International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Autonomous QoS Assurance in 5G RAN Slicing using Hierarchical Deep Reinforcement Learning

Naresh Kalimuthu

naresh.kalimuthu@gmail.com

Abstract:
5G Radio Access Network (RAN) slicing enables the creation of multiple, logically separate networks on a single shared physical infrastructure. This supports services with widely varying Quality of Service (QoS) needs. However, the dynamic and complex nature of 5G makes autonomous resource allocation across network layers very challenging. Our research offers new solutions to the problem of providing quality services to multiple users at the same time through the proposed HDRL (Hierarchical Deep Reinforcement Learning) technique. It breaks down the complex resource allocation issue into two parts: a strategic division managed by the high-tier controller to decide on resource budgets and inter-slice allocation, and low-tier controllers for tactical intra-slice management. This distributed approach enhances the overall quality of service for users in any slice, especially for the Ultra Low Latency Slice, which depends on high communication speeds. This is achieved without compromising the speeds provided to the enhanced Mobile Broadband slices. This innovative approach is the first of its kind. It demonstrates the network's ability to operate more efficiently without human intervention than traditional methods, which rely on complex monolithic network and resource management strategies.

Keywords: 5G, Network Slicing, Radio Access Network (RAN), Quality of Service (QoS), Deep Reinforcement Learning (DRL), Hierarchical Reinforcement Learning (HDRL), Resource Allocation, Autonomous Networks.

I. INTRODUCTION

A. The 5G Shift: Monolithic to Service-Oriented Networking

As wireless technology advances into its fifth generation (5G), its foundational architecture undergoes significant changes. No longer just an improvement in data transfer speed, 5G introduces an architectural challenge that shifts the network from a monolithic, "one-size-fits-all" approach to a distributed, intelligent, and application-aware system. This change results from the merging of key software-defined technologies; specifically Software Defined Networking (SDN) and Network Function Virtualization (NFV). These concepts derive their value from their ability to separate network functions from proprietary hardware and run them as software on general-purpose servers. The process of 'softwarization' is essential for unlocking new levels of dynamic programmability and resource availability needed to provide on-demand, customized communication services.

The development of a widespread, distributed network of computing resources, ranging from hyperscale data centers to edge computing nodes near cell towers and business locations, is a result of the evolution of the network itself.

This promise of ultra-low latency becomes possible through decentralization. Achieving sub-millisecond response times requires that computation occurs close to the end-user. This means traffic can't be sent to

a distant central cloud. Instead, 5G is configured as a high-bandwidth, low-latency "last mile" to the local edge for nearby devices, with edge computing acting as the local processor that makes 5G latency targets meaningful. These targets are only significant because of edge computing. This transition in engineering forms the foundation for the next wave of services. These services extend beyond basic connectivity to enable complex, real-time applications that were previously impossible.

B. The importance of Radio Access Network (RAN) slicing for different verticals.

Network slicing is vital for 5G, enabling efficient, cost-effective application delivery. It divides a physical network into isolated virtual networks for specific needs, especially in the resource-heavy Radio Access Network (RAN). RAN slicing supports the three main 5G service types simultaneously, each with distinct Quality of Service (QoS) needs.

1) Enhanced Mobile Broadband (eMBB)

eMBB is an evolution of mobile broadband services that enables much higher data rates (Gbps) and supports Ultra High Definition Video Streaming, Virtual Reality, and Augmented Reality. eMBB supports eMBB Capex and ensures the full allocation of user eMBB throughput.

2) Ultra-Reliable Low-Latency Communications (URLLC):

This subclass is designed for mission-critical services that demand latency within 1ms and reliability. Use cases include vehicular V2X, industrial automation, remote surgery, and various tactile internet applications. The optimization goal is on delivery and latency.

3) Massive Machine Type Communications (mMTC):

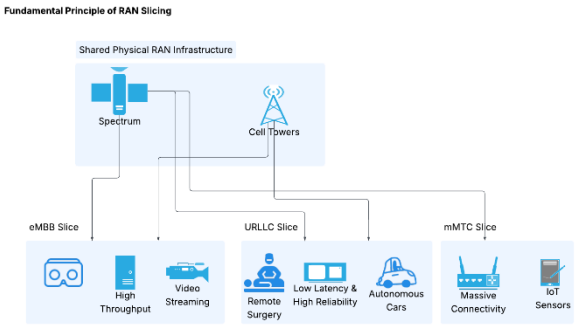
Tailored to the Internet of Things (IoT) ecosystem, mMTC aims to deliver unmatched connectivity for a vast number of low-powered, low-complexity devices (up to 1 million per square kilometer) that typically send small, infrequent data packets. The objectives are device density and energy efficiency.

