This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3288637

Slice Sandwich: Jagged Slicing Multi-tier Dynamic Resources for Diversified V2X Services

Yu Liu, Zirui Zhuang, Qi Qi, *Senior Member, IEEE*, Jingyu Wang, *Senior Member, IEEE*, Dezhi Chen, Lu Lu, Hongwei Yang, Jianxin Liao, and Zhu Han, *Fellow, IEEE*

Abstract—With the advancement of intelligent transportation systems, a series of diversified V2X applications come into being, which have different key performance indicators (KPIs) and transmission features. Moreover, multi-tier computing as a new system-level architecture distributes computing and communication capabilities anywhere between the cloud and the end-user. Unfortunately, the existing network paradigm for V2X services adopts a one-shot allocation of resources ignoring the inherent differences of V2X service. To cope with these problems, three types of refined network slices for V2X services are first proposed to simultaneously support heterogeneous service characteristics without excessively splitting resources. Considering the spatiotemporal correlation between service traffic and physical resources, a jagged slicing in multi-tier dynamic resources, which forms a “slice sandwich” brightly, is realized by a dual timescale intelligent resource management scheme. The inter-slice resource configuration is based on neural bandits with upper confidence bounds at each large-time period, while the exclusive resources are managed elastically by deep Q-learning in terms of the real-time changing network state in the small slot. We developed a simulation environment by Simulation of Urban Mobility (SUMO) including real-world road conditions and traffic models. The experiment results demonstrate that the proposed scheme can effectively guarantee KPIs of V2X services and improve the system revenue compared with benchmark algorithms.

Index Terms—V2X services, network slicing, multi-tier computing, resource allocation, multi-armed bandits, deep Q-learning.

1 Introduction

With the increase in population and the development of urbanization, the transportation system is facing unprecedented pressure [1]. Many V2X (vehicle-to-everything) services have emerged to adapt to complex traffic situations and offer enjoyable driving experiences. Up to now, the Third Generation Partnership Project (3GPP) has defined 57 use cases of V2X [2], [3], containing V2V (vehicle to vehicle) services, V2P (vehicle to pedestrian) services, V2I (vehicle to infrastructure) services, and V2N (vehicle to network) services. Different from conventional services for stationary or low-speed equipment, V2X services own exclusive transmission features and key performance indicators (KPIs). To reflect how various V2X services influence the performance of the internet of vehicles (IoV), representative use cases are detailedly summarized in Table 1. It is not difficult to see that there are extremely diversified and even conflicting service characteristics among use cases, which poses critical pressure on the networking infrastructure [4].

Network slicing has emerged as a promising paradigm to meet diverse service demands. It enables multiple independent logical networks (i.e., slices) to run on a common physical network infrastructure [5], [9]. However, as V2X applications advance, the predefined slice for ultra-reliable low-latency communications (URLLC) can hardly meet more and more stringent and heterogeneous service characteristics by one-shot resource allocation [6], [7], [8]. However, as V2X applications advance, the predefined slice for ultra-reliable low-latency communications (URLLC) can hardly meet more and more stringent and heterogeneous service characteristics by one-shot resource allocation [6], [7], [8]. Taking the advancements in existing studies, three types of slices are proposed to accommodate existing and future V2X use cases without excessively segmenting network resources. Specifically, the slices for *basic road safety services*, *enhanced road safety services*, and *non-safety related services* are used to deliver basic driving information, achieve high-level automatic driving, and improve driving comfort and efficiency, respectively. The illustration of representative use cases and their corresponding slices are depicted in Fig. 1.

Unfortunately, constrained by computing capability or transmission delay, it is difficult to process multiple tasks by a single paradigm [11], [12], [13]. Multi-tier computing as a new system-level computing architecture provides a new resolution for the problem. It involves three tiers with the users at tier one, edge cloud at tier two, and remote cloud at tier three [14]. By reasonably orchestrating available resources along its continuum, the strict KPIs of each slice are expected to be met. However, exploiting this hierarchical computing architecture for service provisioning entails joint allocation of multi-dimensional resources [15], [16]. Besides, the high mobility of vehicles introduces more complexity to resource management [17], [18], [19]. How to effectively allocate multi-tier resources to multiple slices according to time-varying network conditions is a thorny problem.

To cope with the problem, existing studies usually adopt hierarchical resource allocation methods [20], [21], [22], [23], [24]. Although these studies obtained certain results in improving resource utilization, they are not applicable to the IoV. That is because they ignored the exclusive characteristics of V2X services and the importance of multi-tier dynamic resources. Thus, considering the spatiotemporal correlation between service traffic and physical resources [25], a Two-Time-scale Resource Management Scheme (2Ts-RMS) is proposed. Specifically, the scheme is divided into two stages, namely inter-slice resource configuration and intra-slice resource scheduling. At the beginning of each

TABLE 1
Transmission features and key performance indicators of typical V2X service use cases.

Slices for V2X services	Use Case	Payload (Bytes)	Frequency (Hz)	Max End-to-End latency (ms)	Reliability (%)
Basic Road Safety	Forward Collision Warning	50-300	10	100	90
	Emergency Stop	400	10	100	90
Enhanced Road Safety	Vehicle Platooning	50-6000	2-50	10-25	90-99.99
	Extended Sensors	1600	N/A	3-100	90-99.999
Non safety	Traffic Flow Optimisation	50-300	1-10	1000	N/A
	Software Update	N/A	N/A	300-65535	N/A

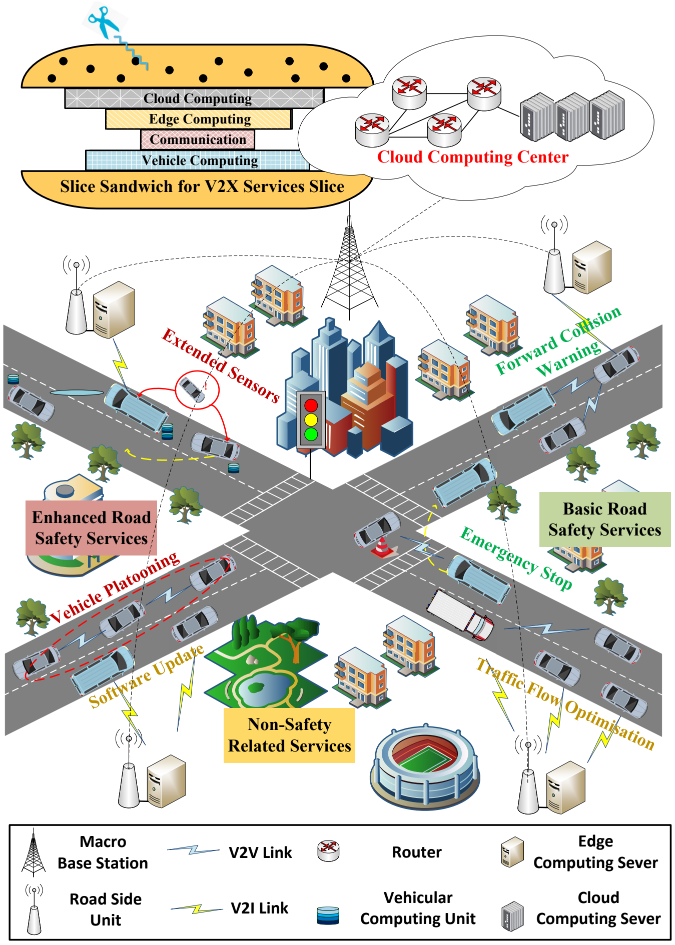


Fig. 1. Illustration of typical V2X applications in vehicular networks. The basic road safety services slice provides position, heading, speed, etc; the enhanced road safety services slice provides raw sensor data, vehicles intention data, coordination, confirmation of future maneuvers, and so on; the non-safety related services slice provides traffic flow optimization and software updates. An exclusive "slice sandwich" for each V2X services slice is made up of jagged multi-dimensional resources.

large timescale (i.e., period), the infrastructure provider (InP) configures resources for service providers (SPs) according to service traffic. Due to the long-term trend of the service traffic, the configuration policy remains unchanged within each large timescale. The SPs create customized slices with obtained multi-dimensional resources. Because the inherent characteristics of slices make its demand for multi-dimensional resources appear jagged, the shape of a "slice sandwich" is naturally formed. Then, to adapt the real-time status of the physical layer, each SP will dynamically schedule available resources at each small timescale (i.e., slot) of a large timescale to provide high-quality services for its subscribers. In this way, system revenue could be maximized while guaranteeing the delay and reliability requirements of mobile users.

It is noted that the resource configuration made in the InP will influence the scheduling process in SPs; meanwhile, the performance of SPs will also affect the decision-making of the InP. The interaction between InP and SPs makes it very challenging to implement conventional mathematical methods to solve the proposed problem. Deep Reinforcement Learning (DRL) as intelligent approaches provide promising solutions to the challenge. In the stage of inter-slice resource configuration, since the status of service requests is only up to the users, it does not change by the selected policy of resource configuration. Thereupon, a Joint Allocation algorithm of Multi-dimensional Resources (JAMR) based on the improved NeuralUCB (Neural bandits with Upper Confidence Bounds) approach is proposed. The algorithm can effectively avoid the curse of dimensionality and learns the unknown system revenue. As for the intra-slice resource scheduling problem, state transitions need to be considered, because the scheduling policies of resource allocation and task offloading will generate different effects on the physical layer states. To adapt to the time-varying physical layer, a Joint Offloading and Resource Allocation algorithm based on the Double Deep Q Network (JORA-DDQN) approach is proposed to obtain optimal scheduling policies. The major contributions of this paper are summarized as follows.

- Three types of refined network slices for V2X services are proposed to simultaneously accommodate multiple V2X services over a common infrastructure.
- In view of differentiated KPIs among slices, a dual Timescale Intelligent Resource Management Scheme (2Ts-IRMS) is proposed to jaggedly divide multi-tier resources into multiple slices in time-varying IoV.
- In order to fit reality, real world road conditions and traffic models are set up in Simulation of Urban Mobility (SUMO). Numerical experiments using Pytorch verify that the proposed scheme can more economically and efficiently utilize network resources.

The rest of this paper is organized as follows. Section 2 presents an overview of the related works. In Section 3, we describe the considered system framework. Section 4 presents the two timescale resource allocation problem. In Section 5, the solutions based on JAMR and JORA-DDQN are proposed. Section 6 evaluates the network performance and compares its performance with some benchmarks. Finally, Section 7 concludes the paper.

2 RELATED WORK

2.1 Network Slicing for V2X services

Up to now, the 3GPP has defined standardized slices to support enhanced mobile broadband (eMBB), URLLC, and massive machine type communication (mMTC) [29], [30]. With the evolution of V2X services, more and more rigorous and heterogeneous KPIs need to be satisfied. Mapping V2X services into existing reference slices or a single V2X slice is no longer appropriate [32], [33]. Network slicing for concrete application scenarios is still emerging, especially for vehicular scenarios [34]. In [35], the authors customized slices for safety and non-safety V2X services, respectively. According to the sensitivity of V2X services to delay, Wu $et\ al.$ proposed delay-sensitive and delay-tolerant slices [36]. As described in Table 1, there are great differences between basic road safety services and enhanced road safety services. One slice for safety or delay-sensitive V2X services is still insufficient to simultaneously cope with the differences.

For dealing with this problem, Campolo $et\ al.$ designed four slices for autonomous driving, tele-operated driving, remote diagnostic, and vehicular infotainment [32]. Similarly, the authors proposed a general network slicing architecture for four typical use cases, namely localization and navigation, transportation safety, autonomous driving, and infotainment services [34]. A common problem of the aforementioned studies is that the validity of the proposed schemes did not be verified. The complexity of slicing management increases with the number of slices. Dividing V2X services into three slices is a more reasonable solution, which is similar to the slicing way for traditional mobile services. In [31], Ge $et\ al.$ proposed three types of service slices, which are used to transmit state-report, event-driven, and entertainment-application messages, respectively. In [39], Cui $et\ al.$ divided the common network infrastructure into three slices to provide short message service, call service, and internet service for vehicles. Different from the existing studies, the proposed slices in this paper fully consider the exclusive characteristics (i.e., transmission features and KPIs) of V2X services. They can cover all V2X use cases defined in [2], [3] without excessively segmenting resources.

2.2 Resource Allocation for Network Slicing

In addition to slicing services with benign granularity, it is important to effectively allocate resources among slices. In [42], the authors developed a fuzzy logic-based resource allocation algorithm to simultaneously satisfy the diversified requirements of V2X services. Although the scheme achieved higher resource utilization, its computation complexity is high as the InP directly allocates its resources to users. Most of the existing studies tend to adopt hierarchical resource allocation methods to reduce the burden of the InP. In [6], Han $et\ al.$ proposed a two-dimension-time-scale resource allocation scheme including inter-slice resource pre-allocation in large time periods and intra-slice resource scheduling in small time slots. The scheme achieves a near-optimal tradeoff among the performance of slices. In [20], Mei $et\ al.$ designed a slicing strategy with two-layer control granularity. The upper-level and lower-level controllers are used to guarantee the quality of services and improve the spectrum efficiency of each slice, respectively. However, these efforts only concentrate on spectrum resource allocation. The significance of computing resources is ignored, which are necessities to satisfy the KPIs of V2X services.

