electronics

MDPI

Article

QoS-Driven Slicing Management for Vehicular Communications

Prohim Tam [1], Seyha Ros [1], Inseok Song [1] and Seokhoon Kim [1,2,*]

[1] Department of Software Convergence, Soonchunhyang University, Asan 31538, Republic of Korea; prohimtam@gmail.com (P.T.); seyha@sch.ac.kr (S.R.); sis5041@sch.ac.kr (I.S.)
[2] Department of Computer Software Engineering, Soonchunhyang University, Asan 31538, Republic of Korea
* Correspondence: seokhoon@sch.ac.kr

Abstract: Network slicing is introduced for elastically instantiating logical network infrastructure isolation to support different application types with diversified quality of service (QoS) class indicators. In particular, vehicular communications are a trending area that consists of massive mission-critical applications in the range of safety-critical, intelligent transport systems, and on-board infotainment. Slicing management can be achieved if the network infrastructure has computing sufficiency, a dynamic control policy, elastic resource virtualization, and cross-tier orchestration. To support the functionality of slicing management, incorporating core network infrastructure with deep learning and reinforcement learning has become a hot topic for researchers and practitioners in analyzing vehicular traffic/resource patterns before orchestrating the steering policies. In this paper, we propose QoS-driven management by considering (edge) resource block utilization, scheduling, and slice instantiation in a three-tier resource placement, namely, small base stations/access points, macro base stations, and core networks. The proposed scheme integrates recurrent neural networks to trigger hidden states of resource availability and predict the output of QoS. The intelligent agent and slice controller, namely, RDQ3N, gathers the resource states from three-tier observations and optimizes the action on allocation and scheduling algorithms. Experiments are conducted on both physical and virtual representational vehicle-to-everything (V2X) environments; furthermore, service requests are set to massive thresholds for rendering V2X congestion flow entries.

Keywords: deep q-networks; quality of service; recurrent neural networks; slicing management; vehicle-to-everything

check for updates

Citation: Tam, P.; Ros, S.; Song, I.; Kim, S. QoS-Driven Slicing Management for Vehicular Communications. *Electronics* **2024**, *13*, 314. https://doi.org/10.3390/electronics13020314

Academic Editor: Yolanda Blanco Fernández

Received: 29 November 2023
Revised: 7 January 2024
Accepted: 9 January 2024
Published: 10 January 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The advent of vehicular communications has led to a new era of vehicle-to-everything (V2X) with enhanced safety, intelligent transportation systems (ITS), improved traffic management, and the realization of autonomous driving [1,2]. The user-centric service demands of the Internet of Vehicles (IoV) in 5G and beyond networks necessitate the deployment of a paradigm shift in logical network architecture and resource-efficient mechanisms such as intelligent slicing orchestration to cater to the diverse quality of service (QoS) requirements in V2X communications. The multi-class requirements are within safety-critical applications, non-safety services in ITS, and on-board infotainment [3,4]. However, the heterogeneous features of vehicular mobility and protocols pose significant challenges to ensuring the applicability of slicing federation and orchestration in real-world use cases.

Before commencing the augmentation of network slicing (NS), it is crucial to have a comprehensive multi-aspect awareness of potential challenges in conventional approaches. There are primary challenges for slicing orchestration in vehicular communications, such as scalability [5], end-to-end (E2E) management and orchestration (MANO) [5], slice optimality [6], core architecture evolutions and APIs [7], mobility management [8], and virtualization sharing [8]. E2E MANO confronts complexities arising from the proliferation of IoV applications from V2X [9], such as vehicle-to-network (V2N), vehicle-to-infrastructure (V2I), vehicle-to-vehicle

(V2V), and vehicle-to-pedestrian (V2P), leading to an increased number of dedicated network slices. To handle the scalability, slice grouping and prioritization have to be either well-fixed or autoscaling. Core architecture evolutions and APIs need rich feature observability and elastic reconfigurability to the interfaces for enabling on-demand, autonomous, predictability, and dynamic configurations. Therefore, the convergence of well-known enabling NS paradigms with prediction-based deep learning and autonomy-based deep reinforcement learning (DRL) introduces innovative schemes (e.g., resource sharing, elastic virtualization, and QoS-driven isolation between slices), which can be evaluated by the reward functions of QoS-aware agents [10–12].

QoS metrics play a significant role in IoV services by dictating the performance and reliability required to support a diverse range of applications [13,14]. These metrics can be captured either reactively or proactively. Reactive capturing, while convenient for analyzing past performance, is less suitable for intelligent NS. In contrast, proactive capturing involves assessing QoS performance prior to data transmission to ensure compliance with predefined class indicators. In the context of vehicular communications, the proactive approach entails evaluating factors such as bandwidth allocation, resource availability, and computational resources before deploying a service. To support proactivity, recurrent neural networks (RNN) and long short-term memory (LSTM) have been implemented by several studies [15–18]; furthermore, integrating with DRL [19] brings promising approaches for long-term self-organizing and adaptive slicing control policies.

Therefore, in this paper, we first gather well-known enabling paradigms in E2E NS for supporting the applicability of our proposed system architecture, and then we emphasize our contributions with QoS-driven system models and reliable environment representation to support the integration of our proposed RNN and deep Q-networks (DQN), namely, the RDQ3N-assisted intelligent slice controller. V2X communications consist of numerous natural parameters; therefore, we normalize the input states, following our sole priority in logical resource slice sharing. We input the processing features to obtain the hidden states of resource availability and allocability in small base stations (SBSs) as edge nodes and macro base stations (MBS) as resource-assisted nodes, later considered DQN states. The output determined the proactive QoS performances of the processing V2X features. RDQ3N has a group-based fixed slice instance of (1) safety-critical, (2) ITS, and (3) on-board infotainment. The reward function jointly considers resource utilization and QoS performances in evaluating the efficiency of action configurations in each timeslot, namely, slice instantiation, resource placement, and scheduling. Experiments are conducted to substantiate the RDQ3N-assisted intelligent slice controller in our built V2X environment, comparing key metrics including average delays per slice, resource efficiency, and reliability with reference schemes. Table 1 presents the acronyms used in this paper.

Table 1. Primary acronyms and descriptions.

Acronym	Description	Acronym	Description
DQN	Deep Q-networks	QoS	Quality of service
DNN	Deep neural networks	RNN	Recurrent neural networks
DRL	Deep reinforcement learning	SDN	Software-defined networking
E2E	End-to-end	SBS	Small base stations
EPC	Evolved packet core	SLAs	Service-level agreements
IoV	Internet of Vehicles	TD	Temporal difference
ITS	Intelligent transport systems	V2I	Vehicle-to-infrastructure
LSTM	Long short-term memory	V2N	Vehicle-to-network
MBS	Macro base stations	V2P	Vehicle-to-pedestrian
MANO	Management and orchestration	V2V	Vehicle-to-vehicle
MEC	Multi-access edge computing	V2X	Vehicle-to-everything
NFV	Network functions virtualization	VNFs	Virtual network functions
NS	Network slicing	VNFM	VNFs manager

2. Preliminary of Intelligent NS in Vehicular Communications

2.1. Primary Enabling Paradigms

Software-defined networking (SDN) [20], network function virtualization (NFV) [21], multi-access edge/cloud computing [22], and virtualization techniques [23] are undeniably the primary enabling paradigms for high-applicability infrastructure in E2E NS. In the operational phase, SDN is dynamically responsible for creating and managing network slices in response to application demands. SDN architecture supports resource allocation, traffic steering, and service-level agreements (SLAs) enforcement. Additionally, SDN ensures the security and isolation of network slices, which requires virtualization from NFV to support efficient and reliable operation of logical isolation networks in a 5G and beyond environment [24,25]. NFV complements SDN by providing software-based network functions that instantiate and orchestrate service function chaining across the E2E network [26]. Together, SDN and NFV provide a holistic approach in terms of new instance creation, management, and optimization of logical network slices. The operational phase can be leveraged closer to edge networks using multi-access edge computing (MEC), which enables better decisions in compute-intensive task offloading [27], therefore reducing backbone latency and improving response times for IoV applications.

Despite the popularity of these paradigms, a converged platform that fully supports joint functionality and operability can be challenging to deploy as a reliable system architecture. The positions of SDN resources, controllers, and applications in NFV are varied in different scenarios [28], which leads to more complexity at the core orchestration level. The cross-tier management of edge-core collaboration is difficult to design, particularly for supporting rich feature state observability in resource-constrained environments.

