ISSN: 2229-6948(ONLINE)
DOI: 10.21917/ijct.2026.0564

MULTI-AGENT REINFORCEMENT LEARNING FRAMEWORK FOR DYNAMIC RESOURCE SLICING AND ADAPTIVE ALLOCATION IN 6G NETWORK CORE

Sreedevi Kadiyala[1] and Chandra Srinivas Potluri[2]

[1]Department of Computer Science and Engineering, Guru Nanak Institutions Technical Campus, India
[2]Department of Computer Science and Engineering, Siddhartha Institute of Engineering and Technology, India

Abstract

The rapid expansion of heterogeneous services in sixth generation (6G) communication networks has increased the complexity of resource orchestration within the network core. Emerging applications such as autonomous systems, immersive communication, and large-scale Internet of Things environments have required highly flexible and efficient resource slicing mechanisms. Conventional resource allocation techniques have relied on static or semi-dynamic policies that have limited adaptability to fluctuating traffic patterns and diverse quality of service requirements. As the network scale has grown and service diversity has intensified, these approaches have faced challenges in maintaining efficient utilization and service reliability. Consequently, the dynamic management of network resources has remained a critical issue in the evolving 6G infrastructure. This study has investigated a dynamic resource slicing mechanism that has utilized Multi-Agent Reinforcement Learning based Adaptive Resource Slicing (MARL-ARS) for the 6G network core environment. The proposed framework has introduced multiple intelligent agents that have interacted with the network environment and that have cooperatively optimized the allocation of bandwidth, computational capacity, and storage resources across different network slices. Each agent has learned an optimal allocation policy through continuous interaction with the system state, while the cooperative learning structure has enabled coordinated decision making among distributed agents. The reinforcement learning mechanism has incorporated reward optimization strategies that have considered network latency, resource utilization efficiency, and service reliability. Through iterative learning, the model has gradually refined its slicing policies and has achieved adaptive resource allocation under varying traffic loads and service demands. The experimental results demonstrate that the proposed MARL-DRS framework significantly improves the performance of dynamic resource slicing in the 6G network core. The system achieves 93% resource utilization under high network load conditions, while the baseline approaches achieve between 78% and 85% utilization. The proposed model also improves the network throughput to 8.6 Gbps, which exceeds the existing approaches that achieve 6.6–7.5 Gbps. The slice allocation accuracy reaches 94% after 35 training episodes, which indicates that the cooperative learning agents effectively interpret the network state and allocate resources accordingly. In addition, the framework reduces the network latency to 35 ms under heavy traffic conditions and maintains a 96% QoS satisfaction rate across heterogeneous service slices.

Keywords:

6G Network Core, Multi-Agent Reinforcement Learning, Dynamic Resource Slicing, Intelligent Resource Allocation, Network Optimization

1. INTRODUCTION

The evolution of wireless communication systems has significantly transformed the structure and operational capabilities of modern network infrastructures. The transition from fifth generation networks toward the sixth generation (6G) paradigm has introduced an unprecedented demand for ultra-reliable, intelligent, and highly adaptive network management mechanisms. 6G communication architectures aim to support massive connectivity, extremely low latency, and high data throughput, which have become essential for emerging applications such as autonomous vehicles, immersive extended reality services, industrial automation, and large-scale Internet of Things ecosystems. These applications have required flexible resource management frameworks that can dynamically adapt to diverse traffic patterns and service-level requirements. Consequently, the concept of network slicing has emerged as a fundamental architectural feature that enables the partitioning of a physical network into multiple logical slices, each of which has been optimized for specific service demands [1].

Network slicing has provided an efficient mechanism that allows service providers to allocate computational, storage, and communication resources in a flexible manner across heterogeneous services. The 6G network core architecture has incorporated virtualization technologies, software-defined networking, and cloud-native infrastructures that have enabled dynamic resource provisioning across multiple slices. Such a design has significantly improved the scalability and operational flexibility of network infrastructures. Several studies have examined resource orchestration mechanisms that have supported efficient slice creation and management in next-generation networks [2]. These mechanisms have emphasized intelligent decision-making frameworks that can allocate resources dynamically while maintaining service quality and system stability. In this context, artificial intelligence and machine learning techniques have become promising tools that can support autonomous network management and optimization.

Recent advancements in reinforcement learning have further expanded the potential of intelligent network management systems. Reinforcement learning techniques have allowed network controllers to learn optimal decision policies through interaction with dynamic environments. The integration of reinforcement learning with network slicing frameworks has enabled the system to continuously observe network conditions, evaluate performance metrics, and adapt resource allocation policies accordingly. Furthermore, the multi-agent paradigm has introduced distributed learning capabilities that allow multiple decision entities to collaborate and optimize resource allocation simultaneously across different network slices. These intelligent frameworks have significantly enhanced the adaptability and efficiency of resource management in complex network environments [3].

Despite the technological advancements in network slicing and intelligent network management, several critical challenges still remain in the deployment of dynamic resource slicing mechanisms within the 6G network core. One of the primary challenges arises from the high variability of network traffic patterns and service demands. Modern communication environments have supported heterogeneous applications that exhibit significantly different quality of service requirements. For example, autonomous control systems have required ultra-low latency communication, while multimedia streaming services have prioritized high bandwidth availability. Such heterogeneity has introduced complexity in resource allocation strategies that must simultaneously satisfy multiple performance constraints [4].

Another challenge has emerged from the distributed nature of modern network infrastructures. The integration of edge computing, cloud platforms, and virtualized network functions has created highly decentralized network architecture. Resource allocation decisions must therefore consider multiple interacting components that operate across different layers of the network. Traditional centralized optimization mechanisms have struggled to scale efficiently under such complex environments. Moreover, the increasing number of network slices has created additional coordination challenges among resource management entities, which has further complicated the optimization process [5].

