International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(15s) 2026 pp. 805-821
https://doi.org/10.70917/ijcisim-2026-2154
Article

Adaptive Slice-Aware Multi-access Edge Computing Scheduling for Low-Latency 5G URLLC Communications

Varunkumar Mishra[1*], Jaymin Bhalani[2], Unnati Shah[3]

[1*]Electronics and Communication Engineering Department, Parul Institute of Technology, Parul University, Vadodara, Gujarat, India.
Varun.mishra@tcetmumbai.in
[2]Department of Electronics and Communication Engineering, Parul Institute of Technology, Parul University, Vadodara, Gujarat, India. jaymin.bhalani25970@paruluniversity.ac.in
[3]Department of Electronics and Communication Engineering, Parul Institute of Technology, Parul University, Vadodara, Gujarat, India.
unnati.shah@paruluniversity.ac.in

Abstract: Ultra-Reliable Low-Latency Communication (URLLC) is one of the most critical types of services in fifth generation (5G) networks for applications which have the need for high speed and reliable communication. Use-cases like autonomous transportation, industrial automation, robotics, and remote healthcare can be negatively affected even by small delay increments in communication. Multi-access Edge Computing (MEC) and network slicing can lower the delay created by centralized processing by bringing computation closer to users and application-specific resources on the network level. Nevertheless, predictable latency in the face of changing network conditions is difficult to achieve due to network congestion, queueing delays, propagation delay, and static resource allocation policy. This paper introduces an Adaptive Slice Aware MEC Scheduling (ASMS) framework that aims to solve the problem of low latency in 5G URLLC scenarios. The suggested solution considers MEC-based processing, network slicing, dynamic resource allocation, intelligent base-station selection, and traffic-aware scheduling. Priority-based decision-making takes place based on the evaluation of network conditions including the current situation in queues, communication latency, channel quality, slice requirements, and Service Level Agreement (SLA) conditions prior to allocating computational and communication resources. The scheduling procedure is ongoing so that resources could be allocated again in case of congestion and violation of SLA. The effectiveness of the proposed ASMS framework was tested in simulations in various conditions of traffic and scheduling using an integrated simulation environment which included MATLAB, NS-3, and Python/SimPy simulators. According to the results, the adaptive MEC-based approach managed to decrease the average communication latency from 2.41 ms to 0.47 ms while staying SLA compliant. At the same time, the improvements were observed in the parameters of throughput, Signal-to-Noise Ratio (SNR), and Bit Error Rate (BER) compared with centralized processing approach.

Keywords: 5G URLLC; MEC-enabled network slicing; slice-aware scheduling; adaptive resource allocation; latency-aware scheduling; QoS optimization; edge computing..

1. Introduction

Rapid proliferation of mission-critical wireless applications has raised the demand for strict latency and reliability requirements as a key design aspect of the next generation of wireless networks, 5G. For example, autonomous vehicles, remote robotic manipulation, industrial automation and telemedicine relieve prompt and efficient information exchange. In this context, communication delays influence the responsiveness and performance of the system. Hence, the introduction of the new category of URLLC 5G services has become relevant to accommodate latency-sensitive applications. [1]

Very low end-to-end latency is a difficult task for a wireless communication network because there are several parameters that influence the total delay. Such factors as propagation delay, transmission processing, queuing, central computation and inefficient resource assignment can cause communication delays. At the same time, all these effects are intensified under varying or heavily loaded traffic conditions. Thus, the ability to provide low latency for a specific set of traffic conditions is not enough. A network should be able to operate with stable performance in changing workloads. [2]

The Multi-access Edge Computing (MEC) technique can become an effective solution for this problem through deploying computing resources closer to the user side. Instead of sending all the service requests to the cloud infrastructure, several processing tasks will be handled at edge servers. As a result, it can lead to the reduction of the communication path and processing delay. Nevertheless, the advantage of edge computing can become minimal in the situation when the competition for the edge resources emerges between multiple users or the traffic among MEC servers becomes unbalanced.

Network slicing is another solution for heterogeneous 5G services due to the logical division of network resources and assigning the specific QoS to each category. Particularly, the URLLC traffic will have higher scheduling priority than the rest of the service types. Nevertheless, static configuration and resource reservation scheme might be insufficient to react to the dynamic changes in traffic load. It is necessary to use some mechanism for adjusting the decision-making process to the current conditions of the network. [3]

There are several research that have analyzed MEC, network slicing, intelligent resource management and other techniques for the reduction of the latency in 5G networks. Nevertheless, the consistent provision of stable URLLC performance in changing traffic conditions is still relevant problem. Separate optimization of the edge processing, slicing or resource management might not consider the interaction between the queue occupancy, channel conditions, slice priority, MEC load and SLA requirements. This problem requires the development of some adaptive scheduling mechanism that will consider all these aspects.

To solve this problem, this research develops the framework called Adaptive Slice-Aware MEC Scheduling (ASMS) for 5G URLLC services. It is based on the combination of MEC processing, network slicing, dynamic resource allocation, beamforming and intelligent selection of the base-station. Unlike the predefined resource allocation policy, this scheduling mechanism analyzes the current network state and allocates resources according to priority and network conditions. In the case of congestion or SLA violation, the ASMS framework will change the allocation and scheduling in order to maintain the proper performance of the service. [4]

The main contributions of this research are as follows:

a) Adaptive Slice-Aware MEC Scheduling (ASMS) algorithm for the allocation of the communication and computational resources to the URLLC services considering the changes in traffic conditions.
b) Priority-based decision mechanism that considers the queue length, communication latency, channel quality, network-slice priority and SLA requirements.
c) Integrated resource-management framework, which consists of MEC, network slicing, beamforming, dynamic resource allocation and intelligent base-station selection.
d) Evaluation of the proposed framework under various traffic conditions and scheduling to analyze the influence of the solution on communication latency and other performance indicators.
e) Simulation results show the reduction of the average communication latency from 2.41 ms to 0.47 ms in compliance with SLA, thus proving the potential of adaptive MEC-based scheduling for the future latency-sensitive 5G and beyond-5G services.