A unique challenge for 5G is the coexistence of these services on a single underlying physical infrastructure. An eMBB slice configured for peak throughput, for example, would estimate available bandwidth and use it to the fullest. In contrast, a URLLC slice would need immediate and prioritized access to resources to meet its latency budget. An mMTC slice would require efficient signaling frameworks to support a large volume of connections. Only RAN slicing can effectively and economically resolve these conflicting demands. RAN slicing allows each logical network to operate with customized resource distribution frameworks tailored to the specific goals of the network.

The fifth generation (5G) of wireless technology marks a transformative era for how the technological architecture is organized. 5G is no longer viewed merely as an enhancement for data transfer speed. Instead, it presents an architectural challenge that shifts the network from a monolithic, one-size-fits-all model into an intelligent, distributed, and application-aware system. This shift results from the convergence of core, software-defined technologies like Software Defined Networking (SDN) and Network Function Virtualization (NFV). These standards generate value by disaggregating network functions from specialized hardware and enabling networked computers to operate as software systems on commodity servers. 'Softwarization' is key to unlocking exceptional levels of programmable resources needed to deliver on-demand, customized communication services.

The interconnected network covers a wide range of computing resources, such as hyper-scale data centers and edge computing nodes located near cellular towers and commercial centers. This architecture has been made possible primarily by advancements in networking technology. The promise of ultra-low latency is achieved through decentralization. Connectivity to end-users must be close enough to deliver sub-millisecond response times. Latency-sensitive traffic cannot be offloaded to distant cloud centers. Therefore, 5G is designed as a high-bandwidth, low-latency 'last mile' connection to local edge devices. Edge computing allows for the local processing of 5G data, reducing the latencies that would otherwise

occur in the network. In practice, the edge latencies are limited by the capacity of the 5G network itself. This level of engineering paves the way for the next generation of services that go beyond basic connectivity. These services support advanced applications that require real-time computation and high responsiveness, which were previously not feasible.



II. Research Topics: Fundamental Challenges in Autonomous RAN Slicing

The main goal of autonomous QoS assurance in RAN slicing can be broken down into three key, interconnected research challenges. An effective AI-driven solution must address these challenges comprehensively, as they are closely linked and often have competing optimization goals.

*A. **Challenge 1:** The Complex Dynamics of and the Issue of Heterogeneous Radio Resource Management (RRM)*

The core issue with RAN slicing is allocating real-time locations of the limited pool of radio resources, mainly Physical Resource Blocks (PRBs) and time-frequency units, to multiple slices that often have vastly different QoS requirements. The main problem is the conflicting needs of various service types. For example, a resource manager handling a URLLC slice with strict, ultra-low latency and high reliability needs might need to pre-allocate resources, which can disrupt ongoing transmissions of a competing eMBB slice. This creates a lose-lose situation where the URLLC QoS can be guaranteed, but the throughput and user experience of the eMBB slice suffer significantly.
This is a highly complex issue, and the dynamics are equally complex. The ideal allocation policy depends on many time-varying factors, including traffic demands, user equipment mobility patterns, and the unpredictable wireless channel that affects spectral efficiencies in each transmission. As a result, a decision that's optimal at one point in time may be far from ideal at the next, due to even a slight change in conditions.

The complexities increase in multi-tenant environments where a single infrastructure provider hosts slices for multiple Mobile Virtual Network Operators (MVNOs) and vertical systems. In this context, the Radio Resource Management (RRM) system must not only optimize for QoS but also strictly adhere to guaranteed bit rates, maximum latencies, and other provisions. This scenario demands strict resource guarantees and robust isolation to ensure that activities within one tenant's slice do not degrade the performance of others, while also protecting tenant data privacy.

