# Intelligent Content Pre-caching Scheme for Platoon-based Edge Vehicular Networks

Yu Wu, Xuming Fang, *Senior Member, IEEE,* Chunbo Luo, *Member, IEEE,* Geyong Min, *Member, IEEE*

***Abstract*—To provide various onboard entertainment services, the ever-increased Internet contents to be exchanged among remote data centers, roadside units (RSUs), and vehicles demand reliable and fast content dissemination in the vehicular networks. Edge pre-caching technology is expected to provide flexible and low-latency content dissemination by allowing edge nodes (i.e., RSUs and vehicles) to pre-cache contents. However, the content dissemination process of edge pre-caching still suffers from high mobility and highly dynamic topology of vehicular networks. The recently proposed platoon-based vehicular network has potentials to mitigate the mobility challenges, but need to deal with multi-hop wireless content dissemination's latency and reliability issues. Additionally, the network resources are limited in edge nodes, whereas various onboard Internet services with different Quality-of-Service (QoS) requirements share the same resource pool by the same network resource scheduling policy, thereby decaying the network performance. Based on the above observations, to cope with the challenging content pre-caching problem under diverse QoS requirements in a platoon-based edge vehicular network, we firstly abstract two isolated virtual content service slices with different QoS requirements based on network slicing technology to provide on-demand customized services. Then, we propose an intelligent deep reinforcement learning (DRL)-based content pre-caching scheme, which optimally matches the available communication resources and limited caching capacities in the edge vehicular network. The scheme jointly considers the impacts of content pre-caching policy and multi-hop wireless transmission on the content pre-caching performance. Simulation results show that our proposed DRL-based content pre-caching scheme achieves a competitive performance of reliability and latency comparing with other state-of-the-art algorithms.**

***Index Terms*—Platoon-based Vehicular Networks, Mobile Edge Caching, Content Pre-caching, Network Slicing, Deep Reinforcement Learning**

## I. Introduction

Many applications and services for current and future vehicular networks are data intensive (e.g., popular series), reliability-sensitive (e.g., financial services), and latency-sensitive (e.g., mobile games) [1]. As a result, the ever-increased Internet contents to be exchanged among vehicles, roadside units (RSU), and remote data centers require reliable and fast content dissemination. However, in rush hours or traffic jams, direct content dissemination from remote data centers in real-time may lead to unprecedented pressure (e.g., severe traffic load, interference, and congestion) on the fronthaul and backhaul links, and may also induce much latency. Regardless of the type of vehicular services, popular contents are likely to be repeatedly requested by different vehicles and dominate mobile data traffic [2]. Owing to the fact that these popular content requests are predictable, we can pre-cache these popular contents (e.g., video streams and music) at network edge nodes (e.g., RSUs and vehicles) during off-peak hours and then provide them during peak traffic hours [1], [3], [4], which is known as edge caching. For those highly concurrent content requests, edge caching nodes can timely retrieve out these contents saved in the edge and transmit them to the requesting vehicle via vehicle-to-vehicle/vehicle-to-RSU (V2V/V2R) communications. Such an edge pre-cached solution can relieve the traffic loads over backhaul links, and reduce network congestion and communication latency [5].

However, the high mobility of vehicles and the highly dynamic topology of vehicular networks will lead to poor link quality, unstable and intermittent connectivity, and even frequent link failure issues of the V2V/V2R communications, thereby impairing the content dissemination instantaneity and reliability of edge pre-caching [6]. Caching multiple content duplications [7] and mobility/trajectory predictions [8], [9] will help to deal with the above issues. Caching multiple content duplications can make more chances for mobile vehicles to access the contents, however, the redundant caching and disseminations will decrease resource utilization and cache efficiency. The contact frequency/probability and contact time duration obtained from mobility predictions are highly related to the content serving probability of pre-cached contents and are very valuable to guide the caching decisions. In addition to these two important methods, the existing paradigm platoon-based driving pattern of intelligent transportation system, proposed for improving road capacity and energy efficiency, has a unique advantage to mitigate the mobility issues by decreasing the relative velocities between vehicles of V2V communications and simplifying V2R communications.

Generally, a vehicle platoon is a group of autonomous vehicles that follow the same path and maintain a common mobility pattern, typically, same speed alignment and a fixed inter-vehicle distance in the order of 10m [10]–[12]. To maintain the platoon-based driving pattern, based on locally sensed/wirelessly collected vehicles' kinematics data (e.g., position, speed, acceleration, and steering information), driving actions of platoon members (PMs) are regulated by a

Yu Wu and Xuming Fang are with the Key Lab of Information Coding and Transmission, Southwest Jiaotong University, Chengdu, Sichuan, 611756, P.R. China.(email: linda_swjtu@126.com, xmfang@swjtu.edu.cn).

Chunbo Luo and Geyong Min are with the Department of Computer Science, College of Engineering, Mathematics, and Physical Sciences, University of Exeter, Exeter, EX4 4QF, U.K. (e-mail:C.Luo@exeter.ac.uk, g.min@exeter.ac.uk).

This work was supported in part by the NSFC under Grant 62071393, NSFC High-Speed Rail Joint Foundation under Grant U1834210, Sichuan Provincial Applied Basic Research Project under Grant 2020YJ0218, Fundamental Research Funds for the Central Universities under Grant 2682021CF019, and China Scholarship Council under 201907000092.

Corresponding author: Xuming Fang (email: xmfang@swjtu.edu.cn).

fully automated control system which installed in the in-front vehicle of the platoon (i.e., platoon leader, PL). Cooperative Adaptive Cruise Control is used for lateral (i.e., steering) and longitudinal (i.e., acceleration and braking) controls, thereby maintaining a desired platoon size, vehicles' velocity, inter-vehicle or inter-platoon distance [13], [14]. When the vehicle's speed changes, there may be platooning maneuvers [10], [15](i.e., re-form, join/ leave, merge/split) introducing disturbance. However, there have been existing studies on platoon management protocols and strategies [16], [17] and simulation results [10], [18] of platoon's ability to adapt to velocities' disturbance and maintain platoon stability.

The traffic flow distribution is reshaped from individual driving pattern to platoon-based driving pattern with unified system parameters for all platoon vehicles. Moreover, traditional V2V and V2R communications are transferred to intra/inter-platoon and platoon-to-RSU communications, in which cooperative communication/caching applications can be implemented to significantly improve the vehicular network performance [11]. For instance, considering a well-formed and stable platoon rather than individual vehicle, platoon's features of small relative speed and relatively constant inter-vehicle distance make the intra-platoon V2V communications a relatively static and stable scenario, avoiding link instability during content dissemination. In addition, from the RSU's perspective, the platoon dynamics are transparent to the RSU and it only observes different associated platoons. Only the PLs are responsible for connecting RSU to provide Internet access for other PMs. In high-mobility and dense scenarios, it is better than all vehicles suffering from the intermittent V2R connections and the severe fronthaul burden caused by the signaling storm of group access and frequent handovers [6], [19].

Although, the above advantages of the edge pre-caching solution and the platoon-based topology can make contents close to the consuming place and mitigate the mobility issue, various onboard Internet services with different Quality-of-Service (QoS) requirements demand different network resource configurations. The performance of the content pre-caching depends on the subsequent corresponding content dissemination process, and the content placement of the pre-caching decision in turn plays a key role in this process. For example, the reliability-sensitive service prefers network resources with good wireless channel condition (e.g., congestion or interference) to decrease signal to bit error ratio (BER) and packet loss ratio, large coverage to support mobility, and large buffer to decrease packet discard ratio. The latency-sensitive service prefers large bandwidth, nearby server, and multiple wireless transmission links. However, these different services share the same resource pool by the same network resource scheduling policy, thereby decaying the network performance. Thanks to software-defined-networking (SDN) and network function virtualization (NFV), network slicing enables the coexistence of several virtual network slices with diverse QoS requirements on top of the same physical infrastructure, shifting "one size fit all" to "one size per service" by providing isolated slices with independent protocol stack splitting and resources allocation policy.

To reach the full potential of edge pre-caching in such a network slicing-enabled platoon-based vehicular network, we need to carefully deal with the following interplays of edge caching, platooning, and network slicing, and propose an intelligent and effective resource management algorithm [20], [21]. Firstly, the interplay of edge caching and platooning raises the multi-hop transmission latency and reliability issues. Unlike most edge caching policies in individual driving pattern just considering 1-hop wireless transmission, after pre-caching popular contents offline in the platoon-based edge vehicular network, the contents will be retrieved via multi-hop wireless transmissions with multi-hop latency and reliability. Secondly, the interplay of edge caching and network slicing introduces the challenge of matching constrained and integrated cross-domain multi-dimensional resources and slices' traffic demand to guarantee resource efficiency for all slices. It should be noticed that in such communication and caching integrated systems, due to different resource constraints of RSUs and vehicles, both communication resources and caching resources may become the network's bottleneck. The caching utilization will decrease due to either low transmission data rate or the cache overflow [22]. Finally, to adapt to the varying nature of the network and satisfy QoS requirements of different slices, we need to intelligently allocate multi-dimensional resources to each slice, in which the performance isolation for QoS is important. Service-oriented optimal resource allocation of virtual networks is known to be an NP-Hard problem. Traditional model-based optimization or queueing theoretic modeling becomes intractable [20].

In this paper, based on the NFV, SDN, and network slicing, we firstly abstract different isolated virtual content service slices (VCSSs) to provide on-demand customized services, and model the service satisfaction utility function of the content pre-caching optimization policy for each VCSS with its specific QoS requirement. Then, to deal with complex multi-dimentional data (e.g., various nodes with different mobility patterns, network resources, and service requirements) and efficiently solve the optimization problem, we prefer a deep reinforcement learning (DRL)-based method rather than traditional mathematical programming methods (e.g., combinatorial or mixed-integer nonlinear programming, graph theory). Compared with the traditional mathematical programming methods, the DRL-based method can interact with the environment, use its multi-dimentional data and then provide foresight pre-caching policy [23], [24]. With deep learning for value function approximation, DRL does not need to model the environment dynamics [25]. Once it is well trained, a DRL agent can perform appropriate control action and pursue the predefined objective [26], [27]. The main contributions of this paper are as follows.

1) Based on the programmable control principle originated from NFV and SDN, and caching theory originated from mobile edge pre-caching, we propose an integrated framework for the platoon-based edge vehicular network. The framework can enable the dynamic orchestration of networking, caching, and communicating resources to improve the content pre-caching performance

of various services.

2) Based on network slicing technology, two VCSSs, which are reliability-constrained content slice (RCCS) and latency-costrained content slice (LCCS), are constructed by an SDN controller. RCCS offers reliability-sensitive content services while LCCS provides latency-sensitive content services, and both share the same communication and caching resources. We formulate the content pre-caching policy to a joint optimization problem for these two VCSSs with the objectives to maximise the reliability and minimize the latency, respectively.
3) To solve the above joint optimization problem, deep Q-Networks (DQN), a DRL-based content pre-caching algorithm with a DNN as Q action-value function approximator, is used to obtain the optimal pre-caching decisions which optimally match with available communication resources under dynamic environment conditions. With the experience reply mechanism, the mini-batch method, and the target Q-network, DQN can prevent local minimum by decreasing the dependence of the samples from the memory of the experience replay, and making the algorithm more stable.

The remainder of this paper is organized as follows. Related work is reviewed in Section II. In Section III, we present the framework of the platoon-based edge vehicular network and describe the details of the system model. Then we formulate the content pre-caching optimization problems of two different VCSSs. We propose the DRL-based content pre-caching algorithm in Section IV. We present the performance evaluation and results analysis in Section V. Finally, Section VI concludes this paper.