To address the multi-dimensional resources allocation issue, Mohammed $et\ al.$ proposed a multi-dimensional resources slicing scheme [50]. Both the InP and SPs adopt the dominant resource fairness (DRF) approach to allocate multi-dimensional resources. In [23], the authors introduced a generalized Kelly mechanism (GKM) to address the multi-dimensional resource allocation issue between the InP and SPs. Meanwhile, each SP utilizes Karush–Kuhn–Tucker (KKT) conditions to derive the optimal scheduling strategy of communication resources. Although these studies make progress in improving the aggregate revenue of SPs, they cannot be directly applied to the IoV with multiple V2X services. On the one hand, when the InP equally treats all slices, it is hard to guarantee road safety in real-world situations. On the other hand, the differentiated characteristics of multi-tier computing resources have not been extensively explored, which will further cut down the system revenue. In our work, we adopt intelligent approaches to economically allocate multi-tier resources to multiple V2X slices while guaranteeing the delay and reliability requirements of mobile users.

2.3 DRL-Enabled Network Slicing

In the dynamic IoV, conventional mathematical models face with high computation complexity and lack adaptability and robustness. Advanced DRL algorithms have been widely applied in network slicing [38]. From the perspective of the effect of the action on the status, DRL can be divided into DRL based on multi-armed bandits (MAB) and DRL based on Markov Decision Process (MDP) [26], [27]. Because the policies of resource allocation and task offloading will generate different effects on the physical layer states, the intra-slice resource scheduling problem is usually formulated as an MDP problem. In [22], Chen $et\ al.$ leveraged the DDQN algorithm to learn the optimal policies of packet scheduling and computation offloading. In [20], the authors further verified the effectiveness of DDQN in jointly optimizing resource allocation and computation offloading. In this paper, considering that the numbers of vehicles and available resources of slices are different, each SP equipped with an exclusive agent implements resource scheduling to guarantee the isolation among slices.

As for the inter-slice resources configuration problem, it is impossible to find the optimal configuration policy before the end of a period. That is because the future status of the IoV is unknowable. In addition, it is impractical to traverse all configuration policies at each period. Taking advantage of the characteristic that the policy of resource configuration will not change the status of service requests, many studies adopt DRL algorithms based on MAB to learn the unknown reward function. In [44], Zanzi $et\ al.$ developed a radio slicing orchestration scheme based on MAB. With no prior knowledge of channel quality statistics, SPs can make adaptive slicing decisions. In [45], Zhao $et\ al.$ formulated resource configuration as a contextual MAB problem and adopted the upper-confidence-bound (UCB) algorithm to solve it. However, these studies assumed a linear relationship between the expected reward

TABLE 2
MAJOR NOTATIONS USED IN THIS PAPER

Notation	Definition
M	Number of VUEs
B	Bandwidth of a RB
Z_l	Packet size of link l
$\mathcal{T}/\mathcal{K}$	Set of slots and periods
$\mathcal{J}/J$	Set/number of RBs
$Y_{m,i}^{v}$	Number of CPU cores of VUE N_m allocated to slice i
Y_i^u	Number of CPU cores of MEC server allocated to slice i
$W_{l,t}$	Queue length of link l at slot t
$\rho_{l,j,t}$	Allocation action of j-th RB for link l at slot t
$e_{l,t}$	Offloading action for link l at slot t
$U_{l,t}$	Service satisfaction of link l at slot t
ψ_i	Penalty factor for slice i
$\varphi_{l,t}$	Reliability of link l at slot t
q_i^{se}	Unit price charging by slice i
q^{cm}	Price of unit communication resource
q^{cp}	Price of unit edge computing resource
q^{cc}	Price to access the cloud computing center
$D_l^{\max}$	Maximum tolerant delay and minimum PRR of link l
$\varphi_l^{\min}$	Minimum PRR of link l
$\mathcal{C}$	Set of multi-dimensional resources configuration for all slices
$c_{i,k}$	Resources configuration for slice i at period k
O_{k,C_k}	Feature vector for resources configuration C_k at period k
$V(C_k)$	Total system revenue corresponding to C_k
P_{k,C_k}	Upper confidence bound of C_k at period k
$s_{l,t},\ a_{l,t}$	State and action space of link l at slot t
$Q^{\pi}(s,a)$	State-action value function under the policy π
$N_l,\ N_{l'}$	Transmitter and receiver of link l
α_d, α_r	Weighting factors for delay and reliability
$r_{l,t}^{v},\ r_{l,t}^{u}$	Available V2V and V2I transmission rate for link l at slot t
$\mathcal{L}_i/L_i$	Set/number of links subscribed to slice i
$\mathcal{J}_i/J_i$	Set/number of RBs allocated to slice i
$Y^u,\ Y_m^v$	Number of CPU cores of MEC server and VUE N_m
$\mathcal{N},\ \mathcal{I},\ \mathcal{L}$	Set of UEs, slices, links
$f_v,\ f_u,\ f_c$	CPU frequency of VUE, MEC server, and cloud server

and the context vector. Furthermore, the effectiveness of DRL algorithms based on MAB will be greatly reduced when the number of candidate actions is large. The curse of dimensionality is inevitable when we jointly consider multi-dimensional resources. Therefore, in this paper, we design a pre-allocation mechanism based on service priories and adopt the NeuralUCB algorithm to obtain an optimal configuration policy of multi-dimensional resources.

3 SYSTEM MODEL

This section describes the system model in detail. Specifically, we first present the network model (Section 3.1) and multi-tier resources model (Section 3.2) of the IoV. Then, we elaborate on the process of transmission (Section 3-3.3) and offloading (Section 3.4) for vehicular tasks. Finally, key performance indicators of V2X services will be presented (Section 3.5). For convenience, Table 2 summarizes the major notations of this paper.

3.1 Network Model

The physical infrastructure of the IoV mainly includes a macro base station (MBS) connected to remote cloud servers, roadside units (RSUs) equipped with MEC servers, and vehicular user equipment (VUEs) with diverse numbers of vehicular computing units. Note that an RSU essentially is a statically logical entity. It supports V2X applications by using the functionality provided by a 3GPP network or user equipment (UE) [47]. Thus, we assume all UEs, which consist of VUEs and RSUs, are within the coverage of the MBS and VUEs can only access the internet via RSUs. Let $\mathcal{N}=\{N_0,N_1,...,N_m,...,N_M\}$ be the set of UEs covered by RSU N_0. Furthermore, we assume that VUE $N_m\in\mathcal{N}(m\neq 0)$ is equipped with Y_m^v central processing unit (CPU) cores. As for RSU N_0, there are Y^u CPU cores deployed at its MEC server. It means that the MEC server can serve at most Y^u VUEs at the same time. f_v and f_u represent the CPU frequency of each CPU core of the VUE and MEC server, respectively.

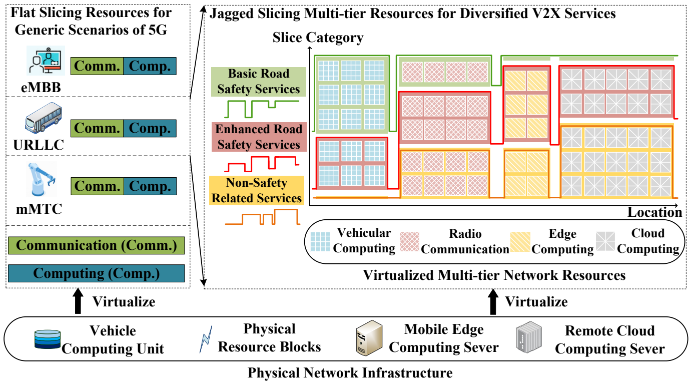

Flat Slicing Resources for Generic Scenarios of 5G
eMBB
URLLC
mMTC
Comm.
Comp.
Communication (Comm.)
Computing (Comp.)
Jagged Slicing Multi-tier Resources for Diversified V2X Services
Slice Category
Basic Road Safety Services
Enhanced Road Safety Services
Non-Safety Related Services
Location
Vehicular Computing
Radio Communication
Edge Computing
Cloud Computing
Virtualized Multi-tier Network Resources
Virtualize
Virtualize
Vehicle Computing Unit
Physical Resource Blocks
Mobile Edge Computing Sever
Remote Cloud Computing Sever
Physical Network Infrastructure

Fig. 2. Illustration of the jagged allocation of virtualized multi-tier resources to refined network slices of V2X services, compared to flat slicing resources to three generic usage scenarios of 5G.

As mentioned in Table 1, there are great differences among V2X services. Therefore, we propose three kinds of network slices to reflect the differences without excessively segmenting resources. Specifically, the three kinds of network slices embrace the slice for basic road safety services, the slice for enhanced road safety services, and the slice for non-safety related services. The specific characteristics and requirements of each slice are described as follows.

- The slice for basic road safety services is mainly aimed at the services that require high timeliness and reliability but low data rates, such as collision warnings and emergency stops. V2V is the prevalent radio access technology to satisfy the requirements of latency and reliability. Note that the packet size of basic safety services is usually small. Thus, instead of offloading tasks to MEC servers, vehicular computing resources are sufficient to process them.
- The slice for enhanced road safety services aims to enable high-level autopilot. Compared to basic road safety services, the slice requires higher reliability, data rate, beacon frequency, and lower latency. Similarly, to effectively transmit messages among vehicles, low-latency V2V communication is the main communication mode. Due to the limited processing capability of VUEs and the long transmission latency of remote cloud servers, a proportion of data processing should be performed in MEC servers.
- The slice for non-safety related services has a low sensibility to delay and reliability, but usually has high requirements of data rate. As a result, it is expected to use multiple access technologies to seek higher throughput and to process tasks in MEC servers or cloud servers.

$$r^{N_{m'}}_{N_m,j,t} = \begin{cases} B\log_2\left(1+\dfrac{p^{N_{m'}}_{N_m}\cdot h^{N_{m'}}_{N_m,j,t}}{\sigma^2}\right), & \text{for the long packet transmission;} \qquad (1a) \\ B\log_2\left(1+\dfrac{p^{N_{m'}}_{N_m}\cdot h^{N_{m'}}_{N_m,j,t}}{\sigma^2}\right)-\sqrt{\dfrac{V^{N_{m'}}_{N_m,j,t}}{\tau_{N_m}}}\cdot\dfrac{G^{-1}(\varpi)}{\ln 2}, & \text{for the short packet transmission,} \qquad (1b) \end{cases}$$

In this paper, an SP corresponds to a slice and provides a class of V2X services. Therefore, we will not distinguish the concepts of slice and SP in the following text. To facilitate analysis, let $\mathcal{L}_i$ be the set of V2V links subscribed to slice $i \in \mathcal{I}$ with $|\mathcal{I}| = 3$. Then, $\mathcal{L} = \cup_{i\in\mathcal{I}}\mathcal{L}_i$ denotes the set of all V2V links across the whole network. Each V2V link $l \in \mathcal{L}$ is composed of a transmitter (VTx) $N_l \in \mathcal{N}$ and a receiver (VRx) $N_{l'} \in \mathcal{N}$.

3.2 Multi-tier Resources Model

As described above, each SP simultaneously needs computing resources and communication resources to service its users. The inherent attributes (i.e., KPIs and transmission features) of V2X services make their demand for multi-dimensional resources appear jagged. Thus, the jagged resource slicing on the multi-tier computing architecture is adopted in this paper. Generally, the architecture tends to use three tiers with users at tier one, edge cloud at tier two, and remote cloud services at tier three. Before determining the most suitable communication method and computing location for any service, the hierarchical and distributed characteristics of multi-dimensional resources should be considered. In the tier of terminal devices, vehicular computing resources usually have relatively small computing capabilities. The purpose of local execution is to reduce communication delay and errors caused by transmission and protocols. Significantly, a VUE can concurrently subscribe to multiple slices in our system model, which is consistent with actual cases. Therefore, let $Y^v_{m,i}$ be the number of vehicular CPU cores allocated to slice i by VUE N_m $(m \neq 0)$.

As for the edge tier, MEC servers have powerful computing capabilities. However, the computing resources of each MEC server are limited. It means that only a part of VUEs can offload their computing tasks to MEC servers by V2I links. To guarantee isolation among slices, the shared edge computing resources Y^u (i.e., CPU cores) and the set of shared wireless communication resources $\mathcal{J}$ with $|\mathcal{J}| = J$ (i.e., physical resource blocks with bandwidth B) are orthogonally divided into three parts. Let $\mathcal{J}_i$ with $|\mathcal{J}_i| = J_i$ be the set of the total wireless communication resources allocated to slice i. Y^u_i is the number of CPU cores of the MEC server allocated to slice i. The cloud tier consists of a large number of remote cloud servers, which has sufficient computing resources. Furthermore, the RSUs are connected to the MBS and cloud computing center via high-speed fronthaul links. Thus, when the VUEs decide to offload computing tasks to remote cloud servers, it is reasonable to ignore the constraint of the number of communication and computing resources. To reflect the usage of cloud computing resources, let Y^c_i be the number of VUEs offloading computing tasks to remote cloud servers. Fig. 2 depicts a diagram of the jagged allocation of virtualized resources to multiple V2X slices, where each slice consists of multi-dimensional resources to form a ”slice sandwich”.