2. Research Gap and Objectives

2.1 GOAL OF RESEARCH: REDUCTION OF PROPAGATION DELAY

This goal is addressed through optimizing physical infrastructure, and the research tries to minimize the time of the travel of a signal from source to destination point. Approaches: The approach taken includes dense deployment of small cells, and the use of the high frequency band for mmWave technology to reduce wavelengths. [5]

2.2 GOAL OF RESEARCH: OPTIMIZATION OF NETWORK ARCHITECTURE

This goal is achieved through designing an efficient network architecture that would minimize latency through efficient data transfer and processing. Approaches: Multi-access edge computing (MEC), routing algorithms, and software-defined networking (SDN). [6]

3. Literature Review

Table 1: Various approaches to reduce latency in 5G URLLC systems from 2019 to 2026.

Sr. No.	Year	Authors / Study	Approach and Main Contribution	Reported Outcome	Identified Limitation
1	2019	D. Feng, J. Luo, and C. Zheng, "Ultra-Reliable Low-Latency Communications (URLLC) in Autonomous Vehicular Networks"	Developed a URLLC-oriented framework for vehicular communication.	Improved communication reliability and reduced latency for vehicular services.	Scalability was not extensively addressed, and MEC integration was absent.
2	2020	P. Frangoudis and A. Ksentini, "Toward Slicing-Enabled Multi-Access Edge Computing in 5G"	Investigated the integration of network slicing with MEC in 5G environments.	Reduced service delay and increased flexibility in slice management.	Adaptive scheduling and congestion-aware resource management were not addressed.
3	2020	A. Filali et al., "Multi-Access Edge Computing (MEC): A Survey"	Reviewed MEC architectures, technologies, and application scenarios.	Highlighted the benefits of processing services at the network edge.	The study was survey-oriented and did not provide a practical implementation framework.
4	2020	Y. Liu, M. Peng, and G. Shou, "Toward Edge Intelligence: Multi-Access Edge Computing for 5G and Internet of Things"	Examined intelligent MEC architectures for IoT environments.	Demonstrated potential improvements in edge intelligence and computational efficiency.	Specific QoS requirements associated with URLLC were not extensively analyzed.
5	2020	R. Chataut and R. Robert, "Massive MIMO Systems for 5G and Beyond Networks"	Reviewed Massive MIMO technologies applicable to 5G and beyond.	Reported benefits in spectral efficiency and communication reliability.	MEC-assisted latency optimization was outside the scope of the study.
6	2021	S. Wijethilaka et al., "Survey on Network Slicing for Internet of Things"	Surveyed network-slicing techniques for IoT services.	Identified improvements in traffic isolation and resource management.	Dynamic adaptation of slices was not sufficiently addressed.

7	2021	J. Lee and K. Yang, “Edge Computing Framework for Low-Latency Applications in 5G”	Developed an edge-computing framework for latency-sensitive 5G applications.	Reduced end-to-end communication delay.	Priority-based scheduling was not implemented.
8	2021	Ericsson, “Massive MIMO Explained: Unlocking 5G Efficiency”	Described the role and benefits of Massive MIMO in 5G networks.	Highlighted improvements in throughput and network efficiency.	URLLC-specific optimization was not considered.
9	2023	Md. Arif Hossain and N. Ansari, “Network Slicing for NOMA-Enabled Edge Computing”	Combined network slicing with NOMA-enabled edge-computing systems.	Improved bandwidth utilization and QoS management.	Congestion-aware scheduling was not comprehensively evaluated.
10	2023	J. Jin, R. Li, and X. Yang, “A Network Slicing Algorithm for Cloud Edge Collaboration Hybrid Computing in 5G and Beyond Networks”	Proposed network slicing for cloud-edge collaborative computing.	Improved resource-orchestration efficiency.	Realistic URLLC latency performance was not evaluated.
11	2023	A. Filali et al., “Dynamic SDN-Based Radio Access Network Slicing with Deep Reinforcement Learning for URLLC Services in 5G and Beyond”	Applied deep reinforcement learning to adaptive RAN slicing for URLLC.	Improved slice management and QoS performance.	The computational complexity of learning-based scheduling can limit real-time deployment.
12	2024	Faezeh B. et al., “MEC Resource Slice Allocation: A Review”	Reviewed resource-allocation mechanisms for MEC network slices.	Identified approaches for improving MEC resource allocation.	Experimental validation under dense-network conditions was limited.
13	2025	Wan et al., “Delay Sensitive Music Transmission	Proposed a delay-sensitive architecture for multimedia	Reduced transmission delay.	Scalability and reliability were not

		Architecture for 5G VANET”	transmission in vehicular networks.		comprehensivel y evaluated.
14	2025	L. Nadeem et al., “Evaluation of Combined Network Slicing and MEC Toward Mobile Robot Applications”	Evaluated the combined use of network slicing and MEC for robotic applications.	Improved latency and traffic isolation.	Multi-user congestion effects were not extensively investigated.
15	2025	Liu and Zhao, “Dynamic Resource Allocation in 5G Using Network Slicing”	Developed a dynamic bandwidth-allocation approach based on network slicing.	Improved resource-utilization efficiency.	MEC orchestration was not incorporated.
16	2024	X. Li, Y. Zhang, and M. Chen, “Deep Reinforcement Learning for Resource Allocation in 5G Wireless Networks”	Applied DRL to adaptive resource management in 5G wireless networks.	Improved scheduling and bandwidth allocation.	Training overhead and computational complexity remain concerns.
17	2024	K. Davaslioglu and E. Ayanoglu, “QoS-Aware Resource Management for Network Slicing in 5G and Beyond”	Developed a QoS-oriented resource-management framework for network slicing.	Improved service differentiation and traffic isolation.	URLLC-specific implementation and evaluation were limited.
18	2024	A. Alsharoa, M. Alouini, and H. Yanikomeroglu, “Dynamic User Association for Latency Minimization in 5G Networks”	Proposed dynamic user association to reduce latency in 5G environments.	Reduced latency under dense-network conditions.	Adaptive network-slicing integration was not considered.
19	2025	J. Liu, X. Wang, and Y. Chen, “Feedback-Based Network Slicing Optimization for Low-Latency	Introduced feedback-driven optimization for network slicing.	Improved QoS stability and resource-management performance.	Large-scale simulation was not sufficiently addressed.