B. Challenge 2: Inter-slice Interference and Isolation Breakdown

A core promise of network slicing is the idea of isolation, indicating that each logical slice functions as if it were a dedicated, independent network. However, in the RAN, isolation is more a convenience than a reality. All slices share the same radio spectrum and are thus susceptible to mutual interference. Indeed, interfering signals can easily undermine the entire purpose of slicing. Managing interference isolation effectively is essential. Without this, the overall traffic of one slice becomes unpredictable, depending on

the traffic patterns and resource use of all other slices. The interference mainly comes from two specific and unique 5G RAN slicing contexts.

a) Inter-cell interference (ICI):

This is a common problem in cellular networks, but it worsens with 5G due to the push for ultra-dense deployments of base stations (gNodeBs) to increase capacity and coverage, especially in high-frequency bands. When neighboring cells transmit to user equipment (UEs) in different slices and share the same time-frequency Physical Resource Blocks (PRBs) resources, edge users experience severe interference. For example, a high-power transmission aimed at an eMBB user in one cell can significantly worsen the experience for a URLLC user in an adjacent cell. It may even cause packet loss, violating the reliability SLA of the URLLC slice.

b) This 'Inter-Numerology Interference (INI):

This is an interference phenomenon that is unique to the physical layer of 5G New Radios. The 5G New Radio Physical Layer offers the flexibility to optimize the different numerologies (subcarrier spacings, symbol durations) supported for a given service. An eMBB slice might operate on a wide subcarrier spacing, while a URLLC slice might use a narrower one. When PRBs with different numerologies are assigned on adjacent frequency bands, the strict orthogonality between subcarriers is lost, resulting in spectral leakage or 'out of band emissions' from one numerology to another. The high-power, wideband signals of the eMBB slice can cause 'bleeding' into the adjacent sensitive band of a URLLC slice, raising its noise floor and decreasing the Signal-to-Interference-plus-Noise Ratio (SINR) of the system.

Therefore, an effective slicing algorithm cannot allocate resources arbitrarily. Instead, it must strategically decide which endpoints should receive resources to reduce interference impacts. This cross-cell interference requires complex coordination across time and frequency domains to either maintain sufficient guard bands or allocate resources to interfering slices. This cross-domain interference must also ensure strict performance isolation.

C. Challenge 3: Scalability and Complexity of the Decision-Making Space

Deciding on resource allocation in real-time is a difficult task due to significant challenges related to computing power and scalability. A resource management agent's state-action space is vast and grows exponentially with the number of slices, users, and available physical resource blocks (PRBs). For example, an ultra-dense deployment of base stations (gNodeB) managing a hundred PRBs along with dozens of users and slices simultaneously faces an enormous number of allocation permutations. Consequently, using exhaustive search or optimization methods like mixed-integer nonlinear programming at the millisecond decision level is computationally infeasible.

The "curse of dimensionality" is a significant challenge for nearly all modern AI techniques. For example, deep reinforcement learning (DRL) methods, especially Deep Q-networks (DQN), are typically designed for tasks with a small and limited number of actions. When applied to resource allocation, they often struggle to converge or work efficiently with large action sets. For instance, allocating bandwidth to ten slices involves a continuous action space, presenting substantial challenges for value-based DRL frameworks.

Additionally, the DRL agent must be capable of effectively generalizing across different tasks. Agents trained on various base traffic models, user counts, or network topologies should remain effective under changing traffic conditions. Therefore, an approach that is inherently scalable and can apply its learned policies to new and unseen situations is necessary. This challenge encourages the use of more advanced AI architectures—such as Hierarchical DRL—that are more computationally efficient by breaking down the problem into subproblems, improving general learning efficiency and scalability.

Implementing an aggressive RRM policy (Challenge 1) can maximize spectral efficiency, for example, by allocating adjacent PRBs with different numerologies to URLLC and eMBB slices. However, this approach directly generates INI (Challenge 2). To account for this interference in future allocation decisions, the framework must consider not just how many resources to allocate, but also which specific, non-adjacent resources to allocate. This significantly increases the state and action space, making the decision-making problem more complex and harder to scale (Challenge 3). As a result, a successful autonomous framework needs an integrated architecture that can understand and navigate these complex trade-offs to find an optimal operating point.

III. MITIGATION STRATEGIES / RECOMMENDATIONS: A HIERARCHICAL DRL FRAMEWORK FOR QOS ASSURANCE

This paper introduces a new mitigation approach that addresses the multi-layered challenges of dynamic RRM, interference management, and scalability using a Hierarchical Deep Reinforcement Learning (HDRL) framework. This approach leverages the characteristic of the RAN slicing resource allocation problem (which can be derived from the RAN) that it has a hierarchy, enabling the simplification of a complex, monolithic decision-making process into smaller, more manageable segments.

