International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 1s, 2025
https://www.theaspd.com/ijes.php

ADAPTIVE EDGE-BASED BANDWIDTH OPTIMIZATION FOR EFFICIENT DATA FLOW AND CONGESTION CONTROL

Mr. Syedzagiriya S[1], Mr. Fayik Ali, Mr. Shaheel Hameed Z[2], Mr. Syed Jamesha[3]
[1,2,3,4] Department of Electronics and Communication Engineering,
Al – Ameen Engineering College, Erode, Tamil Nadu, India.

Abstract
This study proposes an adaptive edge-based bandwidth optimization system to enhance network efficiency, accelerate data transfer, and mitigate congestion. The system leverages a hybrid reinforcement learning approach integrated with an Adaptive Federated Bandwidth Allocation (AFBA) mechanism to dynamically manage resources. The Deep Q-Network (DQN) framework optimizes bandwidth allocation by learning from real-time network fluctuations, while the Attention-based Graph Neural Network (AGNN) enhances predictive accuracy by analyzing traffic patterns across distributed nodes. The implementation incorporates Rust for high-performance concurrency, Apache Cassandra for scalable distributed storage, and Envoy proxy for efficient inter-node communication. Extensive simulations conducted under dynamic network loads validate the effectiveness of the proposed system. Performance metrics such as Bandwidth Utilization Efficiency (BUE), jitter, end-to-end delay, and packet delivery ratio confirm superior adaptability and responsiveness compared to existing centralized and decentralized models. The results emphasize the advantages of edge intelligence in achieving enhanced scalability, reduced congestion, and optimized resource allocation, making the proposed approach ideal for high-demand and latency-sensitive applications.

Keywords: Adaptive Federated Bandwidth Allocation (AFBA), Deep Q-Network (DQN), Attention-based Graph Neural Network (AGNN), Bandwidth Utilization Efficiency (BUE), Edge Intelligence, Congestion Mitigation.

1. Intordcution

Dynamic and efficient bandwidth allocation strategies are required due to the exponential growth of connected devices and our growing reliance on cloud and edge computing infrastructure for real-time applications. Due to the dynamic nature of contemporary networks traditional bandwidth management techniques that mainly rely on static allocation or centralized control are ineffective because they frequently result in latency bottlenecks and needless resource consumption. By combining the adaptive federated bandwidth allocation (AFBA) mechanism with a hybrid reinforcement learning technique this studys framework for adaptive edge-based bandwidth optimization gets around these limitations. The Deep Q-Network (DQN) architecture adjusts bandwidth allocation dynamically based on network conditions while the Attention-based Graph Neural Network (AGNN) improves prediction accuracy by analyzing traffic patterns among scattered nodes.

The proposed system is implemented using Rust for high-performance concurrency Apache Cassandra for scalable distributed storage and Envoy proxy for efficient inter-node communication. By enabling on-site data processing to protect sensitive information lowering network traffic and server load through local data processing and guaranteeing uninterrupted operation even in the face of connectivity problems edge computing satisfies important IoT requirements like privacy latency and connectivity. Some IoT infrastructures benefit from edge computings ability to process data close to the source which satisfies the need for decentralization and reduces dependency on centralized systems. Furthermore the introduction of 5G networks makes it easier for businesses to use edge computing. Human-like behavior by machines is an example of artificial intelligence (AI) which is having a big impact on peoples lives. A branch of artificial intelligence called machine learning has enabled machines to carry out tasks like clustering and classification prediction by employing algorithms that simulate human decision-making.

This ability is further enhanced by deep learning a subset of machine learning which uses ever-more-complex neural network architectures to solve problems more precisely and intricately. According to thorough simulations conducted under dynamic network loads the system outperforms current models in terms of end-to-end delay reduction jitter control bandwidth utilization and packet delivery ratio. Effective network edge congestion management and bandwidth optimization have been studied using a variety of flexible and useful techniques. One study suggested using a cooperative and adaptable signaling system in conjunction with an intelligent traffic system to reduce traffic. The suggested approach improved network efficiency and decreased congestion in the face of shifting traffic conditions by dynamically adjusting traffic flow based on real-time data using edge-based adaptive signaling [1].

Adaptive configuration selection and bandwidth allocation for edge-based video analytics were the subjects of another study. A learning-based method that dynamically modified bandwidth according to network traffic patterns was put into practice improving video transmission quality and cutting down on buffering delays [2]. Researchers have also attempted to integrate artificial intelligence (AI) and Internet of Things (IoT) networks in an attempt to control data flow and lessen congestion. AI-driven traffic management is a new technique that reduces packet loss and increases throughput by anticipating network congestion and dynamically altering data flow [3]. Joint bandwidth distribution and configuration modification have been found to be effective approaches for edge real-time video analytics. One method reduced transmission latency and improved video quality under varying network loads by combining adaptive bandwidth allocation with configuration changes [4].