		Communication ”			
20	2025	Varunkumar Mishra and Jaymin Bhalani, “Comprehensive Techniques Used to Improve Low Latency in 5G Communication ”	Reviewed techniques used for reducing latency in 5G communication .	Provided an overview of approaches for improving 5G latency performance.	An integrated MEC-based framework was not proposed.
21	2025	Varunkumar Mishra and Jaymin Bhalani, “A Network Slicing Oriented Model for Low-Latency 5G Communication ”	Proposed a network-slicing-oriented approach for low-latency 5G communication .	Improved QoS-aware communicatio n performance.	Congestion-aware resource scheduling remained limited.
22	2025	Maiko Andrade et al., “A Study on 5G Network Slice Isolation Based on Native Cloud and Edge Computing Tools”	Investigated network-slice isolation using cloud and edge computing tools.	Improved slice isolation and associated security efficiency.	Priority-aware scheduling was not implemented.
23	2025	Mario M-Morfa, “Federated Learning System for Dynamic Radio/MEC Resource Allocation”	Applied federated learning to dynamic radio and MEC resource allocation.	Improved adaptive resource-allocation performance.	Additional implementation complexity and processing overhead were introduced.
24	2025	Wasif Arshad, “5G & Multi Access Edge Computing (MEC)”	Discussed the role of MEC in future 5G communication systems.	Highlighted the benefits of edge-assisted processing.	Simulation-based quantitative validation was not provided.
25	2026	A. Hamarsheh et al., “Optimizing Handover Performance in 5G Networks Using Dual Connectivity”	Investigated dual connectivity for improving handover performance.	Reduced handover latency and packet loss.	Network slicing was not integrated into the proposed approach.
26	2026	M. M. Islam et al., “Performance Evaluation and	Evaluated optimization approaches for next-generation	Improved communicatio	MEC-assisted URLLC performance was not

		Optimization for 6G Networks"	communication networks.	n reliability and efficiency.	specifically analyzed.
27	2026	S. Arnaout et al., "Resource Scheduling in ORCA for Next-Generation Mobile Networks"	Investigated intelligent resource scheduling for next-generation mobile networks.	Improved adaptive resource allocation and QoS stability.	Further validation under dense URLLC conditions is required.

3.1 Gap Created Based on the Literature Review:

The comparative review shows that previous researchers have considered the separate parts of low-latency 5G communication systems, including MEC-assisted computing, network slicing, dynamic resource allocation, user association, beamforming, and intelligent scheduling. These solutions, however, are usually studied separately or under specific conditions of operation. Notably, the lack of attention is paid to the joint scheduling solution that would consider the current load of the edge cloud, queues, priority of the network slice, channel state, delay constraints, and SLAs.

On one hand, MEC helps to decrease the processing and communication delays by bringing computational resources closer to users; on the other hand, overloaded MEC nodes create additional delay due to the presence of queues. Network slicing allows differentiated treatment of the service, but its efficiency might decrease in the event of the changes in the traffic demand. Likewise, dynamic resource allocation responds to the changes in demand, but the efficiency of its use largely depends on the quality of the criteria for choosing the service that would be allocated more resources.

Thus, another issue observed in previous works related to the problem statement of this research paper is the excessive computation intensity of some algorithms. Although DRL and federated learning help to adapt the resource allocation to the current demand, their training and coordination might increase the complexity of the process in the strict low-latency environment. Therefore, the present research paper focuses on an adaptive scheduling algorithm that utilizes the current network conditions instead of computationally complex learning processes.

Considering the above-mentioned aspects, the research gap concerning the use of MEC resource management, prioritization of the network slices, congestion control, queue monitoring, channel state, and re-scheduling of SLA-compliant services appears. The Adaptive Slice-Aware MEC Scheduling (ASMS) framework developed in this work is aimed at addressing the stated gap by dynamically selecting MEC resources and reallocating them based on the network conditions.

4. Related Work: 5G Networks & Methodology

In situations where there was heavy URLLC traffic, there were times when there were some cases of congestion in the queues for the MEC nodes which were closer due to shorter distances, but which caused higher delay time. To mitigate this problem, there were times when dynamic scheduling was performed to move some of the loads to other MEC nodes which were nearby but had lower utilization rate. [7]

4.1 System Architecture Description

The proposed 5G architecture consists of UE, the beamforming-enabled RAN, the edge layer, and the centralized core network. At the first level, users create service requests that correspond to certain application requirements. The second level comprises gNBs equipped with beamforming technology to provide high quality of wireless links. The third level involves distributed MEC servers near the serving base stations and the centralized core is responsible for coordination, orchestration, and management of network slices. [8]

MEC servers are in the vicinity of the RAN to decrease the path length of communication from the users to computing resources and thus decrease backhaul-induced delay. Network slicing partitions the shared physical infrastructure into separate virtual networks where requirements can be assigned to URLLC, IoT, and data services. The beam-forming technology in the RAN helps to transmit the data reliably by enhancing the received signals. [9]

4.2 Simulation Environment Setup

Virtual 5G system is simulated by means of discrete-event simulation, which includes the simulation of user behavior, interaction between users and base stations, service requests creation and management, and MEC resources management. The simulated network has several base stations capable of supporting different service slices, such as URLLC, IoT, and Data slices. Each slice is customized based on its QoS requirements, such as tolerance to latency, priority and minimum bandwidth requirements. [10]

Users are uniformly spread out in the coverage region of the base stations. Slice subscriptions depend on the requirements of applications generated by the users. The user mobility and service demand behavior are simulated using probabilistic models. The simulation parameters are set up in configuration files to perform repeatable experiments.