The limitations of existing resource allocation frameworks have highlighted several research problems in the context of 6G network core design. Many conventional resource slicing approaches have relied on rule-based or static allocation strategies that have lacked adaptability to rapidly changing network conditions. Such approaches have resulted in inefficient resource utilization and suboptimal service performance. Additionally, centralized control mechanisms have created bottlenecks that have limited the scalability of resource management systems in large-scale network environments [6].

Another significant problem has arisen from the inability of existing methods to capture the complex interactions among multiple network slices. In a real-world network environment, resource allocation decisions made for one slice may influence the performance of other slices. Conventional optimization algorithms have often treated slice management as independent tasks, which has ignored these interdependencies and has led to inefficient resource distribution [7]. Furthermore, the absence of cooperative decision-making mechanisms has prevented existing frameworks from achieving coordinated optimization across distributed network entities.

In addition, the increasing complexity of 6G networks has required adaptive learning models that can continuously refine their resource allocation strategies based on real-time network observations. Many existing machine learning approaches have focused on single-agent reinforcement learning models that have struggled to scale effectively in multi-slice environments. These models have encountered difficulties in maintaining stable learning performance when multiple decision variables and network constraints have been involved simultaneously [8]. Therefore, the development of a cooperative and scalable learning framework remains an essential requirement for efficient dynamic resource slicing in the 6G network core.

Based on these limitations, this study has investigated an intelligent resource slicing framework that has leveraged multi-agent reinforcement learning to optimize dynamic resource allocation within the 6G network core. The primary objective of this research has been to design a distributed learning architecture that enables multiple agents to collaboratively manage network resources across heterogeneous slices. The proposed framework aims to improve resource utilization efficiency, reduce network latency, and enhance service reliability through adaptive decision-making mechanisms. Another objective has focused on developing a learning-based resource orchestration strategy that can respond effectively to rapidly changing traffic conditions while maintaining coordination among multiple network slices.

The novelty of this work lies in the integration of cooperative multi-agent reinforcement learning with dynamic resource slicing mechanisms in the 6G network core. Unlike conventional approaches that rely on centralized optimization models, the proposed framework has introduced distributed agents that interact with the network environment and learn coordinated resource allocation policies. This collaborative learning structure has enabled the system to capture complex interdependencies among network slices while maintaining scalability and adaptability.

The major contributions of this research can be summarized as follows:

- This study has developed a multi-agent reinforcement learning-based dynamic resource slicing framework that enables intelligent allocation of bandwidth, computing power, and storage resources within the 6G network core.
- This work has introduced a cooperative learning mechanism that allows distributed agents to coordinate their decisions, which has improved overall network efficiency, slice allocation accuracy, and service quality under dynamic traffic conditions.

2. RELATED WORKS

Several studies have examined intelligent resource allocation mechanisms for next-generation communication networks. In recent years, researchers have investigated machine learning and reinforcement learning techniques to address the growing complexity of network slicing and resource orchestration in advanced wireless systems.

[9] have proposed a reinforcement learning-based resource allocation model for network slicing in software-defined networks. The authors have designed a Q-learning framework that has enabled dynamic bandwidth allocation across multiple service slices. The experimental results have demonstrated that the learning-based strategy has improved resource utilization compared with conventional heuristic allocation algorithms. However, the study has relied on a centralized learning model that has limited the scalability of the system in large-scale network environments.

[10] have introduced a deep reinforcement learning framework for slice resource management in 5G core networks. The proposed approach has utilized deep Q-networks to learn optimal resource allocation policies that adapt to varying traffic conditions. The system has incorporated network state observations such as bandwidth demand, latency requirements, and slice priority levels. Simulation results have shown that the approach has significantly improved service-level agreement satisfaction rates. Nevertheless, the single-agent learning structure has struggled to effectively capture interactions among multiple network slices.

[11] have investigated the application of multi-agent reinforcement learning for distributed network resource management. Their study has presented a cooperative learning model in which multiple agents have interacted with a shared network environment and have jointly optimized bandwidth allocation policies. The framework has demonstrated improved scalability compared with centralized reinforcement learning models. However, the study has primarily focused on edge computing environments rather than the complete 6G network core architecture.

[12] have proposed an intelligent network slicing framework that has integrated deep learning with software-defined networking controllers. The authors have designed a prediction-based resource management strategy that has estimated future network traffic patterns and has adjusted slice allocation accordingly. The predictive model has improved traffic management efficiency and has reduced network congestion. Despite these improvements, the approach has depended heavily on accurate traffic prediction models, which has limited its adaptability in highly dynamic network conditions.

[13] have examined resource orchestration techniques in cloud-native 6G core networks. Their research has presented a virtualization-based resource allocation model that has utilized containerized network functions for efficient service deployment. The framework has improved network flexibility and scalability by enabling dynamic slice instantiation. However, the resource allocation mechanism has relied on rule-based policies that have lacked intelligent learning capabilities.

[14] have proposed a distributed reinforcement learning model for adaptive resource management in ultra-dense wireless networks. The system has introduced decentralized agents that have learned optimal channel allocation strategies through environmental interaction. The experimental analysis has demonstrated that the distributed learning architecture has reduced interference levels and has improved network throughput. Nevertheless, the model has not explicitly addressed the complexity of resource slicing in 6G network cores.

[15] have explored artificial intelligence-driven network slicing techniques for future wireless communication systems. Their study has evaluated several machine learning algorithms that have optimized slice resource allocation under dynamic traffic loads. The results have indicated that learning-based strategies have significantly improved network efficiency compared with traditional optimization algorithms. However, the work has primarily focused on algorithmic comparisons and has not developed a cooperative multi-agent framework for coordinated resource slicing.