## II. Related Work

### A. Mobile Edge Caching

Recent efforts have been made to study the edge caching. The edge caching policy can be resource reservation-based approach or resource share-based approach [28]. Dedicated resources can be reserved in advance to guarantee slice isolation and adapt to dynamic traffic demand in the reservation-based approach. The resource reservation focuses on how to maximize the profit between the revenue from users and the resource reservation cost from the economic aspects, and how to balance the reserved resource utilization (i.e., minimizing the influences of both over and under reservation) and QoS satisfaction of network slices [29]–[31]. Two timescale resource reservation approach including long timescale inter-slices resource reservation and short timescale intra-slice resource allocation are investigated [32]. In addition, all the resources can be reserved to become guaranteed resources [33], [34], or keep an amount of resources unreserved and sharable on a best-effort basis [20], [35]. Usually, guaranteed resources are available in a large timescale (e.g., slice lifetime) and the unreserved auxiliary resources are for short timescale on-demand request [35]. However, there is a reservation-allocation-utilization process (RAUP) in the above reservation-based approach with following issues: 1) To ensure slice isolation, the reserved storage space of a slice cannot be used by any other slices, thereby existing over/under reservation problems. 2) The ability of adaption to varying user demands has a limitation. Only if the allocated resource is less than the reserved resource, it can adapt to the sudden increase of the user demand of the slice. 3) The RAUP may involve different levels/timescales of resource reservation/allocation which is time consuming and redundant. Hence, different from the above reservation-based approach, we consider a share-based approach in which an intelligent DRL-based algorithm directly making pre-caching decisions at one-step without RAUP, thereby achieving high-efficiency.

Edge caching also can be in an online way or in an offline way. Online caching policy is real-time content caching and dissemination only taking place after a user requesting a content. This method is very sensitive to the real-time network status and may bring high latency, low reliability, and high traffic loads over fronthaul/backhaul links due to limited communication/caching resources from edge networks to core networks, burst traffic, and the instable wireless links caused by vehicles' mobilities [36]. Offline pre-caching policy caches popular contents in advance of a user request at edge networks (e.g., BSs/RSUs/vehicles) in a proactive way during off-peak hours [36]. Since edge nodes can timely transmit contents to requesting vehicles, this policy is suitable for wireless networks with temporal-spatial varying network utilizations and can deal with highly concurrent content requests. However, offline proactive pre-caching relies on the accurate prediction of user traffic demands, content popularity, and user mobility [37]. Hence, there are many researches focus on mobility prediction (e.g., by Markov renewal process [8], Markov chain [38], LSTM [9]) and user demands/content popularity prediction (e.g., recommendation and push-based [37]). Based on these researches, our work focuses on an offline content pre-caching policy. Different from mobility prediction-based method, we explore the potential of platoon-based vehicular networks to overcome the high-mobility issues.

### B. Vehicular Edge Caching

Y. He *et al.*, in [39]–[45] studied the joint communication, caching and computing optimization with objective of maximizing the system profit from an economic point of view. M. Li, *et al.*, formulated the joint optimization of the networking, caching, and computing resources as a POMDP to make offloading decision and the selection of caching and computing node with the aim of minimizing the network cost and computation time [46]. S. Zhang *et al.*, proposed an air-ground integrated vehicular network slicing framework with multi-dimensional heterogeneous resources, in which high-altitude platforms (HAPs) proactively push contents to vehicles through large-area broadcast, while the ground RSUs provide high-rate unicast services on demand. Their main purpose was to minimize RSU transmission rate with delay requirements by investigating the tradeoff among RSU transmission rate, HAP broadcast rate, and vehicles' caching capabilities [22]. Y. Zhang *et al.*, proposed an online vehicular caching based on a two-dimensional Markov process of the interactions between caching vehicles and mobile users

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3178099, IEEE Internet of Things Journal



to optimize network energy efficiency [47]. L. Hou *et al.*, proposed an optimal caching strategy utilizing heuristic Q-learning solution together with LSTM-based mobility prediction to minimize the latency of caching services [9]. S. Zhang *et al.*, proposed a cache-assisted lazy update and delivery scheme to balance content freshness and service latency in vehicular networks [42]. J. Ma *et al.*, proposed a content placement strategy which jointly considered the caching at the vehicular layer and RSU layer to minimize the average latency [48]. Y. Hui *et al.*, proposed a 2-hops relay-based content dissemination scheme which applies first-price sealed-bid auction [49].

However, there are obvious differences between the existing caching methods of vehicular network and the proposed pre-caching method of platoon-based vehicular network. Firstly, most of the existing methods aim to maximize the system profit from an economic point of view [39]–[41], [45], improving energy efficiency [47], or only minimizing latency [9], [48], [49], whereas our purpose is to maximize the QoS satisfactions of transmission latency and reliability for different virtual service slices. Secondly, some of the methods focus on separated content caching [49] and content disseminations [50], i.e., relay selection on V2V networks, and donot exploit the joint influences of content placement and disseminations. Thirdly, others of them study the content caching with single-hop content retrieval from neighbor vehicles/RSUs/BSs without exploring the joint impacts of content placement, channel condition, and transmission bandwidth in the multi-hop retrieval scenario. Whether considering the effects of wireless transmission hops determined by the content pre-caching strategy on the content retrieval latency and reliability makes our objective and constraints of the optimization problem quite different from the existing works. Finally, there are some cluster-based edge caching researches on user-centric networks [51]–[53], where BSs/RSUs form a cluster and serve for one user/vehicle, not the vehicles form a cluster/platoon. Hence, the existing caching methods of vehicular network cannot be directly applied to our slice-enabled platoon-based vehicular network.

### C. Edge Caching and Network Slicing

There have been extensive works on resource allocation with/without network slicing, including separated or joint communication, computation, and caching (i.e., 1C, 2C or 3C) resource allocation. The scenarios, network requirements, challenges, and corresponding solutions are all very different with their own research purposes. For example, the resource managements include communication resource allocation [41], [54], [55], caching policy, joint communication and caching resources allocation [55], [56], and joint communication, computing and caching resources management [39], [40], [57]. There are also researches on slice admission control, device association, computing offloading, joint communication and computing resources management which are not highly related to our work. Furthermore, these important issues are investigated in various scenarios (e.g., Core networks [58], RAN [57], [59]–[61], IoV [39], [40], [45], [54]–[56], [62]) with different selected optimization objective and constraints, such as network profit [39], [43], [45], cache hit ratio [60], service latency [55]–[57], [63], [64], transmission rate [60], [64], energy efficiency [63], [65], spectrum efficiency [61]. Most of existing solutions formulate the 2C/3C resources allocation as the convex optimization problem [64] or constrained (mix-)integer (non-)linear Programming nonconvex optimization problem [56], [61], which is NP-hard. These optimization problems are solved by the alternating direction method of multipliers (ADMM) [55], [64], Lyapunov optimization method [56], matching theory [55], and Deep Learning/RL/DRL-based algorithm [39], [40], [45], [54], [57], [60], [61], [63], [64].

The differences between our work and the above existing works are as follows. Firstly, they rarely investigated the reliability performance of joint communication and caching resources management. The reliability performance can be measured by accumulated BER, handoff failed probability, packet loss or discard ratio caused by multi-hop transmission, bad channel condition (e.g., handoff and interference) or network congestion. Secondly, most of existing works focused on the channel allocation and transmitting power allocation and did not explore much on the communication aspects of both transmission hops and bandwidth. Thirdly, for those mobile edge caching and communication joint optimization without network slicing, they gave the same resource scheduling strategy that optimized the same objective for all kinds of services, which was not scalable. Fourthly, the optimization objective of some existing works may include more than one metric to support various QoS requirements of different services but the utility was in the form of the sum of weighted metrics with constant weight set, which was not scalable. Finally, they chose different DRL algorithms for different reasons, and their definitions of the state space, action space, and reward function were also quite different for their own purposes.

## III. SYSTEM MODEL

In this section, we propose a network slicing framework for the platoon-based edge vehicular network and illustrate the details of the network model, content requesting, pre-caching, retrieval process, wireless propagation model, and utility model of the sliced edge vehicular network.

### A. Network Slicing Framework

According to [66], we propose a network slicing framework for the platoon-based edge vehicular network as shown in Fig.1. The network is divided into the core networks and mobile edge networks. The core network refers to a central SDN controller with network control and management functions and its global database storing original contents and historical information backups of the whole network. The mobile edge network includes RSUs and platoons, which are equipped with storages to provide edge caching function. In the edge vehicular network, we consider a set of RSUs $\mathbf{B} = \{1, ..., b, ..., B\}$ along the highway road segment, and a set of linear platoons $\mathbf{K_b} = \{1, ..., k, ..., K_b\}$ associates with RSU $b$. RSUs are connected via ideal optical connections,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3178099, IEEE Internet of Things Journal



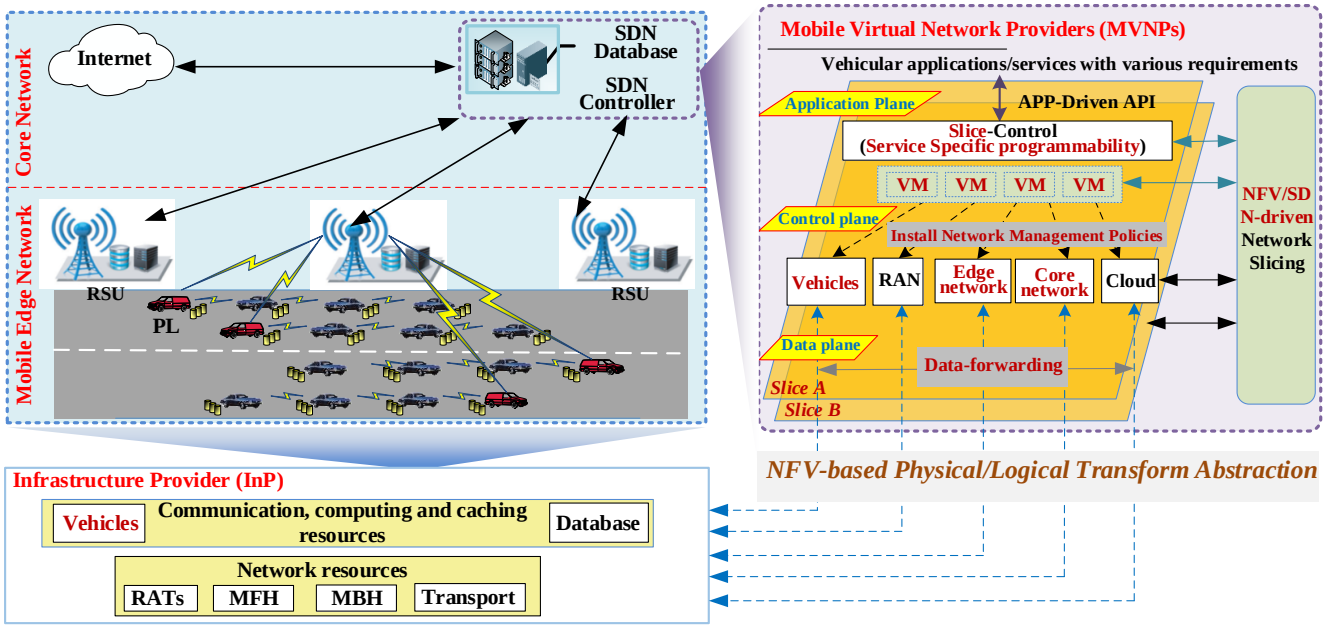


Fig. 1. Network Slicing Framework for Platoon-based Edge Vehicular Networks

and each platoon $k$ has the same length and contains a set of vehicles $\mathbf{U_{b,k}} = \{1, ..., u, ..., U_{b,k}\}$. The caching capacity of RSUs and vehicles are $C_b$ and $C_u$, respectively, and the caching capacity is divided into equal-sized caching slots.