2.2. Add-on Paradigms for Intelligence

On top of primary enabling paradigms, self-organizing flexibility and autonomy have been added to NS MANO capability using artificial intelligence [29], blockchain [30], federated learning [31], and edge intelligence [32]. In terms of artificial intelligence choices, many studies [33–38] have introduced intelligent NS design for V2X services with RNN/LSTM as early feature prediction and DRL as a policy maker. In [34], the authors reviewed how different research studies apply RNN and its variant, LSTM, to predict traffic and manage resources. RNN/LSTM outperforms other techniques in traffic prediction for vehicular-specific slices and supports predicting user mobility for inter-slice resource allocation. However, the computational cost of RNN/LSTM has led researchers to explore lighter variants like simplified or soft-gated recurrent units, which demonstrated comparable accuracy with reduced computation time. Moreover, RNN-based QoS prediction in IoV was presented in [35]. The proposed RNN utilized a multi-layer gated recurrent unit structure to capture QoS variations and model the evolving IoV environment. By leveraging experience features and objective factors, the proposed RNN and collaborative filtering can handle data sparsity issues and enhance the precision of QoS predictions.

Beyond RNN/LSTM, DRL can emerge as a pivotal add-on paradigm in the realization of intelligent and efficient resource management for 5G/6G NS, where the significant challenges of tasks such as admission control, resource allocation, scheduling, and orchestration demand autonomous decision making and policy making [36,37]. In [38], the authors introduced a novel multi-agent-based dynamic slicing framework for vehicular networks. The proposed DRL-assisted approach leveraged a multi-agent deep deterministic policy gradient to enable base stations to cooperatively allocate spectrum and computing resources. Experimental results demonstrated the effectiveness of the framework in achieving significant cost reductions and high QoS satisfaction rates. Furthermore, the DRL-based approach for resource optimization in UAVs and IoV networks necessitates utilizing 5G/6G infrastructure [39–41]. An online MEC-based scheme was also developed for slice selection, coverage selection, resource block, and power allocation [42].

Building upon the promising advancements in intelligent NS using RNN/LSTM and DRL, the role of the core orchestrator should be expanded and integrated into cross-tier management of the edge-core framework to achieve E2E NS for reliable vehicular communications.

2.3. Slice Instances in Vehicular Communications

NS is employed to facilitate diverse QoS requirements, particularly in massive use cases such as V2X services. However, the creation of slice instances necessitates adherence to a standardized set of specific criteria. With well-defined slice instances, scalability can be effectively covered and extended to a wide range of applications. There are several well-known studies that categorize slice instances and use cases differently. Table 2 presents the expected performances and critical requirements of use cases and possible slices of V2X services [5,43].

Table 2. Expected performances of slice instances from complementary studies [5,43].

Slice Instances	Latency	Data Rate	Scalability (Per Vehicle)	Ref.
Autonomous vehicles	1 ms	10 Mb/s	50–200	[5]
Tele-operated vehicles	20 ms	1Mb/s or 25 Mb/s	50–100	
Remote diagnostic	Not a concern	Not a concern	10–20	
Vehicle infotainment system	100 ms	0.5–15 Mb/s	5–15	
Slice Instances	**Latency**	**Data Rate**	**Reliability**	**Ref.**
Localization and navigation	10–100 ms	1 Mb/s	Low to high	[43]
Driving/transportation safety	50–100 ms	1 Mb/s	99.9%	
Autonomous driving	1 ms	10 Mb/s	Close to 100%	
Infotainment services	Up to 100 ms	15 Mb/s	Fair	

Each study classifies the slice criteria differently based on its own system models and target optimization objectives. In [44], the authors proposed two slices as follows: (1) slice 1 (security services) aims for low latency and high reliability (high-quality V2V links for safety information), and (2) slice 2 (infotainment service) focuses on high transmission rates (the roadside unit provides infotainment services via high-quality V2I links).

The primary target of optimization in this study [44] was to maximize infotainment slice throughput while ensuring latency constraints for the security slice. The authors considered traffic admission control, spectrum allocation, and transmit power allocation. In [45], the authors proposed a network clustering and slicing algorithm to group vehicles based on traffic demands. Slice grouping was based on use case requirements that prioritize safety messages for autonomous driving and video streaming for infotainment. Furthermore, in [46], the authors presented a road-state-based adaptive NS framework for vehicular networks, which prioritizes emergency traffic during incidents while maintaining acceptable QoS for non-emergency services. The study employed a Mininet-WIFI emulator to create a simulated vehicular network, the ONOS SDN controller to manage network resources, and the OpenVirteX NS tool (Open Network Foundation (ONF), Korea University, Seoul, Republic of Korea) to implement the proposed scheme.

3. Proposed System Framework

3.1. Working Flow

Our proposed framework runs on SDN and NFV platforms from edge to core. The network slice request originates from the NFV orchestrator, which acts as the central control plane for managing and orchestrating network slices. The NFV orchestrator receives the request from the V2X service provider or application, which specifies the desired network slice characteristics, such as latency, bandwidth, and QoS requirements. Figure 1 illustrates

our proposed system framework and its working flow. Overall, there are three tiers in this edge-core framework, as follows:

- ***Tier-1***. We represent SBSs with three possible general entities, denoted as $S = \{1, 2, \ldots s\}$, to symbolize eNodeB, gNodeB, or access points as vehicular NFV-enabled edge computing and network connectivity. eNodeB or gNodeB, the base station component of LTE and 5G networks, can be deployed with NFV infrastructure. Traditionally, eNodeB and gNodeB have been implemented as dedicated hardware appliances, but NFV enables the virtualization of eNodeB functions onto general-purpose hardware, such as servers or (edge) cloud platforms. In our framework, tier-1 is prioritized for low-latency slicing services since it is a computing resource-constrained environment. Vehicle nodes are denoted as $V = \{1, 2, \ldots v\}$. Upon receiving the service request at timeslot-t, denoted as $E_t = \{1_t, 2_t, \ldots e_t\}$, the SDN controller collaborates with the NFV orchestrator to determine the optimal network path for V2X traffic within the requested service e_t. The controller observes and considers the capabilities of the SBS resources, current topology states, and traffic patterns, denoted as $X_t = \{1_t, 2_t, \ldots x_t\}$. X_t is input to the main RNN execution, part of RDQ3N. The SDN controller translated the determined path into flow rules, which are instructions to direct traffic in tier-1 enabling slicing-aware traffic forwarding.
- ***Tier-2***. We denote MBS as $M = \{1, 2, \ldots m\}$ with higher resource capabilities. In this tier, the SDN controller continuously monitors the network load and traffic patterns within the network slice. The controller adjusts traffic flow across MBS-m in tier-2, which ensures that the load is distributed evenly and that V2X traffic receives the QoS-driven priority based on the sliced modelling. As node-v moves across the network, the controller coordinates with the mobility management entity to manage handover between MBSs within the network slice, while ensuring that V2X communication remains uninterrupted and that QoS expectation is maintained during handover.
- ***Tier-3.*** We refer to the evolved packet core (EPC), 5G core networks, or core data center. The SDN controller establishes and manages connectivity to external entities for the network slice, therefore enabling V2X communication with entities outside the slice. The core entity collects hidden states h_t and QoS metrics QoS_t proactively from our proposed RDQ3N (main RNN) to build a virtual representational environment. The NFV orchestrator, in conjunction with the virtual network functions manager (VNFM), orchestrates the allocation of NFV resources, such as virtual network functions (VNFs) and computing resources, to the defined slice instances. The VNFM deploys and configures the required VNFs onto the allocated SBS resources in tier-1, while ensuring that the required network functionalities are provided within the slice.

To enhance the realm of NS, the core orchestration serves as the backbone of the network and orchestrates the allocation of resources and services across the various slices. Hence, interface N2 establishes a robust communication channel between the core network controller and access point, and is responsible for orchestrating slicing, often utilizing a protocol such as the next-generation application protocol (NGAP), enabling communication control and user-plane critical information exchange, such as slicing configuration, resource allocation, and user mobility management. Thus, it allows the core network to manage and direct traffic control, ensuring the optimal allocation of resources for diverse applications of slicing. Simultaneously, the interface connects the core network to SBS, which extends network convergence and capacity. For the xn interface, we aim to provide the corresponding slice-specific configurations, resource allocations, and traffic prioritization across the network, ensuring that each slice receives the necessary resource and operates. Lastly, access points to the SBS interface ensure handovers and consist of service delivery within network, which handles data forwarding and load balancing across multiple SBSs within a slice.