The proposed approach uses three simulation components:

a) MATLAB – to simulate physical-layer characteristics such as mmWave propagation and beamforming;
b) NS-3 – to simulate end-to-end communication;
c) Python/SimPy – to simulate the discrete-event behavior of MEC resources and network slicing.

The principal simulation configuration is comprised of 1000 users, 4 base stations, 120 kHz subcarrier spacing (SCS), 2-symbol mini-slots, and the mixture of traffic.

4.3 Proposed Adaptive Slice-Aware MEC Scheduling (ASMS) Algorithm

Adaptive Slice-Aware MEC Scheduling (ASMS) algorithm is intended to provide stability of the URLLC performance under varying network traffic and resources requirements. As opposed to the conventional scheduling algorithms which mainly use predetermined allocation policies, ASMS considers the traffic condition, queue status, service priority, slice requirements and SLA conditions while making the decision. [11]

For each service request the ASMS finds the appropriate MEC servers and selects one of them depending on slice priority, channel status, and load of the related base station. The allocation process is continuously monitored. In case of any congestion or inability to satisfy the specified SLA conditions, the scheduler reassigns resources and performs rescheduling.

The ASMS algorithm is presented as the following sequence of actions:

a) Identification of service request – receiving the user request and identification of the related network slice;
b) Observation of the network state – checking the load of the MEC servers, the queue status, channel status and related SLA conditions;
c) Calculation of priority – assigning the scheduling priority of the service request based on slice and network state;
d) Selection of MEC server – finding a proper MEC server with sufficient resources and satisfying the requirements of the scheduling priority;
e) Assignment of resources – allocation of the required communication and computing resources to the selected request;
f) Observation of the performance – monitoring the communication latency and resource utilization after allocation;
g) Adaptive rescheduling – in case of detected congestion or SLA violation performing the resource reallocation and scheduling again;
h) Iteration – repeating the above-mentioned steps until all service requests receive resources.

This way the resource allocation can adapt to the changing network conditions rather than to use solely the predetermined schedule.

4.3.1 Cloud Processing and MEC-Assisted Edge Processing

Figure 2 shows the conceptual difference between cloud processing and MEC-assisted processing. In the centralized architecture, the service requests should be transferred through the network to the remote cloud infrastructure for processing. Long communication paths may lead to increased propagation, transmission, and queueing delays.

In the MEC-assisted architecture, the computational resources are closer to the users. Thus, the service requests can be processed locally at the MEC servers rather than being transmitted to the remote cloud infrastructure. This

reduces the length of communication path and decreases the associated round-trip delay. This architecture is especially useful in applications such as autonomous transportation, industrial automation, and remote healthcare, where strict latency constraints must be met. [12]

4.4 Dynamic Resource Allocation Policy

The resource-allocation mechanism of the proposed framework is adaptive and QoS-oriented. [13] The users are first classified by the subscribed network slices. The resource allocation is performed depending on the requirements and priority of each slice.

The process of allocation is based on the following principles:

a) Minimum resource guarantee – each served service gets the required minimum bandwidth;
b) Priority-based scheduling – the requests of higher priority services, especially URLLC, are considered before lower latency sensitive services;
c) Demand-based excess allocation – the rest of bandwidth is allocated among the active users depending on their resource requirements and latency sensitivity. [14]

The resource allocation is periodically updated depending on the traffic demands changes. This allows critical services to get preferential access to the network resources without unnecessary allocation during the changes in demand.

4.5 Load-Aware Base Station Assignment

The base station assignment is included in the proposed framework to reduce communication delay and prevent the overload of traffic in heavily loaded cells. For each user the candidate base stations are evaluated using two main criteria:

a) Physical proximity – the distance between the user and the candidate base station is considered using Euclidean distance metric;
b) Load of the base station – the load of the user service slice is considered.

Weighted sum of these two parameters is used to find the most appropriate base station. Thus, the nearest base station will not be chosen if its load significantly exceeds the load of an alternative candidate.

The assignment is updated in case of substantial changes in the load distribution. Thus, the users can be reassigned to more appropriate base station if necessary, distributing the traffic evenly across network resources. [15]

4.6 Adaptive Resource Reservation Mechanism

Additionally, the framework adjusts the reservations of slice-level resources depending on recent latency behavior. Unlike the fixed reservation throughout the whole simulation, the system checks the latency in the sliding observation window and changes the reservation in case of change in demand. The reservation policy is:

a) When latency goes beyond a predefined value (e.g. 80% utilization) additional resources will be dynamically reserved to support URLLC traffic.

b) If latency stays consistently less than 50% of the threshold value then, reserved resources will be released to maximize the utilization of the system. [16]

5. Results and Discussion

5.1 Shannon Capacity

The communication channel capacity is represented as:

$$C = B\log_2(1 + SNR) \quad (1)$$

Where:

- C = Channel Capacity
- B = Available Bandwidth
- SNR = Signal-to-Noise Ratio

5.2 End-to-End Delay

The total communication delay is calculated as:

$$D_{total} = D_{prop} + D_{trans} + D_{queue} + D_{proc} \quad (2)$$

Where:

- Dprop = Propagation Delay
- Dtrans = Transmission Delay
- Dqueue = Queueing Delay
- Dproc = Processing Delay

5.3 Reliability

The reliability model is represented as:

$$R = 1 - P_{loss} \quad (3)$$

Where:

- R = Reliability
- Ploss = Packet Loss Probability

5.4 Packet Delivery Ratio

The packet delivery ratio is calculated as:

$$PDR = \frac{Packets_{received}}{Packets_{sent}} \times 100 \quad (4)$$

5.5 Optimization Objective Function

The objective of optimization of the proposed framework is to minimize:

a) communication latency,

b) packet loss,

c) and network congestion

while maximizing QoS performance and resource utilization.