We further assume that the entire physical infrastructures and network resources (i.e., database, RSUs, vehicles, fronthaul/backhaul networks, and their caching, computing and communicating capabilities) are owned by a single infrastructure provider (InP). NFV virtualizes the network functions of these underlying physical infrastructures and network resources to logical and programmable network functions (e.g., virtual storages and networks) in the form of virtual machines (VMs), and this process is known as the NFV-based physical/logical transform abstraction as shown in Fig.1. Supported by NFV, OpenFlow-based SDN splits the application plane, the control plane, and the data plane. The data plane is composed of switches (e.g., RSUs and vehicles), which cache/forward data according to relevant policies. Network functions (e.g., resource and mobility management) run in the VMs of the application plane. Through the SDN Northbound Interface, the application plane obtains network resources from or sends data to the lower layer. The control plane periodically updates new network policies on the switches of the data plane through the SDN Southbound Interface and maintains a global database of the data plane and application plane. Then based on the network slicing technology, the SDN controller constructs VCSSs (i.e., mobile virtual network providers, MVNPs), each of which supports a specific type of content service with its guaranteed QoS (e.g., transmission reliability or latency) [39], [43], [67]–[69]. The detailed principle, technology and practice of SDN, NFV, and network slicing can refer to [70], [71]. The set of VCSSs is defined as $\mathbf{V} = \{1, ..., v, ..., V\}$, each slice $v$ has its own unique set of content segments $\mathbf{O_v} = \{1, ...o_v, ..., O_v\}$. The size of a content segment is represented by $O_s$ and is equal to a caching slot [64]. Particularly, we consider two VCSSs which are RCSS and LCSS in this paper.

Specifically, the main roles of the central SDN controller are as follows:

1) Map the physical network resources to virtual logical resources via the technologies of SDN and NFV.
2) Determine and adjust network management and slicing policies according to the data processing and analyzing results of the global database and install these policies to switches and local controllers (i.e., RSUs).
3) Cooperate with local controllers to construct VCSSs on-the-top of the virtual network resources [57].
4) Allocate isolated virtual resources to each VCSS based on its QoS requirement.

To cooperate with the central SDN controller for resource virtualization and network slicing, there is a local controller placed at the RSU. It collects and monitors local information (e.g., available communicating and caching resources, channel condition, content requests, and positions and mobilities of its associated platoons) of the edge vehicular networks and sends data reports to the SDN database for future data processing, analysis, and policy adjustments.

### B. Content Requesting, Pre-caching, Retrieval Process

The whole content requesting, pre-caching, and retrieval process is described as follows:

*1) Online content requesting phase:* During this process, vehicles initiate content requests to the PL. Then the PL maintains a requesting queue and executes *online content retrieval phase*. Meanwhile, the PL records these historical content requests together with other local information (e.g.,

available communicating and caching resources, channel condition, and platoon's position and mobility) to form a content request profile (CRP) and then uploads it to its associated RSU. The RSU collects these local historical CRPs from all its associated platoons and then uploads them to the SDN database and controller to predict future CRPs and network status. We define the set of vehicles requesting contents from the slice $v$ as $\mathbf{U_v} = \{u^v_{b,k} | \forall b \in \mathbf{B}, \forall k \in \mathbf{K_b}, \forall u \in \mathbf{U_{b,k}}\} = \{1, ..., u_v, ..., U_v\}, v \in \mathbf{V}$, where $u^v_{b,k}$ represents the status of vehicle $u$ of platoon $k$ requesting contents from slice $v$. We can predict the upcoming content request and its popularity from the data processing and analysis of historical CRPs recorded in the SDN global database. The content request arrival process is assumed to follow a Poisson process with an average rate of $\mu$ (requests/s). The content popularity is assumed to be a Zipf-like distribution with a parameter $\alpha$. Hence, the predicted popularity of the $o$-th popular content is $p_o = \frac{1}{o^\alpha \sum_{c=1}^{O} \frac{1}{c^\alpha}}$. We assume that the average arrival rate of requests and predicted content popularities are recorded in the predicted CRP, and keep static in a relatively long period [72]. Hence, we have a content pre-caching task for each predicted content request.

*2) Offline content pre-caching phase:* During this process, according to the predicted CRPs and network status, local controllers in RSUs make pre-caching decisions according to the installed content pre-caching policy. Through fronthaul and backhaul connections with the backbone, the RSU and its associated platoon pre-cache forthcoming popular contents during off-peak traffic hours. For any pre-caching task, the pre-caching policy only involves vehicles of the same platoon $u' \in \mathbf{U_{b,k}}$ and the platoon's associated RSU $b$, that is to say other RSUs have no effect on the pre-caching task's decision of the vehicle $u_{b,k}$. The pre-caching policy is controlled by two binary parameters, $Z_b$ and $Z_{k,u}$. $Z_b = 1$ indicates that content is cached in RSU $b$, and 0 otherwise. $Z_{k,u} = 1$ indicates that content is cached in vehicle $u$ of platoon $k$, and 0 otherwise.

We apply the Last Recently Used content replacement mechanism for RSUs' or vehicles' storages. Besides, since vehicles in the same platoon can communicate via V2V wireless communications, we abstract all vehicles' caching capacities of a platoon into a logically centralized caching unit which is managed by the PL. Compared with using each vehicle's caching capacity independently in a distributed way, this can improve the caching utilization for its fewer content duplications, more content diversity, and less frequent content removal [73]. The impact of the multi-hop V2V wireless communications on the final content pre-caching performance (i.e., reliability and latency) will be explored later.

*3) Online content retrieval phase:* During this phase, a vehicle can retrieve the pre-cached contents via V2R or V2V communications in the edge vehicular networks [74]. It should be noted that the edge vehicular network only allows content retrieval from RSU or within the platoon, that is to say there is no content retrieval between platoons, thereby avoiding operation on unstable V2V links of inter-platoon communications caused by platoons' relative mobility. The content retrieval phase includes the following stages:

*a) Wireless Intra-Platoon Content Retrieval, WIPCR:* If the vehicle's requested content is pre-cached in the platoon, it will be retrieved from vehicles in the same platoon via V2V communications.

*b) Wireless Cellular Content Retrieval, WCCR:* If WIPCR fails, the PL tries to retrieve the desired content from the associated RSU via V2R communications.

If both WIPCR and WCCR fail, in other words, if the content is not pre-cached in the edge vehicular networks, it is needed to originally retrieve the desired content from the remote database to the requesting vehicle. According to whether the content is pre-cached and where the content is pre-cached (i.e., RSU, or vehicles within the platoon), the whole content retrieval phase may be a combination of the two stages. Hence, the online content retrieval phase will experience different QoS performance due to different offline pre-caching decisions.

### C. Wireless Propagation Model

*1) Sub-6GHz wireless propagation model for WCCR stage:* In the WCCR stage, the stable communication duration (SCD) is calculated, that is, the time duration that the platoon stays within the coverage of RSU. SCD can be obtained by the positions of the platoon and the RSU, the coverage of RSU, and the platoon's constant speed, which are collected by local controllers at RSUs [74]. Within the SCD, there is no V2R link failure caused by the relative mobility between the platoon and the RSU. For the platoon's handover between RSUs, the neighbor RSUs are connected via ideal wired connections, and pre-cached contents can be transmitted among RSUs before handover happens. In this paper, we focus on the content pre-caching policy rather than the handover management. Each RSU has a total bandwidth $W_b$ (Hz) in sub-6GHz (e.g., 2.4GHz) for V2R communications. Let $Y_{b,k} \in [0, 1]$ be the percentage of bandwidth RSU $b$ allocated to the platoon $k$, and then $W^c_{b,k} = Y_{b,k} W_b$ represents the bandwidth of platoon $k$ for its V2R communications during SCD. Due to platoons driving in a highway road segment and we assume that the spectrum allocation of inter-RSUs and intra-RSU are all independent and orthogonal, the inter-RSUs and inter-platoon interference can be ignored. Then, the effective WCCR data rate of platoon $k$, which the requesting vehicle $u$ belongs to, can be expressed as $R^c_{b,k}$.

$$R^c_{b,k} = W^c_{b,k} log_2(1 + SNR_{b,k}) \tag{1}$$

$$SNR_{b,k} = \frac{P_{b,k} G_{b,k}}{P_N} \tag{2}$$

where $SNR_{b,k}$ is the signal-to-noise-ratio. $P_{b,k}$ is the transmission power in mW. $G_{b,k}$ is the channel gain. $P_N$ is the thermal noise power in mW. And

$$P_N[mW] = W^c_{b,k}[Hz] * 10^{\frac{N_0[dBm/Hz]}{10}} \tag{3}$$

$$G_{b,k} = 1/L_{b,k}[dB] \tag{4}$$

where $N_0$ is the background noise power spectrum density, as a standard temperature of 17 °C, the thermal noise level is $N_0 = -174[dBm/Hz]$ [74]. $\eta$ is log-norm shadow fading with mean zero and standard deviation $\sigma = 7dB$. $L_{b,k}[dB]$ is the path loss and can be obtained by

$$L_{b,k}[dB] = 20log_{10}(d) + 20log_{10}(F) + 32.4 \tag{5}$$

where $d$ is the distance between the receiver and transmitter in $km$. $F$ is the frequency band in $MHz$ [64].

*2) mmWave wireless propagation model for WIPCR stage:* In the WIPCR stage, each RSU allocates independent and orthogonal mmWave spectrums (e.g., 60GHz) to their associated platoons for their intra-platoon V2V communications. PL controls WIPCR in the platoon, and the spectrum is time-division multiplexed within the platoon. In the relatively static V2V communication scenario, front and rear vehicles have line-of-sight path and use mmWave directional narrow beam for transmission. Hence, the inter/intra-platoon interferences can be ignored. The bandwidth that RSU $b$ allocated to its platoon $k$ for WIPCR is denoted by $W^p_{b,k}$. Then, the effective WIPCR data rate can be expressed as $R^p_{b,k}$

$$R^p_{b,k} = W^p_{b,k} log_2(1 + SNR_{uu'}) \tag{6}$$

where the $SNR_{uu'}$ is the $SNR$ between vehicles.

We assume that mmWave V2V links use directional beams with antenna directional transmit gain and receive gain $G_x, x \in \{t, r\}$ [74],

$$G_x = \begin{cases} \frac{2\pi-(2\pi-\varphi_x)g_s}{\varphi_x}, & |\theta_x| \le \frac{\varphi_x}{2} \\ g_s, & otherwise \end{cases} \tag{7}$$

where $\theta_x$ is the beam offset angle to the mainlobe. $\varphi_x$ represents half-power beamwidth of the transmit beam or the receive beam. $0 < g_s \ll 1$ is the sidelobe gain.

We can have $SNR_{uu'}$ by

$$SNR_{uu'} = \frac{P_{uu'} G_{uu'} G_t G_r}{P'_N} \tag{8}$$

where $P_{uu'}$ is the transmission power in mW. $G_t$ and $G_r$ are transmit antenna gain and receive antenna gain, respectively. $G_{uu'}$ is the channel gain and $P'_N$ is the thermal noise power in mW, they can be obtained by Eq. (3)-(5). For mmWave frequency, it should be noted that $d$ and $F$ in Eq. (5) are in m and in GHz, respectively.

It should be noted that the physical topology of platoon vehicles may not be linear, but it can be abstracted into a logical linear topology. The linear network topology represents how platoon vehicles exchange information and not their positions on the road. Different platoon's physical topologies will affect the communication interferences, whereas the platoons length will affect the number of V2V transmission hops experienced in the content retrieval process. The spectrum allocations are all independent and orthogonal in the V2R communications, and intra-platoon V2V communications use mmWave directional narrow beam in a time-division multiplexed way, thereby the inter/intra-RSU and inter/intra-platoon interferences can be ignored. This is because the aim of this paper is the optimal pre-caching policy rather than the interference management. Hence, the linear platoon topology is typical and the length of platoon can be different.

### *D. Utility Models of the Two VCSSs*

In this section, for a pre-caching task of content $o$ in the offline pre-caching phase, the pre-caching performance is represented by the online content retrieval satisfaction (i.e., reliability or latency satisfaction) for vehicle $u$ to retrieve content $o$ from RCSS or LCSS. The pre-caching performance is related to the content popularity and content serving probability of the caching aspect and the transmission hops and transmission bandwidth of the communication aspect. The content serving probability $p_s$ is defined as the probability of the pre-cached contents responding for content requests. Except for the content popularity, other factors are all determined by the content pre-caching decision. In this paper, we focus on the edge pre-caching performance and thus we omit the case of original content retrieval from remote database when formulating the utility models.