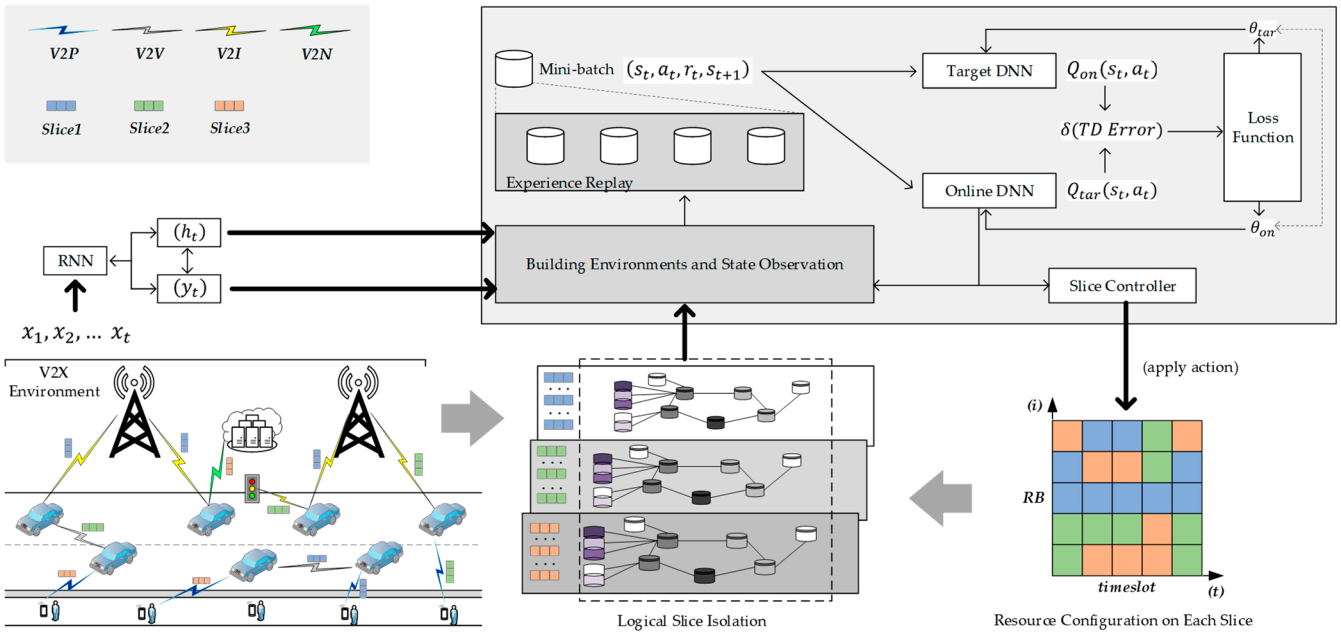


Figure 1. System framework between the V2X environment and RDQ3N components.

This subsection presents the collaborative working flow from the infrastructure layer to SDN and NFV (tier-1, tier-2, and tier-3) in enabling the dynamic provisioning and management of E2E network slices. The details of the main RNN and DQN to achieve an RDQ3N-assisted intelligent slice controller will be presented in Section 4. The extension of Figure 1 description will be given in Section 4 on the target and online deep neural networks (DNN), as well as how it associates with the slice controller to adjust resource configuration for each logical slice isolation.

3.2. Slice Instances

Based on our preliminary studies in Section 2, we propose three slice instances: (1) safety-critical service, (2) ITS-supported service, and (3) on-board infotainment services. These instances correspond to slices 1, 2, and 3, respectively. The expected performance of each slice is given in Table 3. The slicing indices are denoted as L_i, where i ranges from 1 to 3. However, due to multi-service involvement, we cannot establish fixed and nonduplicated criteria for each slice. For instance, autonomous driving can be considered both safety-critical and ITS-supported, depending on the use cases. Autonomous systems require planning a safe path that avoids collisions and adheres to traffic rules, which necessitates their consideration as a safety-critical slice instance. Simultaneously, autonomous systems also coordinate vehicle movements to reduce congestion and improve traffic flow, which makes it an ITS-supported slice instance as well. Therefore, our slicing model is a balanced trade-off that determines as follows:

- ***Slice 1***. The safety-critical slice instance is prioritized with ultra-reliable transmission of mission-critical services, while ensuring rapid latency with a maximum delay of 1–20 ms. The data rate ranges from 1 Mb/s to 10 Mb/s, which emphasizes vehicle-cooperative awareness messages that require low bandwidth but more dense requests. Scalability can be between 50 and 200. Reliability expectation is targeted as high. If the agent outputs actions on slice identification as L_1, the allocation metrics in MBS-m and scheduling in SBS-s will be higher priority compared to L_2 and L_3.
- ***Slice 2***. The ITS-supported slice instance acknowledges the importance of assisting applications that enhance traffic management and efficiency. With a maximum delay of 1–50 ms, slice 2 balances low-latency requirements with broader application coverage. The data rate is in the same range as with safety-critical applications. Scalability is specified between 50 and 100. The reliability level for ITS-supported applications is

classified as medium compared to slice 1. The guaranteed bit rates can be arranged less than L_1 in congestion states.

- ***Slice 3***. The on-board infotainment slice instance caters to non-safety-related services that prioritize high data rates for non-mission-critical streaming entertainment. With a maximum delay of up to 100 ms, slice 3 allows for a more relaxed transmission timeframe. The data rate spans from 0.5 Mb/s to 15 Mb/s, accommodating the varying bandwidth needs of heavy-traffic streaming data for infotainment content. Scalability is less, ranging from 5 to 15. The reliability level for this slice is classified as low since it can afford the unavoidable loss in a congested and constrained timeslot.

Table 3. Slice instances for our approach.

Slice Instances	Max. Delay	Data Rate	Scalability (Per Vehicle)	Reliability
Safety-critical	1–20 ms	1–10 Mb/s	50–200	High
ITS-supported	1–50 ms	1–10 Mb/s	50–100	Low–High
On-board infotainment	Up to 100 ms	0.5–15 Mb/s	5–15	Low

4. RDQ3N-Assisted Intelligent Slice Controller

4.1. Environment Components

In our study, we synchronized the physical and logical infrastructure to support the integration of RNN and DQN as RDQ3N for the intelligent slice controller. We gathered the parameters that matter to our virtual environment for building RDQ3N agent policies. Let x_t denote the state of vehicular edge networks that later input to the main RNN component. h_t is a partial feature in state observations that is input to build the virtual environment. Let s_t denote the state of the vehicular core networks at time t, including resource block conditions and V2X service request demands. The action a_t is a three-tuple adjustment on slice identification $L_i(e_t)$ of the requested services, resource allocation $\rho_t(m|e_t)$, and scheduling capacity $\varphi_t(s|e_t)$ in that timeslot-t. Equation (1) presents the action spaces of the RDQ3N framework. The proposed agent at time t obtains the output of QoS predictions as QoS_t from the main RNN component. After deploying a_t to s_t, the agent evaluates a reward r_t that considering the performances of QoS metrics, denoted as QoS_t, including expected latency L_t, expected throughput T_t, expected reliability (loss ratio) P_t, and resource efficiency R_t.

$$a_t \leftarrow \{L_i(e_t), \rho_t(m|e_t),\ \varphi_t(s|e_t)\} \tag{1}$$

4.2. QoS-Driven Models

The primary target is to evaluate the actions from RDQ3N that guide SDN control and NFV orchestration to achieve better QoS performance. The QoS-driven model is considered as the goal of our approach, represented as G_t in Equation (2), which aims to optimize policy π_t in maximizing the cumulative discounted reward. k denotes the possible future timesteps in the learning process. As k-step grow larger, the discount factor γ^k and the future rewards (r_{t+k+1}) have less impact on the RDQ3N policy making; therefore, our RDQ3N agent gains the capability of balancing the short- and long-term decision weights. The reward r_t, as given in Equation (3), reflects the performances of a_t on slice identifications and resource placement, including (1) sub-reward on throughput r_t^{TP}, (2) sub-reward on latency r_t^{LC}, and (4) sub-reward on resource efficiency r_t^{RP}.

$$G_t = \sum_{k=0}^{N} \gamma^k r_{t+k+1},\ \forall k \in N \tag{2}$$

$$r_t = r_t^{TP} + r_t^{LC} + r_t^{RP} \tag{3}$$

4.3. Algorithm Flows

There are three primary phases in our study, including (1) RDQ3N predictions on hidden state and slicing QoS, (2) action selections in RDQ3N, and (3) slice controller operations.