3. PROPOSED METHODOLOGY

This study has proposed a Multi-Agent Reinforcement Learning based Dynamic Resource Slicing Framework (MARL-DRS) for the 6G network core environment. The framework has utilized multiple cooperative agents that have interacted with the network infrastructure to dynamically allocate communication, computational, and storage resources among network slices. Each agent has observed the system state that has represented traffic demand, slice priority, and available infrastructure capacity. Through repeated interactions with the environment, the agents have learned an optimal allocation policy that has maximized the network performance objectives such as throughput, latency reduction, and efficient resource utilization. The cooperative learning mechanism has enabled distributed agents to share environmental feedback and that has improved the stability of the learning process. The overall framework has achieved adaptive slice management that has responded efficiently to heterogeneous service demands in the dynamic 6G network core.

The proposed framework has followed the following operational steps:

- Network Environment Initialization and Slice Formation
- State Space Representation and Observation Modeling
- Multi-Agent Policy Learning through Reinforcement Mechanism
- Dynamic Resource Allocation Optimization
- Cooperative Reward Evaluation and Policy Update
- Convergence and Adaptive Resource Slicing Deployment

3.1 NETWORK ENVIRONMENT INITIALIZATION AND SLICE FORMATION

The proposed framework begins with the initialization of the virtualized 6G network core environment that is the infrastructure responsible for dynamic service orchestration. The architecture consists of a set of physical nodes $N=\{n_1,n_2,...,n_N\}$that have provided computational, storage, and communication resources for multiple service slices. Each node has maintained resource capacities such as bandwidth B_n, computing power C_n, and storage capacity S_n. The network has supported a set of logical slices $S=\{s_1,s_2,...,s_K\}$, where each slice corresponds to a particular service category such as ultra-reliable low latency communication, massive machine-type communication, or enhanced mobile broadband. Each slice s_{kh} as required a minimum resource demand vector: $R_k=(b_k,c_k,s_k)$, where, b_k=bandwidth demand, c_k=computational demand, s_k=storage demand. The overall network resource capacity across nodes has been expressed as

$$B_{total} = \sum_{n=1}^{N} B_n \quad (1)$$

$$C_{total} = \sum_{n=1}^{N} C_n \quad (2)$$

$$S_{total} = \sum_{n=1}^{N} S_n \quad (3)$$

During the initialization phase, the slicing controller has partitioned the physical infrastructure into logical resource pools. These pools have supported dynamic allocation for incoming service requests. The slice allocation matrix has been defined as

$$A = \begin{bmatrix} a_{11} & a_{12} & ... & a_{1K} \\ a_{21} & a_{22} & ... & a_{2K} \\ ... & ... & ... & ... \\ a_{N1} & a_{N2} & ... & a_{NK} \end{bmatrix} \quad (4)$$

where a_{nk} is the portion of resources allocated from node n to slice k. The initialization process has ensured that

$$\sum_{k=1}^{K} a_{nk} \leq R_n \tag{5}$$

This guarantees that the allocation does not exceed the available node capacity. Through this structured representation, the system establishes the operational environment in which learning agents can observe resource availability and service demand.

3.2 STATE SPACE REPRESENTATION AND OBSERVATION MODELING

The next stage constructs the state representation that enables intelligent agents to observe the network environment. The state vector captures the dynamic characteristics of network slices and infrastructure resources. The global network state at time $t$is represented as

$$S_t = \{D_t, U_t, L_t, C_t\} \tag{6}$$

where

D_t=slice demand vector

U_t=resource utilization level

L_t=latency conditions

C_t=infrastructure capacity state

The slice demand vector has been expressed as

$$D_t = [d_1(t), d_2(t), ..., d_K(t)] \tag{7}$$

where $d_K(t)$ denotes the traffic demand generated by slice *k*.

Resource utilization is calculated as

$$U_t = \frac{\sum_{k=1}^{K} R_k(t)}{R_{total}} \tag{8}$$

where $R_k(t)$ is resources allocated to slice kat time *t*.

Network latency, which strongly influences service quality, has been estimated as

$$L_t = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{\lambda_k}{\mu_k - \lambda_k}\right) \tag{9}$$

where, λ_k=arrival rate of slice traffic and μ_k=service rate of the network slice

The observation function for each agent is defined as

$$O_i(t) = f(S_t, R_i) \tag{10}$$

where R_i is the local resource view available to agent *i*.

This modeling approach allows each learning agent to obtain a partially observable representation of the global system state. Such representation is essential for decentralized decision making in a distributed network core environment.

3.3 MULTI-AGENT POLICY LEARNING THROUGH REINFORCEMENT MECHANISM

The proposed framework utilizes multi-agent reinforcement learning in which several autonomous agents interact with the environment to learn optimal resource allocation policies. Each agent $i$maintains a policy function $\pi_i(a_i \mid s)$ which determines the probability of selecting action $a_i$given the observed state s.

The objective of the reinforcement learning process is to maximize the expected cumulative reward

$$J(\pi) = E_\pi\left[\sum_{t=0}^{T} \gamma^t r_t\right] \tag{11}$$

where

γ=discount factor

r_t=reward obtained at time step *t*

The agents update their policy through the Q-learning framework, which estimates the optimal state-action value function

$$Q(s,a) = E\left[r_t + \gamma \max_{a'} Q(s',a')\right] \tag{12}$$

The iterative update rule is given as

$$Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha\left[r_t + \gamma \max_{a'} Q_t(s',a')\right] \tag{13}$$

where α =learning rate.

In the multi-agent scenario, each agent maintains its own Q-value function $Q_i(s,a_i)$ while interacting with other agents that share the environment.

To maintain coordination among agents, the framework introduces a joint policy formulation

$$\Pi(a \mid s) = \prod_{i=1}^{M} \pi_i(a_i \mid s) \tag{14}$$

where *M* is the total number of agents.

This cooperative learning structure allows distributed agents to simultaneously optimize the global resource slicing strategy.