*1) Reliability Satisfaction of the RCCS:* We map the uncertain wireless propagation environment of each hop as BER (i.e., $0 \le P_{v2r}, P_{v2v} \ll 1$ for WCCR phase and WIPCR phase, respectively), which can be inferred by the SNR. Since the relevant mobility between platoon vehicles is approximately 0, while relevant mobility between vehicle and RSU is much higher. $P_{v2v}$ of intra-platoon V2V links in the WIPCR phase are same, and $P_{v2r}$ of V2R links in the WCCR phase is larger than $P_{v2v}$.

If the requested content is pre-cached within the platoon (i.e. $Z_{k,u} = 1$), we formulate the reliability as $P_{cor}$ which is calculated as

$$\begin{aligned} P_{cor} &= (1 - P_{v2v})^{h_{v2v}} \\ &= (1 - P_{v2v})(1 - P_{v2v})(1 - P_{v2v}) \dots \\ &= (1 - 2P_{v2v} + \underbrace{P^2_{v2v}}_{P_{v2v} \ll 1, P^2_{v2v} \cong 0})(1 - P_{v2v}) \dots \\ &\cong (1 - 2P_{v2v})(1 - P_{v2v})(1 - P_{v2v}) \dots \\ &\cong (1 - 3P_{v2v} + \underbrace{2P^2_{v2v}}_{P_{v2v} \ll 1, P^2_{v2v} \cong 0}) \dots \\ &\cong 1 - h_{v2v} P_{v2v} \end{aligned} \tag{9}$$

where $h_{v2v}$ is the number of V2V hops from the vehicle who pre-cached the requested content (i.e., replying vehicle) to the requesting vehicle.

If the desired content is pre-cached in the associated RSU (i.e. $Z_b = 1$), $P_{cor}$ can be calculated as

$$\begin{aligned} P_{cor} &= (1 - P_{v2r})(1 - P_{v2v})^{h_{v2v}} \\ &= (1 - P_{v2r})(1 - P_{v2v})(1 - P_{v2v}) \dots \\ &= (1 - P_{v2r} - P_{v2v} + \underbrace{P_{v2r} P_{v2v}}_{\cong 0})(1 - P_{v2v}) \dots \\ &\cong (1 - P_{v2r} - P_{v2v})(1 - P_{v2v}) \dots ] \\ &\cong (1 - P_{v2r} - 2P_{v2v} + \underbrace{P_{v2r} P_{v2v} + P^2_{v2v}}_{\cong 0}) \dots \\ &\cong (1 - P_{v2r} - 2P_{v2v}) \dots \\ &\cong 1 - P_{v2r} - h_{v2v} P_{v2v} \end{aligned} \tag{10}$$

However, improving $P_{cor}$ of non-popular contents does not contribute a lot to the system reliability performance while wasting caching resources. Moreover, contents prefer to be pre-cached in a certain place to improve their serving

probabilities $p_s$ for upcoming requests. Hence, the multi-hop reliability utility function is weighted by $p_o$ and $p_s$ to demonstrate their influences on reliability performance:

$$P_{rel} = p_o p_s P_{cor} \tag{11}$$

Then, if we consider a pre-caching task of predicted request of content $o$ from vehicle $u$, after the offline pre-caching phase, the online content retrieval reliability satisfaction of vehicle $u$ retrieving the pre-cached content $o$ from RCSS (i.e., slice $v_1$) is defined as $Sat_{u,o}^{v_1}$.

$$Sat_{u,o}^{v_1} = \xi(P_{rel}) = \frac{1}{1+\exp(-\kappa\left(\mathrm{P_{rel}} - \tau_{\mathrm{rel}}\right))} \tag{12}$$

where $\kappa$ is the steepness constant, $\tau_{rel}$ is the reliability satisfaction requirement. Here, we use the sigmoid function to normalize the values of $P_{rel}$, thereby making them range from 0 to 1 with the average around the reliability satisfaction requirement $\tau_{rel}$.

*2) Latency Satisfaction of the LCCS:* According to the two content retrieval stages, we divide the content retrieval latency into two parts: (a) *WCCR latency* $d_{b,k} = \frac{O_s}{R_{uk}^c}$, which is the time needed to retrieve the content from the associated RSU to the PL, and (b) *WIPCR latency* $d_{k,u} = h_{v2v}\frac{O_s}{R_{uk}^p}$, which is the time needed to retrieve the content from the replying vehicle to the requesting vehicle by multi-hop V2V communications.

With considering the influences of $p_o$ and $p_s$, the multi-hop latency utility function is formally calculated as:

$$d_u = \begin{cases} d_{k,u} p_o p_s, & if\ Z_{k,u} = 1 \\ (d_{b,k} + d_{k,u}) p_o p_s, & if\ Z_b = 1 \end{cases} \tag{13}$$

As such, the latency satisfaction of vehicle $u$ retrieving content $o$ from LCSS (i.e., slice $v_2$) is

$$Sat_{u,o}^{v_2} = \xi(d_u) = \frac{1}{1+\exp(-\kappa(\mathrm{q_d} - \mathrm{d_u}))} \tag{14}$$

where $q_d$ is the latency satisfaction requirement.

Then, to maximize the satisfaction of each slice, we formulate the content pre-caching optimization problem as follows:

$$\begin{aligned} max \quad & Sat^v \\ & = \frac{1}{V \times U_v \times O_v} \sum_{v\in\{v_1,v_2\}} \sum_{u\in\mathbf{U_v}} \sum_{o\in\mathbf{O_v}} Sat_{u,o}^v \end{aligned} \tag{15}$$

$$s.t. \quad P_{rel} \geq \tau_{rel} \tag{15a}$$

$$d_o \leq q_d \tag{15b}$$

$$\sum_{k\in K} W_{b,k}^c \leq W_b, \forall b \tag{15c}$$

where $Sat^v$ is the overall satisfaction utility function.

## IV. DRL-BASED PRE-CACHING SCHEME FOR MULTI-SLICED EDGE VEHICULAR NETWORKS

Usually, the above joint optimization problem is solved by traditional mathematical programming methods. Nonetheless, these methods may suffer from the following issues [23], [24], [75]:

1) The increasing diversity and complexity of resources and service requirements make it challenging to model the problem and balance the performance even with many imperfect assumptions that some key information factors are given.
2) It is difficult to adapt to the highly complex dynamic environment of edge vehicular caching and communicating.
3) Most mathematical programming methods are non-convex and NP-hard problems. There is no efficient algorithm to solve them with polynomial time complexity and being executed in real-time.
4) Except for Lyapunov optimization, most of them are built-in one-shot optimization, so they do not apply well for long-term performance.

Hence, in this paper we model the above joint optimization problem as a Markov decision process (MDP), which is a powerful dynamic optimization theory to obtain the optimal resource control policy in terms of the long-term average performance. In MDP, a decision maker or agent can interact with the environment by making a sequence of actions to optimize a predefined system performance criterion. However, MDP needs prior knowledge of the environment model (i.e., state transition probability and immediate reward). It also suffers from the curse-of-dimensionality problem that the MDP model's state space and the computation complexity increase exponentially with the growing number of vehicles. Therefore, we use deep Q-Networks (DQN), a DRL-based method, to solve the optimization problem with its good generality and scalability. DQN does not need the priori knowledge of the environment model. It chooses the action according to the current observed environment and the samples of the system states and rewards from the experience replay policy [76].

Without considering the interference management and handover management, the scenario of multiple RSUs and multiple platoons is simplified into many independent scenarios of 1 RSU and multiple platoons. Based on the optimization problem, we define the state, action and reward of DQN as follows:

### A. System State Space

System state $\mathbf{S}$ is a finite state space consisting a set of parameters that can describe the environment. According to the predicted CRP and the network status, for the upcoming requesting content $o \in \mathbf{O_v}$ recorded in the predicted CRP, we define the system state as $s_o (s_o \in \mathbf{S})$:

$$s_o = \{v, u, k, b, p_o, W_{b,k}^c, \mathbf{W_{b,k}^P}, C_b^a, \mathbf{C_{u_k}^a}\} \tag{16}$$

where $v$, $u$, $k$ and $b$ are the index of the VCSS, the requesting vehicle, its platoon, and its associated RSU, respectively. $p_o$ is the requested content's popularity. $W_{b,k}^c$ and $\mathbf{W_{b,k}^P}$ are the bandwidths of RSU $b$ allocated to the requesting platoon $k (k \in \mathbf{K})$ for its V2R and intra-platoon V2V communications. $C_b^a$ and $\mathbf{C_{u_k}^a}$ are the available caching capacities of RSU $b$, and the vehicle $u_k (u_k \in \mathbf{U_k})$ of the requesting platoon $k$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3178099, IEEE Internet of Things Journal



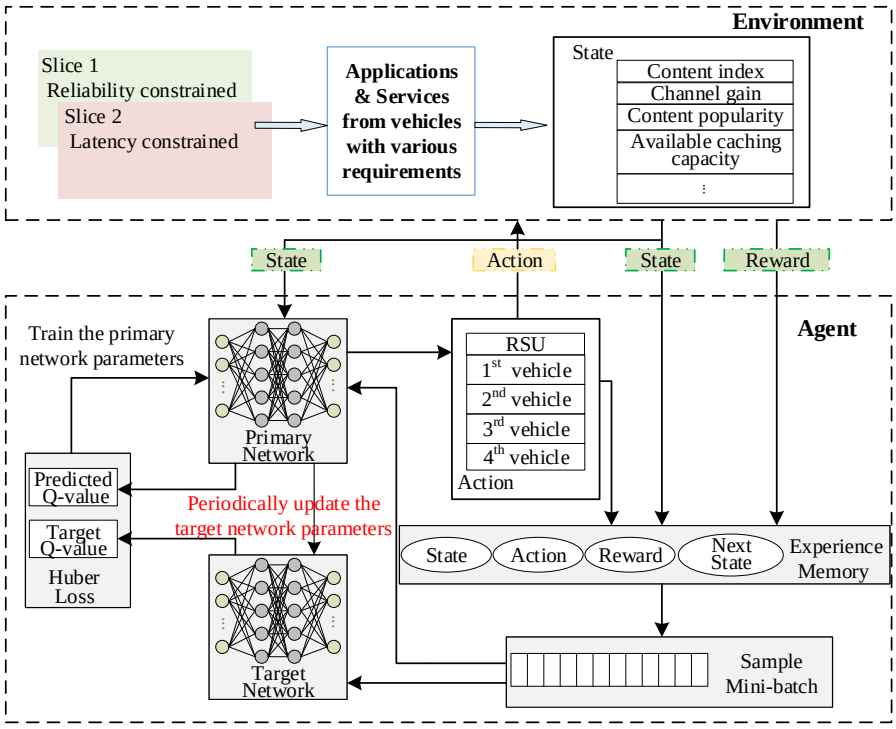


Fig. 2. An illustration of the Deep Q-Network based optimization framework

### B. System Action Space

System action $\mathbf{A}$ is a finite action space from which the agent makes a content pre-caching action $a_o (a_o \in \mathbf{A})$ under current system state $s_o$, and it is defined as:

$$a_o = \{a_{k,b}, a_{k,1}, ..., a_{k,u_k}\} \tag{17}$$

where $a_{k,b} \in \{0,1\}$ and $a_{k,b} = 1$ denotes that the content is pre-cached in the requesting platoon's associated RSU $b$, and 0 otherwise. $a_{k,u_k} \in \{0,1\}$ and $a_{k,u_k} = 1$ represents pre-caching the content locally on the vehicle $u_k$ of the requesting platoon $k$, and 0 otherwise.

### C. Reward Function

Reward $\mathbf{R}$ is an immediate reward matrix that gives immediate reward signals $r_o(s_o, a_o)$ or $r_o$, $r_o \in \mathbf{R}$ for short, to indicate which action is good in an immediate sense. After the agent taking a pre-caching action $a_o$ for the upcoming requesting content $o$ under the environment state $s_o$, the content will be retrieved by the requesting vehicle, and the agent will obtain an immediate reward $r_o$ as defined in the utility functions, which are the reliability satisfaction $Sat_{u,o}^{v_1}$ and latency satisfaction $Sat_{u,o}^{v_2}$.