3.3 Signal Transmission Model

In conventional services, the data rate of large-sized packets can be directly calculated through the Shannon formula. However, unlike conventional services, the packet size of most V2X services is short, which ranges from 32 to 200 bytes [20]. Since the negative effect of channel dispersion and coding length, the data rate of a short packet cannot be accurately obtained by the Shannon formula. In [48], based on finite block-length theory, a new method used to approximately calculate the data rate of short packets is proposed. Therefore, the available data rate between VTx $N_m \in \mathcal{N}$ and VRx $N_{m'} \in \mathcal{N}$ on the resource block $j \in \mathcal{J}$ at slot t can be calculated as formula (1a) or formula (1b), where σ^2 is the power of additive white Gaussian noise on each resource block (RB). $p^{N_{m'}}_{N_m}$ is the transmission power when N_m communicates to $N_{m'}$. $h^{N_{m'}}_{N_m,j,t}$ denotes the channel coefficient on RB j at slot t, which contains path loss, Rayleigh fading and shadowing effect. As for the short packet transmission in formula (1b), $V^{N_{m'}}_{N_m,j,t}$ is used to reflect the random variability of the channel. It is calculated as

$$V^{N_{m'}}_{N_m,j,t} = 1-\left(1+\frac{p^{N_{m'}}_{N_m}\cdot h^{N_{m'}}_{N_m,j,t}}{\sigma^2}\right)^{-2}. \tag{2}$$

$G^{-1}(\cdot)$ and ϖ are the inverse of the Gaussian Q-function and the effective decoding error probability, respectively. τ_{N_m} is the number of transmit symbols. Both of them are used to reflect the influence of coding on short packet transmission.

In addition, during the phase of data transmission, each V2V link maintains an individual queue to buffer the arriving packets. The packet is delivered according to the first-come-first-serve policy [22]. As for link l, the dynamic evolution of its queue can be written as

$$W_{l,t+1} = \min\{W_{l,t} - \Delta t\cdot r^v_{l,t}/Z_l + A_{l,t}, W^{\max}_l\}, \tag{3}$$

where $W_{l,t}$ is the queue length (i.e., number of packets) at slot t. $A_{l,t}$ denotes the instantaneous packet arrival. $W^{\max}_l$ is the maximum length of the buffer queue, and Z_l is the total packet size (in bits). Δt refers to the duration of each slot. During the duration, the quality of wireless channels keeps stable. As for $r^v_{l,t}$, it depicts the total rate capacity from VTx N_l to VRx $N_{l'}$ at slot t, which can be expressed as

$$r^v_{l,t} = \sum_{j\in\mathcal{J}} \rho_{l,j,t}\cdot r^{N_{l'}}_{N_l,j,t}. \tag{4}$$

Term $\rho_{l,j,t}$ is a binary variable. $\rho_{l,j,t} = 1$ denotes the j-th RB is allocated to link l at slot t, otherwise $\rho_{l,j,t} = 0$.

3.4 Task Offloading Model

In this paper, we consider a hybrid computation offloading scenario [21]. The computing task of a vehicle can be executed locally. It can also select to be offloaded to the MEC server by V2I communication or the remote cloud computing servers through relayed V2I and high-speed fronthaul links. As for the computing task of link l at slot t, let $e_{l,t} \in \{0\ 1\ 2\}$ be its offloading action. Specifically, $e_{l,t} = 0$ represents local execution, $e_{l,t} = 1$ indicates offloading the computing task to the MEC server, and $e_{l,t} = 2$ means that the offloading position of the task is the remote cloud computing center. Considering the output size of the computing task is much smaller than the input size of the computing task, the download time of processed data is ignored [21], [43]. Thus, at slot t, the processing time for the b-th packet of link $l \in \mathcal{L}_i$ can be described as:

$$D^{cp}_{l,b,t} = \begin{cases} \dfrac{Z_l \cdot \beta_l}{Y^v_{m,i} \cdot f_v}, & e_{l,t} = 0; \quad \text{(5a)} \\ \dfrac{Z_l \cdot \beta_l}{f_u} + \dfrac{Z_l}{r^u_{l,t}}, & e_{l,t} = 1; \quad \text{(5b)} \\ \dfrac{Z_l \cdot \beta_l}{f_c} + \dfrac{Z_l}{r^u_{l,t}} + t_c, & e_{l,t} = 2, \quad \text{(5c)} \end{cases}$$

where β_l denotes that the input packet requires β_l cycles/bit for processing. Term f_c is the CPU frequency of each CPU core of the remote cloud server, and t_c is the network delay between RSU N_0 and the cloud computing center. It is worth noting that $r^u_{l,t}$ is the available transmission rate for link l to upload data to RSU N_0, which is denoted as

$$r^u_{l,t} = \sum_{j \in \mathcal{J}} \rho_{l,j,t} \cdot r^{N_0}_{N_l,j,t}. \tag{6}$$

3.5 Key Performance Indicators

As defined in [3], whole end-to-end (E2E) communication refers to the process that transfers a given piece of information from a source to a destination at the application level. Generally, the E2E delay consists of waiting time in the queue, transmission time, network latency, and processing latency [49]. In this paper, we have assumed that all VUEs are in the coverage of the RSUs and they can only grasp data from RSUs. Consequently, it is reasonable to ignore the network delay during the process of data receiving. Thus, we mainly consider waiting, transmission, and processing delays. At slot t, the E2E delay of the b-th packet of link l can be written as

$$D^{\text{E2E}}_{l,b,t} = D^{cw}_{l,b,t} + D^{ct}_{l,b,t} + D^{cp}_{l,b,t}, \tag{7}$$

where $D^{cw}_{l,b,t}$ denotes the queuing delay at VTx N_l, and $D^{ct}_{l,b,t}$ refers to the transmission time between VTx N_l and VRx $N_{l'}$. To reflect the delay state of link l at slot t, let $D_{l,t}$ be the average packet delay of queue $W_{l,t}$.

In addition to delay, reliability is another key performance indicator [10]. From the view of service provisioning, the probability of receiving or dropping data packets is usually used as a measure of reliability [46]. When the delay of a packet exceeds the maximum tolerant delay, we consider the packet as dropout, otherwise as receiving. In this paper, we choose the packet reception ratio (PRR) as the index to evaluate reliability, which can be expressed as

$$\varphi_{l,t} = \Pr\{D^{\text{E2E}}_{l,b,t} < D^{\max}_{l}\}, \tag{8}$$

where $D^{\max}_{l}$ is the maximum tolerant E2E delay of link l.

4 PROBLEM FORMULATION

In this paper, the resource allocation problem is decomposed into two stages. First, at the beginning of each large-time period k, the InP jaggedly allocates shared physical resources to SPs (Section 4.1). Then, at each small-time slot t, each SP independently manages acquired resources to provide services for its users (Section 4.2). Fig. 3 helps illuminate the process of two-time-scale resource allocation.

4.1 Large Timescale Problem Formulation

At the beginning of each period $k \in \mathcal{K}$, the multi-tier resources are jaggedly allocated to slices to maximize system revenue. The revenue consists of the fees charged by SPs and the fees paid for accessing resources. As for slice $i \in \mathcal{I}$ with resources configuration $c_{i,k} = \{c^v_{i,k}, c^u_{i,k}, c^c_{i,k}\}$, the fee charged by it is the value of the valuation function $v(c_{i,k})$. Term $c^v_{i,k} = \{Y^v_{m,i,k} | \forall m \in [1, M]\}$, $c^u_{i,k} = \{J_{i,k}, Y^u_{i,k}\}$, and $c^c_{i,k} = \{Y^c_{i,k}\}$ represent the resource configuration for the terminal tier, edge tier, and cloud tier, respectively. From the perspective of privacy and security, the computing resource of vehicles only can be used by themselves. Thus, it is reasonable to ignore the cost of using vehicular computing resources. Let q^{cc} be the price to access the cloud computing center for each user. q^{cm} and q^{cp} are treated as the price to utilize unit communication resources and unit edge computing resources, respectively. Therefore, corresponding to the set of multi-dimensional resources configuration for all slices $C_k = \{c_{i,k} | \forall i \in \mathcal{I}\} \in \mathcal{C}$, the total system revenue $V(C_k)$ is expressed as

$$V(C_k) = \sum_{i \in \mathcal{I}} \left[v(c_{i,k}) - q^{cm} J_{i,k} - q^{cu} Y^u_{i,k} - q^{cc} Y^c_{i,k} \right] \tag{9}$$

Based on the above analysis, the problem that the InP allocates multi-tier resources to SPs can be formulated as:

$$\begin{aligned}
&\text{P1} : \max_{C_k} \ [V(C_k)] \\
&\text{subject to :} \\
&\text{C1} : \sum_{i \in \mathcal{I}} Y^v_{m,i,k} \le Y^v_m; \ \forall m \in [1, M], k \in \mathcal{K}, \\
&\text{C2} : \sum_{i \in \mathcal{I}} J_{i,k} \le J; \ \forall k \in \mathcal{K}, \\
&\text{C3} : \sum_{i \in \mathcal{I}} Y^u_{i,k} \le Y^u; \ \forall k \in \mathcal{K}, \\
&\text{C4} : Y^v_{m,i,k} \in [0, Y^v_m]; \ \forall m \in [1, M], i \in \mathcal{I}, k \in \mathcal{K}, \\
&\text{C5} : J_{i,k} \in [0, J]; \ \forall i \in \mathcal{I}, k \in \mathcal{K}, \\
&\text{C6} : Y^u_{i,k} \in [0, Y^u]; \ \forall i \neq i' \in \mathcal{I}, k \in \mathcal{K}, \\
&\text{C7} : \mathcal{Y}^v_{m,i,k} \cap \mathcal{Y}^v_{m,i',k} = \varnothing; \ \forall m \in [1, M], i \neq i' \in \mathcal{I}, k \in \mathcal{K}, \\
&\text{C8} : \mathcal{J}_{i,k} \cap \mathcal{J}_{i',k} = \varnothing; \ \forall i \neq i' \in \mathcal{I}, k \in \mathcal{K}, \\
&\text{C9} : \mathcal{Y}^u_{i,k} \cap \mathcal{Y}^u_{i',k} = \varnothing; \ \forall i \neq i' \in \mathcal{I}, k \in \mathcal{K},
\end{aligned} \tag{10}$$

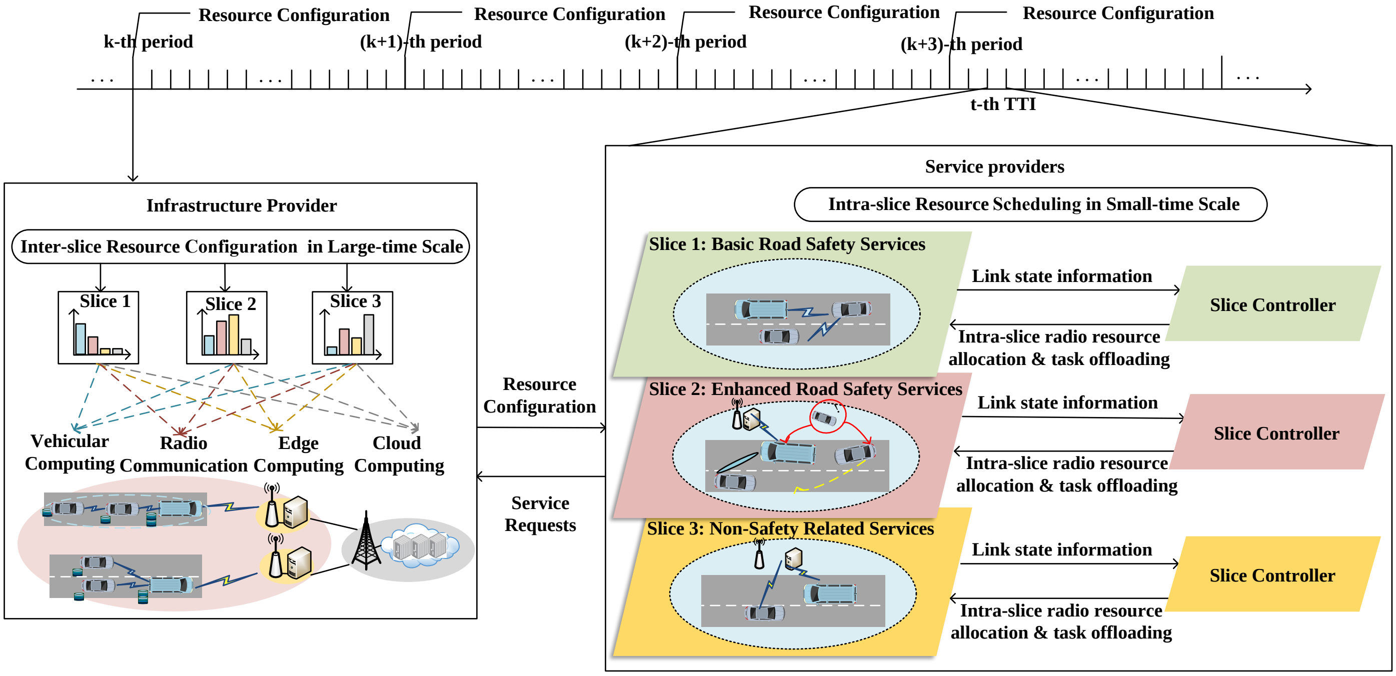


Fig. 3. Schematic of two-time scales resource allocation. First, at the beginning of each large-time period, the InP allocates shared physical resources to SPs (inter-slice resource configuration). Then, each SP elastically assigns exclusive resources to its users at each small slot (intra-slice resource scheduling).

where $\mathcal{Y}^u_{i,k}$ is the set of edge computing resources allocated to slice i at period k. Similarly, $\mathcal{Y}^v_{m,i,k}$ is the set of vehicular computing resources allocated to slice i from VUE N_m at period k. C1-C3 respectively guarantee that allocated resources do not exceed the capacity of vehicular computing resources Y^v_m, communication resources J, and edge computing resources Y^u. C4-C6 are the constraints to the value of $Y^v_{m,i,k}$, $J_{i,k}$, and $Y^u_{i,k}$, respectively. These constraints ensure that the number of resources allocated to each SP must be a non-negative integer. C7-C9 ensure isolation among all slices. It is noteworthy that the cloud tier contains a large number of cloud servers and the RSUs connect to the cloud computing center via high-speed fronthaul links. As a result, resource constraints of offloading tasks from the edge tier to the cloud tier can be ignored.