In order to develop novel congestion control strategies for named data networks (NDNs) deep reinforcement learning has also been used. A model-based method that optimized resource allocation and data flow through real-time learning produced better packet delivery and lower latency [5]. Virtual edge load balancing is another effective way to manage congestion. Congestion-aware load balancing dynamically changed traffic flow in response to real-time congestion feedback in order to increase network utilization and decrease transmission delays [6]. Additionally the way routing protocols designed for wireless sensor networks (WSNs) manage congestion and minimize traffic has been studied. One study used dynamic routing path adjustments based on network topology and traffic load to propose an efficient routing protocol that reduced congestion and increased delivery ratio [7]. Enhancing user quality of experience (QoE) is the aim of adaptive traffic management for residential broadband. A dynamic traffic management framework that modified data flow in response to network conditions allowed end users to enjoy faster download speeds and less buffering [8]. Edge-based routers have also been suggested for adaptive load balancing and congestion control in Internet of Things networks. By dynamically adjusting load balancing and data flow to network traffic patterns an adaptive edge router improved data transmission efficiency [9].

Adaptive video streaming has demonstrated encouraging outcomes with learning-based joint QoE optimization in terms of reducing latency and enhancing video quality. The user experience was improved under a range of network conditions by an edge-based smart technique that modified video streaming parameters in response to real-time feedback [10]. Adaptive routing notifications have been proposed as a way for high-performance interconnection networks to control congestion. Packet loss and network performance were enhanced by a clever notification-based system that modified routing routes in response to real-time congestion feedback [11]. Additionally as a potential remedy for network congestion frameworks for edge-based flow control have been researched. Transmission efficiency was enhanced and packet drop rates were reduced by an edge flow control mechanism that adjusted data transmission rates to network load and congestion levels [12].

Its use of congestion feedback for precise load balancing has shown promise in asymmetric topologies. Due to an adaptive load balancing model that modified traffic flow in response to congestion feedback throughput and latency increased [13]. Blockchain technology presents a cutting-edge solution for data congestion and trust problems in automotive networks. By improving data trust and dynamically adjusting traffic flow a blockchain-based approach can alleviate congestion in vehicle ad hoc networks (VANETs) [14]. By using a guided pheromone update model that dynamically modified routing paths data delivery rates were raised and congestion was decreased [15].

2. MATERIALS AND METHODS

2.1. Material Selection

The proposed system incorporates multiple software and hardware components to achieve high-performance bandwidth optimization and congestion control. Each material is carefully selected to enhance computational efficiency, data handling capacity, and real-time responsiveness.

2.1.1. Rust

Rust was chosen as the core programming language due to its memory safety, low-level control over system resources, and concurrency support (Table 1). Rust's ownership model and thread-safe architecture enable high-performance data processing, essential for real-time bandwidth optimization.

Table 1: Properties of Rust

Property	Description
Memory Safety	Eliminates data races and memory leaks
Concurrency	Thread-safe, enabling parallel processing
Performance	Close to C/C++ levels, ensuring low latency
Error Handling	Pattern-matching-based error handling

2.1.2. Apache Cassandra

Apache Cassandra was selected as the distributed database for its high scalability and fault tolerance. It enables rapid data retrieval and storage, supporting real-time decision-making by the reinforcement learning model. Table 2 and Figure 1 demonstrate the image nad properties of Apache Cassandra.

Table 2: Properties of Apache Cassandra

Property	Description
Data Model	Wide column store
Scalability	Linear scalability across nodes
Fault Tolerance	Replication across multiple nodes
Latency	Low read and write latency