### D. Algorithm Design

The goal of DQN is to find an optimal pre-caching place $a_o = \pi(s_o)$ for the upcoming requesting content, so as to maximize the long-term reward. Unlike immediate reward signals, the long-term return is defined as a value function in the form of a prediction of the expected, accumulative, discounted, future reward. There are two kinds of value functions which are state-value function $v(s_o)$ and action-value function (i.e., Q-value) $Q(s_o, a_o)$ measuring how good each state, or state-action pair is. DQN uses a DNN with parameters $\boldsymbol{\theta}$ to approximate the mapping from the current system state $s_o$ to the value function $Q(s_o, a_o|\boldsymbol{\theta})$ of all possible actions. In state $s_o$, DQN chooses the action $a_o$ by $\epsilon$-greedy policy with $\epsilon$ decreasing linearly. The $\epsilon$-greedy policy means that the agent chooses the action with the largest Q-value $Q(s_o, a_o|\boldsymbol{\theta})$ with a probability of $(1-\epsilon)$, and equally chooses the other actions with a probability of $\epsilon$ [58]. Then, the action is executed with returning a reward $r_o$ and the next state $s'$. DQN uses experience reply to store the agent's experience $(s_o, a_o, r_o, s')$ at each time step in a reply memory. Then it uses a random sampled mini-batch of transitions $(s_o, a_o, r_o, s')$ from the replay memory to train DNN. We use gradient descent approach to update the parameters $\boldsymbol{\theta}$ of DNN which minimize the Huber Loss between the current predicted Q-value $Q(s_o, a_o|\boldsymbol{\theta})$ and target Q-value $r_o + \gamma \max_{a'} Q(s', a')$, where $\gamma$ is a discount factor reflecting the present value of future rewards. The Huber Loss function is defined as:

$$L_\delta(y', y) = \begin{cases} \frac{1}{2}(y'-y)^2 & \text{for } |y'-y| \le \delta \\ \delta|y'-y| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \tag{18}$$

where $y'$ is the target Q-value and $y$ is the current predicted Q-value. In this way, The predicted Q-value directly approximates to the optimal Q-value $Q^\star$. Experience reply and mini-batch can prevent local minimum by decreasing the dependence of the collected experiences.

Furthermore, DQN uses a second neural network to calculate the target Q-value with independent parameters $\boldsymbol{\theta'}$ that are only updated periodically (i.e., every $T_u$ time steps) instead

of every single time step to reduce the correlations between the predicted Q-value and the target Q-value, and thus the network becomes more stable. The modified target Q-value is $r_o + \gamma \max_{a'} \hat{Q}(s', a'|\boldsymbol{\theta'})$. With experience replay and fixed Q target network, DQN can learn more from the past experience and improve its stability and convergence. Fig. 2 illustrates how our proposed DQN works, and we give the pseudo code of our DRL-based content pre-caching algorithm in Algorithm 1.

**Algorithm 1** :DRL-based content pre-caching algorithm

Initialize the experience replay buffer $\boldsymbol{M}$ of size $S_M$; the mini-batch B of size $S_B < S_M$; a primary DQN and a target DQN with two sets of parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta'} = \boldsymbol{\theta}$ .
  // Initialize the replay memory $\boldsymbol{M}$ by random policy
**if** the replay memory is not Full **then**
  The agent observes the state $s_o$ and then randomly selects a pre-caching action $a_o(s_o)$;
  Apply the action $a_o(s_o)$ to the environment to obtain the reward $r_o(s_o, a_o)$ and observe the next state $s'$;
  Store the experience tuple $(s_o, a_o, r_o, s')$ in the experience replay buffer $\boldsymbol{M}$;
**end if**
**for** all available episodes **do**
  **for** all time steps **do**
    The agent observes a state $s_o$ and then picks a pre-caching action $a_o(s_o)$ by the $\epsilon$-greedy policy;
    Execute the action $a_o(s_o)$ to the environment to obtain the reward $r_o(s_o, a_o)$ and observe the next state $s'$;
    Store the experience tuple $(s_o, a_o, r_o, s')$ in the experience replay buffer $\boldsymbol{M}$ to update it;
      // Updates the DQNs' parameters
    Randomly sample mini-batch samples $\boldsymbol{B}$ from memory $\boldsymbol{M}$, i.e., sampling $(s_o, a_o, r_o, s')$ from memory $\boldsymbol{M}$;
    **for** all $B$ samples $(s_o, a_o, r_o, s')$ in the mini-batch transitions **do**
      **if** the current episode ends at time step $i+1$ **then**
        set the target Q-value $y' = r_o$
      **else**
        set $y' = r(s_o, a_o) + \gamma \max_{a'} \hat{Q}(s', a'|\boldsymbol{\theta'})$;
      **end if**
      Conduct gradient descent with Huber Loss to update parameters $\boldsymbol{\theta}$ of primary network;
    **end for**
    The agent regularly reset $\boldsymbol{\theta'}(i+1) = \boldsymbol{\theta}(i)$ every $T_u$ time steps, and otherwise $\boldsymbol{\theta'}(i+1) = \boldsymbol{\theta'}(i)$;
    Update scheduling time step index by $i = i + 1$;
  **end for**
**end for**

### *E. Complexity Analysis*

The complexity of our proposed algorithm mainly contains two parts, the one is the complexity of Q-network to predict the Q-values, and the other is the complexity of training the Q-network. In the simulation, we assume that there is only one content pre-caching task in a time slot. For each task, the agent predicts all the Q-values in terms of content pre-caching actions according to the system state. In our proposed algorithm, the Q-network predictor contains two hidden layers with the number of neurons $L_1$ and $L_2$, respectively. From Eq. (16), the dimension of state space is $(7 + |\boldsymbol{K_b}| + |\boldsymbol{U_{b,k}}|)$. After obtaining the platoon index $k$ of a received pre-caching task from the observed state, from Eq. (17), the dimension of action space is $(1 + |\boldsymbol{U_{b,k}}|)$. Then, the complexity of Q-value generation is $O\left((7 + |\boldsymbol{K_b}| + |\boldsymbol{U_{b,k}}|) L_1 + L_1 L_2 + L_2 (1 + |\boldsymbol{U_{b,k}}|)\right)$. Since the size of hidden layers in our system is fixed, the complexity of Q-value generation can be given as $O(8 + |\boldsymbol{K_b}| + 2|\boldsymbol{U_{b,k}}|) = O(N), N = \max\{8, |\boldsymbol{K_b}|, 2|\boldsymbol{U_{b,k}}|\}$. In the training process of Q-network, there are $T$ time slots in a period, then the Q-network will be trained in $T$ time steps. In each time step, the complexity is similar to the Q-value predictor. Finally, in a period, the complexity of our proposed algorithm is $O(NT)$.

It should be noted that our major innovation/contribution is not on the DRL-based algorithm itself, but on problem formulation of multiple slices' content pre-caching optimization in the platoon-based edge vehicular network and its transformation to a standard DRL problem for solution through the corresponding definitions for the state, action and reward in Eqs. (16-17).

## V. SIMULATION AND ANALYSIS

We adopt the DRL-based algorithm to solve the optimization problem Eq. (15) and evaluate the content dissemination performances of RCCS and LCCS in the platoon-based edge vehicular network. The experiment parameters are shown in Table I. Unless explicitly stated otherwise, these simulation parameters are used to obtain all the results. According to [77], the edge caching scheme can be classified into three types, namely, infrastructure supported, user device/vehicle sharing enabled, and a hybrid type. Although there are no pre-caching schemes devised for the considered platoon-based vehicular networks, there are hybrid edge caching schemes that investigating the cooperation caching of individual vehicles (e.g., fog caching) and RSUs (i.e., edge caching). The main difference of our paper is that we consider platoons instead of individual vehicles. In addition, since the performance improvement of our work is the result of the pre-caching optimization policy rather than the improvement of DRL algorithm itself, it seems not vital to compare our method with other DRL algorithms. Therefore, in this paper, to demonstrate the effectiveness of our proposed content pre-caching policy, we compare the pre-caching performance of the proposed DRL-based method with four state-of-the-art methods.

1) **Platoon-based RSU Supported Method (PRS Method)** [78], [79]: The method always chooses RSU as the pre-caching places for all contents.
2) **Platoon-based Vehicle Sharing Method (PVS Method)** [80], [81]: The method randomly chooses platoon vehicles as the pre-caching places for all contents.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3178099, IEEE Internet of Things Journal



3) **Platoon-based Random Hybrid Method (PRH Method):** The method pre-caches content either in the RSU or in platoon vehicles in a hybrid way, and chooses pre-caching places for contents randomly.
4) **Individual-driving Optimized Hybrid Method (IOH Method)** [82], [83]: The method optimally chooses either RSU or individual vehicles in a hybrid way as the pre-caching places for contents to maximize the objective performance.

In addition, for comparison, we define that the Platoon-based Optimized Hybrid Method (POH Method) is the proposed method in which contents are pre-cached either in the RSU or platoon vehicles in a hybrid way, and chooses optimal pre-caching places for contents to maximize our objective performance. The POH-Eval Method is the evaluation process after the DQN model is well trained.

TABLE I
SIMULATION PARAMETERS

| Symbol | Value |
|---|---|
| $\|\boldsymbol{V}\|$ | 2 |
| $\|K\|$ | 4 |
| $\|\boldsymbol{U}_k\|$ | 4 |
| $C_b$, slots | 100 |
| $C_{uk}$, slots | 10 |
| $O_s$, Mb | 1 |
| $\mu$ | 10 |
| $\alpha$ | 1.5 |
| $f_c$ for V2R, GHz | 2.4 |
| $f_c$ for V2V, GHz | 60 |
| $W_b$, MHz | 20 |
| $Y_{kb}$ | [0.1,0.2,0.3,0.4] |
| $W_p^k$, GHz | 2.16 |
| $P_b$, mW | 31.62 |
| $P_{uu'}$, mW | 5 |
| $\varphi_x, x \in \{t, r\}$ | $5^\circ$ |
| $P_{v2r}$ | 0.0022 |
| $P_{v2v}$ | 0.0011 |
| $p_s^u$ | 4 |
| $p_s^b$ | [2,12] |
| $\kappa$ | 2 |
| **Parameters for DNN** | |
| Huber loss, $\delta$ | 1 |
| Learning rate, $\alpha_{lr}$ | 0.00025 |
| Memory size, $S_M$ | 5000 |
| Batch size, $S_B$ | 64 |
| Discount factor, $\gamma$ | 0.99 |
| $\epsilon_{\max}$ | 1 |
| $\epsilon_{\min}$ | 0.01 |
| Decay speed of epsilon, $\lambda$ | 0.001 |
| Target NN's parameters updating period, $T_u$ | 100 time steps |

### *A. The Reliability Satisfaction of the RCCS*

We mainly study the impacts of transmission hops, transmission bandwidth, content popularity, and content serving probability on the reliability satisfaction. It should be noted that the content serving probability $p_s$ can be affected by the vehicles/platoons mobilities or the temporal-spatial traffic densities and can be obtained/inferred by mobility/trajectory predictions of real trace experiments or simulators (i.e., One Simulator [84]). However, its time-varying value should not affect the effectiveness of our proposed optimization model and the DRL-based algorithm. Hence, the content serving probability of the platoon vehicles and the RSU are set to different values in Fig.3 to validate their different influences.

Fig.3 shows the accumulative reward varying with the number of content requests, in which the content serving probability of RSU is higher than that of platoon vehicles in the first 5000 content requests. We can see that the reliability satisfactions of the PVS Method and the PRS Method are mainly influenced by their different transmission hops and content serving probabilities. The PVS Method pre-caches all contents in the platoon vehicles, so it has fewer transmission hops than the PRS Method in most cases. However, due to RSU's content serving probability is higher than the platoon vehicles' in the first 5000 content requests, from Eq. (11), in this case, we can know that the content serving probability plays a more significant role than the transmission hops on the reliability satisfaction. Hence, the PRS Method's reliability satisfaction is better than the PVS Method's. In the next 5000 content requests, the content serving probability of RSU gradually decreases to be lower than that of platoon vehicles. We can see that the accumulative reliability satisfaction of PRS Method also decreases to be lower than that of the PVS Method gradually, and the performance of PRH Method is always between them for the reason that the caching decisions of PRH Method are combinations of caching decisions of PVS Method and PRS Method. The reliability satisfactions of the PRS Method and the PVS Method vary according to different content serving probability settings. However, no matter how the parameters are set, the proposed POH Method always have the best reliability satisfaction adaptive to the content serving probability.