4.2 Small Timescale Problem Formulation

After determining the resource configuration for all slices, each slice utilizes acquired multi-dimensional resources to provide services for its subscribers to maximize the fee charged by it. As for slice $i \in \mathcal{I}$, q^{se}_i is the unit price to charge link l for realizing service satisfaction $U_{l,t}$, which consumes computing resources to process arriving tasks and communication resources to transmit queued packets. Since both delay and reliability are KPIs to weigh the quality of V2X services, the service satisfaction $U_{l,t}$ of link l at slot t can be described as:

$$U_{l,t} = \alpha_d \cdot U^{(1)}_l(D^{\text{E2E}}_{l,t}) + \alpha_r \cdot U^{(2)}_l(\varphi_{l,t}), \tag{11}$$

where α_d and α_r are weighting factors that balance the importance between delay and reliability. Based on theoretical KPIs and actual indexes, the satisfaction of links shows a negative exponential. Besides, to reflect the negative impact of violating KPIs, a penalty factor ψ_i is introduced into service satisfaction. In general, penalty factors for safety-related services are larger than non-safety-related services. Thus, for link $l \in \mathcal{L}_i$, its delay satisfaction and reliability satisfaction are written as

$$U^{(1)}_l(D^{\text{E2E}}_{l,t}) = \begin{cases} \exp(-D^{\text{E2E}}_{l,t}), & D^{\text{E2E}}_{l,t} \le D^{\max}_l; \quad \text{(12a)} \\ \exp(-D^{\text{E2E}}_{l,t}) - \psi_i, & D^{\text{E2E}}_{l,t} > D^{\max}_l, \quad \text{(12b)} \end{cases}$$

$$U^{(2)}_l(\varphi_{l,t}) = \begin{cases} \exp(-(1-\varphi_{l,t})), & \varphi_{l,t} \ge \varphi^{\min}_l; \quad \text{(13a)} \\ \exp(-(1-\varphi_{l,t})) - \psi_i, & \varphi_{l,t} < \varphi^{\min}_l, \quad \text{(13b)} \end{cases}$$

where $\varphi^{\min}_l$ is minimum PRR of link l. Note that each period is composed of T slots and C_k remains unchanged during the duration of period k. Once resource configuration is determined by the InP, the remaining problem for each SP is how to maximize the long-term satisfaction of all service requests. Thus, during period k, the problem that SP $i \in \mathcal{I}$ schedules resources among its links $\mathcal{L}_i$ is formulated as:

$$\begin{aligned} \text{P2}\ :\ & \max_{e,\rho}\,[v(c_{i,k})] = \max_{e,\rho} \left[q^{se}_i \sum_{t=1}^{T} \sum_{l \in \mathcal{L}_{i,k}} U_{l,t} \right] \\ & \text{subject to}: \\ & \text{C10}: \sum_{l \in \mathcal{L}_{i,k}} \rho_{l,j,t} \le 1; \forall t \in [1,T], j \in \mathcal{J}_{i,k}, \\ & \text{C11}: \sum_{l \in \mathcal{L}_{i,k}} \sum_{j \in \mathcal{J}_{i,k}} \rho_{l,j,t} \le J_{i,k}; \forall t \in [1,T], \\ & \text{C12}: e_{l,t} \in [0,1,2]; \forall t \in [1,T], l \in \mathcal{L}_{i,k}, \\ & \text{C13}: \sum_{l \in \mathcal{L}_{i,k}} e_{l,t} = 1 \le Y^u_{i,k}; \forall t \in [1,T], \end{aligned} \tag{14}$$

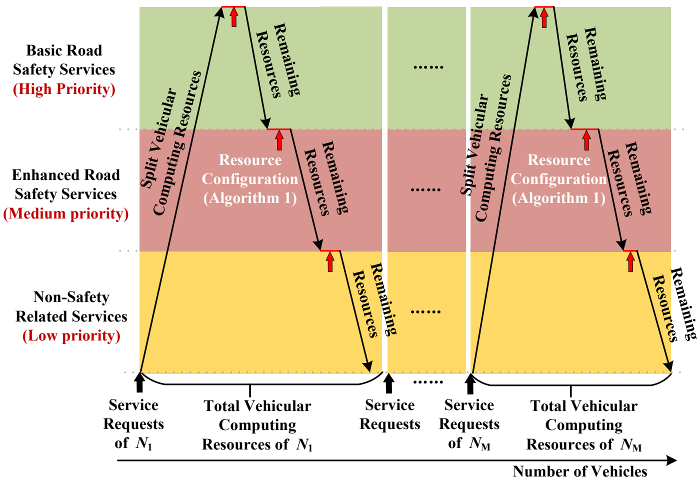

Basic Road Safety Services
(High Priority)
Enhanced Road Safety Services
(Medium priority)
Non-Safety Related Services
(Low priority)
Split Vehicular Computing Resources
Remaining Resources
Resource Configuration (Algorithm 1)
Remaining Resources
Remaining Resources
......
......
......
......
Split Vehicular Computing Resources
Remaining Resources
Resource Configuration (Algorithm 1)
Remaining Resources
Remaining Resources
Service Requests of N_1
Total Vehicular Computing Resources of N_1
Service Requests
Service Requests of N_M
Total Vehicular Computing Resources of N_M
Number of Vehicles

Fig. 4. The flow chart of vehicular computing resources configuration.

where $\boldsymbol{e} = \{e_{l,t}|\forall t \in [1, T], l \in \mathcal{L}_{i,k}\}$ is the set of offloading actions. $\rho = \{\rho_{l,j,t}|\forall t \in [1, T], l \in \mathcal{L}_{i,k}, j \in \mathcal{J}_{i,k}\}$ is the set of the allocation of RBs. C10 refers that each RB only can be assigned to a link at each slot t. C11 indicates that the allocated communication resources cannot exceed the obtained communication resources $J_{i,k}$. C12 implies that the computing task of link l can be handled in only one way, such as being executed locally, offloaded to the MEC server, or remote cloud computing servers. C13 indicates that the allocated edge computing resources to all links cannot exceed the acquired edge computing resources $Y^u_{i,k}$.

5 Dual Timescale Intelligent Resource Management Scheme

The optimization problems described in Section 4 are difficult to solve as they are NP-hard problems. In addition, the IoV needs an intelligent resource management scheme to adapt to dynamic network conditions. Hence, in this section, a novel 2Ts-IRMS is proposed to address the resource allocation problem in the IoV. Specifically, we adopt the proposed JAMR algorithm to address inter-slice resource configuration at each large-time period (Section 5.1) while the JORA-DDQN algorithm is used to solve intra-slice resource scheduling at each small-time slot (Section 5.2).

5.1 Inter-slice Resource Configuration

At large timescales, a central question is how the InP allocates multi-tier resources to SPs to maximize system revenue. Obviously, it is impossible to find the optimal resource configuration of P1 in (10) before the end of period k. That is because the future state of the IoV is unknowable and cannot be obtained in advance. In addition, we cannot acquire all $V(C_k)$ $(\forall C_k \in \mathcal{C})$ for each period in practice. Luckily, within a specific region, the long-term trend of network conditions can be characterized by service requests [20]. Meanwhile, the selection of resource configuration will not change the state of service requests. Consequently, it is reasonable to formulate P1 as a contextual MAB problem. However, the computation complexity is extremely high when the InP simultaneously allocates all network resources. To address this challenge, based on service requests, the problem of multi-tier resource allocation is approximately decomposed into several subproblems. Each subproblem focuses on the characteristics of resources at different tiers.

Algorithm 1 Inter-slice vehicular computing resources configuration based on service priorities

```
for $m = 1, 2, ..., M$ do
  Input: $Y^v_m$.
  for $i = i_1, i_2, i_3$ do
    Input: $\mathcal{L}_{i,k}$.
    Compute $Y^{v,\text{req}}_{m,i,k}$ by (14).
    Compute $Y^{v,\text{rem}}_{m,i,k}$ by (15).
    if $Y^{v,\text{rem}}_{m,i,k} < Y^{v,\text{req}}_{m,i,k}$
      Let $Y^v_{m,i,k} = Y^{v,\text{rem}}_{m,i,k}$;
      Let $E^v_{m,i,k} = 0$;
    else
      Let $Y^v_{m,i,k} = Y^{v,\text{req}}_{m,i,k}$.
      Let $E^v_{m,i,k} = 1$;
  end for
end for
Return $E^v_{i,k}$.
```

5.1.1 Vehicular computing resources

Different from other application scenarios, the IoV contains a large number of safety-related services. It is necessary to guarantee their resource requirements to avoid traffic accidents, at first. As described in Section 3, when vehicular computing resources are sufficient, local execution is the first choice to process the computing tasks of safety-related services. It can avoid unnecessary transmission delay and error. Thus, based on the consideration of service priorities, we first explore vehicular computing resource allocation among slices. Obviously, safety-related services have a higher service priority than non-safety-related services in practical IoV. As for safety-related services, we consider that basic road safety services have a higher service priority than enhanced road safety services. That is because basic driving functions should be guaranteed at first under the condition of insufficient computing resources. To facilitate analysis, the slice for basic road safety services, the slice for enhanced road safety services, and the slice for non-safety services are denoted as i_1, i_2, and i_3, respectively. Indexes are used to reflect their service priority and vehicular computing resources are orderly allocated based on this priority. The specific flow of inter-slice vehicular computing resources configuration is shown in Fig. 4.

It is noteworthy that the real-time policies of task offloading and resource scheduling at small timescales have little dependence on vehicular computing resource configuration. Furthermore, the computing resources of each vehicle only can be used by themselves. Therefore, service requests can be considered the only influencing factor for vehicular computing resource configuration. As for VUE $N_m (m \neq 0)$, the number of required vehicular computing resources $Y^{v,\text{req}}_{m,i,k}$ for slice $i \in \mathcal{I}$ at period k can be approximately calculated according to:

$$Y^{v,req}_{m,i,k} = \left\lceil \frac{\sum\limits_{l\in\mathcal{L}_{i,k}} W^{\max}_l \cdot Z_l \cdot \beta_l \cdot \Lambda_{(l'=m)}}{f_v} \right\rceil , \tag{15}$$

where $\lceil x \rceil = \min\{X \in \mathbb{Z} | X \geq x\}$ is the ceiling function of x. $\mathcal{L}_{i,k}$ is the link set of slice i at period k. $\Lambda_{(l'=m)}$ indicates whether the condition $(l' = m)$ is satisfied. Specifically, $\Lambda_{(l'=m)} = 1$ denotes VRx $N_{l'}$ of link l is VUE N_m; otherwise, $\Lambda_{(l'=m)} = 0$. Since vehicular computing resources are allocated to different slices according to service priorities in turn, the number of vehicular computing resources that each slice can be used to allocate is different. Let $Y^{v,\text{rem}}_{m,i,k}$ be the number of available vehicular computing resources of VUE N_m to slice $i \in \mathcal{I}$, which can be expressed as:

$$Y^{v,\text{rem}}_{m,i,k} = \begin{cases} Y^v_m, & i = i_1; \quad (16\text{a}) \\ 0, & i \neq i_1; Y^{v,\text{rem}}_{m,i',k} < Y^{v,\text{req}}_{m,i',k}; \quad (16\text{b}) \\ Y^{v,\text{rem}}_{m,i',k} - Y^{v,\text{req}}_{m,i',k}, & i \neq i_1; Y^{v,\text{rem}}_{m,i',k} \geq Y^{v,\text{req}}_{m,i',k}, \quad (16\text{c}) \end{cases}$$

where $i' \in \mathcal{I}$ refers to the slice whose priority is one level higher than slice i. After the number of available and required vehicular computing resources of all slices are determined, the number of allocated vehicular computing resources $Y^v_{m,i,k}$ from VUE N_m to slice i at period k can be obtained. Furthermore, to reflect whether the vehicular computing resources requirements of slice i are satisfied, we define the state set of vehicular computing resources allocation at period k as $E^v_{i,k} = \{E^v_{m,i,k} | \forall m \in [1, M]\}$. $E^v_{m,i,k} = 1$ indicates the vehicular computing resources requirement of slice i to VUE N_m is satisfied, otherwise $E^v_{m,i,k} = 0$. The detail of inter-slice vehicular computing resources configuration based on service priorities is described in Algorithm 1.

5.1.2 Edge computing resources & Radio communication resources

In order to ensure that each computing task has a processing location and to avoid wasting resources, the multi-tier computing resources configuration of slice $i \in \mathcal{I}$ meets:

$$L_{i,k} = Y^v_{i,k} + Y^u_{i,k} + Y^c_{i,k} = \sum_{m=1}^{M} E^v_{m,i,k} + Y^u_{i,k} + Y^c_{i,k}, \tag{17}$$

where $L_{i,k} = |\mathcal{L}_{i,k}|$ is the total number of pending computing tasks at period k. $Y^v_{i,k}$ denotes the number of VUEs with sufficient vehicular computing resources allocated to slice i to process relevant computing tasks. After the vehicular computing resources configuration is determined, the problem of large timescales is transformed into the problem that the InP adjusts the configuration of edge computing resources and radio communication resources among slices.