3.4 DYNAMIC RESOURCE ALLOCATION OPTIMIZATION

Once the policy learning stage identifies optimal decision strategies, the system performs dynamic resource allocation across network slices. The allocation objective function maximizes network performance metrics:

$$\max_A F = \omega_1 T + \omega_2 U - \omega_3 L \tag{15}$$

where

T=network throughput

U=resource utilization

L=network latency

$\omega_1,\omega_2,\omega_3$=weighting parameters.

Network throughput is estimated as

$$T = \sum_{k=1}^{K} \min(b_k, B_k) \tag{16}$$

where B_k is allocated bandwidth.

Resource utilization is expressed as

$$U = \frac{\sum_{k=1}^{K}(b_k + c_k + s_k)}{B_{total} + C_{total} + S_{total}} \tag{17}$$

Latency cost is computed through queueing delay models:

$$L = \sum_{k=1}^{K} \frac{\lambda_k}{\mu_k - \lambda_k} \tag{18}$$

The optimization process is allowing that

$$\sum_{k=1}^{K} b_k \leq B_{total} \tag{19}$$

$$\sum_{k=1}^{K} c_k \leq C_{total} \tag{20}$$

$$\sum_{k=1}^{K} s_k \leq S_{total} \tag{21}$$

The resource slicing policy obtained from reinforcement learning determines the action vector $a_t = (a_1, a_2, ..., a_K)$, which specifies the amount of resources allocated to each slice. This mechanism allows the system to adapt resource distribution dynamically based on real-time traffic demand and network conditions.

3.5 COOPERATIVE REWARD EVALUATION AND POLICY UPDATE

The reinforcement learning agents evaluate the effectiveness of resource allocation decisions using a cooperative reward function. The global reward is formulated as

$$R_t = \beta_1 U_t + \beta_2 T_t - \beta_3 L_t \tag{22}$$

where β_1,β_2,β_3 are reward coefficients.

This reward encourages policies that increase utilization and throughput while minimizing latency. The distributed agents update their policies using gradient-based optimization:

$$\nabla_{\theta_i} J(\theta_i) = E\left[\nabla_{\theta_i} \log \pi_{\theta_i}(a_i \mid s) Q_i(s, a_i)\right] \tag{23}$$

where θ_i denotes the parameters of the policy network for agent i.

Through repeated interactions with the network environment, the agents continuously refine their policies and improve resource allocation strategies.

3.6 CONVERGENCE AND ADAPTIVE RESOURCE SLICING DEPLOYMENT

The final stage evaluates the convergence behavior of the learning system. The training process continues until the policy value stabilizes. Convergence is evaluated through the policy difference metric:

$$\Delta_t = \left\| \pi_{t+1} - \pi_t \right\| \tag{24}$$

When $\Delta_t < \epsilon$, the policy has reached convergence. Once convergence has been achieved, the trained policy is deployed for real-time dynamic resource slicing in the network core. The deployed slicing policy continuously monitors the network state S_t and produces the optimal allocation action $a_t = \pi(S_t)$, which dynamically adjusts slice resources according to service demand.

This adaptive decision mechanism is allowing an efficient utilization of infrastructure resources while maintaining quality-of-service guarantees for heterogeneous applications within the 6G network ecosystem.

4. RESULTS AND DISCUSSION

The experimental evaluation is conducted in a simulated 6G network core environment that replicates the dynamic conditions of heterogeneous service slicing. The implementation utilizes the MATLAB R2023b simulation environment together with the Network Simulator 3 (NS-3) framework. MATLAB supports the reinforcement learning training modules and the optimization routines, while NS-3 simulates the network topology, traffic generation, and slice-level resource behavior.

The simulation environment models a virtualized 6G network core that contains multiple service slices and distributed infrastructure nodes. Each slice generates traffic patterns that represent ultra-reliable low latency communication, enhanced mobile broadband, and massive machine-type communication services. The multi-agent reinforcement learning model interacts with the network environment that provides state observations such as traffic demand, queue delay, bandwidth utilization, and computational capacity. The agents continuously update their allocation policies that maximize the reward function which reflects throughput, latency reduction, and efficient resource usage.

The experiments run on a workstation equipped with an Intel Core i7 processor, 32 GB RAM, and an NVIDIA RTX-3060 GPU. The computing system accelerates the reinforcement learning training stage that requires iterative policy optimization and environment interaction. The simulation executes several training episodes in which the agents explore and exploit resource allocation strategies. The experimental framework records the performance metrics for each training cycle, and the evaluation compares the proposed model with baseline slicing algorithms that appear in the related works section.

4.1 EXPERIMENTAL SETUP AND PARAMETERS

The experimental configuration uses several parameters that control the behavior of the learning model and the network environment. These parameters determine the slice traffic generation, reinforcement learning settings, and resource capacity distribution. The Table.1 is summarizing the experimental parameters used in the simulation environment.

Table.1. Experimental Parameters for Dynamic Resource Slicing Simulation

Parameter	Description	Value
Number of network nodes	Core infrastructure nodes	25
Number of network slices	Logical service slices	6
Maximum bandwidth capacity	Total available bandwidth	10 Gbps
Computing capacity per node	Processing capability	200 GFLOPS
Storage capacity per node	Data storage limit	2 TB
Learning rate	Reinforcement learning parameter	0.001
Discount factor	Future reward weighting	0.95

Training episodes	Reinforcement learning iterations	1000
Simulation duration	Total simulation time	2000 s
Traffic arrival rate	Packet arrival intensity	0.6–0.9

As shown in Table.1, the simulation includes twenty-five network nodes that form the core infrastructure responsible for handling the slice resource allocation. Each node provides communication, computing, and storage resources that the slicing controller dynamically allocates to the service slices. The reinforcement learning parameters such as the learning rate and the discount factor control the policy optimization process. The simulation runs for one thousand training episodes that allow the multi-agent system to explore and refine the optimal slicing strategy.