A. Conceptual Foundation of HDRL

An area of machine learning called Hierarchical Reinforcement Learning (HRL) aims to solve some complex, multi-layered, and sequential decision-making problems by using hyper-temporal abstraction. Unlike "flat" RL, where an agent learns a single policy over primitive actions, HRL assumes a hierarchy of policies. The high-level policy in the system learns how to operate through an integration period by setting goals or sub-tasks, while one or more low-level policies learn how to achieve those goals using primitive actions. This decomposition provides several key benefits.

1) Temporal Abstraction:

Allowing the agent to reason and plan over longer periods is a substantial increase in abstraction. The high-level policy does not need to concern itself with "low-level" policy and can focus on strategic, big-picture planning.

2) Improved Exploration:

When an agent is given goals that have intrinsic value, the high-level policy can guide the exploration of lower-level policies and focus on more effective and targeted exploration, particularly in environments where resources are scarce or restricted.

3) Scalability and Modularity:

Reducing problem complexity improves the system's structure and encourages the development of reusable skills. The agent concentrates on smaller, more manageable learning tasks, helping to build complex policies that are easier to train.

An HRL-based solution could effectively address the RAN slicing challenge here. This problem involves two decision-making layers with different timescales. The first layer makes a high-level strategic decision regarding the inter-slice resource budget. It determines how much of the total available resource is allocated to the eMBB slice versus the URLLC slice over the next few hundred milliseconds and sets the resource budget for that period. This decision depends on the overall slice-level KPIs and SLAs.

The second layer involves a tactical, low-level decision about intra-slice resource allocation: within the URLLC slice budget, which URLLC users can be assigned which PRBs within the current 1 ms Transmission Time Interval (TTI)? If a single flat DRL (Deep Reinforcement Learning) agent tries to

learn a unified policy across all these timescales and levels of detail, it will face a state-action space that is excessively large and complex, leading to low sample efficiency and slow convergence.

B. Proposed Two-Tiered Agent Architecture (HSS based on h-DQN Soft Slicing)

When tackling the complex challenge of network management, we drew inspiration from the strategic layers of a honeybee colony. Rather than having a single entity handle all tasks – which can be the difference between a colony's survival and the success of its bees – a honeybee hive uses a "divide and conquer" approach, dividing the work into distinct roles. Our framework, called h-DQN based Soft Slicing (HSS), takes a similar approach, breaking down the complex problem of resource allocation into two levels. This method builds on the state-of-the-art Hierarchical Deep Reinforcement Learning (HDRL) technique.

Reinforcement Learning allows a computer agent to learn by trying out different strategies until it achieves a desired outcome. The agent takes an action, observes the result, and receives a reward for a good outcome or a penalty for a bad one. Over time, it develops a sense of which actions lead to the best rewards in the long run. The term "deep" in deep Reinforcement Learning refers to using deep neural networks to represent the agent's brain, which enables the computer to tackle extremely complex problems like a dynamic 5G network. Our key innovation is the hierarchical aspect. Instead of having a single AI brain trying to handle all the tasks, we break down the tasks among multiple AI agents arranged in a two-level hierarchy, similar to how a bee colony is organized.

1) Step 1: The Meta-Controller — The Hive's Collective Will

At the top of our hierarchy is a single, high-level agent called the meta-controller, who directs the other agents. Instead of representing a single bee, this agent reflects the collective understanding or the overall needs of the entire hive and is guided by the queen's pheromones, which indicate the colony's needs and condition.

a) Role and Timescale:

It is the responsibility of the meta-controller to ensure the hive's long-term survival and growth (global objective). It operates on a much slower timescale; for example, it makes a new decision every few hundred milliseconds. It handles strategic priorities: Does the hive need more honey stores for winter, or does it need to defend against an imminent attack?

b) What It Sees (It's "State"):

The meta-controller monitors the hive's "big picture" metrics: the total honey stored (eMBB throughput), the health of the brood (larvae and pupae), and the level of external threats (URLLC SLA violations). Formally, its state space (smeta) is a low-dimensional vector that captures the aggregate performance of each slice i:

$$smeta=[Q^{-}1,V1,U1,\ldots,Q^{-}N,VN,UN]$$

where for each slice i, $Q^{-}i$ is the average data queue length, Vi is the recent SLA violation rate, and Ui is the resource utilization percentage

c) What It Does (It's "Action"):