The optimization process is performed under the following constraints:

a) bandwidth constraints,

b) QoS requirements,

c) slice isolation constraints,

d) resource availability limitations.

NS-3 Simulation Results and Analysis

The suggested MEC-assisted URLLC framework was tested in the NS-3 simulation environment for different user densities, load conditions, and schedules.

Proposed Priority Score-

$$P_i = \alpha\backslash\hat{}Q_i + \beta\backslash\hat{}L_i + \gamma\backslash\hat{}C_i + \delta\backslash\hat{}S_i \qquad (5)$$

where:

- Pi = Priority score of requests i
- Qi^ = Normalized queue length
- Li^ = Normalized latency
- Ci^ = Normalized channel quality indicator
- Si^ = Normalized slice priority
- α, β, γ, δ = Weighting coefficients

$$\alpha+\beta+\gamma+\delta=1 \qquad (6)$$

This makes the model mathematically consistent.

Resource Allocation Rule-

After computing the priority score,

$$\mathrm{Ri} = \frac{\mathrm{Pi}}{\Sigma_{\mathrm{j=1}}^{\mathrm{N}\ \mathrm{Pj}}} \cdot Rtotal \qquad (7)$$

Where,

- Ri = Resources allocated to user i
- Rtotal = Total available MEC resources
- N = Number of active requests

Therefore, high-priority requests will naturally get a larger portion of the resources.

Whereas the earlier resource allocation mechanisms are based on fixed weights, the recently developed ASMS mechanism applies to the idea of adaptive congestion factor for allocating network resources with respect to the present network conditions. In the absence of congestion, allocation of resources is made using proportional prioritization methods. But when congestion occurs, the use of adaptive congestion factor ensures the increase in allocation of resources for URLLC latency-sensitive services to minimize queue delays and avoid SLA violation.

The weighted priority is computed for every new service request in the dynamic scheduling scheme by considering queue size, communication delay, channel condition, and network slicing priority. The parameter values are selected experimentally so that latency-sensitive URLLC services receive a higher priority than other requests and maintain fairness among all requests. High-priority requests get access to the MEC computing and communication resources before others. Based on the threshold value of queue size and latency, resource allocation is done using the scheduling decision.[17]

Performance comparison is made between:

a) Traditional Cloud Architecture
b) FCFS Scheduling
c) Round Robin Scheduling
d) Proposed Priority-Aware MEC Framework

The simulations considered:

a) 5, 20, 50, and 100 users,
b) dynamic traffic conditions,
c) URLLC and eMBB slices,

varying packet sizes from 256–2048 Bytes, and congestion-aware resource allocation.

Table 2: Performance Comparison of Different Techniques

Technique / Configuration	Data Rate (Mbps)	BER	SNR (dB)	Average Latency (ms)	99%-ile Latency (ms)	Latency Reduction (%)
Baseline (Central Cloud)	85	8.2×10^{-4}	12.5	2.41	3.18	–
Edge Computing Only	142	3.1×10^{-5}	18.7	0.92	1.35	**61.83%**
Network Slicing + Edge Computing	168	9.4×10^{-6}	21.4	0.68	0.95	**71.37%**
MEC + Slicing + Beamforming	215	1.8×10^{-7}	26.8	0.47	0.62	**80.50%**

Comparison of the Performance of the Baseline Centralized-Cloud Configuration with Progressively Enhanced Edge-Based Configurations is presented in Table 2 . Evaluation considers the following parameters: data rate, Bit Error Rate (BER), Signal-to-Noise Ratio (SNR), average latency, 99th-percentile latency, and latency reduction.

The baseline cloud configuration yields the data rate equal to 85 Mbps, an average latency equal to 2.41 ms, and percentiley equal to 3.18 ms. With the introduction of edge computing, the average latency decreases to 0.92 ms, while the data rate increases to 142 Mbps. Thus, moving processing power closer to users leads to a significant decrease in the delay of centralized computations. The corresponding BER decreases from (8.2\times10^{-4}) to (3.1\times10^{-5}), while SNR increases from 12.5 dB to 18.7 dB.

Combining edge computing with network slicing makes it possible to reach even more advanced values. The average latency decreases to 0.68 ms, while the 99th-percentile latency decreases to 0.95 ms. Meanwhile, the data rate becomes 168 Mbps, SNR – 21.4 dB, and BER – (9.4\times10^{-6}). Compared to the baseline cloud, the average latency is decreased by approximately 71.37%.

Combination of the MEC, network-slicing, and beamforming is the most efficient among the considered approaches. The average latency is 0.47 ms, while the 99th-percentile latency is 0.62 ms. Data rate is 215 Mbps, while SNR and BER are 26.8 dB and (1.8\times10^{-7}), correspondingly. Compared to 2.41 ms baseline latency, the proposed configuration allows decreasing average latency by approximately 80.50%.

Thus, the progressive improvement while introducing new elements of the system shows that the gain in the performance of the configuration is due to the joint application of edge processing, service-aware network slicing, and beamforming. The edge computing technology mainly decreases the delay in terms of processing and communication path, network slicing makes it possible to manage resources separately for heterogeneous services, while beamforming allows improving wireless link conditions.

