigger the retransmission of an entire sequence of data, severely penalizing throughput [20].

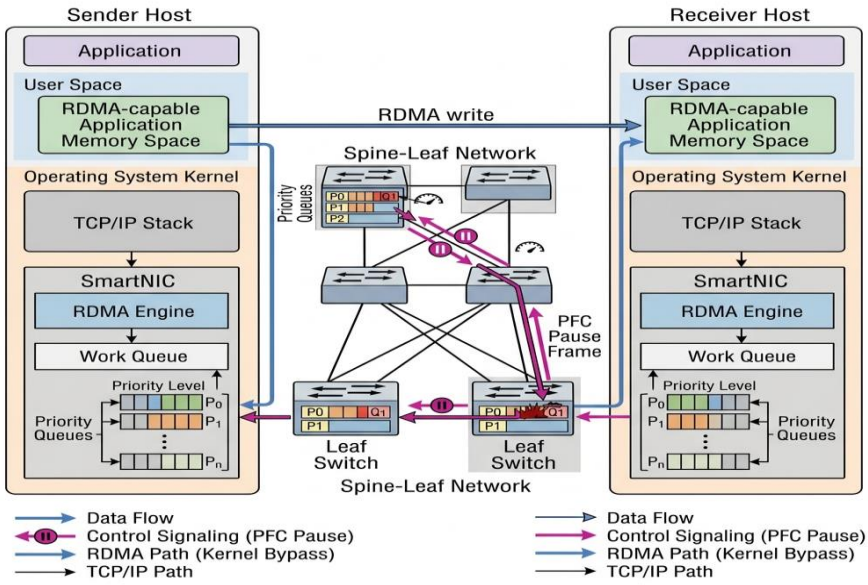


Figure 1: Architecture of RDMA in Modern Data Center Networks

However, Priority based Flow Control introduces severe collateral damage during periods of persistent congestion. If multiple flows converge on a single egress port, an event known as an incast, the buffer fills rapidly. The switch emits pause frames to all ingress ports contributing to the congestion [21]. Those ingress ports, now unable to forward traffic, quickly fill their own buffers and emit pause frames to their upstream connections. This phenomenon, congestion spreading, can quickly degrade the performance of the entire network fabric, affecting flows that do not even traverse the initially congested link. Furthermore, if a cyclic dependency forms in the routing topology, a Priority based Flow Control deadlock occurs, permanently halting all traffic on the affected virtual lanes until the network administrator intervenes or a watchdog timer drops the stalled packets, violating the lossless guarantee [22].

3. Literature Review and Related Work

3.1 Heuristic Congestion Control Protocols

The networking research community has invested heavily in developing congestion control algorithms specifically for RoCEv2 environments to prevent the invocation of Priority based Flow Control. Data Center Quantized Congestion Notification is the most widely deployed standard, implemented directly in the hardware of modern network interface cards. It utilizes Explicit Congestion Notification marks generated by switches to signal impending congestion. When the queue depth at a switch exceeds a configured threshold, the switch probabilistically marks packets [23]. The receiver echoes these marks back to the sender in a notification packet, prompting the sender to reduce its transmission rate. While effective in steady state scenarios, the probabilistic nature of the marking and the reactive rate adjustment often fail to respond quickly enough to microbursts, leading to buffer overflows or unnecessary rate throttling [24].

Alternative heuristic approaches, such as Timely and Swift, leverage round trip time measurements rather than explicit switch feedback. These delay based algorithms continuously monitor the latency of data packets and acknowledgments. An increase in round trip time indicates queuing delay in the network, prompting a reduction in the sending rate. The primary advantage of delay based methods is that they do not require specific switch configurations or hardware support for explicit marking [25]. However, in highly utilized multi tenant data centers, baseline network latency can fluctuate due to background processing overheads on shared

switches, causing delay based algorithms to misinterpret transient latency spikes as persistent congestion, thereby artificially suppressing throughput [26].