In Fig.3, the reliability satisfaction of the IOH Method is the worst for the following reasons: 1) There is higher relative speed of V2V communications in the individual driving pattern than the platoon-based driving pattern. Furthermore, the V2V communications in the individual driving is based on the Carrier Sense Multiple Access/Collision Detect (CSMA/CD) rather than centralized control by PL in platoon-based driving pattern. Hence, the channel condition (e.g., collision, congestion, interference) of V2V communications in the individual driving pattern is worse than that in the platoon-based driving pattern. Hence, the SINR and BER is higher, thereby the reliability satisfaction decreases. 2) The higher relative speed of V2V communications in the individual driving pattern results in shorter SCD. The content sharing between vehicles may fail if the SCD is shorter than the content transmitting time, thereby decreasing the content serving probability and the reliability satisfaction. 3) The storages of the individual driving vehicles are used independently in a distributed way. The storage is much smaller than the wirelessly connected and centralized managed storages of the platoon vehicles, which is abstracted as a logically centralized caching unit. As a result, IOH has more content duplications, less content diversity, more frequent content removal than the platoon-based methods. The pre-cached contents may be removed before serving for any content requests, thereby decreasing the content serving probability and the reliability satisfaction.

Fig. 4 shows the impact of content popularity on the reliability satisfaction when the content serving probability of RSU

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3178099, IEEE Internet of Things Journal



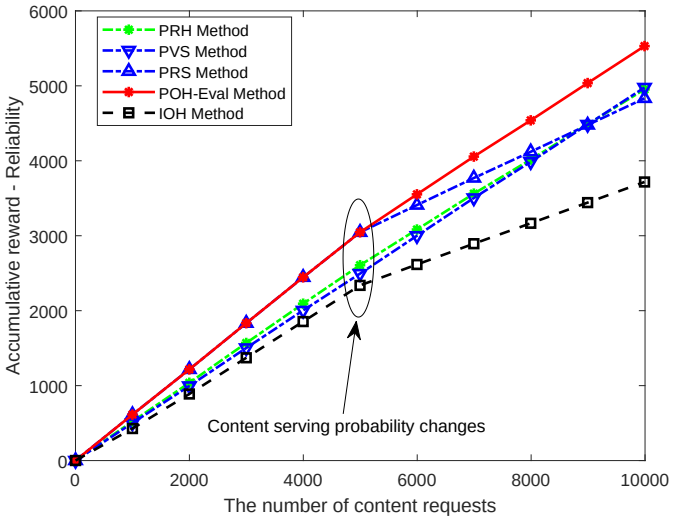


Fig. 3. Accumulative reward (reliability satisfaction utility) with the number of content requests.

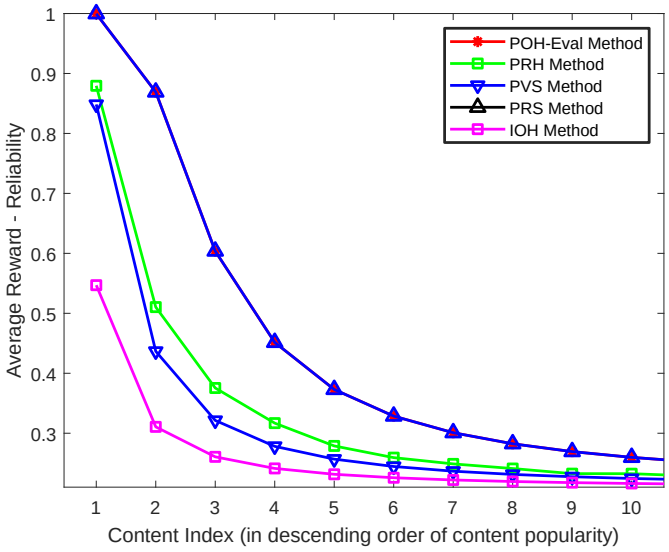


Fig. 4. Average reward (reliability satisfaction utility) with the content popularity.

is higher than that of platoon vehicles. The average reward of all five methods decreases with the decreasing content popularity. Because contents with higher popularity would be requested more frequently than those with lower popularity, we consider the content popularity in the reliability utility function (i.e., Eq. 11) to improve the reliability satisfaction of the contents with high content popularity. Meantime, for contents with certain fixed popularity, we can have the same conclusions with Fig.3. POH-Eval Method converges to the PRS Method, which has the best performance due to its high content serving probability setting, and both of them consistently outperforms PRH Method. As we have analyzed before, besides their different inherent transmission hops, PRS Method and PVS Method's reliability satisfactions mainly depend on their different content serving probability. Hence, compared with these two methods with unstable reliability satisfaction relying on parameter setting, the POH-Eval Method always has a competitive reliability satisfaction.

We further reveal the reason why POH-Eval Method outperforms PRH Method in Fig.5, in which the content serving probability of RSU is higher than that of platoon vehicles. In both Fig.5 (a) and (b), we can see that caching contents in RSU can obtain a higher average reward than caching them in platoon vehicles. Consistent with this, the majority of pre-caching decisions of POH-Eval Method is to store contents in the RSU. On the contrary, PRH Method caches most contents in platoon vehicles which have smaller average reward.

We also study the latency performance of RCCS in Fig.6. Among all five methods, PRS Method has the largest latency and PVS Method has the smallest latency. This is because compared with PVS Method, PRS Method experiences more transmission hops and it uses smaller bandwidth in sub-6GHz, thereby suffering from lower content retrieval rate and large latency. The caching decisions of POH-Eval Method converges to the PRS Method and to be the worst ones. It is because that on RCCS, POH-Eval Method's optimization objective is reliability satisfaction rather than latency satisfaction. PRH are combinations of caching decisions of PRS Method and PVS Method. Hence, its latency performance is between PRS Method and PVS Method.

### *B. The Latency Satisfaction of the LCCS*

We evaluate the average latency with different content popularities in Fig.7. It is shown that PRS Method has the largest latency while PVS Method has the smallest latency. This is because of the influences of their different transmission hops and transmission bandwidths, as we analyzed before in Fig.6. The POH-Eval Method's latency approaches the PVS Method's and to be the best ones. We have the PRH Method with its latency performance in the middle of the PVS Method and PRS Method. IOH Method of Fig.7 has shorter latency than that of Fig.6, for the reason that Fig.7 is to optimize LCCS while Fig.6 is to optimize RCCS. The POH-Eval Method is always better than the PRH with shorter latency. That is because, in this LCCS, the optimization objective of POH-Eval Method is always the latency satisfaction. In conclusion, among all five methods, the POH-Eval Method can simultaneously allocate resources for independent virtual slices with different QoS requirements according to independent policies, thereby achieving the best slice's performance satisfaction.

### *C. The Overall Satisfaction of RCCS and LCCS*

Finally, we give the overall accumulative reward defined in Eq. (17), in which the content serving probability of RSU is higher than that of platoon vehicles. In Fig. 8, the POH-Eval Method converges to the PRS Method after well trained. As analyzed before, the performance of PVS Method and PRS Method highly relies on their content serving probability settings. However, the POH-Eval Method always has the best satisfactions of reliability and latency, and its overall accumulative reward also comes first among the five methods.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3178099, IEEE Internet of Things Journal



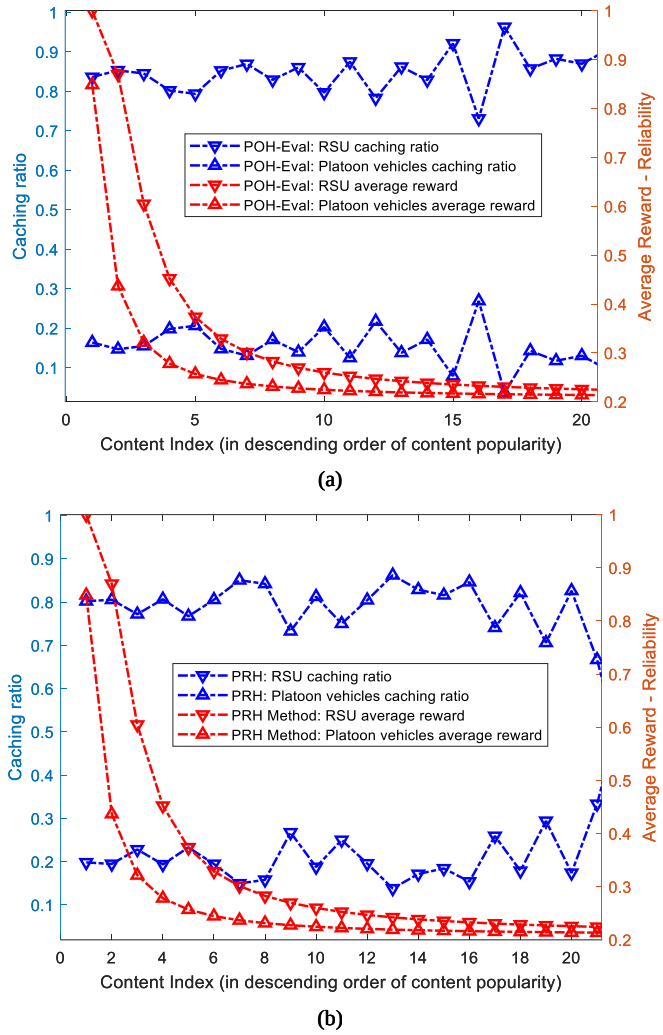


Fig. 5. Caching ratio and average reward with the content popularity. (a) POH-Eval Method; (b) PRH Method.

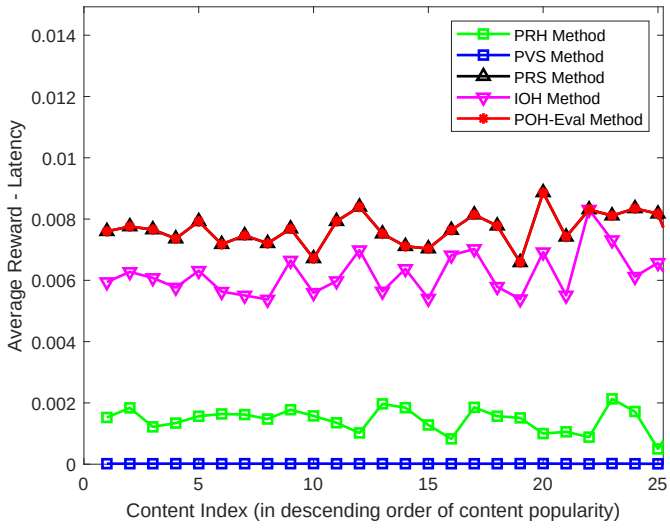


Fig. 6. Latency performance of reliability-constrained slices

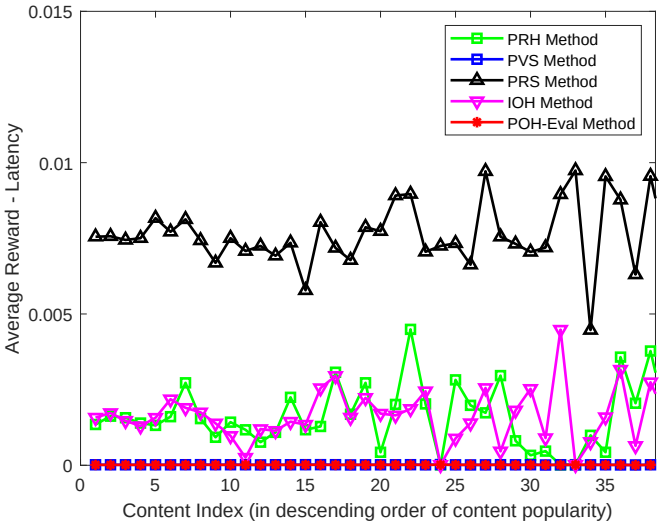


Fig. 7. Average latency (s) with content popularity.

