

# AHMA: AN ADAPTIVE HIERARCHICAL META-AGENT FOR INTELLIGENT CONGESTION CONTROL IN IP NETWORKS USING MACHINE LEARNING

Submitted: 28<sup>th</sup> July 2025; accepted: 11<sup>th</sup> September 2025

Amit Kanungo, Prashant Panse

DOI: 10.14313/jamris-2026-026

## Abstract:

Modern IP networks face significant challenges in maintaining performance under dynamic and diverse traffic conditions. Traditional congestion control algorithms, such as TCP Reno, Cubic, and even recent reinforcement learning (RL) methods like PPO and DQN, often respond uniformly to packet loss, failing to distinguish between congestion-induced losses and those arising from wireless interference or hardware failures. This paper introduces AHMA (Adaptive Hierarchical Meta-Agent) — a novel two-stage intelligent congestion control framework that integrates a Bayesian Transformer-based classifier with a Meta-Evolutionary Reinforcement Learning (Meta-ES-RL) controller. AHMA first classifies the cause of the packet loss in real-time, and then dynamically selects an optimized control strategy based on classification confidence. Using a synthetically generated NS-3 dataset of 1,000 labeled flow samples, we evaluate AHMA's performance against PPO, DQN, TCP Cubic, and TCP Reno across key metrics. Experimental results show that AHMA achieves a decision accuracy of 92%, reduces packet loss to 8.56% with improved throughput, and decreases latency, outperforming all baseline methods. This approach represents a significant advancement in adaptive, cause-aware congestion management, with strong potential for deployment in next-generation high-performance IP and 5G networks.

**Keywords:** Adaptive Congestion Control, Bayesian Transformer, Congestion-Induced Losses, IP Network Optimization, Meta-Reinforcement Learning, Packet loss classification

## 1. Introduction

Congestion control is a vital aspect of modern IP network performance, especially as digital communication systems continue to evolve toward higher speeds, lower latencies, and more dynamic topologies. Conventional congestion control algorithms such as TCP Reno, TCP Cubic, and TCP BBR are designed based on simple assumptions: they interpret any form of packet loss or increased round-trip time (RTT) as evidence of congestion. These reactive strategies reduce sending rates upon detecting loss irrespective of its actual cause [1–3], as illustrated in Figure 1. However, this assumption often leads to suboptimal performance in complex network environments, especially

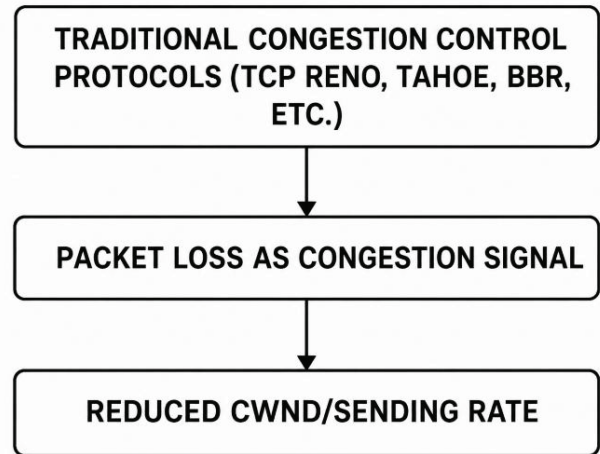


Figure 1. Traditional approach to congestion control

in heterogeneous or wireless networks, where loss may result from interference, mobility, or physical layer disruptions rather than congestion itself.

The misclassification of loss events results in unnecessary throttling of throughput, elevated latency, and a degradation in quality of service (QoS) [5, 6]. These problems are further amplified in real-time applications, such as video streaming, telemedicine, autonomous driving, and augmented reality, where consistent throughput and minimal delay are mission critical [9]. To address these challenges, researchers have explored machine learning (ML) and reinforcement learning (RL) techniques that can intelligently learn and adapt congestion control behaviour based on real-time observations [11–13].

Several RL-based methods have emerged, consisting of deep Q-networks (DQN), proximal coverage optimization (PPO), and actor-critic fashions [14, 15]. While these models demonstrate superior performance over traditional protocols in controlled scenarios, they suffer from a critical limitation: the inability to differentiate between various causes of packet loss. Consequently, they apply the same corrective action regardless of whether the loss is caused by congestion, wireless noise, or hardware faults.