In the first phase, in the context of vehicular edge networks, the hidden resource state at timeslot-t (h_t) represents the aggregated features of the local/edge resource availability, traffic patterns, and congestion queue conditions within tier-1. The calculation of h_t involves a linear transformation of the current input state x_t and the previous hidden resource state h_{t-1} (temporal dynamics of the vehicular environment). Equation (4) represents the output o_t of the transformation in a single hidden unit, which is the weighted sum of the tier-1 states that represents the aggregated input to the hidden state. U denotes the size of the input vector x_t, and V is the size of the hidden state vector h_{t-1}. $W_{hx_u} \cdot x_{t_u}$ signifies the impact of the tier-1 x_t on the hidden resource state, and $W_{hh_v} \cdot h_{t-1_v}$ captures the temporal dependencies from h_{t-1}. The bias b_h is a constant vector added to the weighted sum that provides an additional degree of freedom and control over the h_t calculation. Equation (5) shows that the hidden resource state h_t is then passed through the hyperbolic tangent ($tanh$) activation function. h_t encapsulates significant features of current states in tier-1 by considering both instantaneous conditions and experience replays.

$$o_t = \sum_{u=1}^{U} W_{hx_u} \cdot x_{t_u} + \sum_{v=1}^{V} W_{hh_v} \cdot h_{t-1_v} + b_h \tag{4}$$

$$h_t = \tanh(o_t) \tag{5}$$

The main DQN component, a primary RDQ3N agent, leverages h_t along with other relevant states in tier-2 to assist action selection in the second phase. The convergence of RNN and DQN components enables the proposed RDQ3N agent to learn and adapt the slicing policies with optimal actions based on the temporal expansion of hidden resource states in the vehicular edge and core environments. After obtaining h_t, transformation is computed for output QoS_t following Equations (6) and (7). For a single output unit in predicting QoS for vehicular services, we expand the output transformation z_t, and later use activation functions to set a fix label class of V2X-QoS criticality, namely, QoS_{t_u} of the input requested service demands. P denotes the size of the output vector QoS_t.

$$z_t = \sum_{v=1}^{V} W_{Qh_v} \cdot h_{t_v} + b_Q \tag{6}$$

$$QoS_{t_u} = \frac{e^{z_{t_u}}}{\sum_{p=1}^{P} e^{z_{t_p}}} \tag{7}$$

In the second phase, the hidden state observation and prediction (h_t and QoS_t) serve as features to the main DQN component of the RDQ3N-assisted intelligent slice controller. RDQ3N executes the main DQN class to obtain the optimal policy π^* on the slicing management. Algorithm 1 presents the integration of the main DQN to RDQ3N agent on the built vehicular environment. We build a virtual representation based on the predicted and hidden state resources in tier-1 and tier-2, following current resource allocations for each slice instance. We transform the observed physical states in tier-1 into a functional state representation that encodes the resource availability and current slice allocations. In the main DQN, we determine the early (hyper) parameters on both online and target DNNs. In pseudocode line 4, RDQ3N iterates through training steps. The state observations are randomized from batches of RNN outputs of the requested service, including e_t and x_t. The RDQ3N agent sets the slice identification of e_t, whether it belongs to L_1, L_2, or L_3. The resource allocation to the identified slice is obtained and re-configured (CPU, memory, and bandwidth), depending on the current policy π and partial ε-greedy. The performance might be ineffective during the training phase; however, after obtaining π^*, the performance will be optimized accordingly. RDQ3N also schedules the tasks or steering for the selected

slice, associated to intelligent slice controller with ***SliceController()*** function. After applying (1) $L_i(e_t)$ on service request clustering, (2) $\rho_t(m|e_t)$ on tier-2 allocation, and (3) $\varphi_t(s|e_t)$ on tier-1 scheduling, the RDQ3N agent monitors the physical infrastructure response and formulates s_{t+1} and $r_t\left(.|r_t^{TP}, r_t^{LC}, r_t^{RP}\right)$. Experience replay B is a compute-intensive input for RDQ3N, which requires random mini-batch sampling to perform the ***OptimizeOnline()*** function. Then, we use the sampling batch one-by-one to train predicted-Q as $Q_{pre}(s_t, a_t)$ and improve the online network parameters θ_{on}. We give the general target-Q formulation as $Q_{tar}(s_t, a_t)$, and we aim to identify the temporal difference (TD) error δ and minimize the loss of the networks. We exchange the weights θ_{tar} for periodic improvement using θ_{on}. The hyperparameter τ determines the balance between early and updated values.

Algorithm 1. Training RDQ3N for QoS-Driven Slicing Management

Initialize the main DQN with early online and target DNNs
Initialize all state observation $\sim s_t(.|h_t \,\&\, QoS_t)$
Initialize experience replay B
For each step **do**:
  Sample s_t of vehicular edge-core systems and NFV-based infrastructure
  For each timeslot t **do**:
    Detect service request $e_t, x_t \rightarrow h_t, QoS_t$
    Select $L_i(e_t), \rho_t(m|e_t),\ \varphi_t(s|e_t)$ based on current π and partial ε-greedy
    SliceController(): deploy a_t to SDN steering and NFV-enabled infrastructure
    Formulate s_{t+1} and $r_t(.|r_t^{TP}, r_t^{LC}, r_t^{RP})$
    $B \leftarrow (s_t, a_t, r_t, s_{t+1})$ // store the batch to experience replay
    Sample mini-batch from B and perform ***OptimizeOnline()***:
      $Q_{tar}(s_t, a_t) = r_t + \gamma \max_{a'} Q_{tar}\left(s_{t+1}, a'\right)$ // general target-Q
      Use online DNN: $Q_{pre}(s_t, a_t)$ // early-Q
      $\delta = Q_{tar}(s_t, a_t) - Q_{pre}(s_t, a_t)$ // TD error
      $Loss = \frac{1}{N}\sum_{k=1}^{N} \delta_k^2$ // loss function
      Minimize Loss: Backpropagation
      $\theta_{on} \leftarrow \theta_{on} - \alpha \nabla \theta_{on} LOSS$ // update weights on online DNN
    Update target DNNs:
      $\theta_{tar} \leftarrow \tau\theta_{online} + (1-\tau)\theta_{tar}$ // update weights on online DNN
    Obtain output reactive QoS metrics $QoS_t(.|L_t + T_t + P_t + U_t)$
    // compared to main RNN output
    Update reward efficiency and ***OptimizeTarget()*** for π^*
    Alter environment to s_{t+1} for next-timeslot
  End for
End for

The reward function r_t emphasizes actions that enhance QoS and resource utilization while penalizing suboptimal adjustments to the experience replay. We assess the stability and effectiveness of learned policies under changing service demands throughout multiple timeslots. Each QoS_t of slicing V2X services is measured by the threshold of the identified slices. Depending on the satisfaction criteria, we update epsilon (ε)-greedy. RDQ3N keeps training until policy π^* is obtained with minimized loss and satisfied QoS expectations.

When obtaining optimal action from RDQ3N, the functionality of the intelligent slice controller consists of flow rule steering, property configuration, and self-organizing adaptation. From the main RNN component of RDQ3N, we continuously monitor the V2X incoming service requests and generate QoS predictions. Action-assisted steering leads to re-routing traffic for specific slices, adjusting resource properties in MEC-equipped MBS, and modifying scheduling parameters in offloading queues of SBS. The intelligent

slice controller translates RDQ3N recommendations into configuration updates for the SDN controller and NFV orchestrator, which update central and local flow entry rules. The intelligent slice controller also continuously monitors the performance of the slices following the implemented configuration, which is evaluated as rewards. Further software execution steps of the proposed framework are given in Figure 2, Section 5.2.

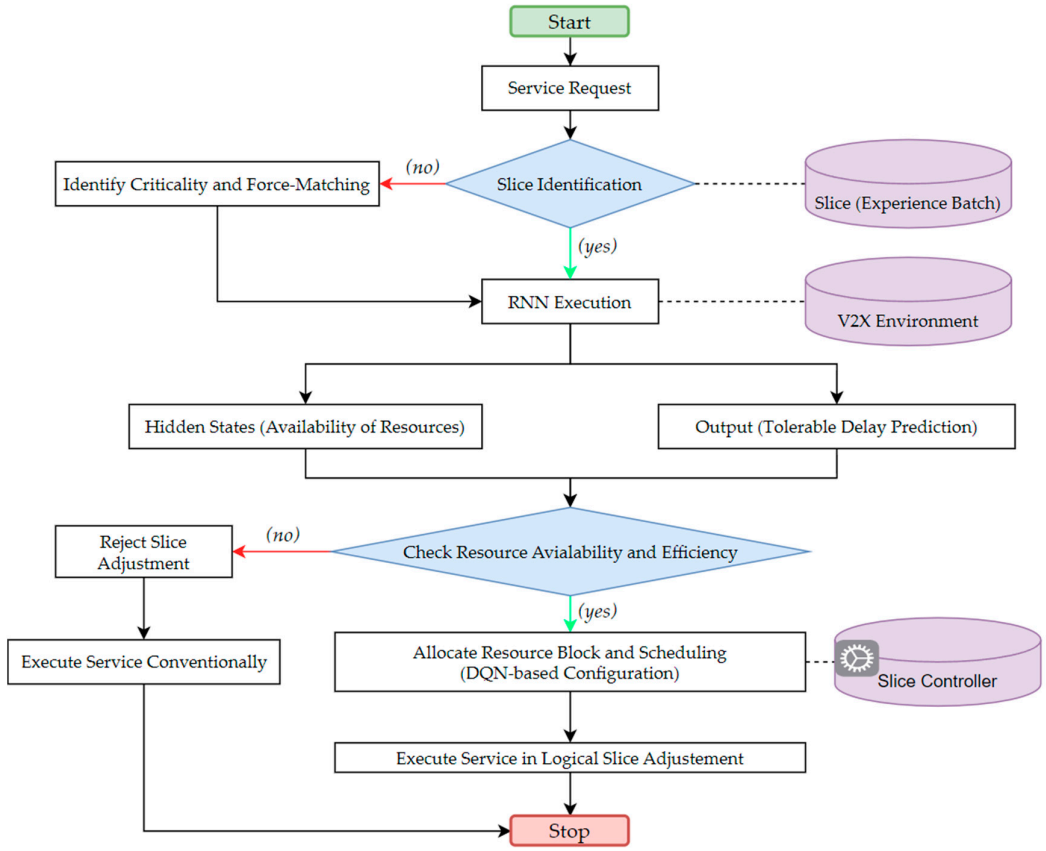


Figure 2. Overview flowchart of the proposed RDQ3N experiment.