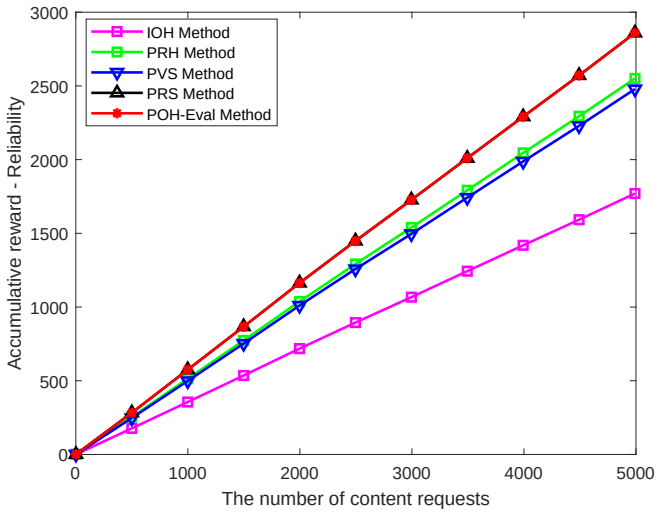


Fig. 8. Accumulative reward.

## VI. Conclusions

In this paper, a platoon-based edge vehicular network is proposed to deal with the mobility challenges in mobile edge caching and communication. To support various vehicular applications with different QoS requirements, we abstract two virtual content service slices (i.e., reliability-constrained content service slice and latency-constrained content service slice) by the flexible programable function of network resource management of SDN and NFV. Then, we study the edge pre-caching optimization problem to achieve reliable and fast content disseminations in the proposed platoon-based edge vehicular network. Since the pre-cached contents are retrieved via multi-hop wireless transmissions with multi-hop latency and reliability issues, the performance of pre-caching policy depends on the optimal matching the caching resource allocation of offline pre-caching phase and the communication resource allocation of the online content requesting phase. Hence, we propose a DRL-based content pre-caching scheme for the two virtual content service slices. Comparing with other state-of-the-art algorithms, we jointly consider the unique advantages of platooning to mitigate mobility challenges, impacts of transmission and bandwidth of the communication

aspect, and the impacts of content popularity and serving probability of the caching aspect on the content pre-caching performance, leading to stable and competitive reliability and latency satisfaction.

## REFERENCES


[1] Sadeghi, F. Sheikholeslami and G. B. Giannakis, "Optimal and Scalable Caching for 5G Using Reinforcement Learning of Space-Time Popularities", *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 180-190, Feb. 2018.

[2] J. Zhang and K. B. Letaief, "Mobile Edge Intelligence and Computing for the Internet of Vehicles", *Proc. IEEE* , vol. 108, no. 2, pp. 246-261, Feb. 2020.

[3] Amadeo, C. Campolo and A. Molinaro, "Information-centric networking for connected vehicles: A survey and future perspectives", *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 98-104, Feb. 2016.

[4] G. Paschos, E. Bastug, I. Land, G. Caire and M. Debbah, "Wireless caching: Technical misconceptions and business barriers", *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 16-22, Aug. 2016.

[5] Y. Hao, Y. Miao, L. Hu, M. S. Hossain, G. Muhammad and S. U. Amin, "Smart-Edge-CoCaCo: AI-Enabled Smart Edge with Joint Computation, Caching, and Communication in Heterogeneous IoT", *IEEE Netw.*, vol. 33, no. 2, pp. 58-64, March/April 2019.

[6] Yawei Zhao, Yu Wu, Yaxiong Feng, Yuxin Zheng and Xuming Fang, "Dynamic channel selections and performance analysis for High-Speed Train WiFi network", *Proc. IEEE Conf. HMWC*, 2015, pp. 31-35.

[7] J. Sahoo, M. A. Salahuddin, R. Glitho, H. Elbiaze and W. Ajib, "A Survey on Replica Server Placement Algorithms for Content Delivery Networks", *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1002-1026, Secondquarter 2017.

[8] Y. Ye, M. Xiao, Z. Zhang and Z. Ma, "Performance analysis of mobility prediction based proactive wireless caching", *Proc. IEEE Conf. Wireless Commun. and Netw. (WCNC 2018)*, pp. 1-6..

[9] L. Hou, L. Lei, K. Zheng and X. Wang, "A Q-Learning-Based Proactive Caching Strategy for Non-Safety Related Services in Vehicular Networks", *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4512-4520, June 2019.

[10] U. Montanaro, S. Fallah, M. Dianati, D. Oxtoby, T. Mizutani and A. Mouzakitis, "On a Fully Self-Organizing Vehicle Platooning Supported by Cloud Computing", *Proc. Int. Conf. Internet Things: Syst., Manag. and Security*, 2018, pp. 295-302.

[11] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, A survey on platoon-based vehicular cyber-physical systems, *IEEE Commun. Surveys Tuts.*, vol.18, no. 1, pp. 263C284, 2016.

[12] T. Robinson, E. Chan, and E. Coelingh, An introduction to the SARTRE platooning programme, *Proc. 17th World Congr. Intell. Transp. Syst.*, 2010, pp. 1C11

[13] C. Campolo, A. Molinaro, G. Araniti and A. O. Berthet, Better Platooning Control Toward Autonomous Driving : An LTE Device-to-Device Communications Strategy That Meets Ultralow Latency Requirements, *IEEE Veh. Technol. Mag.*, vol. 12, no. 1, pp. 30-38, March 2017

[14] T. Renzler, M. Stolz and D. Watzenig, "Decentralized Dynamic Platooning Architecture with V2V Communication Tested in Omnet++", *Proc. IEEE Int. Conf. Connected Veh. and Expo (ICCVE 2019)*, pp. 1-6.

[15] S. Santini, A. Salvi, A. S. Valente, A. Pescap, M. Segata and R. L. Cigno, "Platooning Maneuvers in Vehicular Networks: A Distributed and Consensus-Based Approach", *IEEE Trans. on Intell. Veh.*, vol. 4, no. 1, pp. 59-72, March 2019.

[16] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal, Platoon management with cooperative adaptive cruise control enabled by vanet, *Veh. Commun.*, vol. 2, no. 2, pp. 110C123, 2015.

[17] P. Liu, A. Kurt, and U. Ozguner, Distributed model predictive control for cooperative and flexible vehicle platooning, *IEEE Trans. Control Syst. Technol.*, no. 99, pp. 1C14, 2018.

[18] A. Choudhury et al., "An integrated V2X simulator with applications in vehicle platooning", *Proc. IEEE Int. Conf. Intell. Transport. Syst. (ITSC 2016)*, pp. 1017-1022.

[19] Y. Wu, Y. Zheng, Y. Feng, Y. Zhao and X. Fang, "End-to-End Performance Optimization of Tandem Queuing for High-Speed Train Networks", *Proc. IEEE Conf. Veh. Technol. (VTC 2016-Spring)*, pp. 1-5.

[20] J. Koo, V. B. Mendiratta, M. R. Rahman and A. Walid, "Deep Reinforcement Learning for Network Slicing with Heterogeneous Resource Requirements and Time Varying Traffic Dynamics", *Proc. Int. Conf. Netw.and Service Manag. (CNSM 2019)*, pp. 1-5.

[21] G. Sun, K. Xiong, G. O. Boateng, D. Ayepah-Mensah, G. Liu and W. Jiang, "Autonomous Resource Provisioning and Resource Customization for Mixed Traffics in Virtualized Radio Access Network", *IEEE Syst. J.*, vol. 13, no. 3, pp. 2454-2465, Sept. 2019.

[22] S. Zhang, W. Quan, J. Li, W. Shi, P. Yang and X. Shen, "Air-Ground Integrated Vehicular Network Slicing With Content Pushing and Caching", *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2114-2127, Sept. 2018.

[23] Q. Qi et al., "Knowledge-Driven Service Offloading Decision for Vehicular Edge Computing: A Deep Reinforcement Learning Approach", *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4192-4203, May 2019.

[24] L. Liang, H. Ye, G. Yu and G. Y. Li, "Deep-Learning-Based Wireless Resource Allocation With Application to Vehicular Networks", *Proc. IEEE*, vol. 108, no. 2, pp. 341-356, Feb. 2020.

[25] J. Wang, J. Hu, G. Min, W. Zhan, Q. Ni and N. Georgalas, "Computation Offloading in Multi-Access Edge Computing Using a Deep Sequential Model Based on Reinforcement Learning", *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 64-69, May 2019.

[26] D. Zeng, L. Gu, S. Pan, J. Cai and S. Guo, "Resource Management at the Network Edge: A Deep Reinforcement Learning Approach", *IEEE Netw.*, vol. 33, no. 3, pp. 26-33, May/June 2019.

[27] K. Arulkumaran, M. P. Deisenroth, M. Brundage and A. A. Bharath, "Deep Reinforcement Learning: A Brief Survey",*IEEE Signal Processing Mag.*, vol. 34, no. 6, pp. 26-38, Nov. 2017.

[28] A. Banchs, G. de Veciana, V. Sciancalepore and X. Costa-Perez, "Resource Allocation for Network Slicing in Mobile Networks", *IEEE Access*, vol. 8, pp. 214696-214706, 2020.

[29] J. Monteil, J. Hribar, P. Barnard, Y. Li and L. A. DaSilva, "Resource Reservation within Sliced 5G Networks: A Cost-Reduction Strategy for Service Providers", *Proc. IEEE Int. Conf. on Commun. (ICC 2020)*, pp. 1-6.

[30] H. Chien, Y. Lin, C. Lai and C. Wang, "End-to-End Slicing With Optimized Communication and Computing Resource Allocation in Multi-Tenant 5G Systems", *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2079-2091, Feb. 2020.

[31] D. Bega, M. Gramaglia, M. Fiore, A. Banchs and X. Costa-Prez, "DeepCog: Optimizing Resource Provisioning in Network Slicing With AI-Based Capacity Forecasting", *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 361-376, Feb. 2020.

[32] H. Zhang and V. W. S. Wong, "A Two-Timescale Approach for Network Slicing in C-RAN", *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6656-6669, June 2020.

[33] N. NADDEH, S. B. JEMAA, S. Eddine ELAYOUBI and T. CHAHED, "Proactive RAN Resource Reservation for URLLC Vehicular Slice", *Proc. IEEE Conf. Veh. Technol. (VTC 2021-Spring)*, pp. 1-5.

[34] J. Zhang, S. Chen, X. Wang and Y. Zhu, "DeepReserve: Dynamic Edge Server Reservation for Connected Vehicles with Deep Reinforcement Learning", *Proc. IEEE Conf. Computer Commun. (INFOCOM 2021)*, pp. 1-10.

[35] C. Sexton, N. Marchetti and L. A. DaSilva, "On Provisioning Slices and Overbooking Resources in Service Tailored Networks of the Future", *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2106-2119, Oct. 2020.

[36] S. O. Somuyiwa, D. Gndz and A. Gyorgy, "Reinforcement Learning for Proactive Caching of Contents with Different Demand Probabilities", *Proc. Int. Symposium on Wireless Commun. Syst. (ISWCS 2018)*, pp. 1-6.

[37] D. Liu and C. Yang, "A Deep Reinforcement Learning Approach to Proactive Content Pushing and Recommendation for Mobile Users", *IEEE Access*, vol. 7.

[38] J. Dai and D. Liu, "MAPCaching: A novel mobility aware proactive caching over C-RAN", *Proc. IEEE Int. Symposium on Personal, Indoor, and Mobile Radio Commun. (PIMRC 2017)*, pp. 1-6

[39] Y. He, N. Zhao and H. Yin, "Integrated Networking, Caching, and Computing for Connected Vehicles: A Deep Reinforcement Learning Approach", *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 44-55, Jan. 2018.