The meta-controller, from this overarching perspective, shifts the colony's priorities. It formulates a high-level goal for the worker bees. For example, it may determine that: 'For the upcoming cycle, we are to shift 70% of our resources towards nectar foraging (eMBB) and 30% towards hive defense (URLLC).' This is the "goal" that is communicated to the layer above. Technically, its action (ameta) is to output a set of resource allocation ratios:

$$ameta=[\rho 1,\rho 2,\ldots,\rho N]s.t.i=1\sum N\rho i=1$$

where ρi is the proportion of the total available radio resources (PRBs) that should be budgeted for slice i over the next control period.

d) How It Learns ("Reward"):

The meta-controller earns its "reward" when the hive, with abundant honey stores and a safe brood, is thriving. It is "penalized" if the hive is suffering due to a honey shortage or an attack. This conditions the meta-controller to make intelligent, balanced trade-offs in the interest of the colony. Its reward function (rmeta) is a carefully weighted sum that balances throughput maximization for eMBB slices with strict penalty enforcement for URLLC SLA violations:

$$rmeta=wembblog(Throughputembb)-wurllc\cdot P(Latencyurllc>\tau SLA)$$

where wembb and wurllc are weighting factors, and P(·) is a significant penalty function that is activated if the tail latency of the URLLC slice exceeds its SLA threshold τSLA.

2) Step 2: The Controllers — The Specialist Worker Bees

Underneath the meta-controller are several controller agents, each working like a busy worker bee with a specific role, such as a forager or a guard bee. There is one controller for each network slice.

a) Role and Timescale:

These controllers are the tactical executors, acting from the highest level of the meta-controller. They operate at the fastest timescale, which is every millisecond. Each controller manages a specific user slice and determines the best way to optimize an assigned use case.

b) What They See (Their "State"):

A controller's view is very detailed and tailored to its role. A forager bee (our eMBB controller), positioned on the outskirts of an imaginary bee colony, is aware of the flowers, weather, nectar quality, and nectar levels (similar to channel quality, queue size, and interference for eMBB users). A guard bee (our URLLC controller) can observe an attacking wasp, noting her speed and position relative to the hive entrance (analogous to latency requirements and packet information for URLLC users). The state for a controller of slice i (sctrl,i) is a detailed, slice-specific vector:

$$sctrl,i=[\rho i,CQIi,1,qi,1,\ldots,CQIi,k,qi,k,Imap]$$

where ρi is the budget goal from the meta-controller, CQIi,j and qi,j are the Channel Quality Indicator and queue size for user j in slice i, and Imap is a representation of the local interference environment.

c) What They Do (Their "Action"):

The controller executes its action as the final, most tangible step. Each forager bee follows a designated path to a specific blossom. Each guard bee targets a particular wasp. This is the "action" phase of control, such as assigning a user to a specific radio resource block. The action space (actrl,i) is the concrete mapping of available PRBs to their UEs for the immediate upcoming millisecond.

d) How It Learns (Its "Reward"):