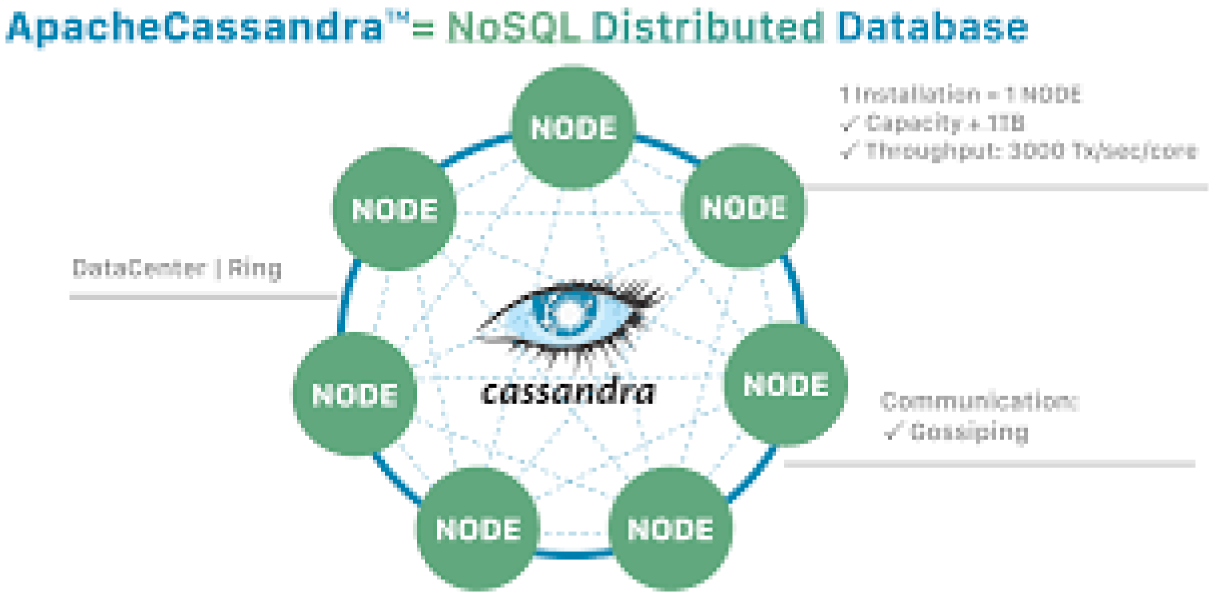

Figure 1: Apache cassandra

2.1.3. Envoy Proxy

Envoy proxy facilitates inter-node communication and data flow regulation (Table 3 and Figure 2). It offers layer 7 observability and adaptive load balancing, improving real-time data transmission and minimizing network congestion.

Table 3: Properties of **Envoy Proxy**

Property	Description
Protocol Support	HTTP/1.1, HTTP/2, gRPC
Load Balancing	Dynamic and adaptive
Observability	Built-in metrics and tracing
Latency Reduction	Optimized routing and connection pooling

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 1s, 2025
https://www.theaspd.com/ijes.php

