






Please cite the Published Version

Shah, Syed Danial Ali , Bashir, Ali Kashif , Al-Otaibi, Yasser D , Dabel, Maryam M. Al  and Ali, Farman  (2025) Dynamic AI-Driven Network Slicing with O-RAN for Continuous Connectivity in Connected Vehicles and Onboard Consumer Electronics. IEEE Transactions on Consumer Electronics. ISSN 0098-3063

DOI: <https://doi.org/10.1109/tce.2025.3527857>

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Version: Accepted Version

Downloaded from: <https://e-space.mmu.ac.uk/638047/>

Usage rights:  In Copyright

Additional Information: This is an accepted manuscript of an article which was published in IEEE Transactions on Consumer Electronics

Enquiries:

If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from <https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines>)

Dynamic AI-Driven Network Slicing with O-RAN for Continuous Connectivity in Connected Vehicles and Onboard Consumer Electronics

Syed Danial Ali Shah, Ali Kashif Bashir, *Senior Member, IEEE*, Yasser D. Al-Otaibi, Maryam M. Al Dabel, and Farman Ali

Abstract—The rise of connected and autonomous vehicles signifies an era of intelligent transportation systems, where robust and continued network connectivity is essential for critical applications and enhanced in-vehicle Consumer Electronics (CE) experiences. Slicing at the network's edge offers tailored and dedicated logical networks for diverse and low-latency vehicular demands, including Advanced Driver Assistance Systems (ADAS) and in-car infotainment. However, seamless migration of network slices as vehicles traverse coverage areas of different network operators presents formidable challenges, such as ensuring continuous connectivity and uninterrupted service for both safety-critical systems and consumer-oriented services. In this paper, we introduced dynamic network slicing for continuous connectivity in connected vehicles and onboard CE using the Open Radio Access Network (O-RAN) framework in a highly dynamic and mobile environment. We implemented an xAPP within O-RAN that enables Deep Reinforcement Learning (DRL) agent to learn optimal policies through interaction with the network, guiding intelligent decisions on slice migration, resource allocation, and handover optimization. We conducted simulations and evaluations to demonstrate the effectiveness of the proposed xAPP in maintaining optimal Quality of Service (QoS), ensuring efficient RAN resource utilization, minimizing service interruptions, and prioritizing safety-critical slices, all while supporting seamless operation of CE within vehicles during mobility.

Index Terms—Edge computing, consumer electronics, deep reinforcement learning, O-RAN, xAPP, in-car infotainment, vehicular networks

I. INTRODUCTION

THE automotive industry is transforming towards connected and autonomous vehicles, driven by advances in sensing technologies, artificial intelligence, and next-generation wireless communication networks [1]. These intel-

ligent transportation systems have the potential to revolutionize road safety, traffic management, and the overall driving experience by enabling a wide range of vehicular applications, such as remote driving assistance, real-time traffic monitoring, infotainment services, and enhanced in-vehicle Consumer Electronics (CE) [2].

However, the successful deployment and availability of these vehicular applications hinges on the availability of reliable and low-latency network connectivity. The traditional cellular networks need enhancements to support these emerging vehicular applications due to vehicular environments' highly dynamic and mobile nature. Radio Access Network (RAN) slicing is a key feature of 5G and beyond networks that allows for the creation of multiple virtual and isolated network resources at the edge of the network, e.g., at the Multi-Access Edge Computing (MEC) server connected to the RAN [3]. Network slice can be tailored to meet the specific QoS requirements of different applications or services, such as low latency, high bandwidth, or enhanced reliability. For example, assistance applications require ultra-low latency and high reliability to enable real-time vehicle control and decision-making. Traffic monitoring applications, on the other hand, may prioritize high bandwidth and consistent connectivity to support the transmission of high-resolution video streams, whereas consumer electronics may emphasize bandwidth and throughput to deliver a seamless multimedia experience to passengers.

However, as vehicles move between different coverage areas of RANs and their respective MEC servers, seamlessly migrating the dedicated network slices while maintaining the required QoS levels, efficiently utilizing network resources, and minimizing service interruptions becomes an exacting challenge. Traditional mobility management techniques fall short of addressing the complexities introduced by network slicing at the MEC and the highly dynamic nature of vehicular environments [4]–[6]. These reactive techniques only trigger network slice mobility actions based on channel conditions, e.g., distance to neighbouring RANs. They treat all types of network slices equally irrespective of their QoS requirements and criticality, resulting in inefficient utilization of RAN and MEC resources. Moreover, as vehicles frequently relocate and change their distance from neighbouring RANs, the traditional schemes that rely solely on proximity to RANs result in frequent migration of network slices and handovers [5]–[7]. This can lead to a ping-pong handover scenario resulting in

Syed Danial Ali Shah (Corresponding Author) is with the School of Electronic and Electrical Engineering, University of Leeds, Leeds LS2 9JT, UK. e-mail: S.Shah@leeds.ac.uk

Ali Kashif Bashir is with the Department of Computing and Mathematics, Manchester Metropolitan University, M15 6BH, UK, and Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, 140401, Punjab, India. e-mail: dr.alikashif.b@ieee.org

Yasser D. Al-Otaibi is with the Department of Information Systems, Faculty of Computing and Information Technology in Rabigh, King Abdulaziz University, Jeddah 21589, Saudi Arabia. e-mail: yalotaibi@kau.edu.sa

Maryam M. Al Dabel is with the Department of Computer Science and Engineering, University of Hafr Al Batin, Saudi Arabia e-mail: maldabel@uhb.edu.sa

Farman Ali (Co-Corresponding Author) is with School of Convergence, Sungkyunkwan University, Seoul, South Korea. e-mail: farman0977@g.skku.edu

significant wastage of network resources.

To address these challenges, we proposed a novel approach that leverages Deep Reinforcement Learning (DRL) within the Open Radio Access Network (O-RAN) framework. We formulated the network slice mobility problem as a Markov Decision Process (MDP), where we define a comprehensive state space that captures different patterns of the vehicular users, e.g., signal strength, network conditions, resource availability, and the current state of the network slices. Our proposed approach incorporates a reward function that dynamically adjusts its weights based on the vehicle type, network slice type, and mobility pattern, enabling the DRL agent to adapt to the highly dynamic vehicular environment and prioritize objectives such as QoS, resource efficiency, service continuity, and slice prioritization accordingly.

The DRL agent is implemented as an xAPP in the Near-Real-Time Radio Intelligent Controller (NRT-RIC) within the O-RAN framework, which learns an optimal policy through interactions with the network environment. The learned policy facilitates the NRT-RIC in making intelligent decisions on network slice migration, resource allocation, and handover optimization for vehicular network slices by leveraging predictive analytics and real-time network state information. We conducted extensive simulations and evaluations in a highly dynamic and mobile environment, where we demonstrated the effectiveness of our proposed approach in maintaining QoS for critical vehicular applications, efficient resource utilization, minimizing service interruptions, and prioritizing safety-critical network slices during vehicle mobility. Additionally, our framework supports the seamless operation of in-vehicle consumer electronics by ensuring reliable and low-latency connectivity for infotainment and other multimedia services. The proposed framework paves the way for intelligent and adaptive network slice management in vehicular networks, enabling reliable and low-latency connectivity for connected and autonomous vehicles.

A. Contributions

The main contributions of this research are summarized as follows:

- We introduced dynamic network slicing for the highly dynamic and mobile environment using the O-RAN framework. To the best of the author's knowledge, this is one of the first works exploring the applications of O-RAN in addressing challenges specific to network slice mobility management in highly mobile environments, e.g., vehicular networks, for continuous connectivity of safety-critical and Automotive Consumer Electronics Applications (CEA), e.g., in-vehicle infotainment.
- We formulated network slice migration across the RANs and their respective MEC servers as an MDP, which considers various factors, including mobility patterns, network conditions, resource availability, and slice state, providing a comprehensive framework for optimizing slice migration decisions. We develop a novel approach using DRL to optimize network slice migration, resource allocation, and handover optimization in a highly

dynamic and mobile environment, leveraging real-time network state information as enabled by the proposed NRT-RIC in the O-RAN framework.

- We implemented the proposed solution as a custom-developed xAPP communicating with the centralized NRT-RIC within the O-RAN framework. The implementation facilitates seamless integration with existing RAN components and enables adaptive and intelligent network slice management.
- We conducted extensive simulations and evaluations to demonstrate the efficacy of our proposed approach in maintaining QoS requirements for various vehicular applications and consumer electronics, optimizing RAN resource utilization, minimizing service interruptions, and prioritizing safety-critical network slices during vehicle mobility.

B. Related Works and Research Gaps

The provision of methods and techniques to enable slicing at the RAN and its associated MEC server, i.e., slicing at the network edge, is essential to facilitate novel services such as Ultra-Reliable Low-Latency Communications (URLLC) for vehicular applications, e.g., automotive CE applications such as Advanced Driver Assist Systems (ADAS), and in-vehicle infotainment. However, this integration is limited, and enabling MEC support for network slicing introduces challenges in coordinating various network entities and functions [15], [21]–[23]. While various studies have addressed the integration of network slicing at the network's edge, focusing on addressing issues such as resource management, slice selection, admission control, and network slice mobility, there remains a need for further exploration, particularly in terms of the network slice mobility challenges in highly dynamic and mobile wireless network environments, e.g., vehicular networks. Effective management of network slice mobility is essential to maintain uninterrupted and continuous connectivity for automotive consumer electronics and safety-critical vehicular applications, particularly during the transition of vehicles between coverage areas of different network operators.