5. Experiment and Results

5.1. Environment Setup

The (hyper) parameters of the simulation are given in Table 4. There are three major entities to discuss in this subsection, as follows:

- ***Integration infrastructure.*** To effectively evaluate the performance of the intelligent slice controller, a comprehensive simulation setup was designed on Mininet-based host core topology and PyTorch-based RDQ3N training. PyTorch was selected as the framework due to its flexibility and dynamic computation graph, while Gymnasium provided support functions for developing the representational V2X environment. Mininet was employed for SDN (RYU controller) and NFV simulation to offer a scalable flow entry installation and customizable resource virtualization.
- ***RDQ3N execution.*** The RNN configuration, with a hidden size of 256, two layers, and an input size of 10, was chosen to handle features in this V2X environment. A dropout rate of 0.2 helped prevent overfitting in our process. The DQN configuration, with a learning rate of 0.001, a discount factor of 0.99, and an exploration rate starting at 0.9, ensured stable training and balanced immediate and future rewards. A large replay buffer size (1,000,000) and batch size (128) facilitated effective experience replay. Target DNN update frequency of 1000 steps provided stability, and an initial exploration phase of 5000 steps helped gather the V2X experiences. The environment parameters, with a small action space (three actions) and a state space of 12 parameters, simplified learning complexity and also provided enough observations for the RDQ3N agent.

Longer training steps for the DQN (100,000) compared to the RNN (10,000) accounted for the criteria of Q convergence. ReLU, Sigmoid, and Tanh activation functions offered non-linearity for diverse action selections and prediction classes throughout the framework. The training steps match the transmission flows generated by the Mininet emulator for 10 h.

- ***Service flow and request operations.*** The slice configuration is three instances, and there are 120 vehicle nodes, 10 tier-1 (SBS), 2 tier-2 nodes (MBS), and service requests at a rate of 1000 per second, aimed at modeling a realistic V2X environment. This comprehensive setup aimed to strike a balance between the real-world environment, computational efficiency, and our proposed QoS-driven model in vehicular communications.

Table 4. Primary (hyper) parameter setup on PyTorch-based RDQ3N and Mininet environments.

Purpose	Specification
RNN and DQN hosts	Python (PyTorch, Gymnasium)
SDN and NFV core hosts	Mininet
SDN controller	RYU
Interfaces between RDQ3N and controller hosts	RESTful APIs
Hidden size	256
RNN number of layers	2
RNN input size	10
RNN dropout	0.2
Learning rate	0.001
Discount factor	0.99
Exploration rate	0.9 (exponential decay: 1000)
Minimum epsilon	0.05
Replay buffer size	1,000,000
Batch size	128
Target DNN update frequency	1000 steps
Target DNN update rate	0.005
Initial exploration steps	5000
Action space	3
State space	12
RNN training steps	10,000
DQN training steps	100,000
Mininet (service flow creation) times	10 h (5 h for training, 5 h for validation)
Action functions	ReLU, Sigmoid, Tanh
Number of slice instances	3
Number of vehicle nodes	120
Tier-1 and tier-2 nodes	(10), (2)
Service requests per second	1000

5.2. Proposed and Reference Schemes

Figure 2 presents the overview flowchart of the proposed RDQ3N framework associated with the V2X environment and slice controller. The process of the proposed scheme can be described as follows.

- ***Initial service request***: The execution phase begins with the reception of a service request from a node-v or vehicular application. The processing request encapsulates the specific requirements of the application.
- ***Slice identification***: The next step involves identifying the appropriate network slice for the received service request, which is achieved through a two-way approach:
    - Slice container (experience batch) or new RNN execution: The system maintains a slice container, also known as an experience batch, which stores slice identifiers and early QoS predictions. The experience batch is leveraged to efficiently identify the most suitable slice for the current service request (less computing response).
- ***Criticality identification and force-matching***: In cases where the slice cannot be identified for the service request, a criticality identification process is triggered. Our system assesses the urgency of the service request and force-matches it with the closest slice instance (class 1, 2, or 3) to ensure real-time service delivery.
- ***RNN-assisted proactivity***: When the slice is identified from the experience batch or force-matching, the main RNN from RDQ3N is employed to predict QoS_t and gather the hidden resource availability states h_t. The main RNN component interacts with the V2X environment (tier-1) to gather real-time input state x_t.
- ***QoS prediction validation***: The predicted QoS output is compared against the maximum tolerable delay associated with the pre-determined slice. If the predicted QoS does not meet the delay threshold, the predicted feature is synchronized to the RDQ3N, the main DQN agent. The feedback mechanism enables the main DQN component in RDQ3N to refine its policy for future slice identifications.
- ***Resource allocation and scheduling***: Once the slice is identified, the execution phase proceeds with resource allocation and scheduling.
    - Resource availability gathering: The main RNN component in RDQ3N, in conjunction with the V2X environment (tier-1), gathers input and output features to maintain hidden states reflecting resource availability and QoS predictions. This real-time resource monitoring enables a major weight in the placement process.
    - Resource availability utilization: The intelligent controller checks the availability and efficiency of the resources associated with the selected slice. If the resources are sufficient and efficiently utilized, the main DQN component in RDQ3N proceeds with current actions $L_i(e_t)$, $\rho_t(m|e_t)$, and $\varphi_t(s|e_t)$.
    - Resource unavailability handling: If the required resources (for serving the current requests) are not available at the specified timeslot, the slice instantiation and adjustment are rejected. In this case, the framework routes the service request to be executed conventionally without logical slice isolation.

For the reference schemes, we selected three policy settings: (1) resource-oriented slicing management using genetic algorithm (RS-GA), (2) priority-based slicing management using weighted fair queuing (PS-WFQ), and (3) multi-agent on tier-2 and tier-3 (MA-SM) to coordinate the slicing management.

5.3. Evaluation Metrics

In this subsection, the results and discussion of the evaluation metrics of the proposed and reference schemes are given. We consider the following metrics as the evaluation of our study to determine the strengths and weaknesses of this framework following the simulation configuration described in Section 5.2. Each metric is explained as follows:

- The loss (train, validation) and accuracy (train, validation) in executing the main RNN associated with tier-1 vehicular edge environment input are captured and illustrated in Figure 3. The training of the RDQ3N (RNN) for predicting early QoS and hidden states based on resource availability using Mininet-generated flow features demonstrates significant effects on slicing management. Initially, the main RNN components obtained high training and validation losses with low accuracy within the first 100 steps.

However, as training progresses, both training and validation losses decrease substantially, and accuracy reaches 98.71% at step [170/10,000]. The model maintains high accuracy and relatively low losses throughout the rest of the training process, with the exception of a slight increase in validation loss (early high peak of 0.1399) at step [2750/10,000]. The problem encountered with validation loss increment suggests a possible degree of overfitting or the need for further regularization. Despite this, the validation accuracy remains high, at an average of 97.18%, indicating that the model is still capable of making reliable QoS predictions. We obtained this result to showcase the applicability of the RDQ3N model to improve NS control in vehicular communications. However, the slight increase in validation loss at the later stage of training suggests that the model may not generalize well to new, namely unseen vehicular communication scenarios. Therefore, we integrated DQN into RDQ3N to tackle the generalization by evaluating the actions taken with three sub-rewards concurrently. With future unsatisfied rewards, RDQ3N activates the actions on hyperparameter tuning, batch normalization, regularization, and cross-validation for better assessment of the model and generalizability.