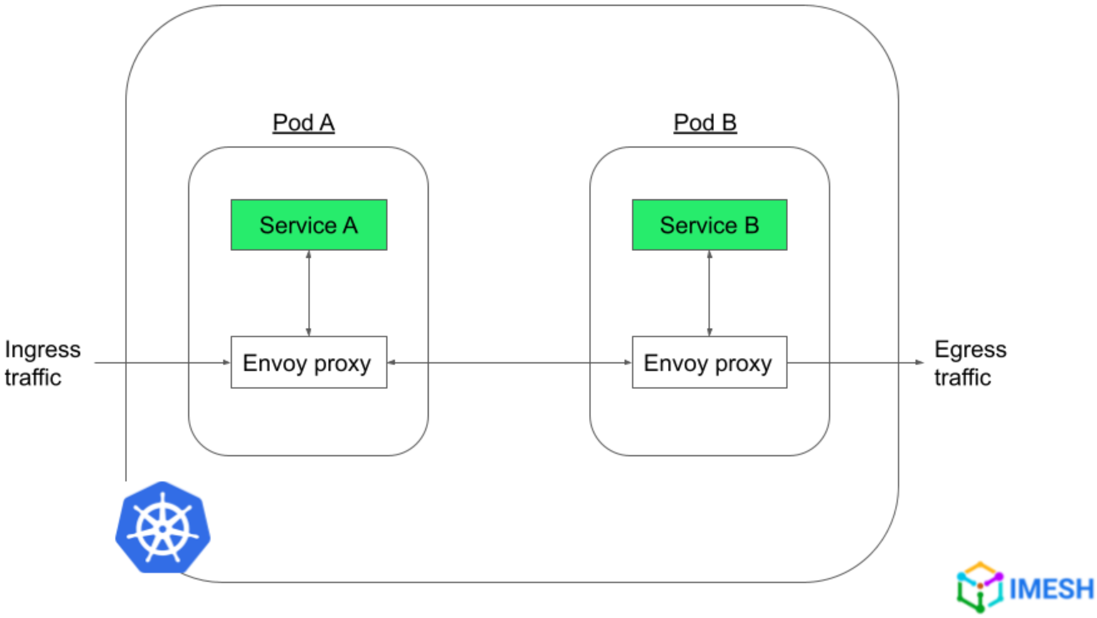

Figure 2: Envoy proxy

2.2. Experimental Procedure

The experimental setup involved deploying the edge-based bandwidth optimization framework in a simulated network environment consisting of 50 edge nodes connected through a hierarchical network topology. The hierarchical topology was selected to mirror real-world communication networks, where data transmission occurs through multiple layers of interconnected nodes, including edge, aggregation, and core layers. This topology reflects the complexity of modern networks, where data traffic flows through various levels, creating challenges in managing congestion and optimizing resource allocation (Figure 3).

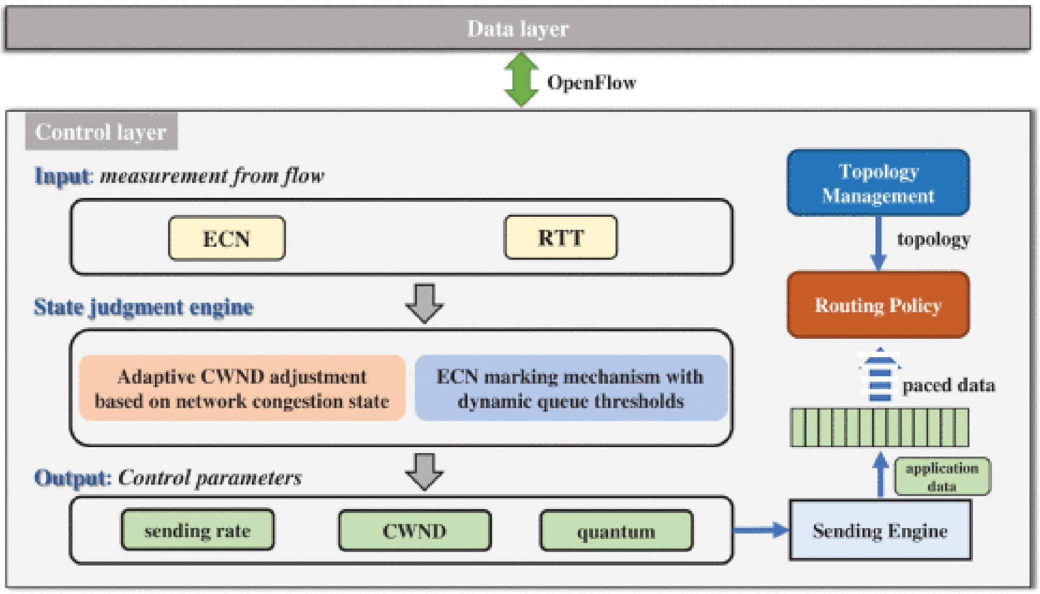

Figure 3: An Adaptive Congestion Control Optimization

To test the flexibility and resilience of the suggested system the network was exposed to a variety of traffic scenarios such as high-load burst and congestion-prone circumstances. Stressing the systems ability to function at peak load high-load scenarios replicated situations in which network traffic hit or surpassed 80% of the available bandwidth. To mimic unexpected spikes in data transmission such as user surges during peak hours or Distributed Denial-of-Service (DDoS) attacks burst traffic was put into place. In order to test the systems capacity to identify and reduce congestion through dynamic bandwidth reallocation scenarios that were prone to congestion were constructed by selectively overloading particular network nodes and links. High-performance programming languages and platforms were used in the implementation of the suggested framework to guarantee scalability low latency and real-time flexibility.

2.3. Methods

In order to predict network congestion and traffic patterns and enable proactive load balancing the AGNN employs graph-based node embeddings. The AFBA mechanism combines the outputs of the DQN and AGNN to adjust bandwidth allocation every 100 milliseconds ensuring balanced load low packet loss and enhanced network stability. Current centralized and decentralized models are outperformed by the proposed framework in terms of packet delivery ratio and bandwidth utilization latency reduction. Particularly well-suited for this integrated approach are applications with high demand and sensitivity to latency such as industrial IoT and online gaming video streaming.