3.2 Machine Learning in Network Scheduling

Recognizing the limitations of static heuristics, recent literature has explored the application of machine learning techniques for network traffic optimization. Early efforts focused on supervised learning approaches, where historical traffic traces were used to train neural networks to predict congestion events and pre-allocate bandwidth. While these predictive models demonstrated theoretical improvements, their reliance on offline training data made them brittle when deployed in operational networks facing novel traffic patterns not represented in the training set. Supervised models struggle to adapt to the inherent non stationarity of data center workloads, where flow size distributions and arrival rates change dramatically throughout the day [27]. Reinforcement learning offers a more resilient alternative by framing traffic scheduling as an active decision making process. In a reinforcement learning setup, an agent continuously observes the network state, executes scheduling actions, and receives a numerical reward based on the resulting performance. Prior works have attempted to apply single agent deep reinforcement learning to centralized software defined networking controllers to optimize routing paths. However, centralized reinforcement learning architectures suffer from high communication overhead and significant latency between state observation and action execution, rendering them unsuitable for the microsecond timescale required by Remote Direct Memory Access traffic. Distributed, multi-agent reinforcement learning approaches are required to push the decision making process to the edge of the network, enabling localized, rapid responses to congestion while maintaining cooperative global optimization.

4. System Architecture and Problem Formulation

4.1 Network Topology and Agent Deployment

This paper proposes a decentralized traffic scheduling architecture deployed over a standard spine leaf data center topology. The spine leaf architecture, characterized by dense interconnectivity and uniform latency between any two end hosts, provides a highly scalable foundation for distributed computing. The proposed system deploys independent reinforcement learning agents at two distinct locations within the network infrastructure: on the programmable network interface cards of the end hosts, and on the edge switches at the leaf layer. This dual deployment strategy addresses both the source rate limitation and the in network traffic prioritization. The network is modeled as a directed graph where the set of vertices represents the end hosts, leaf switches, and spine switches, and the set of edges represents the physical interconnects. The traffic demand is represented by a set of flows, where each flow originates from a source host and terminates at a destination host, carrying a specific volume of data. The primary responsibility of the host based agents is to determine the optimal injection rate for each outgoing flow, while the switch based agents are responsible for dynamically mapping flows to specific priority queues based on their observed characteristics.

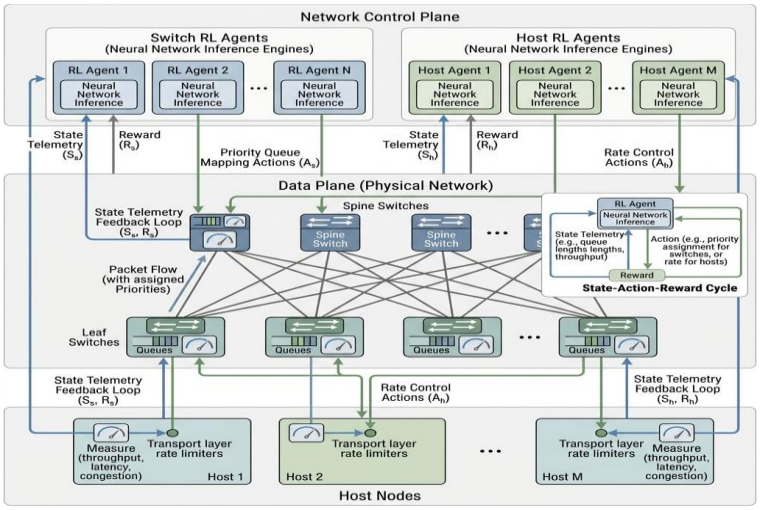


Figure 2The Multi Agent Reinforcement Learning Framework Deployed on a Spine Leaf Switch Architecture

4.2 Mathematical Modeling of Traffic Scheduling

To formalize the optimization objective, we define a comprehensive utility function that balances the competing requirements of different traffic classes. Data center traffic typically consists of two distinct categories: delay sensitive mice flows, such as remote procedure calls and control messages, and throughput sensitive elephant flows, such as virtual machine migrations and deep learning checkpoint synchronizations. The scheduling algorithm must minimize the flow completion time of mice flows while maximizing the aggregate throughput of elephant flows.