- The main reward, given in Equation (3), consists of three potential QoS-driven subcomponents: (1) throughput r_t^{TP}, (2) latency r_t^{LC}, and (3) resource efficiency r_t^{RP}. Figure 4 illustrates each sub-reward within 100,000 steps of RDQ3N (DQN component). The optimal main reward is a total value of 100, which balances each sub-reward from ranges of 20 to 40, depending on the incoming service requests in that particular timeslot. However, throughout our simulation execution (Mininet-generated flows), the congestion states varied, and the reward performance fluctuated to reflect the randomness of the V2X environment. Our proposed QoS-driven settings prioritized latency performance (slices 1 and 2) higher than throughput-maximized slice 3 (infotainment streaming). Therefore, the better results of the sub-reward of latency reflect our expectations compared to the other two sub-rewards. Resource efficiency remains challenging to overcome at every high-peak congestion state, which illustrates a low fluctuation value range between 0 to the highest peak of 17.87 and at the end of 10.09. However, RDQ3N remains effective in showcasing satisfied service request processing, acceptance, and successful ratios compared to reference schemes.
- The average delay per slice, total throughput, acceptance ratios, and successful ratios of the V2X service requests are plotted in Figure 5 in parts (a), (b), (c), and (d), respectively. Within 10 h of service flow generations at a constant rate of 1000 requests per second, our proposed RDQ3N-assisted intelligent slice controller obtained an average delay of 14.24 ms for safety-critical slice (L_1), which is 4.99 ms, 0.99 ms, and 11.31 ms lower than the average delay of MA-SM, PS-WFQ, and RS-GA, respectively. For the other 2 slice instances (L_2 and L_3), the improvements from our scheme achieved (14 ms, 1.61 ms), (15.9 ms, 12.35 ms), and (32.1 ms, 11.37 ms) better than MA-SM, PS-WFQ, and RS-GA, respectively. In Figure 5b, the total throughput represents the efficiency of bandwidth resource allocation and scheduling for each instance. By comparing reference schemes, we can showcase that the RDQ3N improvements are highly notable, with 2255.88 Mb/s, 1671.02 Mb/s, and 3050.12 Mb/s on slices 1, 2, and 3, respectively. PS-WFQ prioritized slice 1 as the most significant serving resource (2254.12 Mb/s); however, PS-WFQ's throughput on slice 3 had the lowest outcome (2921.09 Mb/s) among all the management schemes due to the bias toward less priority. Our scheme aimed to balance (1) priority in constrained states and (2) high transmission rates in non-congested states; therefore, the throughput outcome for all three slices is well preserved. The throughputs of MA-SM and RS-GA are (2200.14 Mb/s, 1662.88 Mb/s, and 3021.12 Mb/s) and (2191.04 Mb/s, 1650.23 Mb/s, 3013.91 Mb/s), on each slice instance, respectively. The improved throughput of RDQ3N is critical for massive machine-type communications, ultra-dense requests, and resource-constrained environments. Figure 5c illustrates the acceptance ratios of initial service requests, which depend on how each scheme optimizes the resource availability, QoS efficiency, and slice instantiation process. Within

the first 4 h flow generation, RDQ3N had an acceptance ratio below 99.00%; however, from the fifth-hour simulation, RDQ3N gained experience batch, environment awareness, and Q convergence that led to better performances from 99.01% to 99.79%. From this point to the end of simulation time, RDQ3N maintained a satisfied acceptance rate on average of 99.46%, which is 0.7016%, 2.6891%, and 1.2491% higher than MA-SM, PS-WFQ, and RS-GA, respectively. From the fifth-hour timeframe, MA-SM, PS-WFQ, and RS-GA achieved only 98.76%, 96.77%, and 98.21% acceptance ratios, respectively. The acceptance ratios illustrate how the slice controller is able to proactively plan the steering path and resource orchestration placement; however, only successful (completion) ratios can truly determine the reliability metric of the management scheme. Figure 5d showcases the success percentage of all accepted service requests. Table 5 gives the successful ratios of each slice between the proposed and reference schemes.

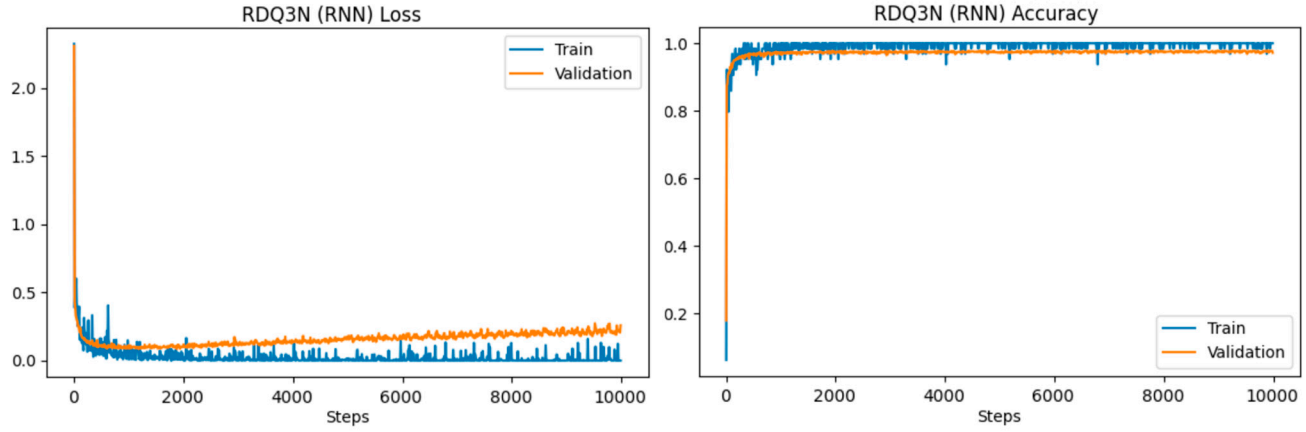


Figure 3. Loss and accuracy on RDQ3N (RNN component) over Mininet-generated flows.

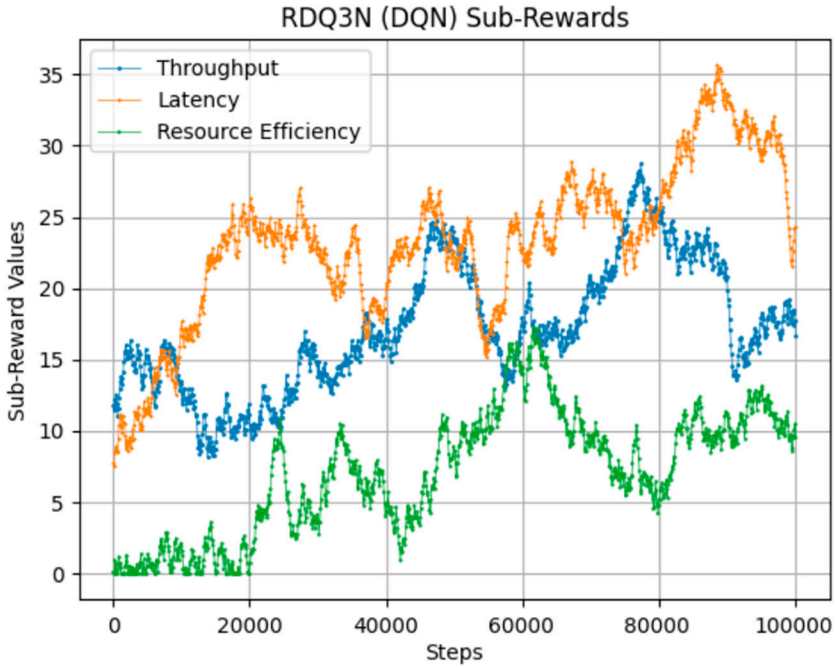


Figure 4. Three sub-rewards of RDQ3N (DQN component) over a virtual representation environment (with tier-2 states and RNN-based hidden state resource availability).