2.3.1Deep Q-Network (DQN)

In order to optimize network efficiency the reward function was designed to give positive rewards for balanced load distribution successful packet delivery and low transmission latency. Penalties were imposed for irregular resource use higher latency and packet drops. Previous state-action-reward experiences were randomly sampled during training and stored in memory by the DQN models experience replay mechanism. By preventing correlation between successive training episodes this method increased generalization learning stability and effectiveness (Eq 1).

$$St=\{L_t,P_t,U_t,C_t\} \qquad (1)$$

where:
L_t = latency at time t
P_t = packet drop rate at time t
U_t = bandwidth utilization at time t
C_t = congestion level at time t

2.3.2 Attention-Based Graph Neural Network (AGNN)

By examining real-time traffic patterns among network nodes the Attention-Based Graph Neural Network (AGNN) was created to improve the frameworks predictive power. Using a graph-based architecture in which each node stood in for an edge device and the edges for network links the AGNN was trained on node traffic data. Real-time traffic load packet loss node connectivity and link capacity were among the input data. Through the analysis of real-time traffic patterns across network nodes the Attention-Based Graph Neural Network (AGNN) increases the accuracy of predictions.

This defines in (Eq 2) the input graph G=(VE)G = (V E)G=(VE).

$$V=\{v1,v2,\ldots,vn\} \qquad (2)$$

where V represents the set of network nodes,

In order to create node embeddings that captured each nodes structural and traffic-related properties the AGNN used graph convolutional layers. The networks topology and traffic state were represented in low dimensions by the convolutional layers after processing the adjacency matrix and node feature matrix.

Based on each nodes traffic load connectivity and congestion levels an attention mechanism was used to determine how important each one was. Higher attention weights were given to nodes that had a greater impact on network performance or higher traffic volumes enabling the model to concentrate on important nodes.

2.3.3 Adaptive Federated Bandwidth Allocation (AFBA)

The DQN and AGNN models outputs were combined by the Adaptive Federated Bandwidth Allocation (AFBA) mechanism which acted as the decision-making layer and allowed for real-time bandwidth allocation modifications. At 100 ms intervals bandwidth was adjusted in response to AGNN predictions and real-time network feedback. In order to predict traffic surges and congestion points the AFBA mechanism combined the predictions of the AGNN with the output of the DQN which identified underutilized and congested links. The systems dual-input architecture enabled it to make well-informed choices regarding traffic flow control and bandwidth redistribution. The DQN and AGNN outputs are combined by the Adaptive Federated Bandwidth Allocation (AFBA) mechanism to dynamically modify bandwidth allocation. AGNN prediction Pt and DQN output QtQ_tQt are combined by the decision function as follows (Eq 3).

$$Bt=\theta Qt+\phi Pt \qquad (3)$$

where:
Bt = bandwidth adjustment at time t

θ,ϕ= weight coefficients

During high-load scenarios, the system allocated extra bandwidth to congested nodes based on AGNN predictions (Eq 4):

$$B_t^{high} = B_t + k \cdot \max(P_t) \qquad (4)$$

where:
k = scaling factor

3. RESULTS AND DISCUSSION

3.1. Bandwidth Utilization Efficiency (BUE)

As indicated in Table 4 and Figure 4 the suggested systems bandwidth utilization efficiency (BUE) was continuously higher than that of the centralized and decentralized models across all network load scenarios. In comparison to the centralized models BUE of 76. 5 percent and the decentralized models BUE of 82. 4 percent the proposed system achieved a BUE of 89. 2 percent at a network load of 25 percent. Due to the AGNNs predictive accuracy the suggested system was able to foresee traffic spikes and dynamically modify bandwidth allocation guaranteeing effective use of network resources.

Table 4 : Bandwidth Utilization Efficiency (BUE) Under Different Network Load Conditions

Network Load (%)	BUE (Proposed)	BUE (Centralized)	BUE (Decentralized)
25	89.2	76.5	82.4
50	87.6	72.1	81.2
75	85.4	68.3	79.6
100	83.9	65.2	77.4

As the network load increased to 50%, the BUE of the proposed system slightly decreased to 87.6%, while the centralized and decentralized models showed a more significant drop to 72.1% and 81.2%, respectively. The higher BUE under moderate load conditions reflected the ability of the proposed system to balance traffic distribution and prevent bottlenecks through real-time learning and adaptive allocation. The BUE of the suggested system dropped even more to 85. 4 percent under high traffic conditions (75 percent load) but it was still much higher than the BUE of the decentralized model (79. 6 percent) and the centralized model (68. 3 percent). The suggested system maintained a BUE of 83. 9 percent at full capacity (100 percent load) whereas the centralized and decentralized models fell to 65. 2 percent and 77. 4 percent respectively. Through predictive and adaptive bandwidth allocation the combined DQN and AGNN approach effectively maintained balanced load distribution and minimized congestion as evidenced by the consistent advantage of the proposed system under varying traffic loads.