In their respective studies, [8], [17], authors proposed Software-Defined Networking (SDN) and Kubernetes-based methodologies for the simultaneous migration of containerized services within a network slice from one edge network to another. However, they did not consider network slice mobility challenges in dynamic environments like vehicular networking. The authors in [14], [15] introduced innovative frameworks emphasizing the necessity of new control functions to advance the current edge networks supporting network slicing. However, these approaches did not consider network slice mobility challenges. In [13], the authors presented a reinforcement learning technique for optimizing slice mobility decisions and managing network slice resources, while [24] proposes a deep reinforcement learning-based solution to address resource allocation issues in MEC-enabled vehicular networks. Similarly, [16] introduces an intelligent network slicing architecture integrating edge computing and employing deep learning for application-specific packet routing towards

TABLE I
PROPOSAL COMPARED TO THE LITERATURE

Reference No.	Slicing at the Network's Edge	Network Slice Mobility	Dynamics of Vehicular Networks	PoC Experiments	O-RAN (xAPP) Implementation	Used Approach
[8]	✓	✓		✓		SDN-based centralized control plane
[9]	✓		✓	✓	✓	Deep reinforcement learning framework
[10]	✓			✓	✓	Network application (xAPP) for network slicing in O-RAN
[11]	✓			✓	✓	Deep reinforcement learning framework
[12]	✓		✓	✓	✓	O-RAN based architectural framework
[13]	✓	✓		✓		Reinforcement learning
[14]	✓					Modules implementing slice control
[15]	✓			✓		Network slicing at the MEC integration framework
[16]	✓			✓		Deep learning
[17]	✓	✓		✓		Velero tool in Kubernetes
[18]	✓			✓		Architectural framework for MEC slicing
[19]	✓		✓	✓		Centralized control plane
[20]	✓					Architectural Framework
Our Proposed	✓	✓	✓	✓	✓	DRL-based xAPP implementation in O-RAN framework

MEC servers. However, their mobility solutions did not consider aspects such as network slice mobility based on available network resources, slice types and their QoS requirements and priority levels.

In [19], the authors proposed a centralized control plane algorithm and virtualized infrastructure to fairly distribute and balance the workload among various network slices across one or multiple edge networks. The authors in [20] introduced a comprehensive end-to-end slicing architecture spanning multiple domains, including RAN, core, and MEC. In [25], an edge computing network is envisioned under network slicing, enabling dynamic allocation of low-latency computational tasks from wireless devices to network slices at MEC servers associated with their respective RANs. The authors in [26] proposed a two-level RAN slicing approach, implementing deep reinforcement learning to optimize RAN's communication and computation resources to meet network applications' stringent latency requirements. However, the above-mentioned research works [19], [20], [25], [26] overlook the challenges posed by network slice mobility arising from enabling slicing at the RAN, i.e., edge of the network. In contrast, our proposed method is the first implementation of network slice mobility management using the O-RAN framework in a highly dynamic and mobile environment such as vehicular networks. Our proposed approach builds on a DRL-based xAPP that seamlessly interacts with the NRT-RIC and effectively manages the RAN control parameters to optimize the network slicing at the RAN and tackle the network slice mobility challenges.

There have been some recent efforts in enabling network slicing in 5G O-RAN [9]–[12]. The authors in [9] proposed a DRL approach that incorporates deep deterministic policy gradient techniques for optimizing resource allocation and inter-slice operations in vehicular networks. In [10], authors proposed a custom network application (xAPP) for network slicing in 5G O-RAN that enables emerging IoT services to co-exist and meet the required service-level agreements. In [11], the authors proposed a DRL-based algorithm to optimize the resource allocation problem for effective resource

management, enabling slicing at the RAN level. The authors in [12] demonstrated the effectiveness of the O-RAN control capabilities in supporting efficient management of the Vehicle-to-Everything (V2X) system. However, none of these works considers the network slice mobility challenges arising from the mobility of the vehicles from the coverage area of one RAN to another. A comparative summary of the most relevant and selected works from the literature is detailed in Table I and compared with our proposal.

II. THE O-RAN ARCHITECTURE

O-RAN is an industry initiative to create open, virtualized, and intelligent Radio Access Networks (RANs) by providing open interfaces and software solutions. O-RAN promote openness and intelligence in next-generation RAN architectures, enabling greater flexibility, innovation, and cost-efficiency for mobile network operators. The following are the key components and interfaces involved in the O-RAN framework.

A. Key Components and Interfaces

The O-RAN architecture consists of several key components and interfaces that coordinate to enable the desired openness and intelligence at the radio level:

1) *O-Cloud*: The O-Cloud is the cloud-native infrastructure hosting the virtualized and containerized network functions, including the O-RAN Software Components, e.g., O-RAN Distributed Units (O-DU), O-RAN Radio Units (O-RU), and O-RAN Central Units (O-CU), as shown in Fig. 1.

2) *O-RAN Software Components*: The O-RAN software components include the O-DU that handles real-time baseband processing and is responsible for functions such as encoding/decoding and MIMO processing; the O-CU that performs non-real-time functions like radio resource control, mobility management, and scheduling; and O-RU that comprises the radio components, including antennas, amplifiers, and digital front-end processing.

3) *Open Interfaces*: O-RAN defines several open interfaces to enable interoperability and multi-vendor ecosystems. These interfaces include open fronthaul, which is the interface between the O-DU and O-RU, enabling the fronthaul network connectivity, and open mid-haul, which is an F1 interface connecting an O-CU to an O-DU. The F1 control plane (F1-C) allows signalling between the O-CU and O-DU, while the F1 user plane (F1-U) is used for transferring the application data. E2 Interface which is an open interface between two endpoints, i.e., the NRT-RIC and the E2 nodes, i.e., DUs and CUs in 5G. Open Southbound Interfaces, which is the interface between the O-RAN Software Components and the RIC.

4) *RAN Intelligent Controllers*: The RICs are a key component of the O-RAN architecture, enabling AI/ML-based intelligent control and optimization of the RAN. There are two types of RICs used in the RAN architecture that include Non-Real-Time RIC (Non-RT-RIC) and Near-Real-Time RIC (NRT-RIC). Non-RT-RIC is used for handling non-real-time operations, e.g., offline model training, policy generation, and enrichment information delivery, whereas NRT-RIC is used for handling near-real-time operations like policy execution, traffic steering, and resource allocation. In this research, we used NRT-RIC for the effective and real-time implementation of RAN control policies.

5) *xApps and rApps*: The RICs host and execute various applications known as "xApps" and "rApps". xApps are AI/ML-based applications that run on the NRT-RIC and provide intelligent functions like prediction, optimization, and policy generation. The xApps work in real-time that can handle those events requiring action from 10 milliseconds (ms) to 1 second. As our proposed scenario, i.e., intelligent network slicing in highly dynamic and mobile network environments, demands a real-time response, we developed our custom xAPP based on DRL that is deployed along with other xApps in the NRT-RIC, as shown in Fig. 1. On the other hand, rApps are applications that run on the Non-RT-RIC and are used for non-real-time network automation.

III. PROPOSED SYSTEM MODEL

Our scenario considers vehicular networks as the environment for our proposed architecture. A vehicular network provides a dynamic and complex environment with strict performance requirements. The scenario consists of sets of O-RAN-compliant O-DUs, O-RUs, and O-CUs, represented as gNodeB $m \in M$, where each gNodeB m is equipped with its MEC server. We considered a centralized intelligent architecture where all the gNodeBs are controlled and managed by a single centralized NRT-RIC and a custom-developed and proposed xAPP, as shown in Fig. 1. MEC servers provide multiple service-specific dedicated slices denoted as $l \in L$. Within this system, U users exist, specifically, vehicles ($u \in U$), each seeking access to a service-specific dedicated slice operating at the nearest RAN and its associated MEC server. Time is segmented into T discrete slots. During each of these time slots ($t \in T$), the mobile user ($u \in U$) requests the service-specific dedicated slice from a RAN ($m \in M$). The high-level architecture for the proposed approach and system model

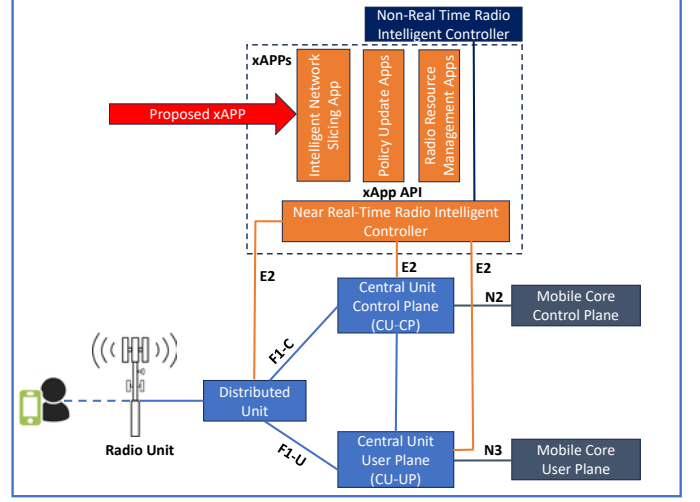


Fig. 1. O-RAN architecture with the proposed xAPP deployment in NRT-RIC

is shown in Fig. 1, where the proposed DRL approach is implemented as an xAPP in the NRT-RIC. Furthermore, the detailed architecture, including all the components involved, is provided in Fig. 2, whereas all the components of the proposed solution are discussed in the following sections.