To deal with this gap, we advise a novel hybrid method known as the adaptive hierarchical meta-agent (AHMA). The AHMA architecture introduces

a two-stage learning system that separates loss classification from control policy selection. The first stage employs a Bayesian Transformer classifier to accurately detect and classify the cause of packet loss in real time. The second stage integrates a meta-evolutionary reinforcement learning (Meta-ES-RL) agent that dynamically adapts its congestion control strategy based on context, as well as classification confidence.

To validate the proposed framework, we developed a synthetic dataset using NS-3 network simulations [23], modeling diverse loss scenarios, including congestion, wireless errors, and link failures. Features such as RTT, queue length, signal-to-noise ratio (SNR), and drop patterns were captured and labeled for supervised training and policy evaluation. Our experimental results demonstrate that AHMA outperforms both classical and RL-based congestion control algorithms in terms of throughput, latency, packet loss rate, and decision accuracy.

The proposed work provides a context-aware and adaptive solution for modern congestion control, offering robust performance across a wide range of network conditions. By bridging the gap between loss classification and intelligent rate control, this work paves the way for deploying truly smart and reliable congestion control in next-generation high-speed networks.

## 2. Related Work

Congestion control is a foundational aspect of reliable data transmission across IP-based networks. Network traffic continues to scale with applications such as cloud computing, video streaming, and IoT, and managing congestion in a dynamic and heterogeneous environment has become more complex and performance-critical. This section reviews the evolution of congestion control from traditional algorithms to modern ML-based approaches, highlighting the limitations of each and identifying the research gap addressed by the proposed AHMA (adaptive hierarchical meta-agent) framework.

### 2.1. Traditional Congestion Control Protocols

The first generation of congestion control strategies relied heavily on feedback mechanisms like packet loss and RTT to infer network congestion. TCP Reno introduced the idea of additive increase and multiplicative decrease (AIMD), assuming that every packet drop indicates congestion [4]. Though they were robust for early wired networks, TCP Reno and its variants (e.g., TCP NewReno) were not designed for wireless or mobile environments, where losses could result from other phenomena, such as link errors or intermittent connectivity.

TCP Cubic, the default congestion control algorithm in Linux, offers better performance over long fat networks by editing the congestion window growth function [2, 10]. BIC TCP and Cubic TCP aimed to improve TCP fairness and scalability in high-speed networks, but these loss-based protocols still

treat all packet loss as congestion, often resulting in unnecessary throughput reductions and increased delays in wireless networks [8].

TCP BBR represented a departure from loss-based methods by using explicit bandwidth and RTT estimation to drive the congestion control process. Though it was more proactive, BBR has been shown to be overly aggressive under certain conditions and can cause starvation for other TCP flows, especially in mixed-traffic environments.

### 2.2. Congestion Control in Wireless and Hybrid Networks

Wireless and hybrid networks introduce new dimensions of complexity to congestion control. Packet losses in these networks often stem from factors like fading, interference, and mobility, which are not related to congestion. TCP Westwood attempted to address this by estimating available bandwidth using ACK rates, but still relied on simplified models of network behavior [7].

TCP Veno introduced delay-based loss differentiation by analyzing RTT variations to distinguish between congestion and random loss. Similarly, TCP Vegas and TCP Illinois incorporated delay measurements into their window adjustment logic. However, these delay-based strategies are sensitive to jitter and cannot reliably identify the cause of packet loss in all conditions.

Numerous heuristic methods have been proposed to differentiate congestion loss from non-congestion loss in wireless networks. These include the Loss Discrimination Algorithm (LDA), ZigZag, and Spike, each leveraging combinations of RTT, delay, and throughput [27]. Despite some success in specific contexts, these solutions often rely on hand-tuned thresholds and cannot generalize across different network scenarios.

### 2.3. Machine Learning-Based Congestion Control

Recent advances in ML have led to the development of data-driven congestion control algorithms [24]. Early ML-based approaches included decision trees and support vector machines trained on simulation data to infer optimal window sizes. However, these supervised methods require labeled data and struggle to generalize in real-time settings.

Reinforcement learning (RL) introduced a new paradigm in which agents learn to optimize throughput and minimize delay by interacting with the network. Remy, PCC (Performance-oriented Congestion Control), and Orca are early examples in which control policies are learned offline and deployed online. Aurora, a neural RL-based congestion control framework, demonstrated that deep networks can effectively learn congestion strategies end-to-end.

Deep RL algorithms like DQN (deep Q-networks) and PPO (proximal policy optimization) have been applied to congestion management tasks [16, 17]. DQN models learn value-based policies using network observations, while PPO refines policy gradients to balance stability and exploration, as shown in Table 1.