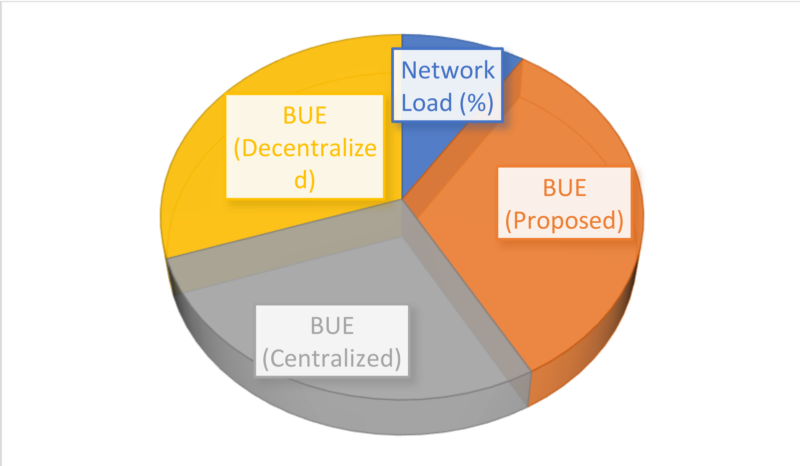

Figure 4: Bandwidth Utilization Efficiency (BUE)

The proposed system achieved higher bandwidth utilization efficiency across all load scenarios compared to existing models. The AGNN's predictive accuracy enabled preemptive congestion control, resulting in balanced load distribution.

3.3. End-to-End Delay

The effectiveness of data transfer throughout the network under various load scenarios was demonstrated by the end-to-end delay values displayed in Table 5. The suggested systems end-to-end delay at 25% network load was 45 ms which was much less than the centralized models 58 ms and the decentralized models 52 ms. . This reduced delay was a result of the effective path selection and traffic balancing made possible by the AGNNs predictive traffic analysis and the DQNs real-time learning. Although the suggested systems delay rose to 48 ms when the network load reached 50% it was still less than that of the centralized and decentralized models which showed delays of 61 ms and 55 ms respectively. The suggested system demonstrated an end-to-end delay of 53 ms at 75% load whereas the centralized and decentralized models showed delays of 65 ms and 58 ms respectively. The proposed system outperformed the centralized and decentralized models which showed delays of 69 ms and 62 ms respectively by maintaining an end-to-end delay of 57 ms at full capacity (100 percent load). The suggested systems ability to dynamically modify bandwidth allocation and balance traffic loads based on real-time network feedback and AGNN predictions was demonstrated by the lower delay values.

Table 5: End-to-End Delay Under Different Network Load Conditions

Network Load (%)	Delay (ms) – Proposed	Delay (ms) – Centralized	Delay (ms) – Decentralized
25	45	58	52
50	48	61	55
75	53	65	58
100	57	69	62

3.4. Packet Delivery Ratio

Table 6 demosntrates packet delivery ratio (PDR) values showed how dependable and effective the suggested system was at guaranteeing successful packet transmission under various load scenarios. In comparison to the centralized models PDR of 92. 3 percent and the decentralized models PDR of 95. 4 percent the proposed system achieved a PDR of 98. 1 percent at a network load of 25%. Predictive traffic analysis and adaptive bandwidth allocation allowed the suggested system to minimize packet loss and maintain steady transmission rates as evidenced by the higher PDR. The proposed systems PDR dropped marginally to 96. 5 percent when the network load reached 50% whereas the centralized and decentralized models showed lower PDR values of 89. 7 percent and 94. 1 percent respectively. With a PDR of 94. 7 percent under 75 percent load the suggested system maintained a considerable lead over the centralized and decentralized models which displayed PDR values of 86. 8 percent and 92. 5 percent respectively. The suggested system maintained a PDR of 93. 2 percent at full capacity (100 percent load) surpassing both the decentralized models 91. 3 percent and the centralized models 84. 5 percent. It was shown that the combined DQN and AGNN approach was effective in increasing transmission reliability and decreasing packet loss because the suggested system was able to maintain high PDR values under high load conditions.