A. Computation Model

We assume that each mobile user demands access to a dedicated slice from the RAN and its associated MEC server to fulfil its QoS requirements. The computational model could be divided into three major steps. Initially, mobile users (interchangeably referred to as vehicles in this manuscript) initiate a request for the service-specific dedicated slice by offloading data of a specific size to be processed by the MEC server through the RAN [27]. Subsequently, the RAN forwards the data for processing to the corresponding MEC server. Lastly, the processed results from the MEC server are returned to the mobile user. Based on the steps outlined, the computational model can be characterized by the processing delay resulting from all these steps, including MEC communication delay, processing delay or MEC slice computation delay, and the downloading delay of the processed result [4].

1) *MEC Communication Delay*: When a mobile user u requests a dedicated slice l from RAN and its associated MEC server m , the communication delay is contingent upon the attained data rate, wireless channel quality, and the magnitude of the requested service [4]. The channel gain between user u and RAN m at time t is represented by g_{um}^t , and the signal-to-noise ratio can be formulated as:

$$\gamma_{um}^t = \frac{p_m g_{um}^t}{I'_M + \sigma^2}, \quad (1)$$

Where p_m denotes the transmit power of the RAN m , σ^2 signifies the noise power, and I'_M represents the accumulated interference from neighboring RANs. The achieved data rate can be defined as:

$$R_{um}^t = b_m \log_2(1 + \gamma_{um}^t), \quad (2)$$

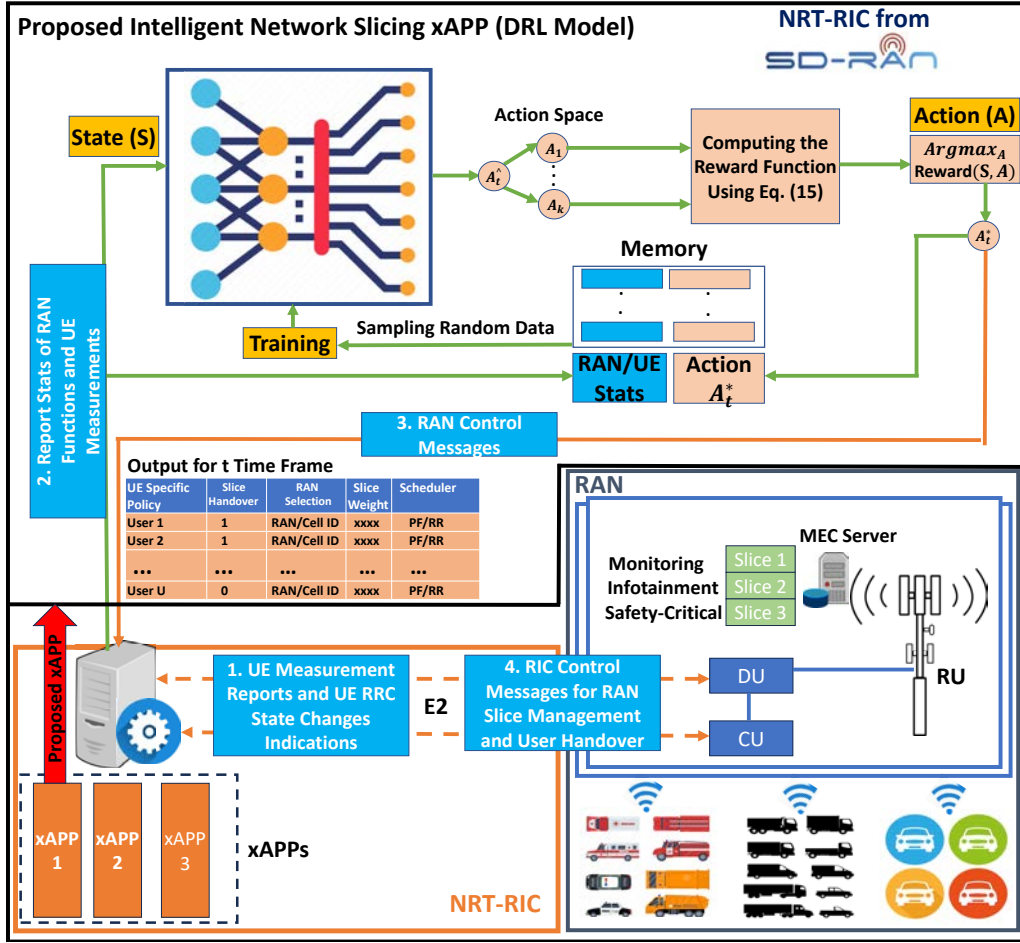


Fig. 2. Proposed O-RAN architecture and its working principles for network slice management in dynamic and mobile network environments

Here, b_m denotes the bandwidth allocated by the RAN m . The delay in communication between the RAN m and the mobile user u , who requests a service-specific dedicated slice at time t , can be characterized as follows.

$$d_{um}^t = \frac{mg_u}{R_{um}^t}, \quad (3)$$

Here, mg_u denotes the magnitude of the service requested by the mobile user u . Additionally, the duration taken to download the processed service requests is mainly dependent on both the size of the processed outcomes and the download data rate of the mobile user u .

2) *Slice Computation Delay*: This delay is dependent on the resources assigned to the dedicated slice tailored for a specific service within a MEC server by the NRT-RIC, along with the computational demands of the mobile user u at time t . The slice computation delay for a particular slice l at RAN m and its associated MEC server at time slot t can be defined as:

$$C_{um}^t = \frac{c_{ul}^t L d_{lm}^t}{RS_{lm}^{max}}, \quad (4)$$

Here, c_{ul}^t denotes the requested computation capacity, measured as RAN time frame rates per second, for MEC slice

l by vehicle u at time t . RS_{lm}^{max} represents the maximum computation capacity allocated to slice l within RAN m , e.g., maximum time frame rates (maximum weight of a network slice in time frame rates). Additionally, $L d_{lm}^t$ represents the current computational load on slice l within RAN m , indicating the number of mobile users utilizing the same service-specific dedicated slice l at time t .

B. Slice Migration Delay (Service Interruption Delay)

The slice migration delay indicates the period of service unavailability when slice mobility is required among different RANs. This delay is defined as the duration required to complete the network slice mobility procedure, which involves initiating a new slice at the target RAN. Here, the target RAN is defined as the potential candidate for network slice migration as the mobile user migrates from the coverage area of one RAN to another. The slice migration delay can be articulated as:

$$M_d = \begin{cases} 0, & \text{if } RAN_{curr} = RAN_{cand}, \\ S_{md}, & \text{if } RAN_{curr} \neq RAN_{cand}. \end{cases} \quad (5)$$

In the case when the mobile user is not handed over towards the candidate RAN, the slice migration delay is returned as 0. There could be several reasons for this scenario to occur, such

as the dedicated slice required by the mobile user does not have any strict latency or service continuity requirements, or the candidate RAN does not have ample resources available to complete the network slice mobility as it needs to prioritize the safety-critical slices and emergency vehicle types, as defined in Table II. Here, s_{md} represents the slice migration delay when the dedicated slice is migrated from the current RAN towards the target RAN, e.g., for network slices with strict latency requirements and high priority. The slice migration delay s_{md} is defined as:

$$S_{md} = \frac{MS^l}{CP_{m_c}^a}, \quad (6)$$

where MS^l represents the size of a MEC slice l , e.g., docker container images, configuration files, that require mobility, and $CP_{m_c}^a$ denotes the availability of the resources at the candidate RAN m_c , i.e., available bandwidth and computational resources.

C. RAN Resources Availability and Slice Utilization

The RAN(s) and their respective MEC servers offer finite network resources, which may be insufficient to handle a surge in traffic stemming from the emergence of IoT services and escalating demands for low-latency dedicated network slices. Thus, optimizing the resource capacity of RAN is essential to facilitate mobile users with their requested slice type and ensure the QoS requirements of each mobile user are met sufficiently. The function for RAN resources availability and slice utilization could be expressed as:

$$f_{m_A} = 1 - \frac{\sum_{l=1}^L RS_{lm}^t}{RS_m^{max}}, \quad (7)$$

where, $\sum_{l=1}^L RS_{lm}^t$ represents the total system resources utilized by all slices L within RAN m at time t , and RS_m^{max} denotes the maximum system resources, indicating the system capacity of RAN m .