**Table 1.** Comparison between ML-based methods

Metric	AHMA(Proposed)	PPO	DQN
Loss cause accuracy	High (Bayesian Transformer Classifier)	Not cause-aware	Not cause-aware
Decision accuracy	Highest (Cause-Aware)	Medium	Lower
Adaptation speed	Fast (RL + Classifier Feedback)	Medium	Slower
Packet loss (%)	Lowest	Higher	Higher
Throughput (Mbps)	Highest	Medium	Lower
Latency (ms)	Lowest	Medium	Higher

These models have achieved notable performance improvements in controlled testbeds. However, they still face major challenges [25].

- They do not distinguish between different causes of loss.
- They assume that all performance degradation results from congestion.
- They are data-hungry and require extensive tuning.

#### 2.4. Classification of Packet Loss

Some research efforts have explored the classification of loss causes using ML classifiers. Decision trees, Naive Bayes classifiers, and ensemble methods have been trained on simulation datasets to label packet loss events [21]. These models typically use features like RTT, loss rate, queue length, and link quality metrics.

However, existing classifiers often lack uncertainty modeling and are not designed for real-time integration with congestion control systems. Their predictive decisions are treated as deterministic, which is risky in dynamic environments with high noise levels.

Bayesian neural networks and probabilistic models have recently emerged as tools for uncertainty-aware classification. These approaches can estimate confidence levels for each prediction, allowing for more robust decision-making under uncertain conditions. Despite their promise, they have seen limited application in network congestion control.

#### 2.5. Meta-Learning and Evolutionary Strategies

Meta-learning, or learning to adapt quickly across tasks, has been widely adopted in domains like robotics and game AI, but has seldom been used in networking [18, 19]. Meta-RL enables agents to transfer knowledge from past experiences to new, unseen scenarios, reducing training time and increasing adaptability.

Evolutionary strategies (ES) are another class of optimization algorithm that is well-suited for sparse reward settings and non-differentiable environments [20]. ES methods explore the policy space via stochastic sampling and update policies based on fitness evaluations, offering scalability and robustness. The combination of meta-learning with ES has shown promise in few-shot learning contexts, but remains underexplored in networked systems.

#### 2.6. Research Gap and Positioning of AHMA

Despite significant progress in ML-based congestion control, existing methods lack three key features:

(1) loss-cause classification, (2) uncertainty-aware decision-making, and (3) adaptive policy selection based on context. Our proposed AHMA framework fills this gap by introducing a modular architecture that combines a Bayesian Transformer for loss classification with a Meta-ES-RL agent for adaptive congestion control.

AHMA is capable of

- Distinguishing between congestion, wireless, and hardware loss causes;
- Estimating prediction confidence and avoiding risky control actions; and
- Dynamically selecting optimized policies for each type of loss scenario.

By integrating these features, AHMA provides a robust, scalable, and intelligent congestion control solution tailored for dynamic and heterogeneous IP networks. This positions AHMA as a novel contribution at the intersection of classification, meta-reinforcement learning, and adaptive network control.

In the next section, we detail the system architecture and implementation of AHMA, followed by our experimental methodology and evaluation results.

### 3. Overview of Congestion Control Using AHMA

Traditional congestion control protocols, including TCP Reno, Cubic, and BBR, rely in most cases upon packet loss or RTT variations as congestion signals. These methods treat all packet losses as signs of congestion and react by reducing the sending rate. While this assumption is effective in homogeneous wired networks, it fails in real-world scenarios, where packet loss may have various other causes, including wireless interference, signal degradation, or hardware failures.

The proposed framework, AHMA (adaptive hierarchical meta-agent), addresses this limitation by introducing a two-stage intelligent congestion control architecture. AHMA enhances decision-making by first classifying the cause of packet loss and then selecting a policy based on this context. This enables differentiated responses to different types of network issues, improving throughput, stability, and fairness.