As described above, we constitute problem P1 in (10) as a contextual MAB problem. Specifically, at the beginning of each period, the InP considered the agent first observes its context in the form of a feature vector. The vector indicates the characteristics of requested services and the allocation status of vehicular computing resources of all slices. At period $k \in [K]$, we define the feature vector for arm $C_k \in \mathcal{C}$ as $O_{k,C_k} = \{o_{k,c_{i,k}} | \forall i \in \mathcal{I}\}$, where $o_{k,c_{i,k}} = \{L_{i,k}, \bar{Z}_{i,k}, \bar{\beta}_{i,k}, \bar{A}_{i,k}, Y^v_{i,k}, J_{i,k}, Y^u_{i,k}\}$ is the context of slice i. Term $\bar{Z}_{i,k}$, $\bar{\beta}_{i,k}$, and $\bar{A}_{i,k}$ are the average packet payload, average computing tasks workload, and average packet beacon frequency of links in slice i, respectively. $L_{i,k}$ is the total number of pending computing tasks at period k. $Y^v_{i,k}$ denotes the number of VUEs that allocate sufficient vehicular computing resources to slice i. $J_{i,k}$ is the number of the total RBs allocated to slice i. $Y^u_{i,k}$ is the number of CPU cores of the MEC server allocated to slice i.

Then, the agent chooses to pull an available arm C from the candidate set of multi-tier resources configuration $\mathcal{C}$ with the aid of context information. After pulling an arm, the agent will observe reward $V_k(C_k)$ from selected arm C_k, but the rewards of the other arms are unknown. Over time, the agent aims to collect enough information about the relationship between the context vectors and rewards so that it can predict the next best arm to play by looking at the current context. However, linear contextual bandits make often fail to fit the relationship between the context vectors and rewards in practice. That is because they assume that the expected reward at each period is linear in the feature vector. Thus, in this paper, NeuralUCB is adopted to solve the resource configuration problem.

The key idea of NeuralUCB is to use a neural network $f(O_{k,C_k}; \omega_{k-1})$ to predict the reward of context O_{k,C_k} and compute upper confidence bounds to guide exploration [28]. Specifically, at period k, upper confidence bound P_{k,C_k} for each arm $C_k \in \mathcal{C}$ can be computed as formula (18), where $g(O_{k,C_k}; \omega_{k-1})$ is the gradient of neural network $f(O_{k,C_k}; \omega_{k-1})$. ω_{k-1} is the parameters of the current neural network. δ and δ' are the width and depth of the neural network, respectively. H^{-1}_k is the inverse of matrix H_k. It is worth noting that the scaling factor μ_k is composed of two parts. One is the confidence radius which is similar to linear UCB. The other one is the function approximation error which is newly added to adapt to the unknown nonlinear function. The exploration parameter ϑ is used to control that the choice is inclined to explore or exploit. The larger ϑ the more inclined the action choice is to explore, otherwise to exploit. The detailed calculation expression of μ_k is shown in (19). Herein, η and η' are step size and the number of gradient descent steps, respectively.

$$P_{k,C_k} = f(O_{k,C_k}; \omega_{k-1}) + \mu_{k-1} \sqrt{g(O_{k,C_k}; \omega_{k-1})^{\mathrm{T}} H^{-1}_{k-1} g(O_{k,C_k}; \omega_{k-1})/\delta}, \tag{18}$$

$$\begin{aligned} \mu_k =& \sqrt{1 + \chi_4 \delta^{-1/6} \sqrt{\log \delta} \delta'^4 k^{7/6} {\chi_1}^{-7/6}} \cdot (\vartheta \sqrt{\log \frac{\det H_k}{\det \chi_1 \mathrm{I}} + \chi_5 \delta^{-1/6} \sqrt{\log \delta} \delta'^4 k^{5/3} {\chi_1}^{-1/6} - 2 \log \chi_2 + \sqrt{\chi_1} \chi_3}) \\ &+ (\chi_1 + \chi_6 k \delta')[(1 - \eta \delta \chi_1)^{\eta'/2} \sqrt{k/\chi_1} + \delta^{-1/6} \sqrt{\log \delta} \delta'^7 k^{5/3} {\chi_1}^{-5/3} (1 + \sqrt{k/\chi_1})], \end{aligned} \tag{19}$$

Algorithm 2 NeuralUCB for edge computing resources and radio communication resources allocation

```
Initialization: $\omega_0$, $H_0 = \chi_1 \mathrm{I}$.
for $k = 1, 2, ..., K$ do
    Observe $O_{k,C_k} (\forall C_k \in \mathcal{C})$.
    for $C_k \in \mathcal{C}$ do
        Compute $P_{k,C_k}$ by (18).
        Let $C_k = \arg\max_{C_k \in \mathcal{C}} P_{k,C_k}$.
    end for
    Play $C_k$ and observe reward $V_k(C_k)$.
    Compute $H_k$ by (20).
    for $j = 0, 1, ..., J-1$ do
        $\omega_{j+1} = \omega_j - \eta \nabla L^{\mathrm{NU}}(\omega_j)$.
    end for
    Return $\omega_j$.
    Compute $\mu_k$ by (19).
end for
```

χ_1, χ_2, and χ_3 are the regularization parameter, confidence parameter, and norm parameter, respectively. χ_4, χ_5, and χ_6 are experimental parameters. After upper confidence bounds for all arms are determined, the arm C_k with the largest P_{k,C_k} is chosen and the agent receives the corresponding reward $V_k(C_k)$. Then, NeuralUCB will update H_k as

$$H_k = H_{k-1} + g(O_{k,C_k}; \omega_{k-1}) g(O_{k,C_k}; \omega_{k-1})^{\mathrm{T}} / \delta. \tag{20}$$

At the end of period k, neural network parameter ω_k is updated by using gradient descent to approximately minimize $L^{\mathrm{NU}}(\omega)$. We define $L^{\mathrm{NU}}(\omega)$ as

$$L^{\mathrm{NU}}(\omega) = \sum_{k=1}^{K} \left(f(O_{k,C_k}; \omega) - V_k(C_k)\right)^2 / 2 + \delta \chi_1 ||\omega - \omega^{(0)}||_2^2 / 2, \tag{21}$$

where $\omega^{(0)}$ is the initial parameter of the neural network. The detail of the NeuralUCB algorithm used to allocate edge computing resources and radio communication resources is described in Algorithm 2 and depicted in Fig. 5.

5.2 Intra-slice Resource Scheduling

In a period, once the resource configuration is determined, the remaining problem is how to effectively allocate resources from an SP to UEs to maximize the satisfaction of all links. The optimization problem P2 in (14) is difficult to solve since the time-varying nature of the physical layer. Besides, the decisions of task offloading and resource scheduling cause changes in link states (e.g., queue characteristics and channel quality), and service satisfaction also depends on link states. Therefore, we utilize the DRL method based on MDP to solve the proposed intra-slice resource scheduling problem. First, we formulate our problem as an MDP to accurately describe the process of resource allocation and task offloading.

At each slot t during the k-th period, links will send the information of service requests and available resources to their subscribed slice $i \in \mathcal{I}$. We define the state space of link $l \in \mathcal{L}_{i,k}$ at slot t as $s_{l,t} = \{W_{l,t}, A_{l,t}, Z_l, \beta_l, D_l^{\max}, \varphi_l^{\min}, r_{l,,t}^{v}, r_{l,t}^{u}, Y_{l,i,k}^{v}\}$. Herein, $W_{l,t}$ is the queue length, $A_{l,t}$ denotes the number of instantaneously arriving packets, Z_l is the total packet size, $D_l^{\max}$ is the maximum E2E delay, and $\varphi_l^{\min}$ is minimum PRR.

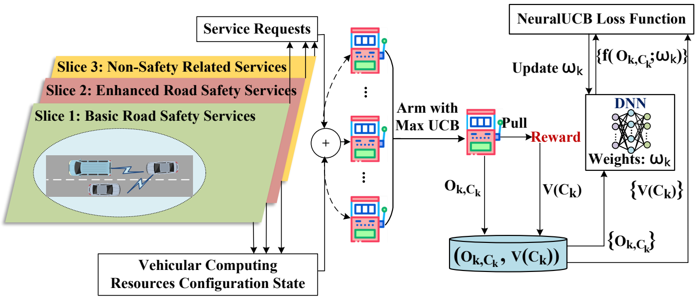

Service Requests
Slice 3: Non-Safety Related Services
Slice 2: Enhanced Road Safety Services
Slice 1: Basic Road Safety Services
Vehicular Computing Resources Configuration State
+
Arm with Max UCB
Pull
Reward
NeuralUCB Loss Function
Update ω_k
$\{f(O_{k,C_k}; \omega_k)\}$
DNN
Weights: ω_k
O_{k,C_k}
$V(C_k)$
$\{V(C_k)\}$
$\{O_{k,C_k}\}$
$(O_{k,C_k}, V(C_k))$

Fig. 5. Edge computing resources and radio communication resources allocation based on NeuralUCB algorithm.

β_l denotes that the input packet requires β_l cycles/bit for processing. $r_{l,,t}^{v}$ and $r_{l,t}^{u}$ are available rates for V2V and V2I transmission, respectively. $Y_{l,i,k}^{v}$ is the number of vehicular computing resources of VUE N_l allocated to slice i. Thus, the state space of slice i at slot t can be defined as: $s_t = \{s_{l,t} | \forall l \in \mathcal{L}_i\} \in \mathcal{S}$.

For link l at slot t, its action space $a_{l,t}$ contains task offloading action and RBs allocation policy, which can be expressed as $a_{l,t} = \{e_{l,t}, \rho_{l,j_1,t}, \rho_{l,j_2,t}, ..., \rho_{l,j_{J_i},t}\}$. Term $e_{l,t}$ is the offloading action of link l at slot t, and $\rho_{l,j,t}$ indicates whether the j-th RB is allocated to link l at slot t. Therefore, the action space of slice i can be defined as: $a_t = \{a_{l,t} | \forall l \in \mathcal{L}_i\} \in \mathcal{A}$.

The goal of resource allocation at this stage is to maximize the satisfaction level of all links within limited resources. Therefore, we set rewards based on the constraint conditions and objective function. After taking action A_t, the reward function is defined as:

$$\begin{aligned} r_t = {} & \ell_1 \cdot \left(\sum_{l \in \mathcal{L}_{i,k}} U_{l,t} \right) \\ & + \ell_2 \cdot \left(\sum_{l \in \mathcal{L}_{i,k}} \rho_{l,j,t} - 1 \right) \cdot \Lambda_{\left(\sum_{l \in \mathcal{L}_{i,k}} \rho_{l,j,t} \leq 1 \right)} \\ & + \ell_3 \cdot \left(\sum_{l \in \mathcal{L}_{i,k}} \sum_{j \in \mathcal{J}_{i,k}} \rho_{l,j,t} - J_{i,k} \right) \cdot \Lambda_{\left(\sum_{l \in \mathcal{L}_{i}} \sum_{j \in \mathcal{J}_{i,k}} \rho_{l,j,t} \leq J_{i,k} \right)} \\ & + \ell_4 \cdot \left(\sum_{l \in \mathcal{L}_{i,k}} (e_{l,t} = 1) - Y_{i,k}^{u} \right) \cdot \Lambda_{\left(\sum_{l \in \mathcal{L}_{i,k}} (e_{l,t} = 1) \leq Y_{i,k}^{u} \right)} \end{aligned} \tag{22}$$

where $\Lambda_{(\cdot)}$ indicates whether the condition $(\cdot)$ is satisfied. Specifically, $\Lambda_{(\cdot)} = 0$ denotes the condition $(\cdot)$ is satisfied, otherwise $\Lambda_{(\cdot)} = -1$.

In the IoV, each slice is regarded as an agent and owns a private neural network. Each agent aims to find the best policy π to maximize the expected cumulative reward $\mathbb{E}[R_t | s, \pi]$ for each state s. The cumulative discounted reward can be expressed as

$$R_t = \sum_{i=0}^{T-1} \gamma^i r_{t+i}, \tag{23}$$

where γ is the discount parameter which reflects the importance of future rewards. The value of γ is restricted between

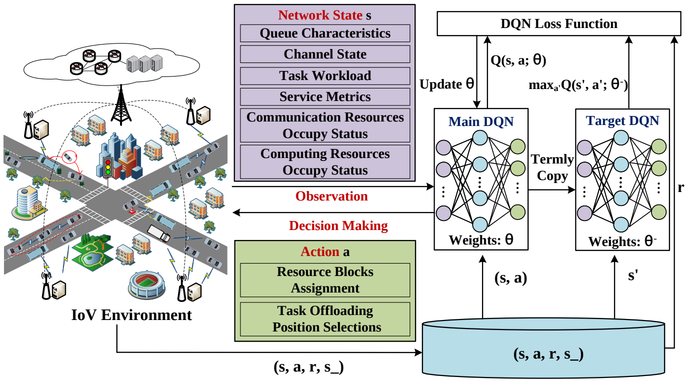


Fig. 6. Intra-slice resource scheduling based on JORA-DDQN algorithm.