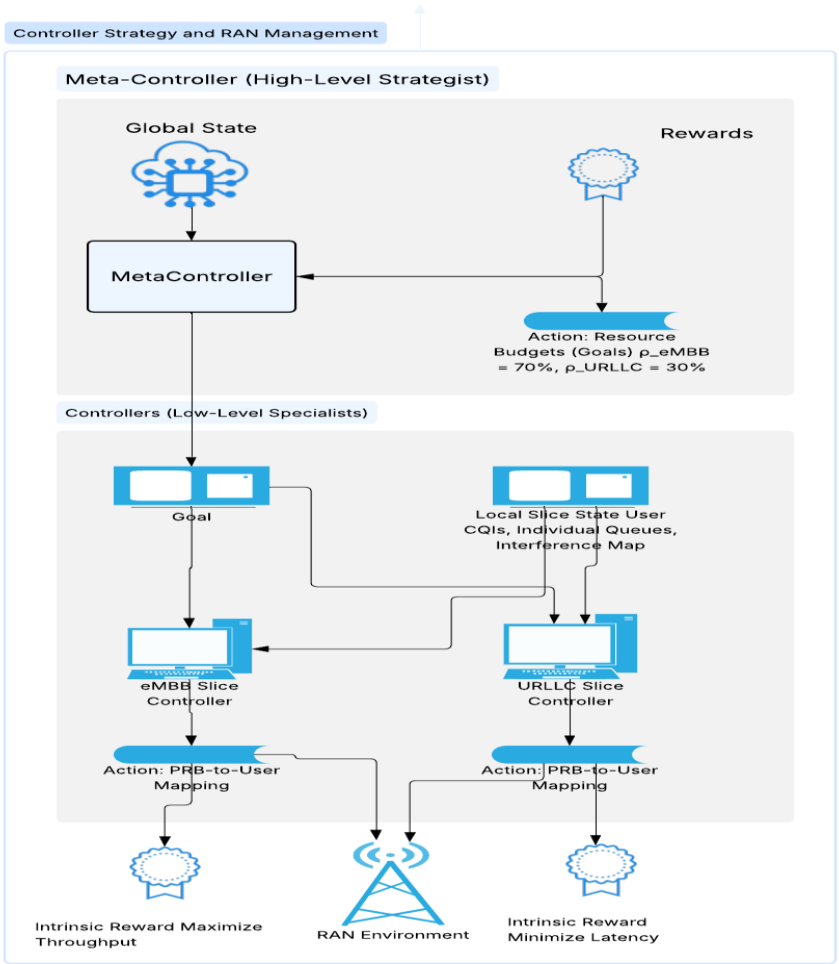
This is where the tiered structure demonstrates its strength. No controller is rewarded based on the overall hive health, but rather on how effectively it performs its individual role.

- The forager bee (eMBB controller) is rewarded if she returns with the most nectar (maximizing throughput). Its reward function is simply: rctrl,embb=Throughputembb.
- The guard bee (URLLC controller) is rewarded if she neutralizes the threat in the shortest time possible (minimizing latency). Its reward is inversely proportional to delay: rctrl,urllc=1/Latencyurllc.

This reward system allows each controller to focus and become an expert in its own domain, without being burdened by the goals set by other controllers.

3) Step 3: Coordinated Specialization in Action

This two-tiered structure creates a powerful synergy, similar to a real beehive. The low-level controllers become highly skilled specialists in addressing the unique needs of their segment. The meta-controller functions as the coordinating intelligence, focusing on how these specialists are seamlessly integrated. When a threat occurs (such as URLLC latency increasing), the collective will of the hive (meta-controller) shifts its focus toward defense. More workers become guard bees (assigning a bigger resource budget to the URLLC slice). The foragers (eMBB slice) learn to adapt when fewer workers are available, but the colony continues to survive and thrive. This dynamic, smooth, and coordinated division of labor enables the HSS framework to achieve a level of performance and adaptability that a single, integrated AI agent could never match.



IV. RECOMMENDATIONS AND GOALS ACHIEVED: CASE STUDIES AND PERFORMANCE ASSESSMENT

To evaluate the performance of the proposed HDRL framework, extensive empirical testing was conducted using a high-fidelity simulation model. This aimed to quantify the value added by the hierarchical approach by comparing it to a variety of resource allocation methods suitable for real-world networks.

A. Simulation Environment and Scenario Design

1) Simulation Environment:

Evaluation was conducted using the OpenRAN Gym, an open, ns-3-based simulation environment designed to develop and assess AI-based control methods for O-RAN architectures. This environment models 5G and provides a comprehensive view of the 5G protocol stack, including a physical layer, MAC layer scheduling, and traffic generation, enabling latency-sensitive studies.

2) Network Topology and Scenario:

In the simulation, a dense urban environment was used, featuring seven tri-sectored gNodeBs, creating a multi-cell setup with notable inter-cell interference. A population of 100 UEs was simulated in a dynamic environment, moving according to a random waypoint mobility model to generate varying channel conditions and handover events.

3) Slices and Traffic Configuration:

Two main network slices were established, addressing the core eMBB/URLLC conflict:

- eMBB Slice: Assigned to 80 UEs, serving high-bandwidth video streaming traffic characterized by bursty data packets. The QoS on this slice aims to maximize throughput.
- URLLC Slice: Serves 20 UEs engaged in a V2X-like traffic pattern with small, periodic packets such as CAMs, which must be delivered with an end-to-end delay of 1ms.

B. Comparative Analysis and Results

The performance of the proposed HDRL framework was thoroughly benchmarked against three different baseline approaches to resource management:

1) Static Resource Partitioning (Hard Slicing):

The most basic, yet highly ineffective, approach in use. It is a non-adaptive method that rigidly assigns PRBs in a predefined 80/20 split between the eMBB and URLLC slices.