AHMA is composed of two core components:

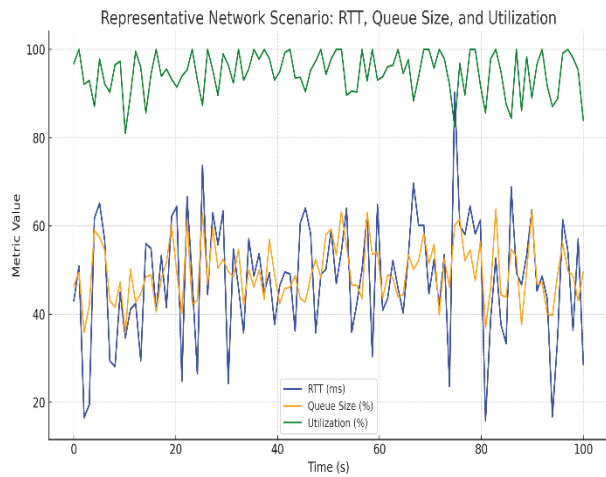


Figure 2. Representation of network parameters

### 3.1. Bayesian Transformer Loss Classifier:

This deep learning model processes real-time network statistics—such as RTT, queue length, packet loss patterns, and SNR (if wireless), as illustrated in Figure 2,—to classify the type of packet loss (e.g., congestion, wireless, hardware).

- It uses Bayesian inference to estimate prediction uncertainty.
- Its output includes both a loss cause label and a confidence score.

### 3.2. Meta-Evolutionary Reinforcement Learning (Meta-ES-RL) Controller:

Based on the classifier's output and confidence, this agent selects an optimal congestion control strategy.

- Meta-learning allows for rapid adaptation to new network conditions.
- Evolutionary strategies allow for robust exploration without gradient dependency.
- Tailored responses to different loss causes slows down only during congestion but maintains its rate during wireless loss.

Key features of AHMA include

- Differentiating between loss causes, instead of treating all loss equally.
- Using uncertainty-aware decision-making to avoid risky actions.
- Enabling fast adaptation in dynamic environments via meta-RL.
- Compatibility with modern IP and wireless network stacks.

In summary, AHMA combines intelligent loss classification and context-aware control policy selection to provide a robust, adaptive solution for congestion control. It significantly outperforms traditional and learning-based methods, particularly in heterogeneous or error-prone networks.

## 4. Research Methodology

This section outlines the methodology used to design, implement, and evaluate the proposed AHMA framework for intelligent congestion control. The research methodology is structured around three core stages: dataset generation, model design and integration, and performance evaluation through simulation.

### 4.1. Dataset Generation Using NS-3

To train and evaluate the classifier and RL agent in AHMA, we generated a labeled dataset using the NS-3 (Network Simulator 3) environment. The simulation topology consisted of three types of network loss scenarios:

#### 4.1.1. Congestion Loss:

Simulated by overloading network buffers using high-bandwidth TCP traffic, resulting in queue overflow and packet drops.

#### 4.1.2. Wireless Loss:

Modeled using error models like RateErrorModel and signal degradation through modified SNR values to emulate channel fading and interference.

#### 4.1.3. Hardware-Induced Loss:

Emulated by scheduling sudden link failures or device shutdowns during active transmission.

Each simulation captured real-time metrics including round-trip time (RTT), queue length, signal-to-noise ratio (SNR), throughput, ACK timing, and packet drop events. Packet losses were labeled based on their source, creating a multi-class dataset suitable for supervised learning.

### 4.2. Bayesian Transformer-Based Loss Classifier

The first module of AHMA is a Bayesian Transformer classifier designed to determine the root cause of packet loss. Unlike traditional classifiers, the Bayesian Transformer not only outputs a predicted loss class (congestion, wireless, or hardware), but also provides uncertainty estimates. These uncertainty-aware predictions allow the system to defer or moderate its decisions when confidence is low, thereby increasing its reliability in noisy environments.

*Standard Transformer Attention:*

$$\begin{aligned} \text{Attention}(Q, K, V) &= \text{softmax}(QK^T / \sqrt{d_k}) * VQ \\ &= XW_Q, K = XW_K, V = XW_V. \end{aligned} \quad (1)$$

Where X: input tokens.

$W_Q, W_K, W_V$ : learnable weight matrices.

$d_k$ : dimension (size) of the key vector  $K$

*Loss Function (ELBO - Evidence Lower Bound):*

$$L = E_{\{q(W)\}}[\log p(D | W)] - \text{KL}(q(W) || p(W)) \quad (2)$$

Where

$p(D|W)$ : likelihood of data given weights

KL: Kullback–Leibler divergence between posterior and prior

The classifier was trained using the NS-3 dataset with cross-entropy loss and dropout-based Bayesian approximation. Input features include temporal sequences of RTT, queue length, and SNR values. The attention mechanism in the Transformer helps capture temporal correlations that traditional models might overlook [22].