IV. NETWORK SLICE TYPES AND PRIORITY OPTIMIZATION

In the context of vehicular networks, different types of network slices are required, e.g., traffic monitoring, infotainment, and safety-critical, each having varying priority levels, depending on their specific QoS requirements and criticality. We assume three types of vehicles, e.g., personal, passenger vehicles (CE), and emergency vehicles, each demanding different slice types, e.g., massive IoT, infotainment, and safety-critical, respectively. Each of the vehicle types exhibits different priorities and QoS requirements, and therefore tuning the reward function so that the DRL agent can learn to adapt to the varying needs of network slices is essential. There we formulated a multi-objective reward function that took into account these parameters of vehicles and adjusted its weight parameters in the reward function to ensure that several types of network slices co-exist in a network while their QoS and priorities are met in a highly dynamic and complex network scenario including the mobility scenarios. The different types

of network slices considered in the paper are presented in Table II, where the sample weights are given for different objectives and demands of vehicles. For example, in the case of vehicular mobility from the coverage area of one RAN to another, it is critical to provide real-time handover of the dedicated safety-critical network slice being used by the emergency vehicle, and therefore, it is given a higher priority reward and high penalty for service interruption as compared to the passenger vehicles (CE) and commercial vehicle types, as shown in Table II. These weights are dynamically adjusted by the proposed DRL agent as it learns to adapt optimal decisions meeting the QoS and priority levels of each vehicle type. Different slice types are considered in the paper, and their priorities are defined in the following subsections.

1) *Emergency Vehicles (Safety-Critical Slice)*: For network slices supporting safety-critical applications and deadline-sensitive tasks [28], e.g., remote driving assistance or real-time traffic monitoring, maintaining QoS and minimizing service interruptions and service failures is essential [29]. The DRL agent assigns higher priorities to the QoS reward and interruption penalty, ensuring that the learned policies prioritize QoS adherence and seamless handovers to maintain the reliability and integrity of safety-critical services.

2) *Passenger Vehicles with Consumer Electronics (Infotainment/Multimedia Slice)*: For network slices supporting high-bandwidth applications, e.g., infotainment services or multimedia streaming, efficient resource utilization is important. In these cases, the DRL agent assigns a higher priority to the resource reward component, encouraging the agent to learn policies that optimize resource allocation and minimize over-provisioning or under-utilization. For these types of passenger vehicles (CE) network slices, which involve consumer electronics with less critical and sensitive requirements, the DRL agent may assign lower priorities to the interruption penalty and priority reward components while focusing on the resource reward component.

3) *Personal Vehicles (Massive IoT Slice)*: For personal vehicles like cars, SUVs, and light trucks, connectivity needs are essential and driven by IoT applications, e.g., location tracking, software updates, and vehicle telemetry data reporting. These use cases involve a massive number of low data rate device connections rather than high bandwidth demands. As such, in this case, the DRL agent prioritizes support for an extremely large number of simultaneous IoT device connections while emphasizing less on relatively lower data rates. The QoS reward component is weighted to ensure sufficient data rates for transferring sensor data and firmware updates. Similarly, the resource reward aims to efficiently multiplex the massive number of IoT connections onto the available network resources. On the other hand, service interruptions are assigned a low penalty weight, as temporary disconnections can be tolerated for passenger vehicles (CE) and IoT services like video streaming and location tracking.

V. INTELLIGENT XAPP FOR NETWORK SLICING MANAGEMENT IN O-RAN-FRAMEWORK

We deployed an intelligent xAPP implementation that can subscribe to RAN functions and UE measurement reports

through its interaction with NRT-RIC and based on these indicators, the xAPP acts as an autonomous agent responsible for orchestrating network slice migration and management decisions in real-time. In this section, we present a comprehensive framework for our intelligent xAPP, focusing on its state space representation, action space definition, reward function design, and dynamic weight adjustment mechanism. In addition, we discussed the key components of the DRL approach, which includes the state transition probability, discount factor, policy, and Q-function. At last, we defined the primary objective of the xAPP, i.e., to learn an optimal policy to maximize cumulative rewards over time.

A. State Space (S)

The state space is represented as the current condition of the vehicular network environment and is defined as:

$$S = (M, Q, N, R, L) \quad (8)$$

where M represents the vehicle mobility patterns and predicted trajectories, Q is the signal strength measurements, e.g., RSRP/RSRQ for the current and neighbouring RANs, N is the network congestion levels, e.g., PRB utilization, buffer occupancy, number of mobile users being served by a RAN in a specific cell, R is the available computing and network resources at the current and target RANs, and L is the current state of the user's network slice, e.g., QoS parameters and slice type requested.

B. Action Space (A)

The action space includes the set of possible actions the DRL agent (xAPP) can take. The actions are represented as:

$$A = \{a_h, a_r, a_s\} \quad (9)$$

where a_h represents the handover actions that include initiating a handover or delaying a handover, a_r is the resource allocation action that includes allocating resources or adjusting resources assigned to a network slice, and a_s is the network slice management action that includes instantiating a network slice or terminating a network slice.

C. Reward Function (R)

We designed a reward function so that it can balance the trade-offs between QoS maintenance, resource efficiency, service continuity, and slice prioritization. The reward function consists of the following major components:

1) *QoS Reward*: This component represents the reward for maintaining the QoS requirements of the network slice, e.g., computational delay/latency and throughput. The QoS reward function is calculated as a function of the QoS parameters before and after a specific action is taken in a given state. A higher QoS reward is obtained when the actual QoS achieved by the mobile user after the action is taken in a given state is closer to the desired QoS target. This component ensures that the DRL agent learns policies that prioritize maintaining

TABLE II
WEIGHT FACTORS OF REWARD FUNCTION FOR DIFFERENT TYPES OF NETWORK SLICES

Vehicle Type	Slice Type	Mobility Pattern	Factor and Weight
Personal Vehicles	Massive IoT	low	QoS Reward: xx Resource Reward: xx Interruption Penalty: xx Priority Reward: xx
Passenger Vehicles (CE)	Multimedia/ Infotainment	Moderate	QoS Reward: xxxx Resource Reward: xx Interruption Penalty: xx Priority Reward: xx
Emergency Vehicles	Safety-critical	High	QoS Reward: xx Resource Reward: xx Interruption Penalty: xx Priority Reward: xx

the required QoS levels for different network slices. The QoS reward is given by:

$$\text{QoS_reward}(s, a) = \alpha \cdot \left(1 - \frac{|\rho_{tgt}^l - \rho_{aft}^l(s, a)|}{\rho_{tgt}^l} \right) \quad (10)$$

where ρ_{tgt}^l represents the desired QoS requirements of a mobile user requesting a particular slice l , and $\rho_{aft}^l(s, a)$ is the actual QoS achieved by the mobile user associated with a particular slice l after the action a is taken in a given state s . Here, α is the scaling factor.

2) *Resource Reward*: The Resource Reward (RSRC) component aims to minimize resource over-provisioning or under-utilization, resulting in efficient utilization of scarce network resources. It is determined based on the resource availability metrics of the RAN before and after an action is taken in a given state. A higher resource reward is obtained when the resource utility after a certain action is taken is close to the optimal resource utilization level but less than the maximum resource availability. This component encourages the DRL agent to learn policies that optimize resource usage and minimize the wastage of scarce network resources.

$$\text{RSRC_reward}(s, a) = \beta \cdot \left(1 - \frac{|\zeta_{opt}^m - \zeta_{aft}^m(s, a)|}{\zeta_{opt}^m} \right) \quad (11)$$

where ζ_{opt}^m represents the optimal resource utilization level of a RAN m , and $\zeta_{aft}^m(s, a)$ is the resource utility of a RAN m when the mobile user is associated with that particular RAN after the action a is taken in a given state s . Here, β is the scaling factor.

3) *Interruption Penalty*: The Interruption Penalty (ITP) component aims to reduce service interruptions, particularly for the slice types that are safety-critical. This component of the reward function penalizes service interruptions or handover

failures, which should be minimized. It is calculated based on the handover success rate or service interruption duration. A higher penalty is imposed when a vehicle type requiring safety-critical slice types is not handed over towards the next optimal candidate RAN to maintain reliable communication and reduce the service interruption duration. This component encourages the DRL agent to learn policies that minimize service disruptions and ensure seamless handovers during network slice migration. The ITP is given as:

$$\text{ITP_penalty}(s, a) = -\iota \cdot \left(1 - \chi_{\text{success}}^l(s, a)\right) \quad (12)$$

where ι is the scaling factor, and $\chi_{\text{success}}^l(s, a)$ is the handover success rate for a particular slice type l when an action a is taken in any given state s . The handover success rate χ can be defined as:

$$\chi_{\text{success}}^l(s, a) = \begin{cases} 0, & \text{if } \text{RAN}_{\text{curr}} = \text{RAN}_{\text{cand}}, \\ 1, & \text{if } \text{RAN}_{\text{curr}} \neq \text{RAN}_{\text{cand}}. \end{cases} \quad (13)$$

4) *Priority Reward*: The priority reward component rewards certain types of network slices or vehicles based on their respective priority levels. A higher reward is obtained for prioritizing critical network slices or high-priority vehicles. This component allows the DRL agent to learn policies that prioritize the migration of essential network slices or user groups with stringent QoS requirements. The priority reward component is defined as:

$$\text{Priority_reward}(s, a) = \delta \cdot \nu^l \quad (14)$$

where δ is the scaling factor, and ν^l is the slice priority level as determined through the slice type l and its associated requirements.