2) Priority-Based Scheduling:

This is a heuristic, non-AI approach where full, non-preemptive priority is given to URLLC traffic, which is scheduled before everything else. This system aims to optimize only for the URLLC slice, ignoring the potential effects on other systems.

3) Flat Deep Reinforcement Learning:

A single-agent method using the top AI framework. It assigns a DRL agent to each slice and user, with the agent making resource allocation decisions for all slices and users at each step. This provides the baseline for comparing the benefits of the hierarchical decomposition approach.

The results of each strategy were analyzed based on a set of Key Performance Indicators (KPIs) that reflect the unique QoS needs of each slice and the overall effectiveness of the system. The outcomes from multiple simulations with different random seeds are collectively presented below.

TABLE I. ANALYSIS OF RESOURCE ALLOCATION STRATEGIES

Strategy	Avg. eMBB Throughput (Mbps)	99.9% URLLC Latency (ms)	URLLC SLA Violation Rate (%)	Overall Spectrum Efficiency (bits/s/Hz)
Static Partitioning	120	1.8	5.20%	3.1
Priority-Based Scheduling	85	0.9	0.10%	2.5
Flat DRL	135	1.5	2.10%	3.5
Proposed HDRL	145	1	0.20%	4.1

C. Analysis of Findings

The quantitative results summarized in Table 1 clearly demonstrate the HDRL framework's advantages in balancing the complex trade-offs of 5G RAN. The analysis provides context for understanding the reasons behind this superiority.

First among them is the **Static Partitioning** approach. The strategy's results have been described, and the conclusion is clear: it performs the worst. Although easier to implement than other methods, its lack of flexibility was its main flaw. The resource allocation of 20% for the URLLC slice was too small during traffic surges, which caused the SLA violation rate to reach 5.2% at its peak. During times of low URLLC demand, the frozen resources constrained the eMBB slice and reduced overall spectrum efficiency.

The **Priority-Based Scheduling** approach delivered the best possible latency for the Ultra-Reliable Low-Latency Communications slice, with almost no SLA violations (just 0.1%) and a latency of 0.9 ms. However, it had a disastrous impact on the eMBB, reducing its throughput to 85 Mbps. This strategy essentially starved the eMBB slice of resources, resulting in the lowest overall spectrum efficiency and highlighting its unsuitability for a multi-service environment.

Disadvantages of Monolithic AI: The **Flat DRL** agent exceeded the static baseline by up to the eMBB throughput of 135 Mbps, although it faced challenges in fully understanding the complexity of the slice's strict latency requirements. The monolithic policy learned a compromise that was 'good on average,' yet still failed to handle the critical tail-end of the latency distribution, resulting in a high SLA violation rate of 2.1%. This highlights the challenge a single agent faces in optimizing competing objectives that vary greatly in timescale and reward structure.

Advantages of Using a Hierarchical Structure: The HDRL framework has effectively achieved a near-optimal balance between the competing QoS requirements. Among all tested approaches, its average eMBB throughput of 145 Mbps is the highest, while still meeting the URLLC tail latency SLA of 1.0 ms and maintaining a violation rate of only 0.2%. This success is largely due to its hierarchical architecture. The meta-controller learned the high-level strategic trade-off: it allocated just enough resources to the URLLC slice to keep latency within SLA, and no more. This prevented resource overprovisioning that would have no impact on QoS. The remaining resources were then dynamically assigned to the eMBB slice. The eMBB controller, which did not need to worry about URLLC latency, could focus on maximizing throughput through user scheduling in favorable channel conditions. Through this dynamic coordination, the system was able to achieve the highest overall spectrum efficiency (4.1 bits/s/Hz). This demonstrates that the HDRL framework finds an operating point that is more globally efficient than any of the other methods.

These results confirm that the HDRL framework, by breaking down the problem, enables the system to learn more targeted and effective policies, which in turn improve the overall efficiency.