Let the total number of flows active in the network at time interval be denoted by an integer index. For each flow, the utility is defined as a function of its current throughput and its accrued latency. The global objective function that the reinforcement learning system seeks to maximize over an extended time horizon is expressed mathematically as follows.

$$max\sum_{t=0}^{T}\sum_{i=1}^{N}(\alpha \cdot log(U_{i,t}+1) - \beta \cdot \frac{L_{i,t}}{S_i})$$

In this equation, the logarithmic term represents the proportional fairness of the throughput allocated to flow at a given time step, ensuring that bandwidth is distributed equitably among large flows without starving smaller ones. The latency term represents the normalized delay, where the latency is divided by the total size of the flow to emphasize the impact of delay on short flows. The coefficients alpha and beta are hyper parameters that allow the system administrator to dictate the relative importance of throughput maximization versus latency minimization based on the specific requirements of the data center tenant.

Table 2 State and Action Space Definitions for Edge and Host Agents

Component Type	State Variables Observed	Action Variables Executed	Decision Frequency
Host Based Agent	Local Queue, ECN Marks, RTT	Flow Rate Limit	Per Millisecond
Leaf Switch Agent	Port Utilization, Buffer Depth	Queue Priority Mapping	Per Millisecond
Spine Switch Agent	Global Telemetry Summary	Load Balancing Weights	Per Ten Milliseconds
Central Controller	Historic Traffic Patterns	Reward Hyper parameters	Per Minute

5. Reinforcement Learning Methodology for Traffic Scheduling

5.1 State Action and Reward Definitions

The formulation of the Markov Decision Process is the most critical step in designing an effective reinforcement learning system for network control. The state space must capture sufficient information for the agent to infer the current degree of congestion without incurring prohibitive telemetry gathering overhead. For the host based agents, the local state observation vector includes the current transmission rate of all active flows, the proportion of recent packets that returned with explicit congestion notification marks, the smoothed round trip time, and the local transmission queue depth. For the switch based agents, the state vector comprises the egress link utilization for all active ports, the instantaneous buffer occupancy of each priority queue, and the rate of incoming explicit congestion notification marks from upstream spine switches. The action space for the host agent is a continuous variable representing the scaling factor applied to the current transmission rate. Rather than outputting an absolute rate, the agent outputs a multiplier, allowing for smoother adjustments and better generalization across different network speeds. The action space for the switch agent is discrete, representing the mapping of incoming packets to one of the available priority queues based on their differentiated services code point values.

The reward function must align with the global optimization objective formulated previously while providing dense, immediate feedback to facilitate rapid convergence. The instantaneous reward computed at each time step for a given agent incorporates penalties for excessive queue buildup and packet drops, which serve as proxies for latency and network instability. The reward formulation is detailed below.

$$R_t = -\gamma \cdot max(0, Q_t - Q_{target}) - \delta \cdot P_{drop} + \omega \cdot Throughput_t$$

Here, the queue deviation penalty strictly penalizes the agent only when the queue depth exceeds a predefined target threshold, encouraging the agent to maintain a small, standing queue to ensure high link utilization without incurring unnecessary queuing delay. The packet drop penalty is heavily weighted to enforce the virtually lossless requirement of the Remote Direct Memory Access environment.

5.2 The Actor Critic Algorithm Design

Given the continuous nature of the rate control action space and the highly concurrent, multi-agent environment, traditional value based methods like Deep Q Networks are insufficient. This methodology employs a Multi Agent Proximal Policy Optimization architecture. Proximal Policy Optimization is an actor critic method that uses a surrogate objective function to prevent destructively large policy updates, ensuring stable and monotonic improvement during the training phase. Each agent maintains two deep neural networks: an actor network that maps state observations to an action probability distribution, and a critic network that estimates the value function of the current state. In the multi agent variant, the system leverages a centralized training with decentralized execution paradigm. During the training phase, the critic network is provided with the global state of the entire network, allowing it to accurately assess the value of an agent's action in the context of the actions taken by all other agents. During deployment, the actor network operates solely on local state observations, enabling rapid, distributed inference without the latency of communicating with a central controller. To ensure stability during training, the Proximal Policy Optimization algorithm relies on a clipped objective function. This clipping mechanism restricts the ratio of the new policy to the old policy, preventing the actor network from deviating too far from its previous behavior in a single update step. The mathematical formulation of the clipping objective is crucial for maintaining the delicate balance of network traffic.