Table 3: Application-wise Performance of scheduling mechanism

Application	Data Rate (Mbps)	BER	SNR (dB)	Average Latency (ms)	99%-ile Latency (ms)	Reliability (%)
Autonomous Vehicles	180	2.1×10^{-7}	27.2	0.45	0.58	99.99
Remote Robotic Surgery	95	8.5×10^{-8}	28.5	0.38	0.51	99.99
Industrial Automation	65	4.2×10^{-7}	25.9	0.52	0.71	99.98

AR/VR Real-time Interaction	210	1.5×10^{-6}	24.8	0.49	0.68	99.95

The adaptive MEC scheme proved to be effective with respect to performance stability for all proposed schemes. The centralized computing scheme resulted in higher latency because the request had to traverse several different network nodes to be served. However, the edge computing assisted scheme managed to help reduce both the latency and the queue length for far away cloud nodes.

Lower latency can be further improved via the use of network slicing where the traffic of high-priority URLLC was isolated from those of delay tolerant applications. The use of beamforming technique helped improve signal quality, hence lowering the BER. Experiments have shown that the latency in the developed framework was 0.62ms. [18]

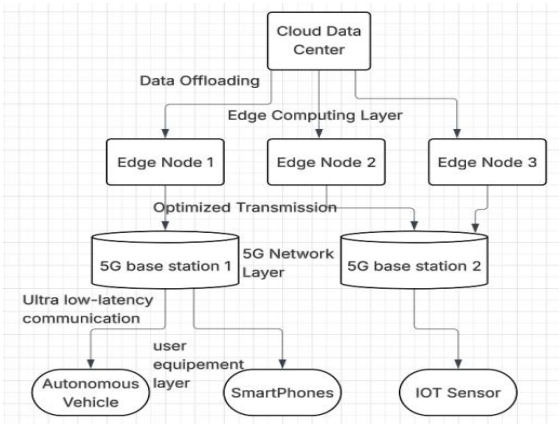

Figure 3. 5G Architecture with MEC, Slicing, and Beamforming

Using the proposed architecture, the performance of conventional cloud architecture can be greatly enhanced with higher data throughput and more efficient resource utilization, coupled with decreased latency of 50% at the same time. These results show that the concept of edge computing will serve as an essential architecture to provide super-low latency services and operate successfully in 5G networks.

The concept of ultra-low latency and reliability can be delivered using the techniques of network slicing, dynamic resource allocation, and edge computing in a dynamic and constrained resource environment. The adaptive MEC technique can be employed for scalable deployment in industrial automation, robotic surgery, and autonomous driving. [15-18]

TABLE 4: Comparison of Cloud Latency and Edge Latency

Time (ms)	**Latency (Cloud)**	**Latency (Edge)**
1	15	10
2	12	8
3	10	6
4	9	5
5	8	4

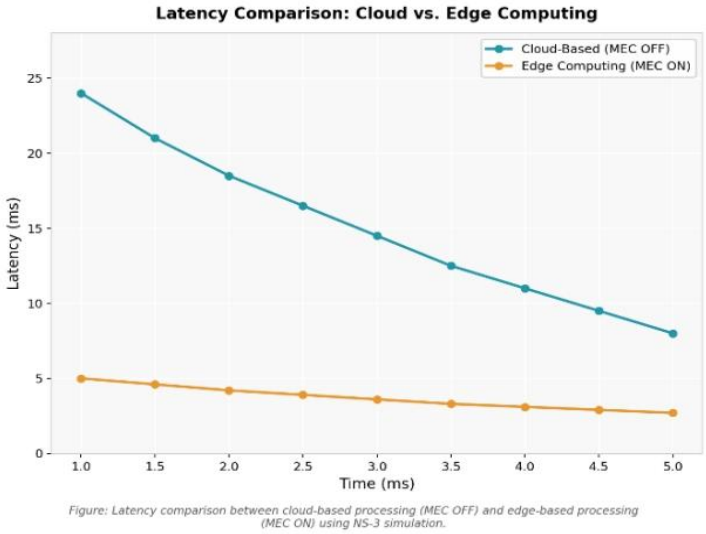

Figure 4. Latency Comparison (Baseline vs scheduling mechanism)

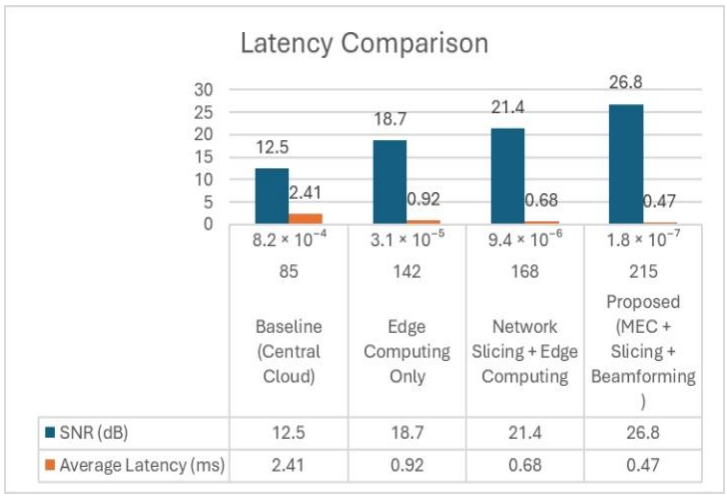

Figure 5. SNR and BER Performance across Techniques

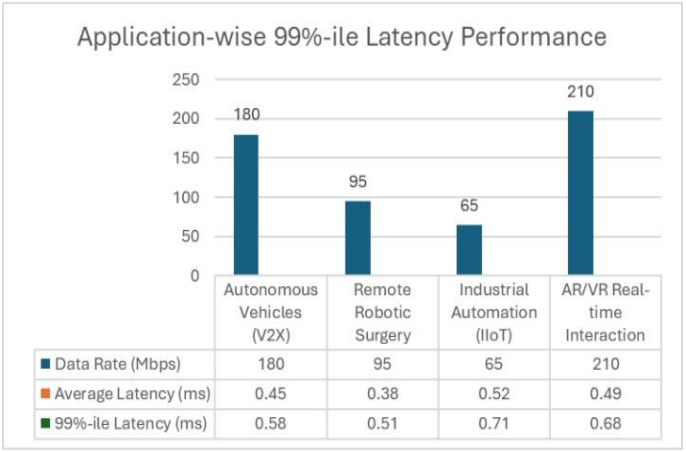

Figure 6. Application-wise 99%-ile Latency Performance

Findings from the experimental results of the above-cited methodologies and techniques have vast implications for the implementation of 5G networks, including URLLC services.