4.2 PERFORMANCE METRICS

The evaluation of the proposed dynamic slicing framework relies on five key performance metrics that measure the efficiency and adaptability of the resource allocation process.

4.2.1 Resource Utilization:

Resource utilization measures how efficiently the network infrastructure resources are consumed during slice allocation. It evaluates the ratio between allocated resources and total available resources. A high utilization value indicates that the slicing mechanism distributes resources efficiently without excessive under-utilization. The metric considers bandwidth, computing power, and storage capacity that the network nodes provide.

4.2.2 Network Throughput:

Network throughput is the total amount of data successfully transmitted across the network slices during a given time period. The throughput reflects the efficiency of the resource allocation policy that supports the traffic demands of multiple slices simultaneously. An effective slicing strategy increases the throughput because the system allocates sufficient resources to each active slice.

4.2.3 Slice Allocation Accuracy:

Slice allocation accuracy measures the correctness of the resource assignment process relative to the actual service demand. This metric evaluates whether the allocated resources match the requirements of each slice. A higher allocation accuracy indicates that the learning agents correctly interpret the network state and that the policy generates appropriate resource decisions.

4.2.4 Network Latency:

Network latency is the communication delay that occurs during data transmission within the network core. The latency measurement considers the queuing delay, processing delay, and propagation delay. A well-optimized slicing policy reduces the latency by allocating resources dynamically to slices that experience higher traffic demand.

4.2.5 Quality of Service Satisfaction Rate:

The Quality of Service satisfaction rate evaluates how frequently the system meets the service requirements defined by each network slice. These requirements include bandwidth guarantees, latency constraints, and reliability conditions. A high satisfaction rate demonstrates that the slicing mechanism maintains consistent service quality across heterogeneous applications.

4.3 DATASET DESCRIPTION

The experimental evaluation utilizes a publicly available network traffic dataset obtained from the Telecom Italia Big Data Challenge Dataset (https://www.kaggle.com/datasets/ocanaydin/italian-telecom-data-2013-1week). This dataset contains real telecommunication traffic records that represent network activity patterns across multiple geographic regions and time intervals. The dataset supports the simulation of realistic traffic demands that the resource slicing system must manage.

The dataset includes network activity information such as call volumes, internet usage, and SMS traffic across different time periods. These records enable the generation of realistic service demands for network slices within the simulation environment.

Table.2. Dataset Description

Attribute	Description
Dataset source	Telecom Italia Big Data Challenge
Total records	~10 million traffic records
Data type	Network traffic activity
Attributes	Call volume, SMS activity, Internet traffic
Time resolution	10-minute intervals
Geographic coverage	Multiple urban regions
Application usage	Traffic demand modeling

The dataset provides traffic intensity values that the simulation uses to generate slice service requests. Each record is network usage within a specific time interval and location. The proposed reinforcement learning model analyzes these traffic patterns that guide the dynamic allocation of network resources.

Three baseline approaches are selected from the related works section to evaluate the performance of the proposed framework. The reinforcement learning resource allocation model proposed by [9] focuses on centralized Q-learning for network slicing. The deep reinforcement learning slicing approach introduced by [10] applies deep Q-networks for adaptive bandwidth allocation. The multi-agent distributed allocation model proposed by [11] employs cooperative agents that coordinate bandwidth distribution in edge network environments.

4.4 RESOURCE UTILIZATION RESULTS OVER NETWORK LOAD

Table.3. Resource Utilization (%) over Network Load

Network Load (%)	Q-Learning Resource Allocation	Deep Q-Network Slice Allocation	Distributed Multi-Agent Allocation	Proposed MARL-DRS
10	68	71	73	78
15	70	73	75	81
20	72	75	77	84

25	74	77	79	87
30	75	79	81	89
35	77	80	83	91
40	78	82	85	93

The results presented in Table.3 show the resource utilization performance across different network load levels. The proposed MARL-DRS framework demonstrates consistently higher utilization compared with the three existing approaches. When the network load reaches 10%, the proposed model achieves 78% utilization, whereas the Q-Learning Resource Allocation method records 68%, the Deep Q-Network Slice Allocation approach records 71%, and the Distributed Multi-Agent Allocation method reaches 73%. As the traffic load increases, the advantage of the proposed method becomes more evident.

At 25% load, the proposed method achieves 87% utilization, which indicates an improvement of 13% compared with the Q-Learning model and 10% compared with the Deep Q-Network strategy. This improvement occurs because the cooperative reinforcement learning agents continuously observe the system state that includes slice demand and resource availability. The agents adjust the allocation policy dynamically, which allows the framework to distribute infrastructure resources more efficiently.

At the highest evaluated load of 40%, the proposed system achieves 93% utilization, while the Distributed Multi-Agent Allocation baseline reaches 85%. The results indicate that the collaborative learning mechanism improves the coordination among agents, which reduces idle resources and enhances infrastructure efficiency. Overall, the numerical results demonstrate that the proposed method maintains stable resource utilization even when the network demand increases.

4.5 NETWORK THROUGHPUT RESULTS OVER TRAFFIC INTENSITY

Table.4. Network Throughput (Gbps) over Traffic Intensity

Traffic Intensity	Q-Learning Resource Allocation	Deep Q-Network Slice Allocation	Distributed Multi-Agent Allocation	Proposed MARL-DRS
10	4.8	5.2	5.5	6.1
15	5.1	5.6	5.9	6.6
20	5.4	5.9	6.3	7.0
25	5.7	6.2	6.6	7.4
30	6.0	6.5	6.9	7.8
35	6.3	6.8	7.2	8.2
40	6.6	7.1	7.5	8.6

The throughput performance presented in Table.4 evaluates the efficiency of the resource slicing strategies in terms of the data transmission capacity. The proposed MARL-DRS approach consistently produces higher throughput across all traffic intensity levels. At 10% traffic intensity, the proposed model delivers 6.1 Gbps, while the Q-Learning approach provides 4.8 Gbps and the Deep Q-Network model produces 5.2 Gbps.

