

Collaborative Learning of Communication Routes in Edge-enabled Multi-access Vehicular Environment

Celimuge Wu, *Senior Member, IEEE*, Zhi Liu, *Senior Member, IEEE*, Fuqiang Liu, *Member, IEEE*, Tsutomu Yoshinaga, *Member, IEEE*, Yusheng Ji, *Senior Member, IEEE*, Jie Li, *Senior Member, IEEE*

Abstract—Some vehicular Internet-of-Things (IoT) applications have a strict requirement on the end-to-end delay where edge computing can be used to provide a short delay for end-users by conducting efficient caching and computing at the edge nodes. However, a fast and efficient communication route creation in multi-access vehicular environment is an underexplored research problem. In this paper, we propose a collaborative learning-based routing scheme for multi-access vehicular edge computing environment. The proposed scheme employs a reinforcement learning algorithm based on end-edge-cloud collaboration to find routes in a proactive manner with a low communication overhead. The routes are also preemptively changed based on the learned information. By integrating the “proactive” and “preemptive” approach, the proposed scheme can achieve a better forwarding of packets as compared with existing alternatives. We conduct extensive and realistic computer simulations to show the performance advantage of the proposed scheme over existing baselines.

Index Terms—Vehicular networks, routing protocol, collaborative learning, multi-access vehicular environment, fuzzy logic, reinforcement learning.

I. INTRODUCTION

Future vehicular Internet-of-Things (IoT) systems involve a huge number of devices in multi-access environments where different types of wireless spectrums should be efficiently utilized [1]–[3]. At the same time, novel services, such as cooperative autonomous driving and intelligent transport systems that demand unprecedented accuracy, latency and bandwidth are emerging. These services have an extreme variance in their resource demand with respect to time, location, context, as well as individual patterns. Current vehicular IoT systems,

such as autonomous driving, only consider the intelligence of a single vehicle agent, and therefore a limitation exists. In order to realize a more intelligent vehicular IoT system, the collaboration between different vehicle/roadside units should be utilized efficiently. This requires an efficient and intelligent networking scheme that can handle the highly mobile and varying features of the environment.

The future vehicular IoT applications can be classified into two main categories in terms of the underlying networking technologies, namely, vehicle-to-infrastructure (V2I) applications and vehicle-to-vehicle (V2V) applications. Most V2I applications, such as vehicular sensor data collections, are traffic-intensive, which means that the applications require a communication approach that could deliver a large amount of data in a short time. In contrast, most vehicle-to-vehicle applications are used to delivery safety messages or control messages between vehicles, which are delay-sensitive. The traffic-intensive applications and delay-sensitive applications have different types of quality-of-service (QoS) or quality-of-experience (QoE) requirements [4], [5], and therefore we should consider the difference in the design of communication protocols.

Meanwhile, there could be multiple types of communication interfaces available simultaneously for each vehicle, resulting in the selection of best communication approach in a multi-access environment particularly important. The edge computing [6]–[9] has been widely discussed for use in data caching and computation offloading. However, the communication route selection in a multi-access vehicular environment was not discussed adequately. Since the communication request and the corresponding QoS requirement are always difficult to predict, we have to reserve some communication resources before a request is made. However, it is difficult to find the best route for each possible communication pair with low overhead, especially when the communication environment itself changes fast with the vehicle movement. It is important to design an intelligent method to handle these spatial-temporal changes of vehicular environment. Recently, artificial intelligence based approaches have been attracting great interest in achieving intelligence in computer systems. However, due to the above mentioned characteristics of vehicular networks, conducting an efficient learning in vehicular environments is a difficult scientific problem. It is important to design a learning scheme that could evaluate and improve actions with low communication overhead.

In this paper, we propose a reinforcement learning based scheme for route selection in multi-access vehicular edge

Copyright (c) 2020 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

C. Wu and T. Yoshinaga are with the Graduate School of Informatics and Engineering, The University of Electro-Communications, 1-5-1, Chofugaoka, Chofu-shi, Tokyo, 182-8585 Japan (e-mail: {celimuge,yoshinaga}@uec.ac.jp).