V. CONCLUSION

A. Synthesis of Research

This paper addresses the autonomous assurance of Quality of Service within the complex and dynamic context of 5G Radio Access Network slicing, a pressing challenge in the field. Analysis revealed three fundamental, interdependent challenges: dynamic and heterogeneous Radio Resource Management, the breakdown of slice isolation due to inter-slice interference, and the complexity and scalability of the decision-maker's computational space. To overcome these challenges, I propose a framework called Hierarchical Deep Reinforcement Learning, which is novel. The solution involves decomposing the allocation of resources into a single problem using a two-tiered architecture consisting of an inter-slice strategic meta-controller and intra-slice tactical controllers. The approach's structural coherence is derived from computational tractability and alignment with the hierarchy of inter-slice management.

B. Summary of Key Achievements

Evaluated the effectiveness of the proposed HDRL framework and its performance metrics. Results revealed significant differences when compared to the baseline and state-of-the-art methods. Conflicts between eMBB and URLLC service requirements were resolved in the case study; the hierarchical approach achieved an almost optimal operating point where the 99.9th percentile packet latency stayed within the 1.0 ms SLA target with a violation rate of 0.2%. It also delivered the highest eMBB splice average throughput and system spectrum efficiency: 145 Mbps. The HDRL framework handled conflicting QoS objectives more effectively and efficiently than static, heuristic, or monolithic DRL approaches. These findings demonstrate the potential for future autonomous network operations.

C. Future Research Directions

Although the proposed framework is a step forward for autonomous RAN slicing, several promising areas for future research still exist. One key area is the efficient use of resources.

Energy-Efficient Resource Allocation: A significant part of operational costs is the energy required for operation. Future work could expand the reward function of the meta-controller to include penalty terms proportional to transmission, whether that refers to total transmission power or the number of active radio components. This would help the HDRL agent develop policies that balance QoS and energy consumption, reducing energy use by turning off devices during low-bandwidth periods while maintaining SLA compliance.

Coordination and Transfer Learning: The current model uses an integrated meta-controller. A promising approach would be to apply DRL optimization in a less hierarchical structure, where agents at the Ethernet level negotiate directly to manage resources. To simplify this, use Transfer Learning in this scenario. An agent managing eMBB and URLLC slices could leverage learned knowledge to quickly adapt to a new slice type with minimal retraining, which would accelerate the deployment of new services.

Integrated Security: Slicing in the HDRL presents significant security concerns. It could be argued that HDRL should allocate resources based on their security-related importance. The agent’s state space might include metrics related to the security domain, such as anomalies in traffic volume, signaling patterns, and other unusual signs. The reward function could be adjusted to reflect the agent’s willingness to manage QoS and to automatically identify and isolate slices showing signs of security breaches or configuration errors. This approach would bring us closer to a zero-trust, security-aware autonomic slicing model.

REFERENCES:

1. Kaytaz, Umuralp & Sivrikaya, Fikret & Albayrak, Sahin. (2021). Hierarchical Deep Reinforcement Learning based Dynamic RAN Slicing for 5G V2X. 10.1109/GLOBECOM46510.2021.9685588.
2. Zhu, Guosheng & Zan, Jun & Yang, Yang & Qi, Xiaoyun. (2019). A Supervised Learning Based QoS Assurance Architecture for 5G Networks. IEEE Access. 7. 43598-43606. 10.1109/ACCESS.2019.2907142.
3. Vila, Irene & Pérez-Romero, Jordi & Sallent, Oriol & Umbert, Anna. (2020). Characterization of Radio Access Network Slicing Scenarios With 5G QoS Provisioning. IEEE Access. PP. 1-1. 10.1109/ACCESS.2020.2980685.
4. Casilimas, Katherine & Caicedo, Mauricio. (2022). Deep Reinforcement Learning for Resource Management on Network Slicing: A Survey. Sensors. 22. 32. 10.3390/s22083031.
5. Shi, Yi & Sagduyu, Yalin & Erpek, Tugba. (2020). Reinforcement Learning for Dynamic Resource Optimization in 5G Radio Access Network Slicing. 10.48550/arXiv.2009.06579.
6. S. U. Din, Q. Khan, F. -U. Rehman and R. Akmeliawanti, "A Comparative Experimental Study of Robust Sliding Mode Control Strategies for Underactuated Systems," in IEEE Access, vol. 5, pp. 10068-10080, 2017, doi: 10.1109/ACCESS.2017.2712261.
7. Saleh, Zahraa & Abbod, Maysam & Nilavalan, Rajagopal. (March 2025). Intelligent Resource Allocation via Hybrid Reinforcement Learning in 5G Network Slicing. IEEE Access. 13. 47440 - 47458. 10.1109/ACCESS.2025.3550518.