As the traffic demand increases to 25%, the throughput of the proposed framework increases to 7.4 Gbps, which indicates a significant improvement compared with 5.7 Gbps achieved by the Q-Learning model. The improvement arises because the multi-agent learning system continuously observes the traffic distribution across slices and allocates additional bandwidth to slices that experience high demand.

At the maximum intensity level of 40%, the proposed model achieves 8.6 Gbps throughput, whereas the Distributed Multi-Agent Allocation method reaches 7.5 Gbps. The improvement of 1.1 Gbps demonstrates that the proposed cooperative policy optimization effectively balances the bandwidth distribution among multiple slices. The reinforcement learning mechanism identifies optimal allocation patterns that maximize network capacity utilization. Therefore, the proposed framework supports higher communication efficiency in dynamic network environments.

4.6 SLICE ALLOCATION ACCURACY RESULTS OVER TRAINING EPISODES

Table.5. Slice Allocation Accuracy (%) over Training Episodes

Training Episodes	Q-Learning Resource Allocation	Deep Q-Network Slice Allocation	Distributed Multi-Agent Allocation	Proposed MARL-DRS
5	72	75	77	82
10	74	78	80	85
15	77	80	82	88
20	79	82	84	90
25	81	84	86	92
30	83	86	88	93
35	84	88	89	94

The slice allocation accuracy results presented in Table.5 demonstrate the learning capability of the proposed framework. The accuracy metric evaluates whether the allocated resources match the service demand of each network slice. The results show that the proposed MARL-DRS model learns the optimal slicing policy more effectively than the baseline algorithms.

At 5 training episodes, the proposed framework achieves 82% accuracy, while the Q-Learning method records 72% accuracy. The higher accuracy indicates that the multi-agent policy effectively interprets the network state that includes traffic demand and slice priority. As the training process progresses, the agents refine their allocation strategy through reinforcement feedback.

At 20 episodes, the proposed method reaches 90% accuracy, which exceeds the Deep Q-Network model by 8%. This improvement occurs because the cooperative agents exchange environment information that improves the global decision strategy. By the final evaluation stage at 35 episodes, the proposed model achieves 94% accuracy, whereas the Distributed Multi-Agent Allocation approach records 89%. The results demonstrate that the proposed reinforcement learning structure provides stable and accurate resource slicing decisions in the network core.

4.7 NETWORK LATENCY RESULTS OVER TRAFFIC LOAD

Table.6. Network Latency (ms) over Traffic Load

Traffic Load (%)	Q-Learning Resource Allocation	Deep Q-Network Slice Allocation	Distributed Multi-Agent Allocation	Proposed MARL-DRS
10	32	29	27	24
15	34	31	29	25
20	36	33	31	27
25	38	35	33	29
30	41	37	35	31
35	43	39	37	33
40	46	42	39	35

The latency analysis in Table.6 measures the delay experienced during data transmission within the network core. Lower latency values indicate a more efficient resource slicing mechanism. The results show that the proposed MARL-DRS model consistently maintains lower delay compared with the existing methods.

At 10% traffic load, the proposed framework produces 24 ms latency, while the Q-Learning approach records 32 ms. The reduction of 8 ms demonstrates that the dynamic allocation policy distributes resources more efficiently to slices that experience higher demand. When the load increases to 25%, the latency of the proposed method increases slightly to 29 ms, but it remains lower than the 38 ms latency observed in the Q-Learning strategy.

At the highest traffic level of 40%, the proposed model records 35 ms latency, while the Distributed Multi-Agent Allocation method produces 39 ms. The results indicate that the cooperative reinforcement learning agents quickly adapt to traffic fluctuations. The adaptive policy adjusts bandwidth and computing allocation dynamically, which reduces queue delay and improves service responsiveness across the network slices.

4.8 QOS SATISFACTION RATE RESULTS OVER NETWORK DEMAND

Table.7. QoS Satisfaction Rate (%) over Network Demand

Network Demand (%)	Q-Learning Resource Allocation	Deep Q-Network Slice Allocation	Distributed Multi-Agent Allocation	Proposed MARL-DRS
10	74	78	80	85
15	76	80	83	88
20	79	83	86	90
25	81	85	88	92
30	83	87	90	94
35	85	89	92	95
40	87	91	93	96

The Quality of Service satisfaction rate shown in Table.7 evaluates how effectively the slicing framework meets the service requirements of different network slices. These requirements include latency constraints, bandwidth guarantees, and reliability conditions. The results indicate that the proposed MARL-DRS framework achieves the highest satisfaction rate across all demand levels.

At 10% network demand, the proposed model achieves 85% QoS satisfaction, while the Q-Learning model records 74%. The improvement results from the cooperative resource allocation strategy that adjusts slice resources dynamically according to traffic patterns. As the demand increases to 30%, the satisfaction rate of the proposed model reaches 94%, whereas the Deep Q-Network approach records 87%.

At the maximum demand level of 40%, the proposed method achieves 96% satisfaction, which exceeds the Distributed Multi-Agent Allocation baseline by 3%. The higher satisfaction rate indicates that the system successfully maintains service guarantees across heterogeneous slices. The reinforcement learning agents continuously monitor network performance metrics and adapt their policies to maintain consistent service quality.