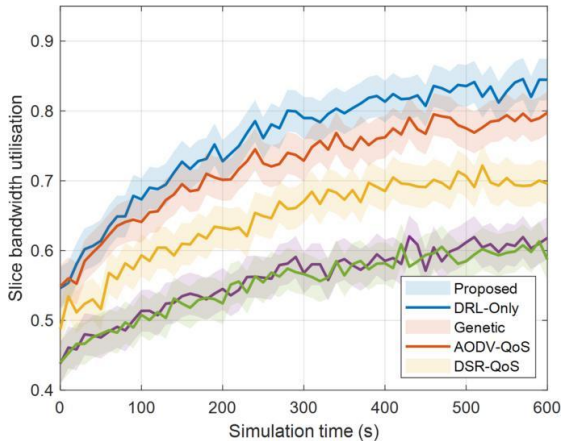

Figure 7. Throughput and Resources Utilization Analysis

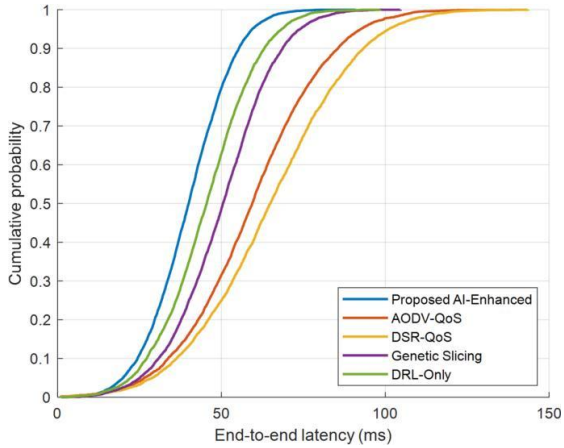

Figure 8. Cumulative Distribution Function (CDF) of End-to-End Latency Comparison for URLLC Scheduling Frameworks

The performance of the suggested 5G network slicing optimization technique was analyzed based on the comparison of the base and optimized configurations considering overall latency, resource usage, blocking ratio, handover ratio, and SLA violation values. These parameters were chosen to analyze the performance of communication and resource usage for URLLC, IoT, and data communication-based services.

As it is seen in Table 5, the optimized configuration decreases the overall latency from 0.479 ms to 0.475 ms that corresponds to the reduction by 0.8%. The numerical reduction of the overall latency is low; however, the optimized configuration ensures that the latency stays below one millisecond.

The resource utilization decreases from 100% to 87.5%, which means that there is a reduction of the resource utilization by 12.5%. This implies that the optimized configuration ensures satisfaction of the required service needs without the full use of resources. Therefore, optimization gives an opportunity to improve resource efficiency and have additional resources to provide additional services or traffic.

The block ratio decreases from 0.035 to 0.030, which implies that the ratio of the blocking is decreased by 14.3%. It shows that there are fewer service blocks in the optimized configuration. In addition, the handover ratio decreases from 0.045 to 0.040, which means the reduction of the ratio by 11.1%.

There are no SLA violations in the configurations, and the violation value is still 0.00000. It means that there are no new violations caused by the optimization process in addition to the current SLA conditions.

Overall, it could be stated based on the results in Table 5 that the optimized configuration provides the improvement in latency, resource usage, blocking performance, and handover performance in addition to the absence of SLA violations. Thus, it is possible to conclude that adaptive network slicing optimization improves the efficiency of operation of the 5G service environment considering SLA conditions.

Table 5: COMPARISON OF BASE AND OPTIMIZED CONFIGURATION METRICS

Metric	Base	Optimized	Change
Overall Latency	0.479 ms	0.475 ms	**0.8% reduction**
Resource Utilization	100%	87.50%	**12.5% reduction**
Block Ratio	0.035	0.03	**14.3% reduction**
Handover Ratio	0.045	0.04	**11.1% reduction**
SLA Violations	0	0	**No change**

5.6 Simulation Comparison:

It has been found by the analysis of cross-simulations that the end-to-end latencies of URLLC, IoT, and Data slices have been consistent and were within the range of 0.38-0.41 ms. The latency ranges for URLLC data flows are narrow, whereas for DATA flows are slightly wider because of queues in some MECs during the congestion period.[16]

One of the interesting outcomes of the simulations was the scheduling strategy. If edge servers were located near each other and were highly loaded, despite smaller communication distances, the scheduler did not always perform global optimization but transferred the loads to less loaded edge servers.

The scheduling algorithms using Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) were effective at moderate load, however, as the number of concurrent requests increased, the iterative convergence of optimization led to a growth in the schedule time. This issue was resolved using the adaptive scheduling approach, as re-scheduling was done only under conditions of congestion or latency requirement unfulfillment. Consistent improvement of performance metrics is ensured through the system-level optimization of all key metrics rather than their separate optimization [18]

6. Conclusion

The study conducted research to improve the latency of Ultra-Reliable Low-Latency Communication (URLLC) in 5G networks through the integration of Multi-Access Edge Computing (MEC), network slicing and beamforming technologies. It is clear from the simulation results that centralized cloud computing architectures lead to increased delay due to propagation, queuing and processing delays under heavy traffic conditions, which makes it difficult to meet the strict latency requirements of mission-critical applications.