$$L^{CLIP} = mathbbE[min(r_t(\theta)A_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)A_t)]$$

In this expression, the probability ratio represents the likelihood of taking a specific action under the current policy relative to the previous policy. The advantage estimate evaluates how much better a specific action is compared to the average action in a given state. The clipping parameter prevents excessive policy updates that could lead to catastrophic network instability, such as a synchronized collapse where all agents simultaneously reduce their sending rates to zero in response to a minor congestion event.

6. Experimental Evaluation and Results Analysis

6.1 Simulation Environment Setup

To rigorously evaluate the proposed multi agent reinforcement learning scheduling framework, we implemented a comprehensive simulation environment utilizing the widely adopted NS three network simulator. The simulator was extended to support accurate modeling of the RoCEv2 protocol suite, including detailed implementations of the Priority based Flow Control mechanism, explicit congestion notification marking at the switch level, and the precise timing of hardware network interface cards. The reinforcement learning agents were implemented using the Ray RLlib framework and bridged to the network simulator through a low latency inter process communication channel, allowing the neural networks to consume telemetry and push policy actions at sub millisecond intervals. The simulated topology consists of a two tier spine leaf architecture containing four spine switches and eight leaf switches, connecting a total of sixty four end host servers. This scale is sufficient to generate complex, multi path routing dynamics and localized congestion hotspots. The access links connecting the servers to the leaf switches are configured at one hundred gigabits per second, while the core links connecting the leaf switches to the spine switches operate at four hundred gigabits per second, representing a standard oversubscription ratio commonly found in production environments.

Table 3Simulation Environment and Network Hyper parameters

Parameter Category	Specific Variable	Configured Value
Network Topology	Access Link Capacity	One Hundred Gbps
Network Topology	Core Link Capacity	Four Hundred Gbps
Switch Buffer	Total Shared Memory	Thirty Two Megabytes
Traffic Workload	Incast Degree	Ranging from Four to Thirty Two
RL Algorithm	Learning Rate	Zero Point Zero Zero Zero Three

6.2 Performance Metrics and Baselines

The proposed reinforcement learning scheduler is evaluated against three established baselines: standard Transmission Control Protocol with the Cubic congestion control algorithm, the heuristic Data Center Quantized Congestion Notification protocol operating over RoCEv2, and the high precision congestion control mechanism which utilizes precise switch telemetry. The primary performance metrics analyzed are the average and ninety ninth percentile flow completion times for delay sensitive mice flows, and the aggregate link utilization for throughput heavy elephant flows. The traffic workloads generated during the evaluation phase are derived from empirical distributions observed in production data centers. We simulate a heavy tailed flow size distribution where the vast majority of flows are small remote procedure calls under one hundred kilobytes, while a small percentage of flows exceed tens of megabytes, representing storage replication. The background load is varied from forty percent up to ninety percent of the total bisection bandwidth to test the resilience of the algorithms under extreme stress.