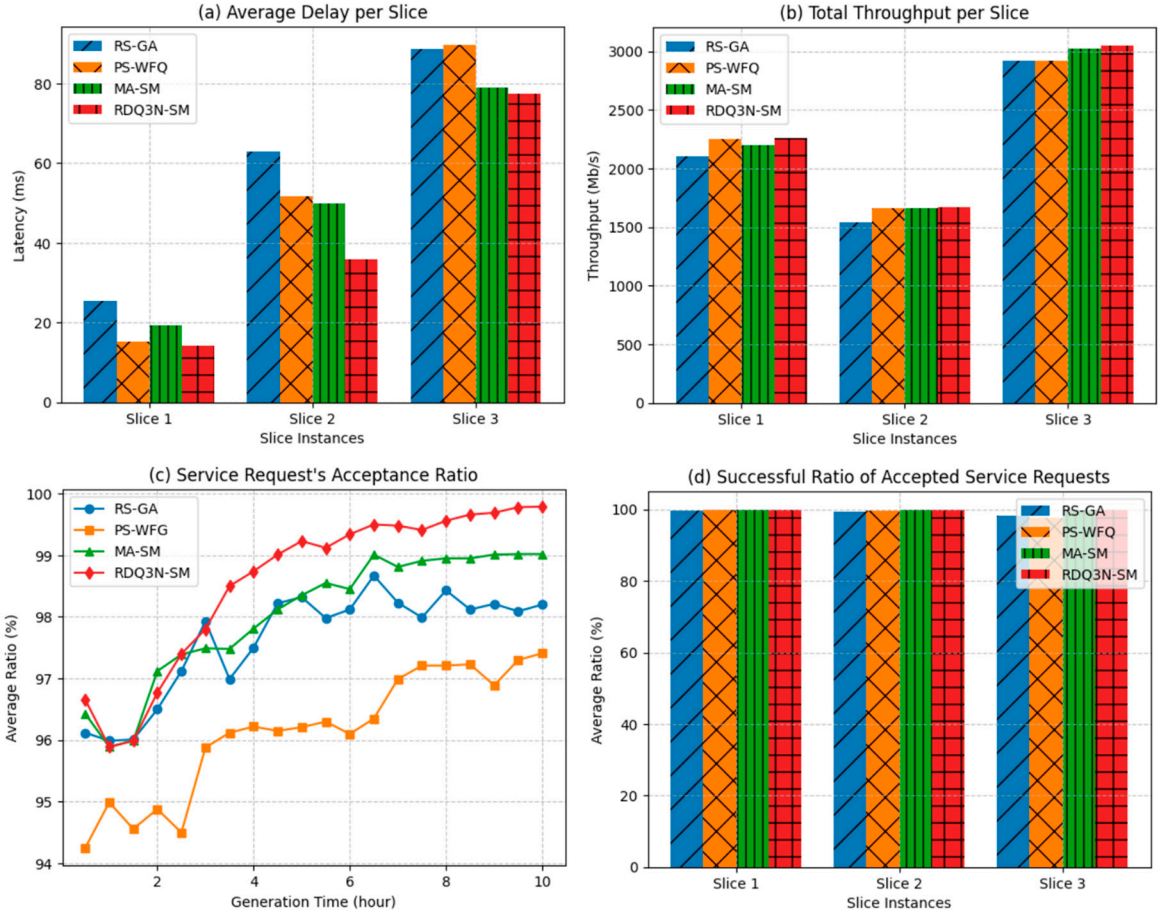


Figure 5. Results on (**a**) average delay per slice, (**b**) total throughput per slice, (**c**) acceptance ratios, and (**d**) successful (completion) ratios between proposed and reference schemes.

Table 5. Successful (completion) ratios of all accepted service requests in each slice.

Condition	RS-GA	PS-WFG	MA-SM	RDQ3N-SM
Slice 1	99.56%	99.92%	99.89%	100.00%
Slice 2	99.45%	99.59%	99.81%	99.95%
Slice 3	98.23%	97.63%	99.79%	99.91%

RDQ3N has demonstrated significant improvements in terms of latency, reliability, and resource efficiency. By leveraging the main RNN and DQN components, RDQ3N effectively predicts early QoS and hidden states based on resource availability, which enables proactive resource orchestration and slice instantiation. The approach simulation led to reduced average delay, improved throughput, and higher acceptance ratios, which enhanced the reliability and performance of V2X services ranging from high safety-critical to non-safety services. The evaluation metrics showcased the key contributions of the RDQ3N scheme in the QoS-driven mechanism to ensure that critical V2X services receive the necessary bandwidth/computing resources while balancing the needs of all slice instances to maximize overall QoS performances.

6. Conclusions and Future Studies

The deployment of RDQ3N in a three-tier vehicular edge-core architecture demonstrated its effectiveness in SDN- and NFV-enabled systems for E2E V2X slicing management. Our experiments showed that RDQ3N enhanced reliability metrics, latency, and resource efficiency for serving massive vehicular services, ranging within safety-critical,

ITS-supported, and infotainment services. The QoS-driven models are formulated with the objective of evaluating core RDQ3N rewards to emphasize the primary goal of the controller and orchestrator settings. The proactivity in predicting QoS and resource availability from the main RNN component made the proposed scheme a compelling solution for achieving diverse V2X QoS requirements. Furthermore, the main DQN component corrected RNN errors through adaptation in a virtual representational environment and optimized actions with the functional capability to allocate and schedule resources in real-time cross-tier orchestration. The intelligent slice controller leveraged the potential outputs and recommendations from RDQ3N and optimized the physical infrastructure (V2X environment) with SDN-based steering and NFV-enabled MEC.

Our future studies will discuss scalability challenges through standardization and interoperability that support proof-of-concept in the intelligent slicing orchestration of vehicular communications systems. We will consider end-to-end integration, hierarchical orchestration, and continuous learning. Thus, we expand the implementation on real-world data that can achieve greater scalability, efficiently manage more service instances, and meet the dynamic demands of a large-scale vehicular network.

Author Contributions: Conceptualization, P.T. and S.K.; methodology, P.T. and S.R.; software, P.T. and S.R.; validation, I.S. and S.R.; formal analysis, I.S. and S.R.; investigation, S.K.; resources, S.K.; data curation, P.T.; writing—original draft preparation, P.T.; writing—review and editing, I.S., S.R. and S.K.; visualization, I.S.; supervision, S.K.; project administration, S.K.; funding acquisition, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by an Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-00167197, Development of Intelligent 5G/6G Infrastructure Technology for the Smart City), in part by the National Research Foundation of Korea (NRF), Ministry of Education, through the Basic Science Research Program under Grant NRF-2020R1I1A3066543, in part by BK21 FOUR (Fostering Outstanding Universities for Research) under Grant 5199990914048, and in part by the Soonchunhyang University Research Fund.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zeadally, S.; Javed, M.A.; Hamida, E.B. Vehicular Communications for ITS: Standardization and Challenges. *IEEE Commun. Stand. Mag.* **2020**, *4*, 11–17. [CrossRef]
2. Arena, F.; Pau, G. An Overview of Vehicular Communications. *Future Internet* **2019**, *11*, 27. [CrossRef]
3. Wang, X.; Li, X.; Jing, T.; Yu, H. An Integrated Dependability Guarantee Provisioning for Cluster-Based IoV Networks with Slicing. *IET Intell. Transp. Syst.* **2023**, *17*, 1752–1768. [CrossRef]
4. Sanchez-Iborra, R.; Santa, J.; Gallego-Madrid, J.; Covaci, S.; Skarmeta, A.F. Empowering the Internet of Vehicles with Multi-RAT 5G Network Slicing. *Sensors* **2019**, *19*, 3107. [CrossRef]
5. Wijethilaka, S.; Liyanage, M. Survey on Network Slicing for Internet of Things Realization in 5G Networks. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1. [CrossRef]
6. Afolabi, I.; Taleb, T.; Samdanis, K.; Ksentini, A.; Flinck, H. Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2429–2453. [CrossRef]
7. Su, R.; Zhang, D.; Venkatesan, R.; Gong, Z.; Li, C.; Ding, F.; Jiang, F.; Zhu, Z. Resource Allocation for Network Slicing in 5G Telecommunication Networks: A Survey of Principles and Models. *IEEE Netw.* **2019**, *33*, 172–179. [CrossRef]
8. Debbabi, F.; Jmal, R.; Fourati, L.C.; Ksentini, A. Algorithmics and Modeling Aspects of Network Slicing in 5G and Beyonds Network: Survey. *IEEE Access* **2020**, *8*, 162748–162762. [CrossRef]
9. Wang, X.; Mao, S.; Gong, M.X. An Overview of 3GPP Cellular Vehicle-To-Everything Standards. *GetMobile: Mob. Comput. Commun.* **2017**, *21*, 19–25. [CrossRef]
10. Mohammed, M.H.; Agrawal, D.; Dixit, P.; Bhattacharyya, B. Deep Learning for Network Slicing and Self-Healing in 5G Systems. In Proceedings of the 2021 IEEE International Conference on Signal Processing, Information, Communication & Systems (SPICSCON), Dhaka, Banglades, 3–4 December 2021.
11. Liu, Q.; Choi, N.; Han, T. Deep Reinforcement Learning for End-To-End Network Slicing: Challenges and Solutions. *IEEE Netw.* **2022**, *37*, 222–228. [CrossRef]