The overall reward function can be written as:

$$\begin{aligned} R(s, a) = & w_1 \cdot \text{QoS_reward}(s, a) \\ & + w_2 \cdot \text{RSRC_reward}(s, a) \\ & + w_3 \cdot \text{ITP_penalty}(s, a) \\ & + w_4 \cdot \text{Priority_reward}(s, a) \end{aligned} \quad (15)$$

Where, w_1, w_2, w_3, w_4 are weights in the reward function assigned to each individual reward components that are dynamically adjusted based on the vehicle type, network slice type, and mobility pattern. The DRL agent learns an optimal policy dynamically assigning weights to different components of the reward function based on the following function:

$$w_i = f_i(\text{vehicle_type}, \text{slice_type}, \text{mobility_pattern}) \quad (16)$$

Here, mobility pattern function $M(t)$ can be defined as a function that captures the vehicle's movement characteristics, e.g., the vehicle's position, channel conditions, speed, and distance from neighbouring RANs. The relationship is defined as:

$$M(t) = g(P_u(t), v_u(t), d_{u, \text{RAN}}(t)) \quad (17)$$

where $M(t)$ represents the mobility pattern at time t , $P_u(t)$ is the vehicle's position, $v_u(t)$ is the vehicle's speed, and

$d_{u, \text{RAN}}(t)$ is the distance from the vehicle to neighbouring RANs. This mobility pattern is captured by the centralized NRT-RIC within the O-RAN framework, enabling it to analyze the mobility patterns across the network. The NRT-RIC uses this information to optimize decision-making, ensuring communication strategies and resource allocation adapt to varying mobility scenarios and QoS requirements of different network slices and applications.

D. DRL Components

Our DRL framework includes several components that play an important role in the training process. These components are defined as follows:

1) *State Transition Probability*: The probability of transitioning from one state s to another state s' after an action a is taken. It is denoted as $P(s'|s, a)$.

2) *Discount Factor*: It determines the importance of future rewards in comparison to immediate rewards. The discount factor ensures that the agent focuses on valuing the immediate rewards more than the future rewards. The discount factor typically ranges between 0 and 1, inclusively ($\gamma \in [0, 1]$).

3) *Policy*: The policy, denoted as $\pi(a|s)$, defines the strategy that the agent, i.e., xAPP implemented in NRT-RIC, follows to select actions in different states.

4) *Q-function*: The Q-function, denoted as $Q(s, a)$, is used for estimation of the expected cumulative discounted reward for taking action a in the state s and following the optimal policy thereafter.

E. Objective

The primary objective of the DRL agent, i.e., xAPP implemented in the NRT-RIC, is to learn an optimal policy, denoted as $\pi^*(s)$, that maximizes the expected cumulative discounted reward over time. It can be expressed as:

$$\pi^*(s) = \arg \max_{\pi} \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \middle| s_0 = s, a \sim \pi(a|s) \right] \quad (18)$$

Here, $\pi^*(s)$ represents the optimal policy that maximizes the expected cumulative reward, γ is the discount factor, $R(s_t, a_t)$ is the reward obtained at time t , s_0 is the initial state s , and π is the policy followed by the agent. The pseudocode for the proposed approach and its working principles is provided in Algorithm 1.

VI. PERFORMANCE EVALUATION

We used the Open Networking Foundation (ONF) SD-RAN Platform [30] as our simulation environment, which aligns with the O-RAN architecture principles. The SD-RAN Platform provides a comprehensive implementation of the disaggregated RAN components, which includes O-RU, O-DU, O-CU, and the NRT-RIC; each of the components aligns closely with the specifications outlined by the O-RAN Alliance. Additionally, the platform incorporates virtual nodes hosted within an edge cloud that facilitate the emulation of real-world deployment scenarios.

Algorithm 1 xAPP Implementation in NRT-RIC (O-RAN) for Network Slice Management

```

procedure INITIALIZE
    Initialize state space  $S$ , action space  $A$ , reward function  $R$ ,
    and discount factor  $\gamma$ 
    Initialize initial policy  $\pi$ 
end procedure
for episode in episodes do
    Initialize state  $s$ 
    while not terminated do
        Select action  $a$  from policy  $\pi(s)$ 
        Execute action  $a$  in the network environment
        Observe next state  $s'$ , reward  $r = R(s, a)$ 
        Update Q-function or policy parameters using DRL al-
        gorithm (e.g., Q-learning, Policy Gradient)
         $s = s'$ 
    end while
end for
procedure OPTIMIZERewardFunction( $s, a$ )
    procedure OVERALLREWARD( $s, a$ )
        return using Eq. (15)
    end procedure
    procedure ADJUSTWEIGHTS
        return using Eq. (16)
    end procedure
end procedure
procedure UPDATEPOLICY
    Use DRL algorithm to learn optimal policy  $\pi^*(s)$ 
end procedure
procedure EXECUTEPOLICY
    Within the O-RAN framework, the xAPP implementing the
    DRL agent interacts with the NRT-RIC to:
    1. Obtain the current network state information (user
    mobility, signal strength, resource availability) through the E2
    interface
    2. Execute the chosen actions (initiate handover, delay
    handover, allocate resources, instantiate/terminate slices) through
    the E2 interface and control mechanisms
end procedure

```

TABLE III
SUMMARY OF SIMULATION PARAMETERS USED IN THE PERFORMANCE
EVALUATION SCENARIO

Parameter	Value
Types of network slices	3
Vehicle's speed	36 km/h
Number of RANs/MECs	3
Controller implementation	Centralized
Number of NRT-RICs	1
Maximum weight of a MEC slice	80-time frame rates
Slice types	Uplink and downlink
Slice scheduler type	Proportional fair
Transmit power of each RAN	40dBm
Vehicles mobility model	Waypoint and direct route
Propagation model	Log distance
DNN training interval	8 batches
DNN training batch size	32
DNN learning rate	0.01
DNN training optimizer	Adam

A. Simulation Scenario

We simulated complex and dynamic scenarios where vehicles move across coverage areas of different RANs, incorporating a small-scale fading model, i.e., the Rayleigh fading model. We designed a setup with three separate RANs, each equipped with disaggregated RAN components. We imple-

mented a centralized control architecture where all the RANs are managed by a centralized NRT-RIC, collectively forming the infrastructure supporting wireless connectivity and service continuity. The vehicles are simulated as mobile entities transitioning between the coverage areas of RANs. For emulating a more realistic mobility patterns, we modelled vehicles moving between RAN coverage areas every 50 seconds. We tested our trained model on 1000 such samples, where we analyzed handover efficiency across various network slices and vehicle types.

The simulation scenarios are designed to evaluate the effectiveness of handover policies tailored to different network slice types. We employed a DRL-based xAPP that communicates with NRT-RIC, managing the RAN components and enabling dynamic adjustment of handover probabilities depending on real-time network conditions and traffic priorities. This dynamic and adaptive approach ensures that optimal handover decisions are made at each time slot to maintain the specific QoS requirements of each network slice in the mobile environment. The DRL agent interacts with the simulation environment in real-time, generating training data dynamically rather than relying on a pre-collected static dataset. The training data consists of real-time Received Signal Strength Indicator (RSSI) values of mobile users collected by the proposed centralized NRT-RIC, which are used as state inputs to the DRL agent, i.e., DRL-based xAPP. The state inputs, the agent's actions, and the corresponding rewards form the state-action-reward-next-state tuples, which are then stored in a replay buffer. The agent observes around 1000 unique RSSI samples while interacting with the environment and stores them in a replay buffer. The samples are reused over multiple episodes, enabling the agent to refine its decision-making process efficiently. The dynamic nature of our environment allows the agent to continuously explore and adapt, creating new trajectories that evolve with its policy, providing diverse training experiences.

We evaluated a fully connected Deep Neural Network (DNN) consisting of one input layer, two hidden layers, and one output layer. The first hidden layer is comprised of 256 neurons, and the second hidden layer contains 512 neurons. We have summarized other DNN configurations, hyperparameters, and the simulation parameters used in this research in Table III. The DRL architecture was determined based on preliminary experiments that aimed to provide adequate model capacity while minimizing the risk of overfitting. We used ReLU activation functions and performed hyperparameter tuning to optimize stability and improve learning efficiency.

B. Performance Metrics: Evaluation and Comparison

We compared our proposal with other approaches in the literature that use programmable and centralized architectural solutions for network slice mobility management. The approaches proposed in [8], [31] use SDN-based migration modules to trigger slice mobility across MEC servers. The approach presented in [8] proposed migration of network slices from the source towards the target MEC server upon vehicle relocation based on resource availability on the MEC servers.