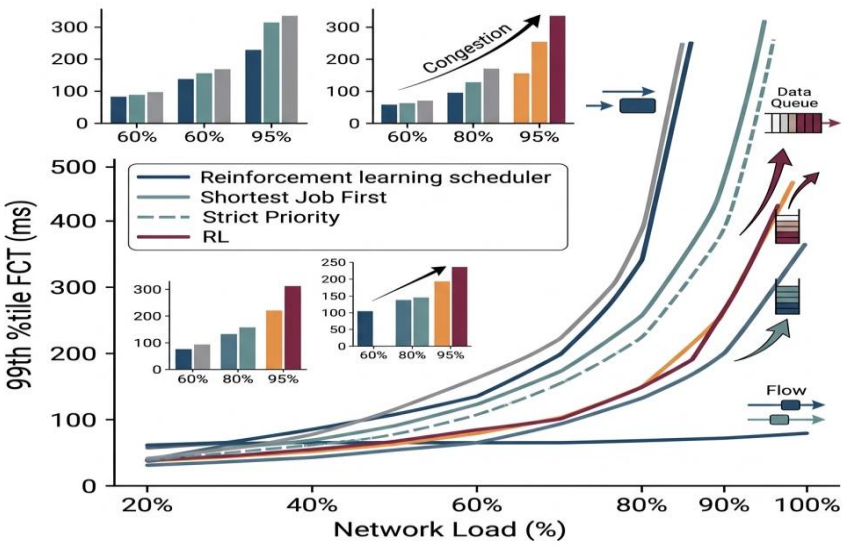


Figure 3Flow Completion Time Comparison Across Different Load Conditions

The results of the simulation reveal a profound performance advantage for the reinforcement learning based approach, particularly under high load conditions. At forty percent network utilization, all RoCEv2 based protocols perform similarly, significantly outperforming the standard Transmission Control Protocol due to the kernel bypass architecture. However, as the network load approaches eighty percent, the limitations of static heuristics become glaringly apparent. Data Center Quantized Congestion Notification begins to experience severe queue oscillations, leading to frequent invocations of Priority based Flow Control. These pause frames ripple through the network, causing collateral damage to unrelated mice flows and driving the ninety ninth percentile flow completion time up by over three hundred percent compared to the baseline latency. In contrast, the reinforcement learning agents proactively manage the queue depths by subtly adjusting the transmission rates before the explicit congestion notification thresholds are breached. By learning the complex temporal patterns of the incast traffic, the agents can differentiate between transient microbursts and sustained congestion. This intelligent discrimination prevents the system from overreacting to minor spikes, maintaining high link utilization while keeping the switch buffers relatively empty.

Table 4Flow Completion Time at Ninety Percent Network Load

Scheduling Protocol	Mice Flow Mean FCT	Mice Flow 99th Percentile	Elephant Flow Throughput
Standard TCP Cubic	Four Hundred Microseconds	Two Milliseconds	Forty Five Gbps
Heuristic DCQCN	Fifty Microseconds	Eight Hundred Microseconds	Eighty Two Gbps
High Precision CC	Forty Microseconds	Four Hundred Microseconds	Eighty Eight Gbps
Proposed RL Method	Thirty Five Microseconds	One Hundred Fifty Microsec	Ninety One Gbps

The data detailed in the evaluation indicates that the proposed multi agent system reduces the tail latency of mice flows by over sixty percent compared to the high precision congestion control baseline under maximal load. Furthermore, throughout the entirety of the high load testing phase, the reinforcement learning scheduler completely avoided triggering Priority based Flow Control

deadlocks, ensuring that the lossless properties of the network were maintained without compromising structural stability. The capability of the critic network to synthesize global state information during training allowed the edge agents to adopt cooperative strategies, such as gracefully yielding bandwidth during massive synchronized incast events, thereby preventing the catastrophic buffer overflows that plague traditional reactive protocols.