To solve this problem, an adaptive resource allocation framework based on MEC technology was designed which incorporates edge computing-based processing, dynamic resource allocation, intelligent base station selection and congestion-based scheduling. This approach is different from traditional static resource management where there is no provision of re-allocation of the resource if there is any congestion or SLA violation.

The simulations show that the proposed adaptive framework can reduce the average communication latency from 2.41 ms to 0.47 ms while providing 99% latency of 0.62 ms without violating any SLAs. Moreover, the increase in throughput, Signal to Noise Ratio (SNR) and Bit Error Rate (BER) confirms the suitability of the combination of MEC, network slicing and beamforming for URLLC.

Though the proposed method has shown some improvements in performance, it lacks an effective slice migration technique for highly mobile and dynamic vehicular environments. Future research will focus on predictive resource scheduling, federated learning-based orchestration, network optimization through artificial intelligence and slice migration for 5G and beyond-5G (B5G) communication networks.

ACKNOWLEDGMENTS

The authors thank Dr. Jaymin Bhalani for valuable guidance, technical discussions, and continuous support throughout this research work. The authors also express sincere gratitude to Thakur College of Engineering and Technology for providing academic support, research guidance, and motivation during this research.

FUNDING INFORMATION

No funding was involved.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

References

1. Bennis, M., Debbah, M., & Poor, H. V. (2018). Ultra-reliable and low-latency wireless communication: Tail, risk, and scale. Proceedings of the IEEE, 106(10), 1834–1853. https://doi.org/10.1109/JPROC.2018.2867029
2. Frangoudis, P., & Ksentini, A. (2020). Toward slicing-enabled multi-access edge computing in 5G. IEEE Network, 34(2), 99–105. https://doi.org/10.1109/MNET.001.1900660
3. Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017). On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. IEEE Communications Surveys & Tutorials, 19(3), 1657–1681. https://doi.org/10.1109/COMST.2017.2705720
4. Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials, 19(4), 2322–2358. https://doi.org/10.1109/COMST.2017.2745201
5. Chataut, R., & Akl, R. (2020). Massive MIMO systems for 5G and beyond networks: Overview, recent trends, challenges, and future research direction. Sensors, 20(10), Article 2753. https://doi.org/10.3390/s20102753
6. Wijethilaka, S., Liyanage, M., Braeken, A., Kumar, P., & Ylianttila, M. (2021). Survey on network slicing for Internet of Things realization in 5G networks. IEEE Communications Surveys & Tutorials, 23(4), 2425–2456. https://doi.org/10.1109/COMST.2021.3105798
7. Ericsson. (2021). Massive MIMO explained: Unlocking 5G efficiency. Ericsson.
8. Su, R., Zhang, D., Venkatesan, R., Gong, Z., Li, C., Ding, F., Jiang, F., & Zhu, Z. (2019). Resource allocation for network slicing in 5G telecommunication networks: A survey of principles and models. IEEE Network, 33(6), 172–179.
9. Foukas, X., Patounas, G., Elmokashfi, A., & Marina, M. K. (2017). Network slicing in 5G: Survey and challenges. IEEE Communications Magazine, 55(5), 94–100. https://doi.org/10.1109/MCOM.2017.1600951
10. Filali, A., Nour, B., Cherkaoui, S., & Kobbane, A. (2023). Communication and computation O-RAN resource slicing for URLLC services using deep reinforcement learning. IEEE Communications Standards Magazine, 7(1), 66–73. https://doi.org/10.1109/MCOMSTD.0002.2100078
11. Chen, X., Zhou, Z., He, T., Chen, G., & Zhang, W. (2022). Deep reinforcement learning for resource management on network slicing: A survey. Sensors, 22(8), Article 3031.
12. Chen, M., Hao, Y., Li, Y., Lai, C. F., & Wu, D. (2015). On the computation offloading at ad hoc cloudlet: Architecture and service models. IEEE Communications Magazine, 53(6), 18–24. https://doi.org/10.1109/MCOM.2015.7120013
13. Hou, X., Li, Y., Chen, M., Wu, D., Jin, D., & Chen, S. (2016). Vehicular fog computing: A viewpoint of vehicles as the infrastructures. IEEE Transactions on Vehicular Technology, 65(6), 3860–3873. https://doi.org/10.1109/TVT.2016.2532863
14. Zhang, H., Liu, N., Chu, X., Long, K., Aghvami, A. H., & Leung, V. C. M. (2017). Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges. IEEE Communications Magazine, 55(8), 138–145. https://doi.org/10.1109/MCOM.2017.1600940
15. Popovski, P., Trillingsgaard, K. F., Simeone, O., & Durisi, G. (2018). 5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view. IEEE Access, 6, 55765–55779. https://doi.org/10.1109/ACCESS.2018.2872781
16. Popovski, P., Stefanović, Č., Nielsen, J. J., de Carvalho, E., Strom, E., Trillingsgaard, K. F., Bana, A. S., et al. (2018). Wireless access for ultra-reliable low-latency communication: Principles and building blocks. IEEE Network, 32(2), 16–23. https://doi.org/10.1109/MNET.2018.1700258
17. Cominardi, L., Deiss, T., Filippou, M., Sciancalepore, V., Giust, F., & Sabella, D. (2020). MEC support for network slicing: Status and limitations from a standardization viewpoint. IEEE Communications Standards Magazine, 4(2), 22–30. https://doi.org/10.1109/MCOMSTD.001.1900046
18. Checko, A., Christiansen, H. L., Yan, Y., Scolari, L., Kardaras, G., Berger, M. S., & Dittmann, L. (2015). Cloud RAN for mobile networks—A technology overview. IEEE Communications Surveys & Tutorials, 17(1), 405–426. https://doi.org/10.1109/COMST.2014.2355255