In [31], the authors introduced a mechanism to identify the specific network slice resources that need to be migrated, where network slices are deployed as independently deployable, stateless microservices. In the event of vehicle mobility, only the identified microservices are migrated to the target MEC server, depending upon its resource availability, to complete the network slice mobility procedure. These approaches only consider network slice mobility management based on MEC resource availability and don't consider any mechanism to identify the network slice type and their QoS requirements. Therefore, network slice mobility is managed uniformly, with all types of network slices treated equally, triggering handovers to MEC servers based on the mobile user's proximity or the resource availability on the MEC servers. In this paper, we refer to these approaches as the SDN-based Network Slice Mobility (SDN-NSM) approach.

We also compared our proposed approach to the conventional network slice mobility management scheme. In this approach, upon relocation, the SDN controller triggers the handover of the vehicle and its associated network slice to the next available RAN and their corresponding MEC server based on the channel conditions and signal strength, i.e., RSSI. The conventional approach also does not consider the network slice type, its QoS requirements, or the resource availability of the target MEC in the network slice mobility management decisions. We refer to the SDN-based Conventional Network Slice Mobility approach as (SDN-CON).

We have selected several performance metrics to assess the viability of the proposed approach for effective network slice mobility management using the O-RAN Framework. We evaluated our proposed approach for the following performance metrics:

1) *Handover Probability of Different Slices*: This performance metric is used to determine the likelihood of initiating and completing handovers for various network slice types. A high handover probability indicates a greater propensity for handover events in case the vehicles move from the coverage area of one RAN to another, resulting in service continuity and efficient network resource utilization. The handover probability convergence graph, as shown in Fig. 3, illustrates the dynamic adjustment of handover policies for different network slice types during the training of a DRL agent. More specifically, as seen in Fig. 3, as the training goes on and the agent learns to adapt dynamic handover policies based on the slice type and vehicular requirements. As seen, the safety-critical slice's handover probability reaches close to 1, indicating a high likelihood of handover initiation to maintain service quality for critical applications in the case of vehicular mobility. This convergence ensures the reliability, availability, and responsiveness of critical services within the 5G networks. In other words, our proposed NRT-RIC prioritises and guarantees uninterrupted connectivity for mission-critical applications, e.g., emergency communication systems, healthcare services, and public safety operations. We simulated congested environments in this simulation scenario, where we distributed load on RANs so that the RANs are highly loaded and can not accommodate all types of network slice handover requests. In such congested scenarios, our proposed NRT-RIC

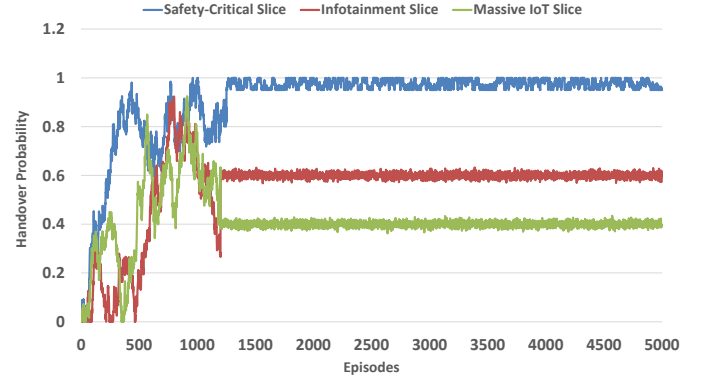


Fig. 3. Handover probability for different network slices with training

consistently prefers handovers of safety-critical network slices to stronger network cells or RANs, ensuring minimal disruption and latency-sensitive transmissions, as seen in Fig. 3.

In contrast, the handover probabilities of passenger vehicles (CE) slices, e.g., infotainment and massive IoT slice, converge to a moderate or lower value, exhibiting a more balanced approach to handover decisions based on traffic priorities, QoS requirements, and resource availability on the neighbouring RANs, as shown in Fig. 3. In addition, as shown in Fig. 3, the proposed approach converges fast, achieving optimal solution in 1200 episodes. This convergence shows the efficiency of the approach in terms of complexity, as it requires only 1200 episodes to find an optimal solution. Once the model is trained, it can effectively provide real-time network slice mobility decisions with response times ranging from 5 to 10 ms, meeting the latency requirements for vehicular communications.

2) *Service Continuity for Different Network Slice Types*: The performance metric is used to evaluate the potential of the proposed approach in enabling a seamless transition of network slices from one RAN to another during handover events. The metrics evaluate the system's ability to maintain uninterrupted connectivity and QoS requirements across different types of network slices. The graph, as shown in Fig. 4, demonstrates the service continuity for different network slice types, e.g., safety-critical, infotainment, and massive IoT, during handover events. We considered a specific scenario where, at $t = 60$ sec, the vehicle requiring a safety-critical slice changes its position by moving from its current serving RAN to the neighbouring candidate RANs. We simulated a highly congested environment where the resources at the candidate RANs are being fully utilized, and there is not enough space to accommodate new slice requests or slice migration requests from the incoming vehicle requesting safety-critical slices. The proposed NRT-RIC xAPP assessing the situation dynamically switches the two other vehicles using less-critical network slices, e.g., infotainment and massive IoT, to the neighbouring RAN, therefore accepting the slice migration request from the vehicle requiring the safety-critical slice. Thus, the safety-critical slice achieves service continuity upon relocation at $t = 60$ sec in our proposed case, as seen in Fig. 4a. In comparison, in the SDN-NSM and SDN-CON cases, the safety-critical slice faces severe degradation of service quality



Fig. 4. Performance comparison of service continuity in congested scenarios across network slices

upon relocation at $t = 60$ sec, as these approaches are unable to identify the network slice types and their QoS requirements, treating all types of network slices with same priority levels, as seen in Fig. 4a.

Consequently, as a result of switching the less-critical network slices, i.e., infotainment and massive IoT slice, to the neighbouring RANs to accommodate the safety-critical slice in our proposed case, the vehicles with less-critical requirements face a slight degradation in service quality but remain within acceptable thresholds, as shown in Fig. 4b and Fig. 4c, respectively. In SDN-NSM and SDN-CON cases, the vehicles with less-critical requirements continue to achieve the same performance levels, but at the expense of severe degradation of the QoS of the safety-critical slice, as shown

in Fig. 4a, Fig. 4b, and Fig. 4c.

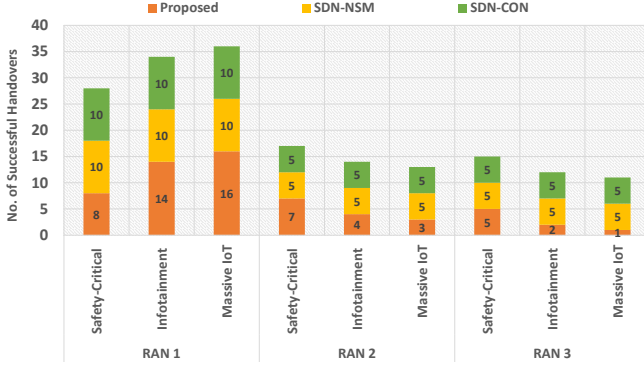
This highlights the ability of the proposed NRT-RIC xAPP to dynamically manage handovers based on slice priorities and resource availability while ensuring that critical services receive precedence and, at the same time, maintaining service continuity for all vehicles. In contrast, the SDN-NSM and SDN-CON approaches don't have any mechanisms in place to dynamically identify different network slice types and their QoS requirements and don't utilize the resources efficiently, e.g., where a network slice requiring fewer resources and high resources are treated equally.

3) Number of Successful Handovers per Different RANs:

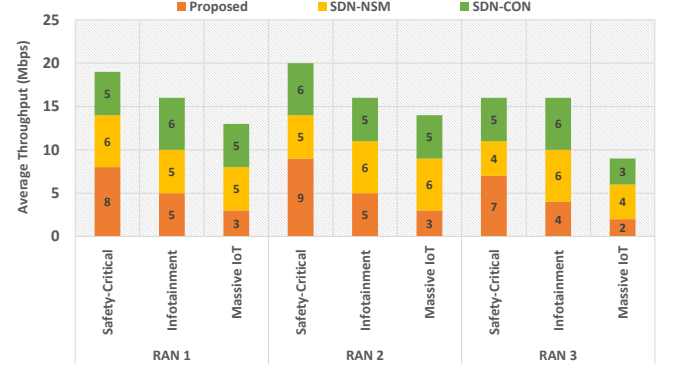
This metric is used to evaluate the effectiveness of handover procedures within each RAN. The metric measures the frequency of successful handover events, reflecting the ability of the proposed system to seamlessly transfer network slices and vehicles associated between different RANs. In this scenario, the trained model is tested using 1000 samples representing the transitioning of vehicles between coverage areas of different RANs approximately every 50 seconds. Fig. 5a offers insight into the effectiveness of the handover process as different vehicle types with different priorities and QoS requirements transition between coverage areas of different RANs. Specifically in our proposed case, the safety-critical vehicle exhibits a higher handover success rate upon relocation in congested scenarios. The proposed NRT-RIC xAPP prioritizes a strategy that aims to prevent frequent handovers, e.g., ping pong handovers, by ensuring that vehicles with less stringent QoS requirements, e.g., massive IoT, remain connected to their original RAN to avoid unnecessary handovers. This can be seen in Fig. 5a, where the vehicles with less critical slice requirements stay connected to their original RAN, i.e., RAN 1, more frequently than the safety-critical network slices. The safety-critical network slices are frequently handed over to the neighbouring RANs upon relocation to maintain service continuity. On the other hand, vehicles with low priority levels are only handed over towards the candidate RANs when they have ample resources available and unused resources, as shown in Fig. 5a. This strategy plays a key role in ensuring the efficient utilization of scarce RAN resources and simultaneously enhances public safety by providing seamless delivery of critical services.