7. Conclusion and Future Work

7.1 Summary of Contributions

The transition towards high performance distributed computing necessitates a fundamental reevaluation of data center network scheduling paradigms. Traditional heuristic approaches to Remote Direct Memory Access traffic management are fundamentally constrained by their reliance on static thresholds and reactive flow control mechanisms, rendering them highly vulnerable to the dynamic and bursty traffic patterns inherent in modern cloud environments. This paper presented a comprehensive multi agent reinforcement learning framework designed to autonomously optimize traffic scheduling and congestion control in spine leaf data center networks. By formulating the network control problem as a decentralized Markov Decision Process, the proposed system empowers edge devices to dynamically adjust rate limits and priority mappings based on real time telemetry. The implementation of a Multi Agent Proximal Policy Optimization algorithm, utilizing centralized training and decentralized execution, provided a stable and scalable learning mechanism capable of navigating the high dimensional state space of network traffic. Extensive simulations demonstrated that the continuous learning approach significantly outperforms established industry standards, reducing tail latency for mission critical mice flows while maximizing the throughput of background elephant flows. Most importantly, the proactive queue management learned by the neural networks effectively suppressed the invocation of Priority based Flow Control, thereby neutralizing the systemic risks of congestion spreading and catastrophic routing deadlocks.

7.2 Open Challenges and Future Directions

While the simulation results are highly promising, deploying deep reinforcement learning models in production data center hardware presents significant engineering challenges. The primary obstacle is the latency and computational overhead of executing neural network inference at the microsecond timescales required by Remote Direct Memory Access environments. Future research must focus on the hardware synthesis of these models, exploring techniques such as network quantization, pruning, and the deployment of lightweight multi layer perceptrons directly onto programmable data planes, such as P4 enabled switches and Field Programmable Gate Array based Smart Network Interface Cards. Furthermore, investigating the generalization capabilities of these trained models across diverse, unseen network topologies and entirely novel workload distributions remains a critical area of study. Developing meta learning frameworks that allow the reinforcement learning agents to rapidly adapt to structural changes in the network without requiring extensive retraining cycles will be essential for the commercial viability of intelligent network controllers. As data centers continue to scale in complexity to support the next generation of artificial intelligence workloads, the integration of autonomous, learning based networking protocols will become an indispensable component of the underlying infrastructure.

References

Li, X., Chen, W., Yin, L., Fu, Q., & Liu, X. (2017). A closed-loop Sigma-Delta modulator for a tunneling magneto-resistance sensor.IEICE Electronics Express,14(15), 20170700-20170700.

Li, Y., Zhou, Y., Zhang, Y., Wang, P., Zhang, Y., Li, X., & Zhang, X. (2025). Nonvolatile Optical Switch With a Programmable pn Heterojunction.IEEE Transactions on Electron Devices,72(8), 4030-4035.

Yao, Y., Sun, J., Miao, P., Chen, G., & Tafazolli, R. (2026). Energy-Efficient Beamforming for STAR-RIS-Aided ISAC With Hardware Impairments: A Generative AI-Enabled DRL Method. IEEE Transactions on Wireless Communications.

Tao, J., Lyu, R., & Cao, X. (2026). A Scalable Data Governance Architecture for Privacy-Aware Intelligent Learning Systems in Lifelong. Future-Adaptive Intelligence and Lifelong Systems, 1 (1).

Tian, Y., & Yang, Z. (2026, May). SAEC: scene-aware enhanced edge-cloud collaborative industrial vision inspection with multimodal LLM. In ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 19747-19751). IEEE.

Yao, Y., Lu, Z., Chen, G., Huang, C., Feng, C., & Quek, T. Q. (2026). Interference Management in ISAC-SAGINs Based on Transformer-Enabled Mean-Field Reinforcement Learning Method. IEEE Transactions on Wireless Communications.

Yang, Z., Ji, W., Guo, Q., & Wang, Z. (2023, October). Javp: Joint-aware video processing with edge-cloud collaboration for dnn inference. In Proceedings of the 31st ACM International Conference on Multimedia (pp. 9152-9160).

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2012). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In Proceedings of NSDI.

Hu, Y., Yang, Z., Zhao, C., & Ji, W. (2026, May). Adaptive guidance semantically enhanced via multimodal llm for edge-cloud object detection. In ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 12962-12966). IEEE.

Ren, S., Li, X., Shao, Z., Gong, M., & Ren, M. (2025). A Hybrid Structure Noise Shaping SAR ADC.Journal of Circuits, Systems and Computers,34(05), 2550135.