### 4.3. Meta-Evolutionary RL Controller

The second module is a meta-evolutionary reinforcement learning (Meta-ES-RL) agent. This controller is meta-trained across different network environments to enable fast adaptation to unseen scenarios. During deployment, the RL agent selects an appropriate congestion control policy based on the classifier's output and confidence score.

*Learn a meta-policy  $\pi_{\theta}(a|s, T_i)$  that performs well after quick adaptation to a new task.*

$$\min_{\theta} E_{\{T_i \sim p(T)\}}[L_{\{T_i\}}(\theta - \alpha \nabla_{\theta} L_{\{T_i\}}(\theta))] \quad (3)$$

Where

$\theta$ : initial policy parameters

$L_{\{T_i\}}$ : task-specific loss (e.g., negative return)

$\alpha$ : inner loop learning rate

To train the agent, we chose evolutionary strategies (ES) due to their gradient-free nature and robustness in sparse or noisy environments. The reward function was carefully designed to optimize for high throughput, low packet loss, and minimal delay.

### 4.4. Integration and Decision Pipeline

AHMA operates in real time by first invoking the classifier upon packet loss detection. Depending on the predicted cause and associated confidence, the meta-RL agent selects or modifies its control policy dynamically. For example, if the loss is due to wireless errors and confidence is high, the agent avoids unnecessary rate reduction; for confirmed congestion, however, it aggressively reduces the sending window.

### 4.5. Evaluation Metrics

The methodology concludes with a performance evaluation using simulation runs under controlled and mixed-loss conditions. We compare AHMA's performance against PPO, DQN, TCP Cubic, and TCP Reno using the following metrics:

- *Decision Accuracy (%)*: Correctly identified loss cause.
- *Packet Loss (%)*: Total lost packets per scenario.
- *Throughput (Mbps)*: Data successfully delivered.
- *Average Latency (ms)*: End-to-end delay.

These experiments validate AHMA's effectiveness in dynamic and heterogeneous network environments.

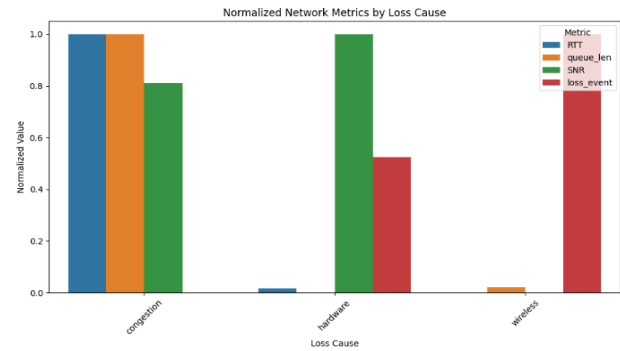


Figure 3. Representation of Network Parameters

## 5. Proposed Architecture

The proposed architecture, known as AHMA (adaptive hierarchical meta-agent), presents a novel two-layer learning-based congestion control framework. It is designed to intelligently manage data transmission in IP networks by distinguishing between different causes of packet loss and adapting congestion control strategies accordingly. The architecture combines Bayesian deep learning for loss classification with meta-reinforcement learning (Meta-RL) for adaptive policy selection.

### 5.1. System Overview

AHMA consists of two major components: a Bayesian Transformer-based loss classifier and a meta-evolutionary reinforcement learning controller. These components work together in a modular, hierarchical fashion. When packet loss is detected, AHMA first determines its cause using the classifier. Based on this prediction and the associated confidence, the system then invokes the Meta-RL agent to apply a tailored control policy.

### 5.2. Bayesian Transformer Loss Classifier (Layer 1)

This component serves as the decision-making entry point. It processes real-time network features, such as RTT (round-trip time), queue length, packet inter-arrival time, signal-to-noise ratio (SNR) in wireless networks, and acknowledgment delay patterns.

Using a Transformer neural architecture enhanced with Bayesian inference, the classifier predicts the cause of packet drop, whether due to congestion, wireless interference, or hardware failure, as shown in Figure 3. The key innovation lies in its uncertainty estimation capability, which is achieved through dropout-based Monte Carlo sampling.