0 and 1. A smaller γ represents that mostly care about the instantaneous reward. In value-based reinforcement learning, the state-action function $Q^{\pi}(s,a)$ named as quality function (Q-function) is commonly used to reflect how good policy π is when taking action a at current state s, denoted as

$$Q^{\pi}(s,a)=\mathbb{E}[R_t|s,\pi]. \tag{24}$$

Q-function $Q(s,a)$ provides the optimum policy π^* by selecting action a that maximizes the Q-value for the state s:

$$\pi^*(s)=\arg\max_{a\in\mathcal{A}} Q(s,a), \forall s\in\mathcal{S}. \tag{25}$$

Based on the definitions above, we can seek out the optimal policy π^* via the recursive nature of the Bellman equation, i.e.,

$$\begin{aligned}Q^{\pi}(s_t,a_t) &\leftarrow Q^{\pi}(s_t,a_t)\\ &+\alpha\left(r_t+\gamma\max_{a_{t+1}\in\mathcal{A}}Q^{\pi}(s_{t+1},a_{t+1})-Q^{\pi}(s_t,a_t)\right).\end{aligned} \tag{26}$$

However, in high-dimensional state spaces, the classic Q-learning method cannot efficiently compute the Q-function for all states. To remedy this problem, DDQN improves the Q-learning by combining the neural networks with Q-learning [41]. Specifically, raw data is input into neural networks as the state. Then, the Q-function is approximated by deep neural networks. It is worth noting that DDQN has two separate networks: the main network and the target network. The main network approximates the Q-function, while the target network gives the temporal difference (TD) target for updating the main network. During the training phase, the main network parameters θ are updated after every action while the target network parameters θ^- are updated after a certain period. At each iteration, the main Q-network is trained towards the target value by minimizing the loss function. We set a mean-squared error (MSE) loss function. The function can measure how closely the $Q(s,a;\theta)$ comes to satisfy the Bellman equation:

$$L^{\mathrm{DDQN}}(\theta)=\frac{1}{2}\cdot\mathbb{E}\left[\left(y_t^{\mathrm{DDQN}}-Q^{\pi}\left(s_t,a_t;\theta_t\right)\right)^2\right], \tag{27}$$

where

$$y_t^{\mathrm{DDQN}}=r_t+\gamma Q^{\pi}\left(s_{t+1},\arg\max_{a_{t+1}\in\mathcal{A}}Q(s_{t+1},a_{t+1};\theta_t);\theta_t^-\right). \tag{28}$$

Algorithm 3 JORA-DDQN for intra-slice resource scheduling

```
Initialization: main network weights $\theta$,
                target network weights $\theta^-$,
                experience replay buffer.
for $episode = 1, 2, \dots, E$ do
  Receive the initial observation $s$;
  for $t = 1, 2, ..., T$ do
    Take action $a_t = \arg\max_a Q^{\pi}(s_t, a; \theta)$ with probability
    $1-\epsilon$ or a random action with probability $\epsilon$ ;
    Get reward $r_t$ and observe next state $s_{t+1}$;
    Store the experience $(s_t, a_t, r_t, s_{t+1})$ into the
    experience replay buffer;
    Get a batch $U$ samples $(s_t, a_t, r_t, s_{t+1})$ from the
    replay memory;
    Calculate the target Q-value $y_t^{\mathrm{target}}$ from the target
    network by (28);
    Update the main network by minimizing the loss
    $L^{\mathrm{DDQN}}(\theta)$ in (27) and perform a gradient descent
    step on $L^{\mathrm{DDQN}}(\theta)$ with respect $\theta$;
    Every G steps, update the target network $\theta^- = \theta$
  end for
end for
```

Once $\{\theta\}$ is determined, our agent will output near-optimal resource allocation strategies and computation offloading decisions using a discrete set of approximate action values. The detail of JORA-DDQN is described in Algorithm 3 and is depicted in Fig. 6.

6 PERFORMANCE EVALUATION

6.1 Simulation Environment

In our simulation, we utilize Pytorch 1.10.0 on Ubuntu 18.04.6 LTS to implement the 2Ts-IRMS algorithm and compare it with multiple comparison algorithms. For experimental purposes, a cellular V2X network environment based on the SUMO platform is established, which consists of a real road network, an MBS, and several VUEs and RSUs. Specifically, to fit the reality, we import the road network around the Beijing University of Posts and Telecommunication from OpenStreetMap to SUMO at first [40]. Then, the whole road network is divided into 9 blocks, which is consistent with the road partitioning strategy of the Manhattan case [51]. An RSU is deployed in the center of each block and can communicate with vehicles within its coverage, which is depicted in Fig. 7. In order to reflect traffic in urban regions as much as possible, vehicles randomly choose lanes of departure, positions of departure, and speed of departure to enter the generated road network and follow the car-following model of Krauss and the lane-changing model of LC2013 for movement [37].

For the communication resources in the IoV, we assume that there are 50 RBs with 180kHz bandwidth to be allocated. For the computing resources, we let the computation capacity (i.e., CPU frequency) of a single CPU core for the VUEs be 10^8 cycles/s, and the number of CPU cores for any VUE be uniformly selected from the set $\{1,\ 2,\ 4,\ 8\}$. Similarly, let the computation capacity of a single CPU core for the RSU be 10^9 cycles/s, and the number of CPU cores for the RSU be fixed to 8. For the services required by the UEs, we assume that there are six typical V2X use cases. The

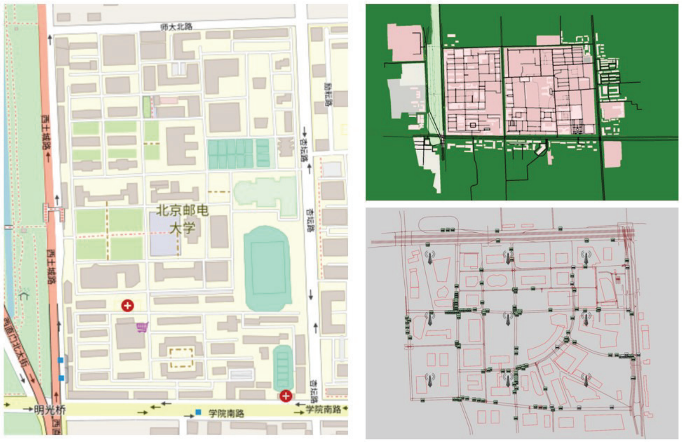


Fig. 7. Real road conditions simulation of Beijing University of Posts and Telecommunications based on SUMO.

TABLE 3
SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
Y_u	8	Y_m^v	$\{1, 2, 4, 8\}$
B	180kHz	$q_1^{se}, q_2^{se}, q_3^{se}$	0.5,0.7,0.9
J	50	ψ_1, ψ_2, ψ_3	0.9,1,0.5
M	20	q^{cm}, q^{cu}, q^{cc}	0.006,0.09,0.03
σ^2	-114dBm	α_d, α_r	0.5,0.5
$W_l^{\max}$	5	f_v, f_u, f_c	$10^8, 10^9, 10^{10}$
V2V Transmit Power	10dBm	V2I Transmit Power	23dBm
Carrier Frequency	5.9GHz	Small Scale Fading	Rayleigh Fading

transmission characteristics and KPIs of the use cases follow Table 1.

In addition, to better evaluate how different V2X services affect the network performance, various combinations of services have been considered in the simulations. By selecting these combinations, we want to test if the proposed 2Ts-RL can satisfy the requirements of multiple services, especially for safety-related services. To simplify, we make slices 1, 2, and 3 represent the slice for basic road safety services, the slice for enhanced road safety services, and the slice for non-safety-related services, respectively. The simulation parameters and neural network parameters are summarized in Table 3 and Table 4, respectively. Afterward, we compare the 2Ts-IRMS algorithm with multiple compared algorithms, which are described as follows:

- Hierarchical resource allocation schemes: The two-timescale bidding resource management scheme (2Ts-BRMS) adopts the generalized Kelly mechanism (GKM) to address the inter-slice multi-dimensional resource configuration problem and allocates resources to users according to channel quality (CQ) [23]. In the two-timescale fair resource management scheme (2Ts-FRMS), both the InP and SPs adopt the dominant resource fairness (DRF) approach to allocate multi-dimensional resources [50].
- Inter-slice resource configuration schemes: The proportional allocation scheme (PA) proportionally allocates resources to slices based on the number of subscribers and average resource requirements [52]. As for the context-aware configuration scheme (CA), it adjusts inter-slice resource configuration based on the localized service requests and traditional UCB algorithm [45].
- Intra-slice resource scheduling schemes: As for communication resource allocation, the queue-aware resource allocation strategy (QA) calculates the queue length of each link. The longer the queue length, the more communication resources are allocated. In the fair resource allocation strategy (FA), the communication resources are equally shared by all links. As for computing resource allocation, the local execution scheme (LE), the edge execution scheme (EE), and the cloud execution scheme (CE) make all tasks to be executed on user terminals, MEC servers, and remote cloud servers, respectively [21].

TABLE 4
NEURAL NETWORK PARAMETERS

Parameter	Value	Parameter	Value
Confidence factor	-1.0	Discount factor	0.9
Regularization factor	1	Buffer size	20000
Exploration probability	0.1	Hidden Layer	3
Learning rate	0.01	Optimizer	Adam
Activation function	ReLU	Minibatch size	100

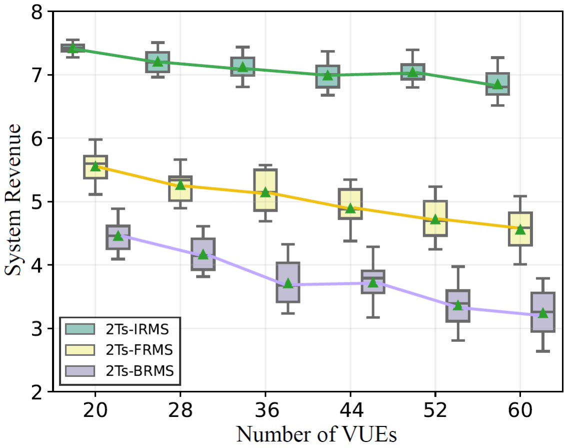
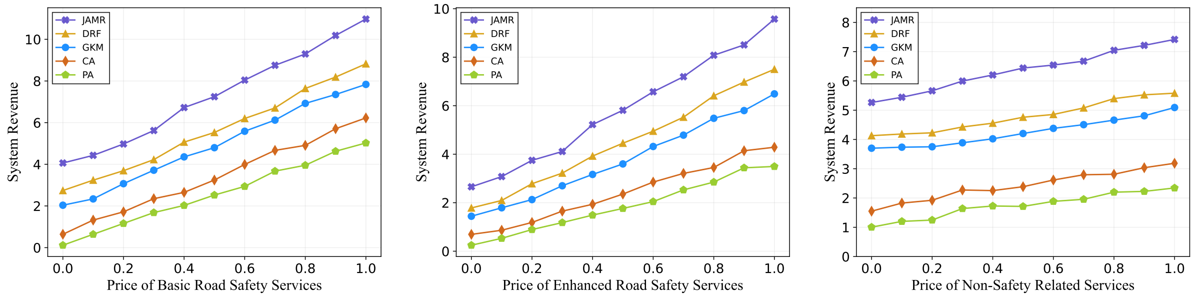


Fig. 8. Comparison of the achieved valuation of hierarchical resource allocation schemes under various combinations of services.

6.2 Simulation Results

Fig. 8 compares the achieved values of the valuation function of three hierarchical resource allocation schemes under various combinations of services. With the fixed number of VUEs in the cellular network, the proportion of users in slices 1, 2, and 3 iterates through all possible combinations. Compared to other comparison algorithms, 2Ts-IRMS has the highest valuation value while owning more stale performance. That is because it can dynamically adjust resource allocation according to the different number of users and service requests. It is noted that there are two main reasons for the low performance of 2Ts-BRMS. On the one hand, in the phase of inter-slice resource configuration, 2Ts-BRMS

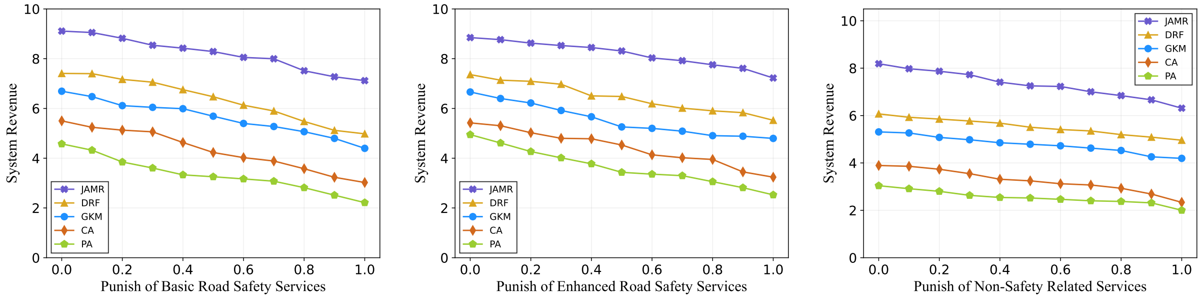


(a)

(b)

(c)

Fig. 9. Comparison of system revenue of multiple inter-slice resource configuration schemes under different unit prices of services. (a) Revenue under different unit prices of basic road safety services; (b) Revenue under different unit prices of enhanced road safety services; (c) Revenue under different unit prices of non-safety related services.

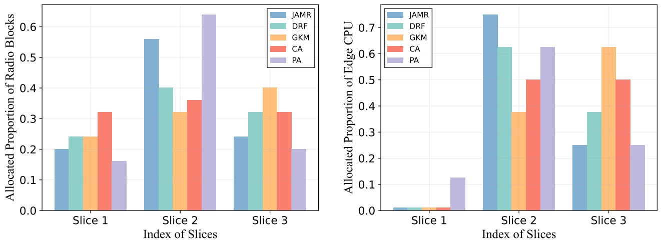


(a)

(b)

(c)

Fig. 10. Comparison of system revenue of multiple inter-slice resource configuration schemes under different punishing values of services. (a) Revenue under different punishing values of basic road safety services; (b) Revenue under different punishing values of enhanced road safety services; (c) Revenue under different punishing values of non-safety related services.