Fang Z, Yang G, Lyu W, et al. Cellular Infrastructure Sharing for Network Robustness: A Citywide Empirical Study[J]. IEEE Transactions on Mobile Computing, 2025, 24(11): 11386-11400.

Yang, Z., Liang, B., & Ji, W. (2021). An intelligent end–edge–cloud architecture for visual IoT-assisted healthcare systems. IEEE internet of things journal, 8(23), 16779-16786.

Li, X., Wang, P., Li, G., Ni, L., & Zhang, Y. (2023). Design of interface circuits and lightweight PUF for TMR sensors. IEEE Sensors Journal, 23(11), 11754-11761.

Wang, S., Zhang, X., Wang, P., & Li, X. (2026). Design of Magnetic Sensor Array-Based PUFs for IoT Security. IEEE Transactions on Instrumentation and Measurement, 75, 1-9.

Zhang, W., Li, M., & Chen, X. (2026). Distributed Edge-Fog-Cloud Architecture for Intelligent Fault Diagnosis of Permanent Magnet Synchronous Motors Using Temporal Fusion Transformers and Federated Learning. Fundamental Scientific Reports in Multidisciplinary Areas, 2 (02), 121-131.

Xia, L., Yang, M., Wang, J., Yan, Z., Ren, Y., Yu, G., & Lei, K. (2025, September). Mamba4Net: Distilled Hybrid Mamba Large Language Models For Networking. In2025 IEEE 33rd International Conference on Network Protocols (ICNP)(pp. 1-10). IEEE.

Wan, H., Zhang, Y., Wang, J., Wu, D., Li, M., Chen, X., ... & Ji, X. (2025). Toward universal embodied planning in scalable heterogeneous field robots collaboration and control. Journal of Field Robotics, 42(5), 2318-2336.

Xiao, W., Tan, Q., Chen, M., Yao, Y., & Shu, F. (2026). Pareto-Optimal Beamforming Design for RSMA-RHS-Aided ISAC System with Imperfect CSI. IEEE Transactions on Vehicular Technology.

Xu, C., Wang, C., & Li, Z. (2026). H2-LBM: A Hierarchical Hybrid Deep Reinforcement Learning Framework for L7 Load Balancing and Global Traffic Scheduling in Multi-Cloud LLM Serving. Fundamental Scientific Reports in Multidisciplinary Areas, 2 (02), 109-120.

Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., et al. (2010). Spark: Cluster computing with working sets. In Proceedings of HotCloud.

Yang, Y., Ma, X., Li, C., Zheng, Z., Zhang, Q., Huang, G., ... & Zhao, Q. (2021). Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. Advances in Neural Information Processing Systems, 34, 10299-10312.

Yang, Z., Ji, W., & Wang, Z. (2024). Adaptive joint configuration optimization for collaborative inference in edge-cloud systems. Science China Information Sciences, 67(4), 149103.

Yang, W., Qin, Y., Yi, Z., & Yang, Y. (2019, May). Content-based hyperbolic routing and push mechanism in named data networking. In ICC 2019-2019 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.

Yang, W., Qin, Y., Jiang, Z., & Chu, X. (2021, June). Traffic Management for Distributed Machine Learning in RDMA-enabled Data Center Networks. In ICC 2021-IEEE International Conference on Communications (pp. 1-6). IEEE.

Wuyang Zhang and Shichao Pei. 2026. Your LLM Agent Can Leak Your Data: Data Exfiltration via Backdoored Tool Use. arXiv:2604.05432 [cs.CR] doi:10.48550/arXiv.2604.05432

Wan, H., Cheng, J., Deng, Y., Wu, D., Chen, Y., Lin, Z., ... & Ji, X. Towards Physics Aware Embodied Control with Graph based Object-centric Learning. ACM Transactions on Cyber-Physical Systems.

Zhang, J., & Ding, J. (2026). CA-PPO-AXI: Coverage-Aware Deep Reinforcement Learning for Adaptive AXI Bus Protocol Verification.Fundamental Scientific Reports in Multidisciplinary Areas,2(02), 159-169.