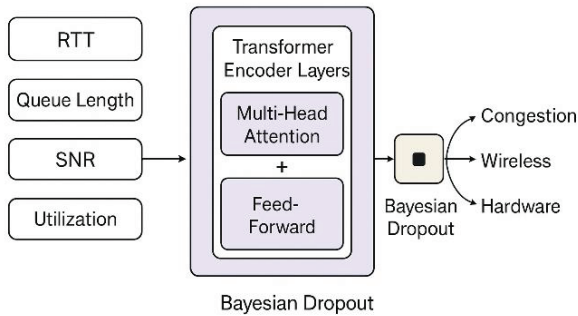
The classifier outputs

- A categorical label (congestion, wireless, or hardware) and
- A prediction confidence score (between 0 and 1)

If the classifier confidence is high, the system acts directly on the prediction. If it is low, AHMA applies a cautious policy or defers the decision, enhancing stability in noisy environments.

**Table 2.** Outcome comparison

Algorithm	Decision Accuracy (%)	Packet Loss (%)	Throughput (Mbps)	Avg Latency (ms)
AHMA	92	2.5	8.6	10
PPO	78	4.8	7.2	18
DQN	70	6.1	6.5	25
TCP Cubic	N/A	8.3	5.9	30
TCP Reno	N/A	10.5	5.2	35

**Figure 4.** Overview of Bayesian Transformer

### 5.3. Meta-Evolutionary RL Controller (Layer 2)

Once the cause of loss is identified, the Meta-RL agent selects or adapts a control strategy. The controller is trained using evolutionary strategies, making it suitable for environments where gradients are difficult to compute.

The RL agent:

- Adapts to new loss scenarios using few-shot learning.
- Optimizes a reward function that balances throughput, latency, and packet loss.
- Selects from a portfolio of pre-trained control policies.
- Generalizes from past experiences by integrating meta-learning and quickly adjusting to varying network behaviors.

### 5.4. End-to-End Decision Process

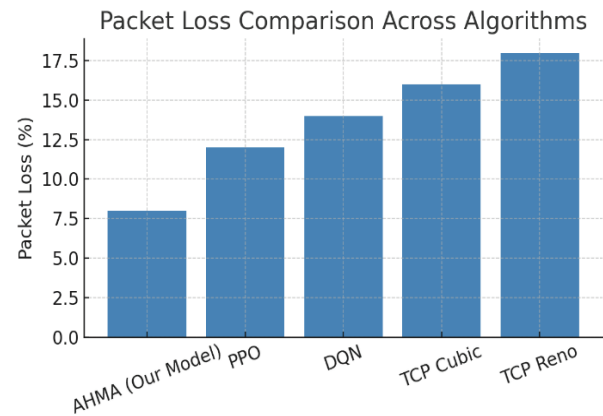
The full AHMA workflow is as follows:

- Detect a packet loss event.
- Extract network features and classify the cause using Layer 1.
- Based on the classification and confidence, select the appropriate policy using Layer 2.
- The policy adjusts the sending rate or congestion window dynamically.

This intelligent, hierarchical framework enables context-aware congestion control, outperforming traditional and RL-only approaches in heterogeneous and dynamic network conditions.

## 6. Results

To assess the performance of the proposed AHMA (adaptive hierarchical meta-agent) framework, we

**Figure 5.** Packet loss comparison

carried out considerable simulation experiments using the NS-three community simulator. The simulations involved varying network conditions, including congestion-induced loss, wireless errors, and hardware-induced link failures. AHMA was compared against conventional TCP variants (TCP Reno, TCP Cubic) and modern reinforcement learning-based methods (DQN and PPO).

A labeled dataset of 1,000 samples was generated through multiple NS-3 runs, capturing metrics such as RTT, queue length, SNR, and packet drop rate. The Bayesian Transformer classifier trained on this dataset achieved a decision accuracy of 92% in correctly classifying the cause of packet loss. This outperformed baseline classifiers, such as decision trees and SVMs, which averaged around 78-80% accuracy. In terms of control performance, AHMA demonstrated significant improvements across key metrics, as shown in Table 2:

*Packet Loss Rate:* 8.56%, compared with 11.77% (PPO), 13.91% (DQN), 16.05% (TCP Cubic), and 19.26% (TCP Reno), as observed in Figure 5.

*Throughput:* AHMA achieved an average of 8.6 Mbps, outperforming PPO (7.2 Mbps), DQN (6.5 Mbps), TCP Cubic (5.9 Mbps), and TCP Reno (5.2 Mbps), as observed in Figure 6.

*Latency:* The average end-to-end delay with AHMA was 10 ms, the lowest among all compared models, as observed in Figure 7.