(a)

(b)

Fig. 11. Resource configuration among slices under different inter-slice resource configuration schemes. (a) Allocated proportion of radio blocks; (b) Allocated proportion of edge cpu cores.

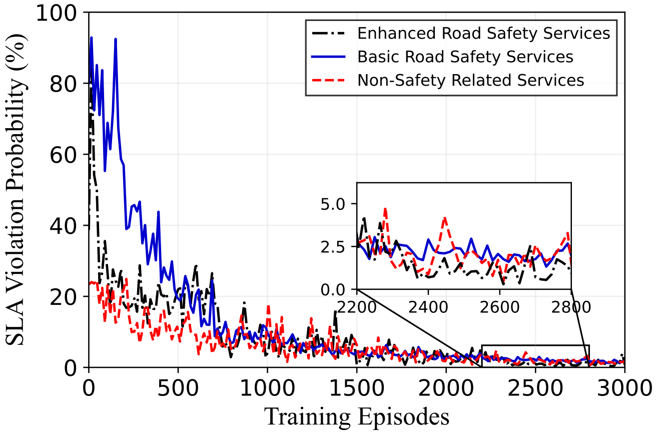
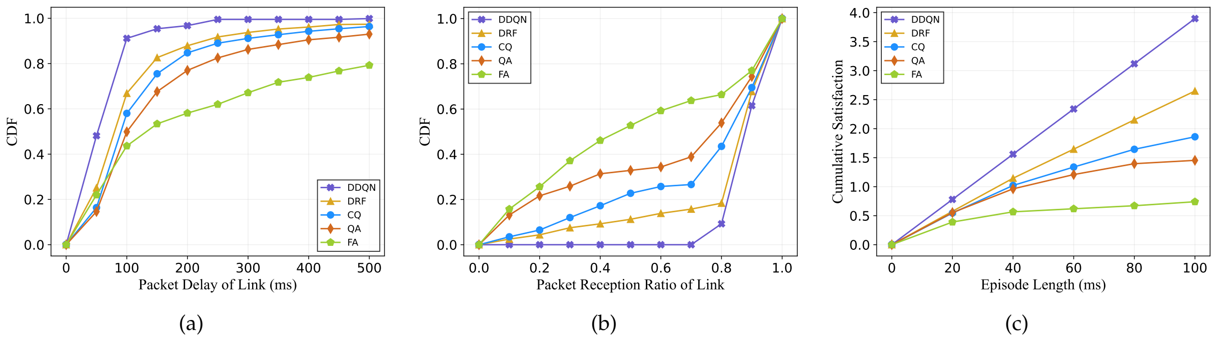


Fig. 12. Convergence performance of the proposed JORA-DDQN in SLA violation probability.

equally treats all slices which leads to the failure to meet safety-related services resulting in a greater negative impact. On the other hand, when SPs allocate resources to users, only considering communication resources affects the quality of services, especially for enhanced road safety services. Besides, the value of the valuation function decreases with the increase in the number of users, because the number of resources is limited and cannot meet too many users.

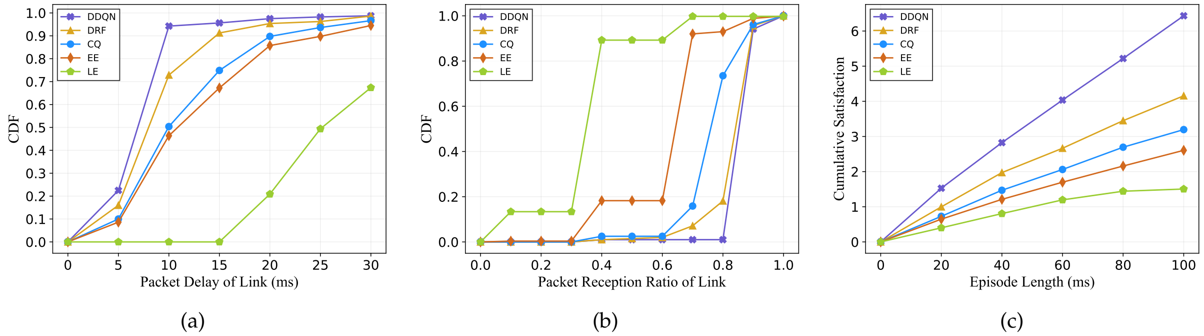
To further analyze the impact of the JAMR approach on inter-slice resource configuration, we evaluate the performance of multiple inter-slice resource configuration schemes by adjusting the unit price of a certain service, which is depicted in Fig. 9. When the unit price of other slices remains unchanged, it can be observed that the system revenue increases with the unit price of the current service. Significantly, JAMR can maintain the highest valuation at any price setting, which further validates its self-adaptive capability. For the three slices proposed, JAMR provides a gain of 24%, 40%, and 76% with respect to DRF, GKM, and CA on average, respectively. The CA scheme with limited fitting ability has a lower revenue since its performance is seriously affected by the nonlinear problem. Furthermore, the fluctuation of the system revenue in the slice for non-safety-related services is obviously smaller than in other slices. The reason for this phenomenon is that the utility of non-safety-related services has a smaller impact on the sys-

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3288637



(a) (b) (c)

Fig. 13. Performance indicators of link in the slice for basic road safety services under different intra-slice resource scheduling schemes. (a) CDF of packet delay of link; (b) CDF of packet reception ration of link; (c) Cumulative satisfaction of links.



(a) (b) (c)

Fig. 14. Performance indicators of link in the slice for enhanced road safety services under different intra-slice resource scheduling schemes. (a) CDF of packet delay of link; (b) CDF of packet reception ration of link; (c) Cumulative satisfaction of links.

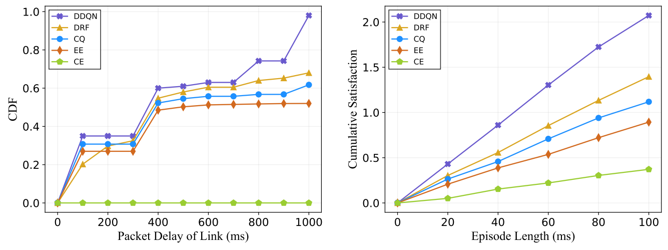
tem performance. Similarly, Fig. 10 depicts the system revenue of multiple inter-slice resource configuration schemes under different punishing values of services. Although the increase in the penalties of services leads to a decrease in the system revenue, JAMR still maintains the highest revenue no matter how the penalties change.

Fig. 11 shows the number of communication resources (i.e., RBs) and edge computing resources (i.e., CPU cores) allocated to different slices under different inter-slice resource configuration schemes. When the number of vehicles remains unchanged at 20 and the proportion of users in slices 1, 2, and 3 is 2:2:1, JAMR assigns 10 RBs to slice 1, 28 RBs to slice 2, and 12 RBs to slice 3. At the same time, 25% of the edge computing resources are allocated to slice 3 and all remaining edge computing resources are allocated to slice 2. Significantly, although the PA scheme allocates enough resources to the slice for enhanced road safety services, the performance of other slices is seriously compromised.

After determining the resource configuration policy for all slices, each SP allocates obtained resources to its subscribers to maximize the long-term satisfaction of all links. To guarantee the isolation among slices, each SP is equipped with an exclusive agent to implement resource scheduling among users based on the proposed JORA-DDQN scheme. To illustrate the convergence performance of the JORA-DDQN scheme in different slices, we plot the variation trend of the SLA violation probability over training episodes for each slice in Fig. 12. In this paper, the SLA mainly refers to delay and reliability. At the beginning of the training, the value of the SLA violation probability is high. With the increase of training episodes, the SLA violation probability gradually decreases. After 1500 episodes, the SLA violation probability is leveling off, which means that all of the slices have converged. Moreover, the slice for enhanced road safety services has the lowest SLA violation probability, which is consistent with KPIs requirements in Table 1.

Fig. 13 depicts the performance of links in the slice for basic road safety services during an episode. It consists of the cumulative distribution functions (CDF) of packet delay, CDF of packet reception ratio, and cumulative satisfaction of links. In view of the characteristics of basic road safety services (i.e., small packet size and high timeliness and reliability), it is more important to allocate radio resources than computing resources. That is because most tasks are sufficient to be processed at terminal devices without occupying the computing resources of the MEC server or remote cloud servers. Thus, the DRF, CA, QA, and FA are selected as benchmark schemes to compare with the JORA-DDQN approach. Meanwhile, to ensure fairness, the task offloading policy of benchmark schemes is consistent with JORA-DDQN. Obviously, due to the flexible resource management paradigm, the proposed JORA-DDQN scheme significantly outperforms benchmark schemes whether in terms of delay, reliability, or cumulative satisfaction. Specifically, the average packet delay of link is 93.033 ms and the algorithm can maintain the packet reception ratio of link at least 90%. Besides, JORA-DDQN has the highest service satisfaction and provides a gain of 50% with respect to DRF.

As for the slice for enhanced road safety services, the packet size is much larger than basic road safety services, and more CPU cycles are needed to process data. The computing resources of the terminal devices are insufficient to



(a) (b)

Fig. 15. Performance indicators of link in the slice for non-safety related services under different intra-slice resource scheduling schemes. (a) CDF of packet delay of link; (b) Cumulative satisfaction of links.

support the simultaneous processing of numerous data. It is necessary to access the MEC server. Thus, the DRF, CA, LE, and EE are selected as benchmark schemes to compare with the JORA-DDQN approach. Similarly, to ensure fairness, the task offloading policy of CA and the RB scheduling policy of LE and EE are consistent with JORA-DDQN. Fig. 14 depicts the performance of links in the slice for enhanced road safety services during an episode. In the proposed scheme, the average packet delay of link is 9.608 ms and the algorithm can maintain the packet reception ratio of link at least 99%. Meanwhile, JORA-DDQN is able to maintain the highest service satisfaction and provides a gain of 52% with respect to DRF. Notably, the LE scheme with the worst performance fails to meet the KPIs requirements of most links.

As described in Section 6.1, the slice for non-safety-related services is expected to offload computing tasks to the MEC server or remote cloud servers. Thus, the DRF, CA, EE, and CE are selected as benchmark schemes to compare with the JORA-DDQN approach. Considering that the slice has a low sensitivity to the reliability, we only draw the curves of delay and satisfaction in Fig. 15. It is observed that JORA-DDQN can effectively use limited resources to reduce task execution delay as much as possible. The CE scheme has poor performance because it will generate additional communication delays.

7 CONCLUSION

In this paper, we propose three types of network slicing to accommodate diversified V2X services over a common physical infrastructure. Specifically, the slice for basic road safety services is used to fulfill the need for imminent warning to nearby entities in time; the slice for enhanced road safety services aims to achieve a higher level of automatic driving; the slice for non-safety related services focuses on improving driving comfort and efficiency of users. Furthermore, in order to take full advantage of multi-tier resources and consider time-varying network conditions, a novel dual timescale intelligent resource management scheme is proposed. First, at the beginning of each period, the InP jaggedly tunes multi-tier resource configuration among slices to improve system revenue. Then, constrained by limited resources obtained from the InP, each SP carries out real-time task offloading and resource scheduling decisions to maximize the long-term service satisfaction of all users. Finally, based on the effect of the action on the states, we propose JAMR and JORA-DDQN algorithms for learning the optimal strategies of proposed problems, respectively. Simulation results show that our proposed 2Ts-IRMS can effectively guarantee the performance requirements of users and improve the system revenue compared with the benchmark algorithms.

REFERENCES

[1] *Global status report on road safety 2018*, World Health Organization, Geneva, Switzerland, 2018.

[2] *Study on LTE support for Vehicle to Everything (V2X) services, Release 14*, document 3GPP TR22.885, Dec. 2015.

[3] *Study on enhancement of 3GPP Support for 5G V2X Services, Release 16.2.0*, document 3GPP TR 22.886, Dec. 2018.

[4] T. Wang, S. Chen, Y. Zhu, A. Tang and X. Wang, ”LinkSlice: Fine-Grained Network Slice Enforcement Based on Deep Reinforcement Learning,” in *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 8, pp. 2378-2394, Aug. 2022.

[5] M. Richart, J. Baliosian, J. Serrat and J. Gorricho, ”Resource Slicing in Virtual Wireless Networks: A Survey,” in *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462-476, Sep. 2016.

[6] Y. Han, X. Tao, X. Zhang and S. Jia, ”Hierarchical Resource Allocation in Multi-Service Wireless Networks With Wireless Network Virtualization,” in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 11811-11827, Oct. 2020.

[7] J. Hwang, L. Nkenyereye, N. Sung, J. Kim and J. Song, ”IoT Service Slicing and Task Offloading for Edge Computing,” in *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11526-11547, Jul. 2021.

[8] N. Modina, R. El Azouzi, F. De Pellegrini, D. S. Menasche and R. Figueiredo, ”Joint Traffic Offloading and Aging Control in 5G IoT Networks,” in *IEEE Transactions on Mobile Computing*, doi: 10.1109/TMC.2022.3154089.

[9] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini and H. Flinck, ”Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions,” in *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2429-2453, third quarter 2018.

[10] J. Martín-Pérez, F. Malandrino, C. F. Chiasserini, M. Groshev and C. J. Bernardos, ”KPI Guarantees in Network Slicing,” in *IEEE/ACM Transactions on Networking*, doi: 10.1109/TNET.2021.3120318, Oct. 2021.

[11] J. Zhao, Q. Li, Y. Gong and K. Zhang, ”Computation Offloading and Resource Allocation For Cloud Assisted Mobile Edge Computing in Vehicular Networks,” in *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944-7956, Aug. 2019.