5. DISCUSSION OF RESULTS

5.1 RESOURCE UTILIZATION ANALYSIS

The results presented in Table 3 demonstrate the effectiveness of the proposed MARL-DRS framework in improving the resource utilization across the network infrastructure. The analysis shows that the proposed model consistently achieves higher utilization compared with the existing approaches. At a 10% network load, the Q-Learning Resource Allocation method records 68% utilization, while the Deep Q-Network Slice Allocation achieves 71% and the Distributed Multi-Agent Allocation reaches 73%. In contrast, the proposed MARL-DRS framework achieves 78% utilization, which indicates a clear improvement in the efficient use of network resources. As the load increases to 25%, the utilization value of the proposed method increases to 87%, whereas the Q-Learning method reaches 74% and the Deep Q-Network approach records 77%. The improvement of approximately 13% over the Q-Learning method demonstrates that the cooperative reinforcement learning strategy effectively manages the slice resource allocation. The agents observe the network state that includes the traffic demand and infrastructure capacity, which allows the system to dynamically allocate resources. At the highest evaluated load of 40%, the proposed model achieves 93% utilization, while the Distributed Multi-Agent Allocation approach reaches 85%. This difference indicates that the collaborative policy learning mechanism improves the coordination among the agents. The framework distributes bandwidth, computing, and storage resources more effectively across slices. Consequently, the proposed system maintains high resource efficiency even when the network demand increases.

5.2 NETWORK THROUGHPUT ANALYSIS

The throughput results presented in Table 4 evaluate the data transmission performance of the network slicing strategies. The proposed MARL-DRS model achieves higher throughput across all traffic intensity levels. At the 10% traffic intensity level, the

Q-Learning Resource Allocation method achieves 4.8 Gbps, while the Deep Q-Network Slice Allocation method achieves 5.2 Gbps and the Distributed Multi-Agent Allocation approach records 5.5 Gbps. The proposed model achieves 6.1 Gbps throughput, which demonstrates a noticeable improvement in the network capacity utilization. When the traffic intensity increases to 25%, the throughput of the proposed framework increases to 7.4 Gbps, whereas the Q-Learning method reaches 5.7 Gbps and the Deep Q-Network method records 6.2 Gbps. The improvement of approximately 1.7 Gbps over the Q-Learning approach indicates that the reinforcement learning agents effectively adapt the resource allocation policy based on the observed traffic conditions. At the highest traffic intensity of 40%, the proposed framework achieves 8.6 Gbps throughput, while the Distributed Multi-Agent Allocation approach records 7.5 Gbps. The numerical improvement of 1.1 Gbps demonstrates that the proposed cooperative learning mechanism successfully balances the resource distribution among network slices. The dynamic allocation policy is allowing an that slices with higher demand receive sufficient bandwidth, which increases the overall communication efficiency of the network core.

5.3 SLICE ALLOCATION ACCURACY ANALYSIS

The slice allocation accuracy results presented in Table 5 illustrate the learning capability of the proposed reinforcement learning framework. The accuracy metric evaluates whether the allocated resources correctly match the requirements of the network slices. At the initial stage with 5 training episodes, the Q-Learning Resource Allocation method achieves 72% accuracy, while the Deep Q-Network Slice Allocation approach records 75% and the Distributed Multi-Agent Allocation method achieves 77%. The proposed MARL-DRS framework achieves 82% accuracy, which demonstrates that the cooperative agents effectively interpret the network state. As the training process progresses to 20 episodes, the proposed model reaches 90% accuracy, while the Q-Learning method records 79% and the Deep Q-Network approach reaches 82%. The improvement of 11% compared with the Q-Learning method indicates that the multi-agent policy learning mechanism improves the allocation decision process. The agents share environmental observations that help in identifying optimal resource distribution patterns. At 35 training episodes, the proposed system achieves 94% accuracy, while the Distributed Multi-Agent Allocation approach records 89%. This result indicates that the learning agents continuously refine the allocation strategy through reinforcement feedback. The cooperative policy optimization mechanism allows the system to maintain accurate slicing decisions in dynamic network environments.

5.4 NETWORK LATENCY ANALYSIS

The latency results shown in Table 6 evaluate the communication delay within the network core. Lower latency values represent better performance because they indicate faster data transmission. At 10% traffic load, the Q-Learning Resource Allocation method records 32 ms latency, while the Deep Q-Network Slice Allocation approach records 29 ms and the Distributed Multi-Agent Allocation method achieves 27 ms. The proposed MARL-DRS framework records 24 ms latency, which indicates a significant reduction in the network delay. When the traffic load increases to 25%, the latency of the proposed framework increases slightly to 29 ms, whereas the Q-Learning method reaches 38 ms and the Deep Q-Network approach records 35 ms. The improvement of 9 ms compared with the Q-Learning method demonstrates that the dynamic resource slicing policy efficiently distributes bandwidth and computational capacity. At the highest load level of 40%, the proposed framework records 35 ms latency, while the Distributed Multi-Agent Allocation method records 39 ms. The difference of 4 ms indicates that the reinforcement learning agents adapt quickly to the changes in traffic demand. The agents allocate resources to slices that experience congestion, which reduces the queuing delay and improves the response time of the network services.

5.5 QOS SATISFACTION RATE ANALYSIS

The Quality of Service satisfaction results presented in Table 7 evaluate the ability of the network slicing framework to meet service requirements across different slices. At 10% network demand, the Q-Learning Resource Allocation method achieves 74% QoS satisfaction, while the Deep Q-Network Slice Allocation approach achieves 78% and the Distributed Multi-Agent Allocation method reaches 80%. The proposed MARL-DRS framework achieves 85% satisfaction, which indicates that the system effectively maintains service quality. When the network demand increases to 25%, the satisfaction rate of the proposed model increases to 92%, whereas the Q-Learning method records 81% and the Deep Q-Network method records 85%. The improvement of approximately 11% compared with the Q-Learning approach demonstrates that cooperative learning agents dynamically allocate resources to meet the slice service requirements. At the maximum network demand of 40%, the proposed framework achieves 96% QoS satisfaction, while the Distributed Multi-Agent Allocation approach records 93%. The numerical improvement demonstrates that the proposed framework maintains reliable service performance even under heavy traffic conditions. The reinforcement learning mechanism continuously monitors the network state and adjusts the slicing policy to maintain service guarantees.