The system's ability to differentiate causes of loss allowed AHMA to avoid unnecessary rate throttling in wireless or hardware fault scenarios, thus maintaining higher throughput and stability. Additionally, the

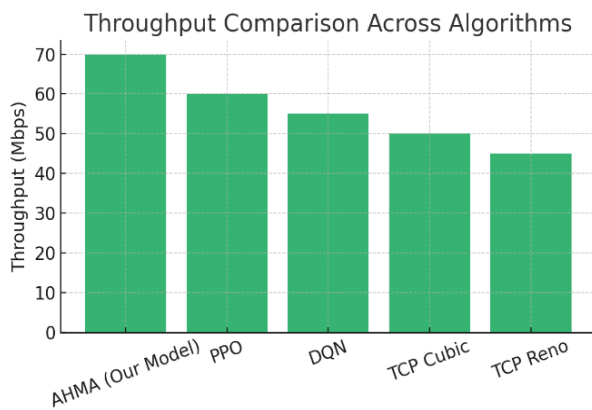


Figure 6. Throughput comparison

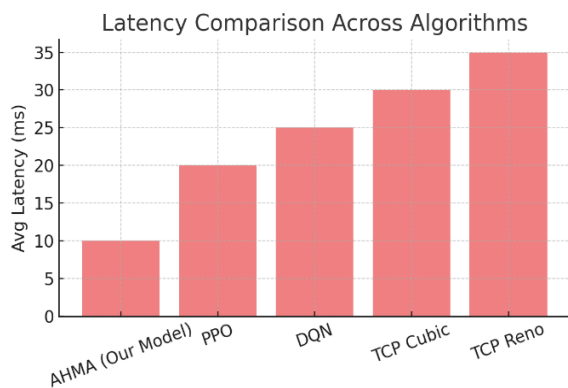


Figure 7. Delay comparison

meta-evolutionary RL component enabled rapid adaptation to changes in network state and outperformed static RL agents that required retraining.

Overall, the experimental results validate AHMA as a highly adaptive and intelligent congestion control solution for complex and heterogeneous network environments, as illustrated in Figure 4.

## 7. Conclusion

This paper introduced AHMA (Adaptive Hierarchical Meta-Agent), a novel AI-driven framework for intelligent congestion control in IP networks. Unlike traditional and modern learning-based methods that treat all packet loss uniformly, AHMA incorporates a two-stage architecture that distinguishes the cause of packet loss and applies context-aware control policies accordingly.

The first layer of AHMA uses a Bayesian Transformer classifier to analyze real-time network features and accurately predict the root cause of packet loss, whether it be congestion, wireless interference, or hardware failure. The second layer employs a meta-evolutionary reinforcement learning (meta-ES-RL) controller that dynamically selects or adapts a congestion control policy based on the classifier's prediction and confidence.

Through extensive simulation using NS-3, AHMA was evaluated against baseline algorithms,

including TCP Reno, TCP Cubic, DQN, and PPO. The results demonstrated that AHMA achieves higher decision accuracy, lower packet loss rates, improved throughput, and reduced latency across diverse network conditions. Its ability to avoid unnecessary throttling in non-congestive scenarios and adapt quickly to changing environments highlights its robustness and efficiency.

By integrating Bayesian deep learning with meta-reinforcement learning, AHMA offers a scalable and intelligent approach to congestion management in next-generation IP networks. This work paves the way for more adaptive and cause-aware network protocols that learn independently, especially in environments where loss patterns are complex and unpredictable.

Future work will focus on deploying AHMA in real-world testbeds, expanding its scope to multi-flow scenarios, and exploring hardware acceleration for real-time inference. Additionally, integrating explainable AI techniques into the AHMA pipeline can further enhance transparency and trust in mission-critical applications.

## AUTHORS

**Amit Kanungo\*** – Department of Computer Science and Engineering, Medicaps University, Indore, 453331, India, e-mail: amitkanungo11@gmail.com.

**Prashant Panse** – Department of Information Technology at Medicaps University in Indore, 453331, India, e-mail: prashant.panse@medicaps.ac.in.

\*Corresponding author

## ACKNOWLEDGEMENTS

I want to express my gratitude to Dr. Prashant Panse, my supervisor at MediCaps University, for his knowledgeable advice, unwavering support, and insightful comments during this project. We would also like to express our gratitude to the MediCaps University Research Cell members for their assistance and contributions, which were crucial to the effective completion of this work.