[12] X. Zhang, Z. Li, C. Lai and J. Zhang, ”Joint Edge Server Placement and Service Placement in Mobile-Edge Computing,” in *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 11261-11274, 1 Jul, 2022.

[13] P. Arthurs, L. Gillam, P. Krause, N. Wang, K. Halder and A. Mouzakitis, ”A Taxonomy and Survey of Edge Cloud Computing for Intelligent Transportation Systems and Connected Vehicles,” in *IEEE Transactions on Intelligent Transportation Systems*, doi: 10.1109/TITS.2021.3084396, Jun. 2021.

[14] P. Arthurs, L. Gillam, P. Krause, N. Wang, K. Halder and A. Mouzakitis, ”A Taxonomy and Survey of Edge Cloud Computing for Intelligent Transportation Systems and Connected Vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1-16, Jun. 2021.

[15] E. El Haber, T. M. Nguyen and C. Assi, ”Joint Optimization of Computational Cost and Devices Energy for Task Offloading in Multi-Tier Edge-Clouds,” in *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3407-3421, May. 2019.

[16] M. M. Gomez, S. Chatterjee, M. J. Abdel-Rahman, A. B. MacKenzie, M. B. H. Weiss and L. DaSilva, ”Market-Driven Stochastic Resource Allocation Framework for Wireless Network Virtualization,” in *IEEE Systems Journal*, vol. 14, no. 1, pp. 489-499, Mar. 2020.

[17] F. Tang, Y. Kawamoto, N. Kato and J. Liu, ”Future Intelligent and Secure Vehicular Network Toward 6G: Machine-Learning Approaches,” in *Proceedings of the IEEE*, vol. 108, no. 2, pp. 292-307, Feb. 2020.

[18] F. Liu, J. Chen, Q. Zhang and B. Li, ”Online MEC Offloading for V2V Networks,” in *IEEE Transactions on Mobile Computing*, 2022, doi: 10.1109/TMC.2022.3186893.

[19] J. Shi, J. Du, Y. Shen, J. Wang, J. Yuan and Z. Han, ”DRL-Based V2V Computation Offloading for Blockchain-Enabled Vehicular Networks,” in *IEEE Transactions on Mobile Computing*, doi: 10.1109/TMC.2022.3153346.

[20] J. Mei, X. Wang, K. Zheng, G. Boudreau, A. B. Sediq and H. Abou-Zeid, ”Intelligent Radio Access Network Slicing for Service Provisioning in 6G: A Hierarchical Deep Reinforcement Learning Approach,” in *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 6063-6078, Sep. 2021.

[21] S. Yu, X. Chen, Z. Zhou, X. Gong and D. Wu, ”When Deep Reinforcement Learning Meets Federated Learning: Intelligent Multi-timescale Resource Management for Multiaccess Edge Computing in 5G Ultradense Network,” in *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2238-2251, Feb. 2021.

[22] X. Chen, Z. Zhifeng, W. Celimuge, B. Mehdi, L. Hang, J. Yusheng and Z. Honggang, ”Multi-Tenant Cross-Slice Resource Orchestration: A Deep Reinforcement Learning Approach,” in *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2377-2392, Oct. 2019.

[23] Y. K. Tun, N. H. Tran, D. T. Ngo, S. R. Pandey, Z. Han and C. S. Hong, ”Wireless Network Slicing: Generalized Kelly Mechanism-Based Resource Allocation,” in *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 8, pp. 1794-1807, Aug. 2019.

[24] F. Zhou, Y. Wu, R. Q. Hu and Y. Qian, ”Computation Rate Maximization in UAV-Enabled Wireless-Powered Mobile-Edge Computing Systems,” in *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1927-1941, Sep. 2018.

[25] Y. Liu, X. Wang, G. Boudreau, A. B. Sediq and H. Abou-zeid, ”Deep Learning Based Hotspot Prediction and Beam Management for Adaptive Virtual Small Cell in 5G Networks,” in *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 1, pp. 83-94, Feb. 2020.

[26] Y. He, N. Zhao and H. Yin, ”Integrated Networking, Caching, and Computing for Connected Vehicles: A Deep Reinforcement Learning Approach,” in *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 44-55, Jan. 2018.

[27] R. Zhou, Z. Xueying, Q. Shixin, L. John, Z. Zhi, H. Hao and L. Zongpeng, ”Online Task Offloading for 5G Small Cell Networks,” in *IEEE Transactions on Mobile Computing*, vol. 21, no. 6, pp. 2103-2115, Jun. 2022.

[28] Zhou D, Li L, Gu Q, “Neural Contextual Bandits with UCB-based Exploration,” *arXiv preprint arXiv: 1911.04462v3*, Jul. 2020.

[29] *Study on Radio Access Network (RAN) Sharing Enhancements, Release 13*, document 3GPP TR 22.852, Sep. 2014.

[30] *Telecommunication Management; Network Sharing; Concepts and Requirements, Release 15*, document 3GPP TS 32.130, Jun. 2018.

[31] X. Ge, ”Ultra-Reliable Low-Latency Communications in Autonomous Vehicular Networks,” in *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5005-5016, May. 2019.

[32] C. Campolo, A. Molinaro, A. Iera and F. Menichella, ”5G Network Slicing for Vehicle-to-Everything Services,” in *IEEE Wireless Communications*, vol. 24, no. 6, pp. 38-45, Dec. 2017.

[33] J. Mei, X. Wang and K. Zheng, ”Intelligent Network Slicing for V2X Services Toward 5G,” in *IEEE Network*, vol. 33, no. 6, pp. 196-204, Nov.-Dec. 2019.

[34] Y. Wu, H. -N. Dai, H. Wang, Z. Xiong and S. Guo, ”A Survey of Intelligent Network Slicing Management for Industrial IoT: Integrated Approaches for Smart Transportation, Smart Energy, and Smart Factory,” in *IEEE Communications Surveys & Tutorials*, doi: 10.1109/COMST.2022.3158270, Mar. 2022.

[35] A. A. Khan, M. Abolhasan, W. Ni, J. Lipman and A. Jamalipour, ”An End-to-End (E2E) Network Slicing Framework for 5G Vehicular Ad-Hoc Networks,” in *IEEE Transactions on Vehicular Technology*, vol. 70, no. 7, pp. 7103-7112, Jul. 2021.

[36] W. Wu, C. Nan, Z. Conghao, L. Mushu, S. Xuemin, Z. Weihua and L. Xu, ”Dynamic RAN Slicing for Service-Oriented Vehicular Networks via Constrained Learning,” in *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2076-2089, July 2021.

[37] F. Lyu, Z. Hongzi, Z. Haibo, X. Wenhao, Z. Ning, L. Minglu and S. Xuemin, ”SS-MAC: A Novel Time Slot-Sharing MAC for Safety Messages Broadcasting in VANETs,” in *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3586-3597, Apr. 2018.

[38] F. Mason, G. Nencioni and A. Zanella, ”Using Distributed Reinforcement Learning for Resource Orchestration in a Network Slicing Scenario,” in *IEEE/ACM Transactions on Networking*, vol. 31, no. 1, pp. 88-102, Feb. 2023.

[39] Y. Cui, X. Huang, P. He, D. Wu and R. Wang, ”QoS Guaranteed Network Slicing Orchestration for Internet of Vehicles,” in *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2022.3147897, Jan. 2022.

[40] N. Kumar, S. Mittal, V. Garg and N. Kumar, ”Deep Reinforcement Learning-Based Traffic Light Scheduling Framework for SDN-Enabled Smart Transportation System,” in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2411-2421, Mar. 2022.

[41] Van Hasselt H, Guez A, Silver D. ”Deep reinforcement learning with double q-learning,” in *Proceedings of the AAAI conference on artificial intelligence*, Arizona, U.S.A, Mar. 2016.

[42] M. Zhang, Y. Dou, P. H. J. Chong, H. C. B. Chan and B. -C. Seet, ”Fuzzy Logic-Based Resource Allocation Algorithm for V2X Communications in 5G Cellular Networks,” in *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2501-2513, Aug. 2021.

[43] X. Huang, L. He, X. Chen, L. Wang and F. Li, ”Revenue and Energy Efficiency-Driven Delay-Constrained Computing Task Offloading and Resource Allocation in a Vehicular Edge Computing Network: A Deep Reinforcement Learning Approach,” in *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8852-8868, Jun. 2022.

[44] L. Zanzi, V. Sciancalepore, A. Garcia-Saavedra, H. D. Schotten and X. Costa-Pérez, ”LACO: A Latency-Driven Network Slicing Orchestration in Beyond-5G Networks,” in *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 667-682, Jan. 2021.

[45] P. Zhao, H. Tian, K. -C. Chen, S. Fan and G. Nie, ”Context-Aware TDD Configuration and Resource Allocation for Mobile Edge Computing,” in *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1118-1131, Feb. 2020.

[46] M. Zhang, Y. Dou, P. H. J. Chong, H. C. B. Chan and B. -C. Seet, ”Fuzzy Logic-Based Resource Allocation Algorithm for V2X Communications in 5G Cellular Networks,” in *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2501-2513, Aug. 2021.

[47] *Overall description of Radio Access Network (RAN) aspects for Vehicle-to-everything (V2X) based on LTE and NR, Release 17.0.0*, document 3GPP TR 37.985, Dec. 2021.

[48] H. Yang, K. Zheng, K. Zhang, J. Mei and Y. Qian, ”Ultra-Reliable and Low-Latency Communications for Connected Vehicles: Challenges and Solutions,” in *IEEE Network*, vol. 34, no. 3, pp. 92-100, Jun. 2020.

[49] D. Harutyunyan, N. Shahriar, R. Boutaba and R. Riggio, ”Latency and Mobility–Aware Service Function Chain Placement in 5G Networks,” in *IEEE Transactions on Mobile Computing*, vol. 21, no. 5, pp. 1697-1709, May. 2022.

[50] T. Mohammed, B. Jedari and M. Di Francesco, ”Efficient and Fair Multi-Resource Allocation in Dynamic Fog Radio Access Network Slicing,” in *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 24600-24614, 15 Dec. 2022.

[51] *Study on LTE-based V2X Services, Release 14*, document 3GPP TR 36.885, Jun. 2016.

[52] D. T. Nguyen, L. B. Le, and V. K. Bhargava, “A market-based framework for multi-resource allocation in fog computing,” in *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1151–1164, Jun. 2019.

[53] F. Mason, G. Nencioni and A. Zanella, ”Using Distributed Reinforcement Learning for Resource Orchestration in a Network Slicing Scenario,” in *IEEE/ACM Transactions on Networking*, vol. 31, no. 1, pp. 88-102, Feb. 2023.

Yu liu is currently working toward a Ph.D. degree with the Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include Internet of vehicles, multi-access edge computing, resource management and orchestration, deep reinforcement learning, etc.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3288637

Zirui Zhuang (Member, IEEE) received his B.S. degree and his Ph.D. degree from Beijing University of Posts and Telecommunications in 2015 and 2020, respectively. He is currently a postdoctoral researcher with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. In 2019, he visited the Department of Electrical and Computer Engineering, University of Houston. His research interests involve network routing and management for nextgeneration network infrastructures, using machine learning and artificial intelligence techniques, including deep learning, reinforcement learning, graph representation, multi-agent system, and Lyapunovbased optimization.

Qi Qi (Senior Member, IEEE) received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2010. She is currently an Associate Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. She has authored or coauthored more than 30 papers in international journal, and is the recipient of two National Natural Science Foundations of China. Her research interests include edge computing, cloud computing, Internet of Things, ubiquitous services, deep learning, and deep reinforcement learning.

Jingyu Wang (Senior Member, IEEE) received his Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2008. He is currently a Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. He is senior member of IEEE/CIC and selected for Beijing Young Talents Program. He has published more than 100 papers in such as the ToN, JSAC, TMC, CVPR, ACL, MM, ICDE, AAAI and so on. His research interests include broad aspects of Intelligent Networks, Edge/Cloud Computing, Machine Learning, AIOps and Self-Driving Network, IoV/IoT, Knowledge-Defined Network and Intent-Driven Networking.

Dezhi Chen is currently working toward a Ph.D. degree with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include UAV control, game theory, and next-generation mobile communication networks using reinforcement learning and artificial intelligence technologies.

Lu Lu received her master degree from Beijing University of Posts and Telecommunications in 2004. She is the deputy director of department of Network and IT technology at China Mobile Research Institute, and the leader of core network group of CCSA TC5. Her research interest covers mobile core network, future network architecture, edge computing etc.)

Hongwei Yang Project Manager at China Mobile Research Institute. His research interest covers Network intelligence and Network performance measurement.

Jianxin Liao (Member, IEEE) obtained his Ph.D degree at University of Electronics Science and Technology of China in 1996. He is currently the dean of Network Intelligence Research Center and the full professor of State Key laboratory of Networking and Switching Technology in Beijing University of Posts and Telecommunications. He is the Specially-invited Professor of the “Yangtse River Scholar Award Program” by the Ministry of Education. His main research interests include mobile intelligent network, service network intelligent, networking architectures and protocols, and multimedia communication.

Zhu Han (Fellow, IEEE) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively.

From 2000 to 2002, he was an R & D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a John and Rebecca Moores Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. Dr. Han was an IEEE Communications Society Distinguished Lecturer from 2015-2018, AAAS fellow since 2019, and ACM distinguished Member since 2019. Dr. Han is a 1 % highly cited researcher since 2017 according to Web of Science. Dr. Han is also the winner of the 2021 IEEE Kiyo Tomiyasu Award, for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: “for contributions to game theory and distributed management of autonomous communication networks.”