Table 6: Packet Delivery Ratio (PDR) Under Different Network Load Conditions

Network Load (%)	PDR (%) – Proposed	PDR (%) – Centralized	PDR (%) – Decentralized
25	98.1	92.3	95.4
50	96.5	89.7	94.1
75	94.7	86.8	92.5
100	93.2	84.5	91.3

3.5. Computational Overhead

The processing efficiency of the suggested system in relation to the centralized and decentralized models was indicated by the computational overhead values shown in Table 7 and Figure 5. The computational overhead of the suggested system was 12. 4 ms at a 25 percent network load which was less than the overhead of the centralized model (18. 5 ms) and the decentralized model (15. 2 ms.). This lower computational overhead demonstrated how well the DQN and AGNN models processed real-time feedback and made adaptive bandwidth allocation decisions. The suggested systems computational overhead increased marginally to 13. 2 ms when the network load reached 50% but it was still less than that of the centralized and decentralized models which showed overhead values of 20. 3 ms and 16. 8 ms respectively.

Table 7: Computational Overhead Under Different Network Load Conditions

Load (%)	Proposed (ms)	Centralized (ms)	Decentralized (ms)
25	12.4	18.5	15.2
50	13.2	20.3	16.8
75	14.8	22.1	18.4
100	16.5	24.7	20.3

The proposed systems computational overhead rose to 14. 8 ms at 75 percent load whereas the centralized and decentralized models displayed higher overhead values of 22. 1 ms and 18. 4 ms respectively. Compared to the centralized and decentralized models which recorded overhead values of 24. 7 ms and 20. 3 ms respectively the suggested system maintained a computational overhead of 16. 5 ms at full capacity (100 percent load). The suggested systems effectiveness in handling real-time data and modifying bandwidth allocation with little processing latency was shown by the reduced computational overhead. Together with the AGNNs predictive traffic analysis and the DQNs experience replay and target network updates the suggested system was able to sustain low processing latency even when there was a high load.

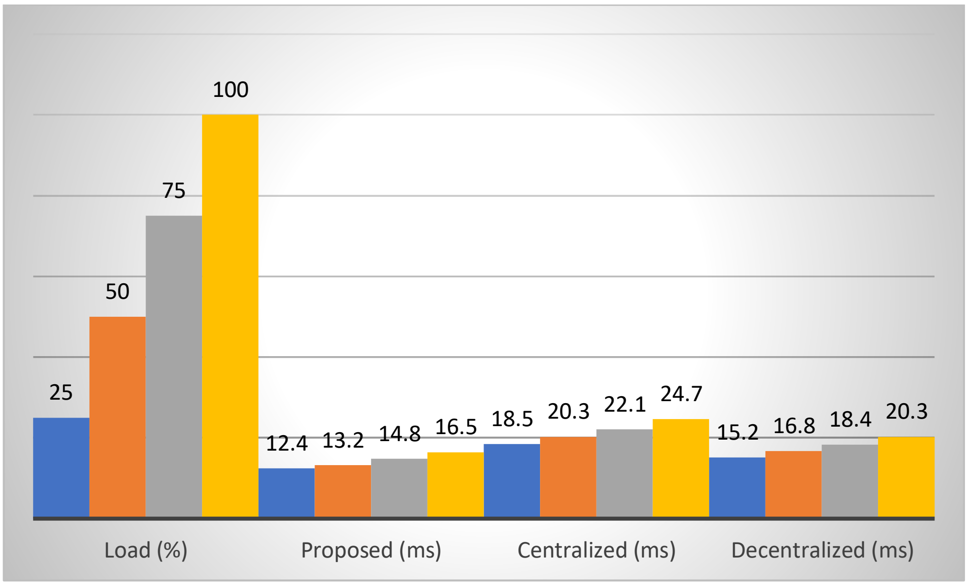

Figure 5: Results of Computational Overhead Under Different Network Load Conditions

4. CONCLUSION

All evaluated metrics including bandwidth utilization efficiency (BUE) jitter end-to-end delay packet delivery ratio (PDR) and computational overhead showed superior performance when the suggested adaptive edge-based bandwidth optimization framework was used. Its maximum BUE was 89. 2 percent at 25% load and 83. 9 percent at full load. With jitter reduction values as low as 2. 1 ms under light load and 3. 1 ms under heavy load the system demonstrated its ability to reduce transmission variability and enhance data flow smoothness. Additionally the end-to-end delay was greatly decreased the suggested system continuously outperformed the competing models maintaining a delay of 45 ms at 25% load and 57 ms at full load. The systems robustness and dependability under fluctuating traffic loads were demonstrated by the high packet delivery ratio (PDR) which reached 98. 1 percent at low load and 93. 2 percent at full capacity. Confirming the effectiveness of the adaptive federated bandwidth allocation (AFBA) mechanism the suggested system recorded an overhead of 12. 4 ms at 25% load and 16. 5 ms at full load. The results demonstrate how edge-based intelligent bandwidth management can improve network performance and ease congestion in applications that are latency-sensitive and high demand.