12. Sarra, B.; Saleh, M.; Saad, H. Ensuring QoS and Efficiency of Vehicular Networks by SDVN-IoV. In Proceedings of the 2020 International Conference on Advanced Aspects of Software Engineering (ICAASE), Constantine, Algeria, 28–30 November 2022.
13. Wu, Q.; Shi, S.; Wan, Z.; Fan, Q.; Fan, P.; Zhang, C. Towards V2I Age-Aware Fairness Access: A DQN Based Intelligent Vehicular Node Training and Test Method. *Chin. J. Electron.* **2023**, *32*, 1230–1244.
14. Wu, Z.; Lu, Z.; Hung, P.C.K.; Huang, S.-C.; Tong, Y.; Wang, Z. QaMeC: A QoS-Driven IoVs Application Optimizing Deployment Scheme in Multimedia Edge Clouds. *Future Gener. Comput. Syst.* **2019**, *92*, 17–28. [CrossRef]
15. Aldhyani, T.H.H.; Alrasheedi, M.; Alqarni, A.A.; Alzahrani, M.Y.; Bamhdi, A.M. Intelligent Hybrid Model to Enhance Time Series Models for Predicting Network Traffic. *IEEE Access* **2020**, *8*, 130431–130451. [CrossRef]
16. Hu, Y.; Zhu, L.; Zhang, J.; Cai, Z.; Han, J. Migration and Energy Aware Network Traffic Prediction Method Based on LSTM in NFV Environment. *KSII Trans. Internet Inf. Syst.* **2023**, *17*, 896–915.
17. Tam, P.; Kang, S.; Ros, S.; Kim, S. Enhancing QoS with LSTM-Based Prediction for Congestion-Aware Aggregation Scheduling in Edge Federated Learning. *Electronics* **2023**, *12*, 3615. [CrossRef]
18. Park, J.; Son, J.G.; Kim, D. Resource Metric Refining Module for AIOps Learning Data in Kubernetes Microservice. *KSII Trans. Internet Inf. Syst.* **2023**, *17*, 1545–1559.
19. Song, I.; Tam, P.; Kang, S.; Ros, S.; Kim, S. DRL-Based Backbone SDN Control Methods in UAV-Assisted Networks for Computational Resource Efficiency. *Electronics* **2023**, *12*, 2984. [CrossRef]
20. *ONF White Paper*; Software-Defined Networking: The New Norm for Networks; Open Networking Foundation: Palo Alto, CA, USA, 2012.
21. *ETSI GS NFV-INF 001*; Network Functions Virtualisation (NFV); Infrastructure Overview. ETSI: Sophia Antipolis, France, 2015.
22. Sabella, D.; Vaillant, A.; Kuure, P.; Rauschenbach, U.; Giust, F. Mobile-Edge Computing Architecture: The Role of MEC in the Internet of Things. *IEEE Consum. Electron. Mag.* **2016**, *5*, 84–91. [CrossRef]
23. Chowdhury, N.M.M.K.; Boutaba, R. A Survey of Network Virtualization. *Comput. Netw.* **2010**, *54*, 862–876. [CrossRef]
24. Ren, Y.; Guo, A.; Song, C. Multi-Slice Joint Task Offloading and Resource Allocation Scheme for Massive MIMO Enabled Network. *KSII Trans. Internet Inf. Syst.* **2023**, *17*, 794–815.
25. Song, I. A Study on 5G Service Methods by Using BOCR Model and ANP. *KSII Trans. Internet Inf. Syst.* **2022**, *16*, 3299–3312.
26. Li, X.; Rao, J.; Zhang, H.; Callard, A. Network Slicing with Elastic SFC. In Proceedings of the 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), Toronto, ON, Canada, 24–27 September 2017.
27. Ros, S.; Eang, C.; Tam, P.; Kim, S. ML/SDN-Based MEC Resource Management for QoS Assurances. In Proceedings of the Advances in Computer Science and Ubiquitous Computing (CUTECSA 2022), Vientiane, Laos, 19–21 December 2022; Park, J.S., Yang, L.T., Pan, Y., Park, J.H., Eds.; Lecture Notes in Electrical Engineering. Springer: Singapore, 2023; Volume 1028.
28. *ETSI GS NFV-EVE 005*; Network Function Virtualisation (NFV); Ecosystem; Report on SDN Usage in NFV Architectural Framework. ETSI: Sophia Antipolis, France, 2015.
29. Wu, W.; Zhou, C.; Li, M.; Wu, H.; Zhou, H.; Zhang, N.; Shen, X.S.; Zhuang, W. AI-Native Network Slicing for 6G Networks. *IEEE Wirel. Commun.* **2022**, *29*, 96–103. [CrossRef]
30. Togou, M.A.; Bi, T.; Dev, K.; McDonnell, K.; Milenovic, A.; Tewari, H.; Muntean, G.-M. DBNS: A Distributed Blockchain-Enabled Network Slicing Framework for 5G Networks. *IEEE Commun. Mag.* **2020**, *58*, 90–96. [CrossRef]
31. Messaoud, S.; Bradai, A.; Ben Ahmed, O.; Quang, P.; Atri, M.; Hossain, M.S. Deep Federated Q-Learning-Based Network Slicing for Industrial IoT. *IEEE Trans. Ind. Inform.* **2020**, *17*, 1. [CrossRef]
32. Li, M.; Gao, J.; Zhou, C.; Shen, X.S.; Zhuang, W. Slicing-Based Artificial Intelligence Service Provisioning on the Network Edge: Balancing AI Service Performance and Resource Consumption of Data Management. *IEEE Veh. Technol. Mag.* **2021**, *16*, 16–26. [CrossRef]
33. Zhou, W.; Islam, A.; Chang, K. Real-Time RL-Based 5G Network Slicing Design and Traffic Model Distribution: Implementation for V2X and EMBB Services. *KSII Trans. Internet Inf. Syst.* **2023**, *17*, 2573–2589.
34. Phyu, H.; Naboulsi, D.; Stanica, R. Machine Learning in Network Slicing–A Survey. *IEEE Access* **2023**, *11*, 39123–39153. [CrossRef]
35. Liang, T.; Chen, M.; Yin, Y.; Zhou, L.; Ying, H. Recurrent Neural Network Based Collaborative Filtering for QoS Prediction in IoV. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 2400–2410. [CrossRef]
36. Tam, P.; Song, I.; Kang, S.; Ros, S.; Kim, S. Graph Neural Networks for Intelligent Modelling in Network Management and Orchestration: A Survey on Communications. *Electronics* **2022**, *11*, 3371. [CrossRef]
37. Hurtado Sánchez, J.A.; Casilimas, K.; Caicedo Rendon, O.M. Deep Reinforcement Learning for Resource Management on Network Slicing: A Survey. *Sensors* **2022**, *22*, 3031. [CrossRef]
38. Cui, Y.; He, S.; Wang, R.; He, P.; Wang, D.; Huang, X. Multi-Agent Reinforcement Learning for Slicing Resource Allocation in Vehicular Networks. *IEEE Trans. Intell. Transp. Syst.* **2023**, 1–12. [CrossRef]
39. Kamal, M.A.; Raza, H.W.; Alam, M.M.; Su'ud, M.M.; Sajak, A. binti A.B. Resource Allocation Schemes for 5G Network: A Systematic Review. *Sensors* **2021**, *21*, 6588. [CrossRef] [PubMed]
40. Pratik, T.; Anurag, T.; Kulkarni, A. GREENSKY: A Fair Energy-Aware Optimization Model for UAVs in Next-Generation Wireless Networks. *Green Energy Intell. Transp.* **2023**, *3*, 100130.
41. Shen, H.; Wang, T.; Heng, Y.; Bai, G. Joint Resource Slicing and Vehicle Association for Drone-Assisted Vehicular Networks. *Drones* **2023**, *7*, 534. [CrossRef]

42. Mlika, Z.; Cherkaoui, S. Network Slicing with MEC and Deep Reinforcement Learning for the Internet of Vehicles. *IEEE Netw.* **2021**, *35*, 132–138. [CrossRef]
43. Wu, Y.; Dai, H.-N.; Wang, H.; Xiong, Z.; Guo, S. A Survey of Intelligent Network Slicing Management for Industrial IoT: Integrated Approaches for Smart Transportation, Smart Energy, and Smart Factory. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 1175–1211. [CrossRef]
44. Li, Y.; Zhang, G.; Ren, C. Network Slicing for IoV: QoS-Oriented Radio Resource Optimization. In Proceedings of the 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 16–26 October 2022.
45. Khan, H.; Luoto, P.; Samarakoon, S.; Bennis, M.; Matti, L. Network Slicing for Vehicular Communication. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e3652. [CrossRef]
46. Smida, K.; Tounsi, H.; Frikha, M.; Song, Y.-Q. FENS: Fog-Enabled Network Slicing in SDN/NFV-Based IoV. *Wirel. Pers. Commun.* **2022**, *128*, 2175–2202. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.