In contrast, as seen in Fig. 5a, for the SDN-NSM and SDN-CON cases, all types of network slices are treated uniformly without taking into account their QoS requirements and priority levels. These approaches trigger frequent handover of non-critical slices as compared to the safety-critical slices, resulting in inefficient utilization of network resources and service quality degradation for safety-critical network slices.

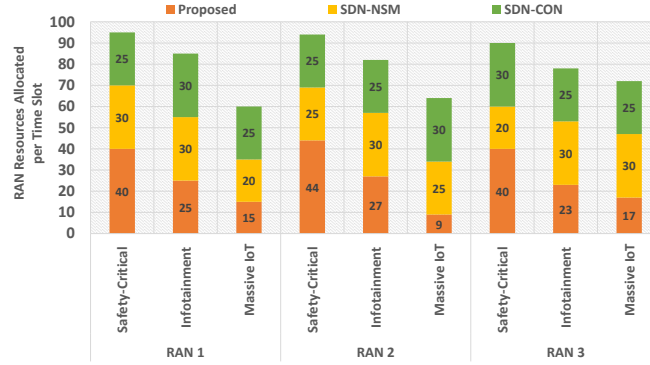
4) QoS Across Different RANs: This performance metric is used to assess the data rate achieved by each network slice across different RANs in case of vehicle mobility events. As seen in Fig. 5b, the average throughput per slice from the perspective of vehicles across three distinct RAN environments shows a trend where vehicles demanding safety-critical tasks consistently achieved higher throughput compared to other vehicle types in our proposed case. In contrast, in the SDN-CON and SDN-NSM cases, the throughput is evenly



(a) Number of successful handovers for different network slice types



(b) Average throughput achieved by different network slices in RAN



(c) Average resources allocated to different network slices by RAN

Fig. 5. Performance evaluation and comparison of network slice mobility management approaches

distributed amongst different types of network slices, as seen in Fig. 5b. This results in inefficient utilization of network resources, which could lead to scenarios where the network slice requiring less throughput is assigned more resources than needed.

In our proposed case, the safety-critical vehicle achieves higher throughput regardless of the RAN it is connected to. The average throughput achieved by different vehicle types is determined based on the performance observed when the trained model is tested with our dataset of 1000 samples. These samples represent the transitioning of vehicles between coverage areas of different RANs approximately every 50 seconds. This metrics shows the efficacy of our proposed NRT-RIC xAPP in coordinating the requirements across different RANs in case of mobility and prioritizing resources to ensure that critical services receive optimal data transfer rates during handover events.

5) *Resources Allocated per Slice Type*: This performance metric evaluates the distribution of RAN resources (weighted time frame rates) [30] among different slice types. Fig. 5c illustrates the number of resources allocated per slice type, showing the effectiveness of the proposed NRT-RIC xAPP in efficiently allocating the network resources based on QoS requirements. As seen in Fig. 5c, the safety-critical slice receives a higher allocation of RAN resources compared to the

infotainment and massive IoT slice. This allocation strategy is adapted to reflect the prioritization of critical services to meet their stringent requirements for reliability and responsiveness. In contrast, other slice types, e.g., massive IoT, receive comparatively fewer RAN resources, as they do not require real-time processes, are less critical, or have low demanding QoS requirements. This dynamic resource allocation approach enabled by the proposed NRT-RIC xAPP optimizes resource utilization within the RAN, allowing efficient delivery of critical services while accommodating the diverse needs of various applications and services.

In comparison, SDN-CON and SDN-NSM approaches lack mechanisms to differentiate between various network slice types and their QoS requirements. As a result, these approaches tend to allocate resources uniformly across all network slice types, often assigning more resources to less critical slices, as shown in Fig. 5c.

6) *Impact of Varying Traffic Density and Congestion Levels*: This metric is used to evaluate the performance of our proposed approach in terms of varying traffic densities and congestion levels. We also compared our proposed approach with SDN-NSM and SDN-CON approaches regarding handover latency across varying vehicle densities, i.e., 10 to 100 vehicles per cell. Handover latency is the delay in transferring a network slice from one RAN and its associated MEC server

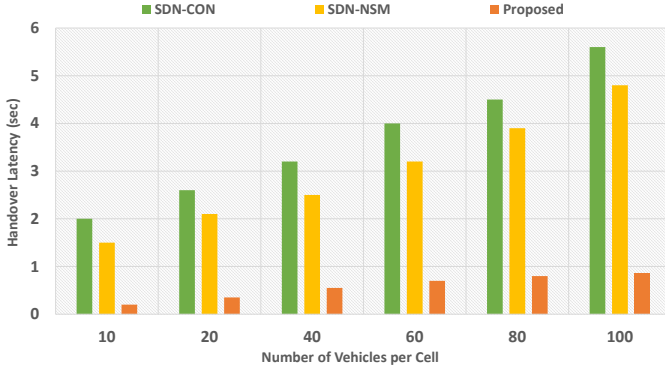


Fig. 6. Handover latency with varying vehicle density per cell

to another as a vehicle moves between coverage areas. As the vehicle density increases, radio access and MEC resources become more congested, resulting in longer delays.

The SDN-CON approach, which uses signal strength and channel conditions for handovers, experienced a significant latency increase, reaching more than 5 seconds at maximum traffic density, as shown in Fig. 6. The SDN-NSM method, which considers resource availability in handover decisions, performed slightly better but faced considerable delays. The higher latency is observed because these conventional approaches only act reactively, i.e., triggering handover only when signal strength drops below a certain level. These approaches are slow to react and identify the need to perform handover, resulting in higher handover latency because of resource contention and queuing delays as the vehicle density increases. In addition, both approaches fail to prioritize network slices based on their QoS requirements and, hence, treat all types of network slices uniformly, e.g., safety-critical slices and less critical ones like infotainment are treated equally, increasing the chances of bottlenecks for safety-critical applications during peak demand, because of their static resource allocation approach.

In contrast, our proposed approach shows superior performance as the proposed NRT-RIC consistently tracks and captures the network topology, i.e., network load and vehicle's position, and proactively performs handover of network slices based on their QoS requirements and priorities; it keeps handover latency within the acceptable range for handovers in 5G and beyond, i.e., less than 1 second, even in high-density scenarios, as shown in Fig. 6.

It is important to note here that the use of mobility models, e.g., waypoint or direct routes, does not impact network slice mobility decisions because the vehicle movement patterns or positions are consistently being captured and analyzed by the centralized NRT-RIC in the proposed O-RAN framework. The NRT-RIC continuously monitors the mobility parameters, e.g., vehicle's position, speed, and distance to neighbouring RAN, and uses this information to make timely handover and resource allocation decisions. As the proposed NRT-RIC dynamically tracks the mobility parameters, the specific movement model, e.g., waypoint or direct route, has minimal effect on the overall network slice mobility management.

VII. CONCLUSION, LIMITATION, AND FUTURE WORK

In conclusion, our study introduces a novel approach to network slicing for highly dynamic and complex networking environments such as vehicular networks using the O-RAN framework. We implemented a DRL-based xAPP within the O-RAN framework that seamlessly interacts with the NRT-RIC to automate intelligent decision-making in optimizing slice migration, resource allocation, and handover processes. Our simulations and performance evaluation demonstrate the significance of the proposed approach in maintaining QoS for different types of network slices, optimizing RAN resource utilization, minimizing service disruptions, and prioritizing safety-critical slices during vehicle mobility. This research opens directions for enhancing future vehicular communication networks' reliability, efficiency, and safety for intelligent and smart transportation systems.

A. Limitation

The proposed approach is an effective network slice mobility approach that efficiently differentiates between network slice types and considers their QoS requirements and priorities in mobility decisions. However, the proposed approach is currently limited in terms of its reliance on capturing the vehicle's current position in real-time and using that information to trigger network slice mobility handovers. The proposed approach currently doesn't have any mechanism to predict the vehicle's movement over upcoming time intervals that could assist in finding the optimal RAN and its associated MEC server and initiating network slice mobility handovers in advance, further reducing the handover latency and ensuring service continuity, in highly dynamic mobile network environments.

B. Future Works

1) *Integrating Federated Learning for RAN*: Integrating federated learning techniques holds promise for enhancing intelligence at the RAN and enabling collaborative decision-making in vehicular communication networks [32]. Future works include exploring the development of a federated learning framework that would enable distributed architecture for NRT-RIC and AI model training across different RANs. This approach could further optimize the decision-making response time of NRT-RICs, thus, facilitating rapid decision-making and enhancing network performance in real-time scenarios.