6. CONCLUSION

This study presents a Multi-Agent Reinforcement Learning based Dynamic Resource Slicing framework for the 6G network core environment. The proposed MARL-DRS model introduces cooperative learning agents that observe the network state and dynamically allocate infrastructure resources across multiple service slices. The reinforcement learning policy optimizes the distribution of bandwidth, computational capacity, and storage resources according to the traffic demand and service priority. The experimental evaluation demonstrates the effectiveness of the proposed framework in improving several network performance metrics. The system achieves 93% resource utilization at a 40% network load, which exceeds the Distributed Multi-Agent Allocation approach that achieves 85% utilization. The throughput performance reaches 8.6 Gbps, while the existing approaches achieve between 6.6 and 7.5 Gbps. The slice allocation accuracy improves to 94% after 35 training episodes, which indicates that the cooperative learning agents correctly interpret the network demand conditions. The latency performance also improves significantly, where the proposed framework maintains 35 ms delay under high traffic load, which remains lower than the delay values of the baseline approaches. In addition, the system achieves a 96% QoS satisfaction rate, which demonstrates its ability to maintain consistent service guarantees for heterogeneous network applications. Overall, the proposed MARL-DRS framework provides an adaptive and efficient solution for dynamic resource slicing in future 6G network infrastructures.

REFERENCES

[1] A. Bouroudi, "Multi-Agent Deep Reinforcement Learning for Resource Allocation and Scheduling in 6G", PhD Dissertation, Department of Mathematics and Telecommunication, University of Rennes, pp. 1-199, 2025.

[2] X. Du, T. Wang, Q. Feng, C. Ye, T. Tao, L. Wang and M. Chen, "Multi-Agent Reinforcement Learning for Dynamic Resource Management in 6G in-X Subnetworks", *IEEE Transactions on Wireless Communications*, Vol. 22, No. 3, pp. 1900-1914, 2022.

[3] K. Praghash, V. Sharma, R.P. Shukla, D. Kumar and M. Manwal, "Fair Resource Allocation in 6G Networks using Reinforcement Learning", *Proceedings of International Conference on Recent Innovation in Smart and Sustainable Technology*, pp. 1-6, 2024.

[4] V. Sharma, R.P. Shukla, D. Kumar and M. Manwal, "Deep Learning-based Resource Allocation Algorithms for 6G Networks", *Proceedings of International Conference on Recent Innovation in Smart and Sustainable Technology*, pp. 1-6, 2024.

[5] K.N.G. Veerappan, "Pilot Vector Perturbation Precoding for Scalable and Efficient Transmission in Ultra-Dense Massive MIMO 6G Networks", *Proceedings of International Conference on India Council Subsections*, pp. 1-6, 2025.

[6] M. Asim Ejaz, G. Wu and T. Iqbal, "Dynamic and Efficient Resource Allocation for 5G End-to-End Network Slicing: A Multi-Agent Deep Reinforcement Learning Approach", *International Journal of Communication Systems*, Vol. 37, No. 17, pp. 1-7, 2024.

[7] O. Alruwaili, M.A. Alrowaily, S.K. Patel and A. Armghan, "Incremental RBF-based Cross-Tier Interference Mitigation for Resource-Constrained Dense IoT Networks in 5G Communication System", *Heliyon*, Vol. 10, No. 12, pp. 1-8, 2024.

[8] R. Krishnamoorthy, C.N. Gireesh Babu, P. Gite, A.O. Khadidos, A.O. Khadidos, O.M. Mirza and S. Selvarajan, "Federated Learning for Dynamic Resource Allocation in 6G Network Slicing", *Wireless Personal Communications*, pp. 1-36, 2026.

[9] K. Mao, "Multi-Agent Reinforcement Learning Driven Resource Game Optimization for Network Slicing in MEC-Enabled HetNets", *Scientific Reports*, Vol. 66, pp. 1-9, 2025.

[10] H. Shen, J. Wang and J. Liu, "Dynamic Resource Reconfiguration for Network Slicing: An Incremental Multi-Agent Reinforcement Learning based Approach", *Proceedings of International Conference on Global Communications*, pp. 3685-3690, 2024.

[11] M. Altıntaş, I. Duru, I. Yazıcı, S.N. Karahan and S. Çimen, "Artificial Intelligence Agents Shaping the Next Generation 6G Network Systems", *IEEE Access*, Vol. 18, pp. 1-8, 2026.

[12] S.S. Mhatre, "AI-Enabled Network Slicing and Resource Management for Open and Programmable Next-Generation (6G) Networks", PhD Dissertation, Department of Signal Theory and Communications, Polytechnic University of Catalonia, pp. 1-149, 2025.

[13] R. Kulkarni, B.S. Pillai and S. Rajendran, "Slice-based 6G Network with Enhanced Manta Ray Deep Reinforcement Learning-Driven Proactive and Robust Resource Management", *Computers, Materials and Continua*, Vol. 84, No. 3, pp. 1-7, 2025.

[14] A.M. Seid, A. Erbad, H.N. Abishu, A. Albaseer, M. Abdallah and M. Guizani, "Blockchain-Empowered Resource Allocation in Multi-UAV-Enabled 5G-RAN: A Multi-Agent Deep Reinforcement Learning Approach", *IEEE Transactions on Cognitive Communications and Networking*, Vol. 9, No. 4, pp. 991-1011, 2023.

[15] F. Rezazadeh, H. Chergui, S. Siddiqui, J. Mangues, H. Song, W. Saad and M. Bennis, "Intelligible Protocol Learning for Resource Allocation in 6G O-RAN Slicing", *IEEE Wireless Communications*, Vol. 31, No. 5, pp. 192-199, 2024.