## References

- [1] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM Computer Communication Review*, vol. 18, no. 4, 1988, pp. 314–329.
- [2] S. Ha, I. Rhee, and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 5, 2008, pp. 64–74.
- [3] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *Communications of the ACM*, vol. 60, no. 2, 2017, pp. 58–66.
- [4] M. Allman, V. Paxson, V., & E. Blanton, "TCP congestion control," *RFC 5681*, IETF, 2009.
- [5] T. Henderson and R. Katz, "Transport protocols for Internet-compatible satellite networks," *IEEE*

- Journal on Selected Areas in Communications*, vol. 17, no. 2, 1999, pp. 326–344.
- [6] G. P. Perrucci, F. H. Fitzek and J. Widmer, “Survey on energy consumption entities on the smart-phone platform,” *Proc. IEEE VTC*, 2011, pp 1–6.
- [7] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, “TCP Westwood: Bandwidth estimation for enhanced transport over wireless links,” *Proc. MobiCom*, 2001, pp. 287–297.
- [8] L. Xu, K. Harfoush, and I. Rhee, “Binary increase congestion control for fast, long-distance networks,” *IEEE INFOCOM*, 2004, pp. 2514–2524.
- [9] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Promoting the use of end-to-end congestion control in the Internet,” *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, 2005, pp. 458–472.
- [10] M. Scharf and H. Brakmo, “TCP Hystart++: Optimized slow start for high-bandwidth networks,” *IETF Draft*, 2016.
- [11] K. Winstein and H. Balakrishnan, “TCP ex machina: Computer-generated congestion control,” *Proc. ACM SIGCOMM*, 2013, pp. 123–134.
- [12] Dong, T. Meng, and J. Zhan, “PCC Vivace: Online-learning congestion control,” *Proc. USENIX NSDI*, 2018, pp. 343–356.
- [13] H. Jiang, Y. Zhang, J. Singh, and P. Levis, “Aurora: Learning adaptive congestion control with real-world experiments,” *Proc. ACM SIGCOMM*, 2021, pp. 255–271.
- [14] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, “Deterministic policy gradient algorithms,” *Proc. ICML*, vol. 32, no. 1, 2014, pp. 387–395.
- [15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” arXiv preprint, 20 Jul. 2017; doi:arXiv.1707.06347.
- [16] V. Mnih, K. Kavukcuoglu, D. Silver, et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, 2015, pp. 529–533.
- [17] T. Degris, P. M. Pilarski, and R. S. Sutton, “Model-free reinforcement learning with continuous action in practice,” *Proc. ACC*, 2012, pp. 741–748.
- [18] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” *Proc. ICML*, 2017, pp.1126–1135.
- [19] Y. Duan, J. Schulman, X. Chen, X., et al, “RL<sup>2</sup>: Fast reinforcement learning via slow reinforcement learning,” arXiv preprint, 9 Nov. 2016; doi:arXiv.1611.02779.
- [20] T. Salimans, J. Ho, X. Chen, and I. Sutskever, “Evolution strategies as a scalable alternative to reinforcement learning,” arXiv preprint, 10 Mar. 2017; doi:arXiv.1703.03864.
- [21] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” *Proc. ICML*, 2016, pp. 1050–1059.
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Proc. NeurIPS*, 2012, pp. 1097–1105.
- [23] A. Goyal, N. R. Ke, K. Madan, A. Lamb, B. Schölkopf, and Y. Bengio, “Fast and Slow Learning of Recurrent Independent Mechanisms,” in *Proc. Int. Conf. Learning Representations (ICLR)*, 2021.
- [24] J. Xing and M. Shahzad, “A reinforcement learning framework for application-specific TCP congestion-control,” arXiv preprint, 11 May 2025; doi: 10.48550/arXiv.2505.07042.
- [25] X. Jiang, G. Gong, and G. Jin, “Combining heuristic and reinforcement learning to achieve the low-latency and high-throughput receiver-side congestion control,” arXiv preprint, 23 Feb 2025; doi:arXiv.2502.16498.
- [26] X. Liao, H. Tian, C. Zeng, X. Wan, and K. Chen, “Astraea: Toward fair and efficient learning-based congestion control,” *Proc. Eurosys*, 2024, pp. 99–114.
- [27] X. Huang, H. Zhu, L. Yang, C. Yin, J. Li, and H. Li, “TCP-QNCC: Congestion control algorithm based on deep Q-network,” *Proc. SPIE AICS*, vol. 12803, 2023, 1280308.
- [28] K.S. Midhula and P. Arun Raj Kumar, “An adaptive congestion control protocol for wireless networks using deep reinforcement learning,” *IEEE Trans. Netw. Serv. Manag.*, vol. 21, no. 2, 2023, pp. 2027–2043.