[40] L. T. Tan and R. Q. Hu, "Mobility-Aware Edge Caching and Computing in Vehicle Networks: A Deep Reinforcement Learning", *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10190-10203, Nov. 2018.

[41] Y. Dai, D. Xu, S. Maharjan, G. Qiao and Y. Zhang, "Artificial Intelligence Empowered Edge Computing and Caching for Internet of Vehicles", *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 12-18, June 2019.

[42] Z. Ning et al., "Partial Computation Offloading and Adaptive Task Scheduling for 5G-enabled Vehicular Networks", *IEEE Trans. Mobile Comput.*, doi: 10.1109/TMC.2020.3025116.

[43] X. Wang, Z. Ning, S. Guo and L. Wang, "Imitation Learning Enabled Task Scheduling for Online Vehicular Edge Computing", *IEEE Trans. Mobile Comput.*, doi: 10.1109/TMC.2020.3012509.

[44] Z. Ning et al., "5G-Enabled UAV-to-Community Offloading: Joint Trajectory Design and Task Scheduling", *IEEE J. Sel. Areas Commun.*, doi: 10.1109/JSAC.2021.3088663.

[45] Z. Ning et al., "Joint Computing and Caching in 5G-Envisioned Internet of Vehicles: A Deep Reinforcement Learning-Based Traffic Control System", *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5201-5212, Aug. 2021.

[46] M. Li, F. R. Yu, P. Si, H. Yao and Y. Zhang, "Software-Defined Vehicular Networks with Caching and Computing for Delay-Tolerant Data Traffic", *Proc. IEEE Int. Conf. on Commun. (ICC 2018)*, pp. 1-6.

[47] Y. Zhang, C. Li, T. H. Luan, Y. Fu, W. Shi and L. Zhu, "A Mobility-Aware Vehicular Caching Scheme in Content Centric Networks: Model and Optimization", *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3100-3112, April 2019.

[48] S. Zhang, J. Li, H. Luo, J. Gao, L. Zhao and X. S. Shen, "Towards Fresh and Low-Latency Content Delivery in Vehicular Networks: An Edge Caching Aspect", *Proc. Int. Conf. Wireless Commun. and Signal Process (WCSP 2018)*, pp. 1-6.

[49] J. Ma, J. Wang, G. Liu and P. Fan, "Low Latency Caching Placement Policy for Cloud-Based VANET with Both Vehicle Caches and RSU Caches", *Proc. IEEE Conf. Global Commun. (GLOBECOM 2017)*, pp. 1-6.

[50] Y. Hui, Z. Su, T. H. Luan and J. Cai, "Content in Motion: An Edge Computing Based Relay Scheme for Content Dissemination in Urban Vehicular Networks", *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 3115-3128, Aug. 2019.

[51] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao and X. Shen, "Cooperative Edge Caching in User-Centric Clustered Mobile Networks", *IEEE Trans. Mobile Comput.*, vol. 17, no. 8, pp. 1791-1805, 1 Aug. 2018.

[52] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao and X. Shen, "Traffic Steering Assisted Mobile Edge Caching: Exploiting Spatial Content Diversity Gain", *Proc. IEEE Conf. Global Commun. (GLOBECOM 2017)*, pp. 1-6.

[53] I. Keshavarzian, Z. Zeinalpour-Yazdi and A. Tadaion, "A clustered caching placement in heterogeneous small cell networks with user mobility", *Proc. IEEE Int. Symposium on Signal Process. and Inf. Technol. (ISSPIT 2015)*, pp. 421-426.

[54] Z. Mlika and S. Cherkaoui, "Network Slicing with MEC and Deep Reinforcement Learning for the Internet of Vehicles", *IEEE Netw.*, vol. 35, no. 3, pp. 132-138, May/June 2021.

[55] J. Zhao, X. Sun, Q. Li and X. Ma, "Edge Caching and Computation Management for Real-Time Internet of Vehicles: An Online and Distributed Approach", *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2183-2197, April 2021.

[56] Z. Ning et al., "Intelligent Edge Computing in Internet of Vehicles: A Joint Computation Offloading and Caching Solution", *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2212-2225, April 2021.

[57] Y. Wei, F. R. Yu, M. Song and Z. Han, "Joint Optimization of Caching, Computing, and Radio Resources for Fog-Enabled IoT Using Natural ActorCCritic Deep Reinforcement Learning", *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2061-2073, April 2019.

[58] R. Li et al., "Deep Reinforcement Learning for Resource Management in Network Slicing", in *IEEE Access*, vol. 6, pp. 74429-74441, 2018.

[59] H. Li et al., "Mobility-Aware Content Caching and User Association for Ultra-Dense Mobile Edge Computing Networks", *Proc. IEEE Conf. Global Commun. (GLOBECOM 2020)*, pp. 1-6.

[60] H. Xiang, S. Yan and M. Peng, "A Deep Reinforcement Learning Based Content Caching and Mode Selection for Slice Instances in Fog Radio Access Networks", *Proc. IEEE Conf. Veh. Technol. (VTC2019-Fall)*, 2019, pp. 1-5.

[61] J. Mei, X. Wang, K. Zheng, G. Boudreau, A. B. Sediq and H. Abouzeid, "Intelligent Radio Access Network Slicing for Service Provisioning in 6G: A Hierarchical Deep Reinforcement Learning Approach", *IEEE Trans. Commun.*, doi:10.1109/TCOMM.2021.3090423.

[62] L. Xu et al., "Socially Driven Joint Optimization of Communication, Caching, and Computing Resources in Vehicular Networks", *IEEE Trans. Wireless Commun.*, doi: 10.1109/TWC.2021.3096881.

[63] S. Nath and J. Wu, "Deep reinforcement learning for dynamic computation offloading and resource allocation in cache-assisted mobile edge computing systems", *Intell. and Converged Netw.*, vol. 1, no. 2, pp. 181-198, Sept. 2020.

[64] G. Sun, H. Al-Ward, G. O. Boateng and G. Liu, "Autonomous Cache Resource Slicing and Content Placement at Virtualized Mobile Edge Network", *IEEE Access*, vol. 7, pp. 84727-84743, 2019.

[65] Z. Chen and Z. Zhou, "Dynamic Task Caching and Computation Offloading for Mobile Edge Computing", *Proc. IEEE Conf. Global Commun. (GLOBECOM 2020)*, pp. 1-6.

[66] TD 208 (PLEN/13), FG IMT-2020: Report on Standards Gap Analysis. https://datatracker.ietf.org/liaison/1457/

[67] Y. He, F. R. Yu, N. Zhao, V. C. M. Leung and H. Yin, "Software-Defined Networks with Mobile Edge Computing and Caching for Smart Cities: A Big Data Deep Reinforcement Learning Approach", *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 31-37, Dec. 2017.

[68] A. C. Baktir, A. Ozgovde and C. Ersoy, "How can edge computing benefit from software-defined networking: A survey use cases and future directions", *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2359-2391, 4th Quart. 2017.

[69] K. Joshi and T. Benson, "Network function virtualization", *IEEE Internet Comput.*, vol. 20, no. 6, pp. 7-9, Nov./Dec. 2016.

[70] X. Hou et al., "Reliable Computation Offloading for Edge-Computing-Enabled Software-Defined IoV", *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7097-7111, Aug. 2020.

[71] Z. Zhao, G. Min, Chapter 5-8 in Edge Computing: Principle, Technology and Practice. Available: http://mobinets.cn/edgebook/contents.html

[72] Z. Ning et al., "When Deep Reinforcement Learning Meets 5G-Enabled Vehicular Networks: A Distributed Offloading Framework for Traffic Big Data", *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1352-1361, Feb. 2020.

[73] Y. Wu, X. Fang and L. Yan, "Performance Analysis of On-board Content Caching and Retrieval for High-Speed Railways", *Proc. IEEE 5G World Forum (5GWF)*, Dresden, Germany, 2019, pp. 542-546.

[74] Y. Wu, L. Yan and X. Fang, "A Low-Latency Content Dissemination Scheme for mmWave Vehicular Networks", *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7921-7933, Oct. 2019.

[75] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen and M. Chen, "In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning", *IEEE Netw.*, vol. 33, no. 5, pp. 156-165, Sept.-Oct. 2019.

[76] L. Lei, H. Xu, X. Xiong, K. Zheng, W. Xiang and X. Wang, "Multiuser Resource Control With Deep Reinforcement Learning in IoT Edge Computing", *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10119-10133, Dec. 2019.

[77] Y. Zhang, "Mobile Edge Caching", in Mobile Edge Computing, Springer, 2022, ch.3, sec.4, pp.28-32. Available: http://link,springer.com/book/10.1007/978-3-030-83944-4

[78] X. Xu, M. Tao, C. Shen, "Collaborative multi-agent multi-armed bandit learning for small-cell caching". *IEEE Trans. Wirel. Commun.*, vol. 19, no. 4, pp. 2570-2585, April 2020.

[79] F. Wang, F. Wang, J. Liu, R. Shea, L. Sun, "Intelligent video caching at network edge: a multiagent deep reinforcement learning approach", *Proc. IEEE Conf. Comput. Commun. (2020)*, pp. 2499-2508,

[80] R. Karasik, O. Simeone, S. Shamai, "How much can D2D communication reduce content delivery latency in fog networks with edge caching", *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2308-2323, April 2020.

[81] K. Zhang, J. Cao, H. Liu, S. Maharjan, Y. Zhang, "Deep reinforcement learning for social-aware edge computing and caching in urban informatics". *IEEE Trans. Ind. Inf.*, vol. 16, no. 8, pp. 5467-5477, Aug. 2020.

[82] W. Wu, N. Zhang, N. Cheng, Y. Tang, K. Aldubaikhy, X. Shen, "Beef up MM Wave dense cellular networks with D2D-assisted cooperative edge caching", *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3890-3904, April 2019.

[83] N. Zhao, X. Liu, Y. Chen, S. Zhang, Z. Li, B. Chen, M. Alouini, "Caching D2D connections in small-cell networks". *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, 12326C12338, Dec. 2018

[84] http://www.netlab.tkk.fi/tutkimus/dtn/theone/pub/the_one


**Yu Wu** (lindaswjtu@126.com) received the B.E. degree in communication engineering from Southwest Jiaotong University, Chengdu, China, where she is currently pursuing the Ph.D. degree with the Key Laboratory of Information Coding and Transmission, School of Information Science and Technology. Her research interests include vehicular network, mobile edge computing, resource optimization, and deep reinforcement learning.

**Xuming Fang** (xmfang@swjtu.edu.cn) is a professor with the School of Information Science and Technology at Southwest Jiaotong University, China. He received the B.E. degree in electrical engineering, the M.E. degree in computer engineering, and the Ph.D. degree in communication engineering from Southwest Jiaotong University, Chengdu, China, in 1984, 1989, and 1999, respectively. He held visiting positions with the Technical University Berlin, Berlin, Germany, and with the University of Texas at Dallas, Richardson, TX, USA. He has published over 200 high-quality research papers in journals and conference publications. His research interests include wireless broadband access control, radio resource management, mobile edge computing, integrated communication and sensing, and broadband wireless access for high-speed railways.

**Chunbo Luo** (C.Luo@exeter.ac.uk) received the Ph.D. degree in high performance cooperative wireless networks from the University of Reading, Reading, U.K., in 2011. His research has been supported by RCUK, Royal Society, EU H2020, NSFC, and industries. His research interest focuses on developing model-based and machine learning algorithms to solve networking and engineering problems, with a particular focus on networked unmanned vehicles. Dr. Luo is a Fellow of the Higher Education Academy.

**Geyong Min** (g.min@exeter.ac.uk) is a Professor of High Performance Computing and Networking in the Department of Computer Science within the College of Engineering, Mathematics and Physical Sciences at the University of Exeter, United Kingdom. He received the Ph.D. degree in Computing Science from the University of Glasgow, United Kingdom, in 2003, and the B.Sc. degree in Computer Science from Huazhong University of Science and Technology, China, in 1995. His research interests include Computer Networks, Wireless Communications, Parallel and Distributed Computing, Ubiquitous Computing, Multimedia Systems, Modelling and Performance Engineering.