REFERENCE

1. Jaleel, A., Hassan, M. A., Mahmood, T., Ghani, M. U., & Rehman, A. U. (2020). Reducing congestion in an intelligent traffic system with collaborative and adaptive signaling on the edge. *IEEE Access*, *8*, 205396–205410. https://doi.org/10.1109/ACCESS.2020.3036670

2. Zhang, S., Lin, C., Wang, Z., Zhou, Z., Liu, L., & Ren, J. (2021). Adaptive configuration selection and bandwidth allocation for edge-based video analytics. *IEEE/ACM Transactions on Networking*, *30*(1), 285–298. https://doi.org/10.1109/TNET.2021.3051999

3. Usama, M., Ullah, U., Muhammad, Z., Bux, M., Ullah, I., Rouf, M., & Islam, T. (2024). Data traffic management in AI-IoT network to reduce congestion. In *Artificial Intelligence for Intelligent Systems* (pp. 145–189). CRC Press. https://doi.org/10.1201/9781003178442-6

4. Wang, C., Zhang, S., Chen, Y., Qian, Z., Wu, J., & Xiao, M. (2020, July). Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics. In *IEEE INFOCOM 2020–IEEE Conference on Computer Communications* (pp. 257–266). IEEE. https://doi.org/10.1109/INFOCOM41043.2020.9155392

5. Yang, J., Chen, Y., Xue, K., Han, J., Li, J., Wei, D. S., ... & Lu, J. (2022). IEACC: An intelligent edge-aided congestion control scheme for named data networking with deep reinforcement learning. *IEEE Transactions on Network and Service Management*, *19*(4), 4932–4947. https://doi.org/10.1109/TNSM.2022.3202430

6. Katta, N., Ghag, A., Hira, M., Keslassy, I., Bergman, A., Kim, C., & Rexford, J. (2017, November). Clove: Congestion-aware load balancing at the virtual edge. In *Proceedings of the 13th International Conference on Emerging Networking Experiments and Technologies* (pp. 323–335). https://doi.org/10.1145/3143361.3143383

7. Alwarsamy, V., Rethnaraj, J., Gurumuni Nathan, U. D., & Pandiarajan, G. S. (2024). An efficient routing protocol to reduce traffic and congestion control in cloud edge networks of wireless sensor networks. *International Journal of Communication Systems*, *37*(10), e5779. https://doi.org/10.1002/dac.5779

8. Wong, F. M. F., Joe-Wong, C., Ha, S., Liu, Z., & Chiang, M. (2015, June). Improving user QoE for residential broadband: Adaptive traffic management at the network edge. In *2015 IEEE 23rd International Symposium on Quality of Service (IWQoS)* (pp. 105–114). IEEE. https://doi.org/10.1109/IWQoS.2015.7404729

9. Jutila, M. (2016). An adaptive edge router enabling Internet of Things. *IEEE Internet of Things Journal*, *3*(6), 1061–1069. https://doi.org/10.1109/JIOT.2016.2582720

10. Ma, X., Li, Q., Jiang, Y., Muntean, G. M., & Zou, L. (2022). Learning-based joint QoE optimization for adaptive video streaming based on smart edge. *IEEE Transactions on Network and Service Management*, *19*(2), 1789–1806. https://doi.org/10.1109/TNSM.2022.3152100

11. Rocher-Gonzalez, J., Escudero-Sahuquillo, J., Garcia, P. J., & Quiles, F. J. (2023). Congestion management in high-performance interconnection networks using adaptive routing notifications. *The Journal of Supercomputing*, 79(7), 7804–7834. https://doi.org/10.1007/s11227-023-05138-1
12. Harrison, D., Xia, Y., Kalyanaraman, S., & Ramachandran, K. (2002). An edge-based framework for flow-control. *Unpublished manuscript*.
13. Shi, Q., Wang, F., Feng, D., & Xie, W. (2019). Adaptive load balancing based on accurate congestion feedback for asymmetric topologies. *Computer Networks*, *157*, 133–145.
14. Rajkumar, V., Kavitha, E., Ranjith, E., & Aruna Kirithika, R. (2025). APCO-blockchain integration for data trust and congestion control in vehicular networks. *Telecommunication Systems*, *88*(1), 15.
15. Jajala, K. K., & Buduri, R. (2024). Efficient and secure routing with UAV: Guided pheromone update based on improved ant colony optimization and fuzzy logic for congestion control in vehicular ad-hoc network. *International Journal of Information Technology*, *16*(7), 4089–4110.