2) *Advancing Self-Optimizing Network Capabilities*: The machine-to-machine communication between IoT devices, vehicular networks, and retail applications forms the principles of self-organizing networks [33]. In the O-RAN initiative, further research is essential to enhance the collaboration between xAPPs and the NRT-RIC, as this synergy could significantly elevate self-optimizing network capabilities in vehicular network environments. Innovative applications and services of the O-RAN framework can be unlocked for proactive network management and predictive optimization of network resources by integrating technologies such as edge computing and predictive analytics with the NRT-RIC and its hosted apps. By continuously enhancing the interactions and communications

between the xAPPS, NRT-RIC, and the disaggregated RAN components, we could enable more efficient, reliable, and responsive vehicular communication networks for intelligent and smart transportation systems.

ACKNOWLEDGMENTS

This research work was funded by Institutional Fund Projects under grant no. (IFPIP: 55-830-1443). Therefore, the authors gratefully acknowledge technical and financial support from the Ministry of Education and King Abdulaziz University, Jeddah, Saudi Arabia.

REFERENCES

- [1] M. N. Avcil, M. Soyuturk, and B. Kantarci, "Fair and efficient resource allocation via vehicle-edge cooperation in 5G-V2X networks," *Vehicular Communications*, p. 100773, Apr. 2024.
- [2] X. Yin, J. Liu, X. Cheng, and X. Xiong, "A C-V2X compatible massive data download scheme based on heterogeneous vehicular network," *IEEE Trans. Consum. Electron.*, vol. 69, no. 4, pp. 962–973, Nov. 2023.
- [3] F. Marzouk, A. Radwan, H. R. Chi, and J. P. Barraca, "Highly flexible and traffic isolating RAN slicing: A consumer IoT-based use case," *IEEE Trans. Consum. Electron.*, vol. 69, no. 4, pp. 709–718, Nov. 2023.
- [4] S. D. A. Shah, M. A. Gregory, S. Li, R. dos Reis Fontes, and L. Hou, "SDN-based service mobility management in MEC-enabled 5G and beyond vehicular networks," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13 425–13 442, Jan. 2022.
- [5] A. Boukerche, A. Magnano, and N. Aljeri, "Mobile IP handover for vehicular networks: Methods, models, and classifications," *ACM Comput. Sur. (CSUR)*, vol. 49, no. 4, pp. 1–34, Feb. 2017.
- [6] N. Aljeri and A. Boukerche, "Mobility management in 5G-enabled vehicular networks: Models, protocols, and classification," *ACM Comput. Sur. (CSUR)*, vol. 53, no. 5, pp. 1–35, Sep. 2020.
- [7] L. Mei, J. Gou, Y. Cai, H. Cao, and Y. Liu, "Realtime mobile bandwidth and handoff predictions in 4G/5G networks," *Comput. Netw.*, vol. 204, p. 108736, Feb. 2022.
- [8] R. A. Addad, T. Taleb, H. Flinck, M. Bagaa, and D. Dutra, "Network slice mobility in next generation mobile systems: Challenges and potential solutions," *IEEE Netw.*, vol. 34, no. 1, pp. 84–93, Jan. 2020.
- [9] B. Hazarika, P. Saikia, K. Singh, and C.-P. Li, "Enhancing vehicular networks With hierarchical O-RAN slicing and federated DRL," *IEEE Trans. on Green Commun. and Netw.*, May 2024.
- [10] S.-p. Yeh, S. Bhattacharya, R. Sharma, and H. Moustafa, "Deep learning for intelligent and automated network slicing in 5G open RAN (ORAN) deployment," *IEEE Open J. of the Commun. Soc.*, Nov. 2023.
- [11] A. Filali, Z. Mlika, and S. Cherkaoui, "Open RAN slicing for MVNOs with deep reinforcement learning," *IEEE Internet Things J.*, Feb. 2024.
- [12] F. Linsalata, E. Moro, F. Gjeci, M. Magarini, U. Spagnolini, and A. Capone, "Addressing control challenges in vehicular networks through O-RAN: a novel architecture and simulation framework," *IEEE Trans. Veh. Tech.*, Jan. 2024.
- [13] R. A. Addad, D. L. C. Dutra, T. Taleb, and H. Flinck, "Toward using reinforcement learning for trigger selection in network slice mobility," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2241–2253, May 2021.
- [14] L. Cominardi, T. Deiss, M. Filippou, V. Sciancalepore, F. Giust, and D. Sabella, "MEC support for network slicing: Status and limitations from a standardization viewpoint," *IEEE Commun. Stand. Mag.*, vol. 4, no. 2, pp. 22–30, Jul. 2020.
- [15] A. Ksentini and P. A. Frangoudis, "Toward slicing-enabled multi-access edge computing in 5G," *IEEE Netw.*, vol. 34, no. 2, pp. 99–105, Apr. 2020.
- [16] P. Du, A. Nakao, L. Zhong, and R. Onishi, "Intelligent network slicing with edge computing for internet of vehicles," *IEEE Access*, vol. 9, pp. 128 106–128 116, Sep. 2021.
- [17] S. D. A. Shah, M. A. Gregory, and S. Li, "Cloud-native network slicing using software defined networking based multi-access edge computing: A survey," *IEEE Access*, vol. 9, pp. 10 903–10 924, Jan. 2021.
- [18] S. D'Oro, L. Bonati, F. Restuccia, M. Polese, M. Zorzi, and T. Melodia, "SI-EDGE: Network slicing at the edge," in *Proc. 21st Int. Symp. on T. Algo. Fdn. and Proto. Des. for Mobile Netw. and Mobile Comput.*, 2020, pp. 1–10.
- [19] K. Hejja, S. Berri, and H. Labiod, "Network slicing with load-balancing for task offloading in vehicular edge computing," *Veh. Commun.*, vol. 34, p. 100419, Apr. 2022.
- [20] S. Bolettieri, D. T. Bui, and R. Bruno, "Towards end-to-end application slicing in multi-access edge computing systems: Architecture discussion and proof-of-concept," *Future Gener. Comput. Syst.*, Nov. 2022.
- [21] MEC ETSI ISG, "Multi-access edge computing (MEC); support for network slicing," ETSI, Sophia-Antipolis, FR, Tech. Rep. GR MEC 024, Nov. 2019.
- [22] MEC ETSI ISG, "Multi-access edge computing (MEC); study on MEC support for V2X use cases," ETSI, Sophia-Antipolis, FR, Tech. Rep. GR MEC 022, Sep. 2020.
- [23] MEC ETSI ISG, "Multi-access edge computing (MEC); study on inter-MEC systems and MEC-cloud systems coordination," ETSI, Sophia-Antipolis, FR, Tech. Rep. GR MEC 035, Jun. 2021.
- [24] Z. Mlika and S. Cherkaoui, "Network slicing with MEC and deep reinforcement learning for the internet of vehicles," *IEEE Netw.*, vol. 35, no. 3, pp. 132–138, Jan. 2021.
- [25] S. Jošilo and G. Dán, "Joint wireless and edge computing resource management with dynamic network slice selection," *IEEE/ACM Trans. on Netw.*, vol. 30, no. 4, pp. 1865–1878, Aug. 2022.
- [26] A. Filali, B. Nour, S. Cherkaoui, and A. Kobbane, "Communication and computation O-RAN resource slicing for URLLC services using deep reinforcement learning," *IEEE Commun. Stand. Mag.*, vol. 7, no. 1, pp. 66–73, Mar. 2023.
- [27] C. Liu and K. Liu, "Toward reliable DNN-based task partitioning and offloading in vehicular edge computing," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3349–3360, Feb. 2024.
- [28] L. Yang, K. Wang, M. Xiao, H. Zhang, M. Li, X. Li, and H. Ji, "Online data driven scheduling for deadline-sensitive tasks of mobile edge computing enabled consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 4142–4154, Feb. 2024.
- [29] C. Tang, G. Yan, H. Wu, and C. Zhu, "Computation offloading and resource allocation in failure-aware vehicular edge computing," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1877–1888, Feb. 2024.
- [30] Open Networking Foundation, "SDRAN-in-a-Box (RiaB)." [Online]. Available: <https://github.com/onosproject/sdran-in-a-box>
- [31] S. D. A. Shah, M. A. Gregory, and S. Li, "Toward network slicing enabled edge computing: A cloud-native approach for slice mobility," *IEEE Internet Things J.*, vol. 11, no. 2, pp. 2684–2700, Jan. 2024.
- [32] M. K. Hasan, N. Jahan, M. Z. A. Nazri, S. Islam, M. A. Khan, A. I. Alzahrani, N. Alalwan, and Y. Nam, "Federated learning for computational offloading and resource management of vehicular edge computing in 6G-V2X network," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3827–3847, Feb. 2024.
- [33] Q. V. Khanh, A. Chehri, Q. N. Minh, V.-H. Nguyen, and N. T. Ban, "An efficient routing algorithm for self-organizing networks in 5G-based intelligent transportation systems," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1757–1765, Feb. 2024.