Z. Liu is with the Department of Mathematical and Systems Engineering, Shizuoka University, No.5. 3-5-1, Johoku Hamamatsu, 432-8561, Japan (e-mail: liu@ieee.org)

F. Liu is with the the School of Electronics and Information Engineering, Tongji University, 4800 Cao'an, Jiading, Shanghai, China, 201804 (e-mail: liufuqiang@tongji.edu.cn)

Y. Ji is with the Information Systems Architecture Research Division, National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan (e-mail: kei@nii.ac.jp)

J. Li is with the Department of Computer Science and Engineering, Shanghai Jiaotong University, 800 Dongchuan RD. Minhang District, Shanghai, China, 200240 (e-mail: lijiecs@sjtu.edu.cn).

Manuscript received 2020.

Corresponding author: Zhi Liu (e-mail: liu@ieee.org).

computing environment. We employ an efficient end-edge-cloud collaboration to fasten the convergence speed of the learning algorithm. The main contributions of this paper are as follows.

- We propose a Q-learning-based routing scheme for multi-access vehicular edge computing environments. The proposed scheme uses a “proactive” approach to find communication routes with a low overhead, reducing the delay in finding a route when a communication request is received. The proposed scheme also employs a “pre-emptive” approach to replace an existing route with a new one by dynamically learning a better route using a reinforcement learning approach.
- We propose a decentralized approach for vehicle edge selection by jointly considering the vehicle velocity, vehicle distribution, and the connectivity between vehicles based on fuzzy logic.
- We achieve end-edge-cloud collaboration approach based on a Q-learning algorithm. Each vehicle agent is able to learn the best route by receiving the feedback from the cloud/edge.
- We consider different QoS requirements posed by different types of applications and select the best next hop route according to the specific requirement of each application.

The remainder of the paper is organized as follows. We first give a brief survey of related work in Section II. Then, the details about the proposed scheme are explained in section III. Section IV shows the simulation results for the evaluation of the proposed scheme, and finally section V draws our conclusions.

II. RELATED WORK

A. Edge computing in vehicular IoT

Most of the existing studies discuss how to conduct efficient caching or computation offloading, and do not seriously discuss how to find a communication peer in a multi-access vehicular environment. Su et al. [10] have proposed a cross-entropy-based caching scheme for vehicular content networks. The content access pattern, vehicle speed, and vehicle density were considered in the content caching at the edge nodes in order to facilitate a timely content delivery. Ale et al. [11] have employed a bidirectional deep recurrent neural network (BRNN) to conduct online proactive caching for edge computing. The BRNN model was used to predict time-series content requests in order to solve the difficulty in content popularity recognition. In [12], the joint optimization of content placement and content delivery in the vehicular edge computing was studied, and a deep deterministic policy gradient framework was used to solve the problem.

Feng et al. [13] have introduced the concept of autonomous vehicular edge computing that utilizes the computational capabilities of vehicles in a decentralized manner. An efficient job caching approach was proposed to improve the scheduling of jobs based on the information exchange between neighbors. Wang et al. [14] have proposed a multi-user non-cooperative game-based approach for computation offloading in vehicular edge computing. They designed a payoff function taking

into account the node distance, application requirements, communication overhead, and the contention for computing resources. In [15], Liu et al. discussed the offloading problem of multiple tasks with task dependency, and proposed a task scheduling algorithm that prioritizes multiple tasks to guarantee the completion time constraint of each task while considering the dependency relationship between tasks. The problem of vehicle edge server selection for the task migration has been discussed in [16]. The problem was formulated as a finite horizon Markov decision process, and a time-aware task offloading approach was proposed to solve the problem.

Tan and Hu et al. [17] have proposed a joint caching and computing approach where the resource allocation was conducted by considering the vehicle mobility and service deadline constraint. In [18], a deep reinforcement learning-based joint optimization of the edge computing and content caching was managed to improve the profits of mobile network operator while ensuring user QoS in 5G-envisioned internet-of-vehicles.

B. QoS control in vehicular environment

Since diverse applications that require different levels of QoS constraints are expected for vehicular networks, the conventional one-fit-all approach fails to satisfy the needs of different users. Wu and Zheng [19] have conducted a theoretical analysis on the uplink local delay between a vehicle and an edge node in a MEC-based VANET using stochastic geometry. The distributions of vehicles and edge nodes were modeled as an independent one-dimensional homogeneous Poisson point process. The analytical result was validated through computer simulations and the dominant factors on the transmission delay were investigated. Zhang et al. [20] have proposed a service-oriented hierarchical soft slicing framework to support multi-dimensional QoS in vehicular networks. Different network slices were constructed to support the context information service and the infotainment service, respectively, in order to differentiate different QoS requirements.

In [21], a grid routing protocol was proposed to guarantee QoS in the perception of complex vehicular IoT environments. The grid identification number was used to calculate the distance between nodes and find the least delay path. Garg et al. [22] have discussed the integration of software defined networking (SDN) and edge computing for QoS guarantee in vehicular networks, and proposed a mobility and QoS-aware SDN framework for autonomous vehicles. Kumar et al. [23] have studied the multimedia content delivery from cloud video streaming servers to moving vehicles. In order to support sufficient QoS for different video streaming cases, a QoS-aware hierarchical web caching strategy was proposed based on two metrics, namely, the load utilization ratio and the query to connectivity ratio. Peng et al. [24] proposed a dynamic spectrum management framework to guarantee QoS by considering the spectrum slicing, spectrum allocation, and transmission power adjustment. Three important issues, specifically, spectrum slicing among base stations, spectrum allocation for vehicles, and transmit power adjustment at base stations, were jointly solved through an alternate concave

search algorithm. In the above mentioned studies, an efficient integration of different communication approaches in a multi-access vehicular environment is not discussed.

C. Communication resource allocation in multi-access vehicular environment

Sun et al. [25] have introduced a data delivery protocol for vehicular networks based on a prediction of vehicular traffic volume. Li et al. [26] have addressed the data routing problem from a vehicle to a fixed destination with multi-hop forwarding. The routing decision was made by using a geographical approach that divides a road segment into smaller grids. However, an efficient use of multiple access technologies was not studied. In [27], an efficient integration of licensed and unlicensed spectrums was discussed. An edge computing approach at vehicles was used to improve the spectrum efficiency in the case where multiple types of communication approaches exist. Nkenyereye et al. [28] have proposed an SDN-based multi-access edge computing approach for vehicular networks. An OpenFlow algorithm was developed to facilitate the packet forwarding process based on the target area and current route condition.

Different types of communications, including broadcast and unicast, coexist in vehicular networks. In [29], an approach for neighbor discovery was proposed. The neighbor discovery approach conducts mobility prediction based on Kalman filter theory. Kuhlmoegen et al. [30] have proposed a packet forwarding scheme that takes advantage of both contention-based forwarding and decentralized congestion control considering the existence of mixed data traffic.

Some studies have discussed the use of physical layer technologies in improving the resource allocation efficiency. Yang et al. [31] have studied the downlink radio resource management problem for ultra-reliable and low latency communications in V2I systems. The problem was discussed by exploiting the benefits of massive MIMO. A non-orthogonal multiple access (NOMA) based resource allocation for vehicular networks has been discussed in [32]. However, the QoS-aware resource allocation problems in a multi-access vehicular environment, especially the route selection and communication approach selection, still need investigations.

III. PROPOSED SCHEME

A. Problem definition and system overview

We assume that each node has three different types of communication interfaces, specifically, cellular, IEEE 802.11p and mmWave interface. These communication approaches do not interfere with each other, and each node can switch between them or utilize all the communication approaches simultaneously. The use of different communication interfaces can improve the wireless resource utilization efficiency while requiring an approach to find the best communication interface for the transmission of each packet. Each node sends periodical *hello* messages with IEEE 802.11p interface in order to exchange information among neighbors, and the *hello* interval

is 1 second by default. The vehicle identifier, position information, vehicle velocity, and some other information (details will be explained later) are attached in the *hello* messages.

The research problems we discuss here are: 1) how to efficiently utilize different types of communication interfaces?; 2) how to find the best route for each vehicle with low overhead in accordance with the requirement and ensure the update of the route when a change of environment occurs?; 3) how to design a learning scheme that could adapt to the dynamically changing vehicular environment?

We propose a reinforcement learning-based routing scheme for multi-access vehicular edge computing environments. The proposed scheme employs a Q-learning algorithm to find the best route for each agent, and uses an end-edge-cloud collaboration approach to achieve intelligence in the route selection. The proposed scheme uses different learning approaches for traffic-intensive applications and delay-sensitive applications. First, we introduce a route selection approach for traffic-intensive applications. Then, we introduce a vehicle-to-vehicle route selection approach for delay-sensitive applications.

In our proposed scheme, IEEE 802.11p is used for transmission of both data and control messages while mmWave is only used to transmit data among neighbors. Basic procedure of the proposed scheme consists of two stages. In the first stage, we choose the vehicle edge nodes¹. In this stage, we select all the vehicle edges, and ensure that they are connected with each other through IEEE 802.11p link. In the second stage, each agent (vehicle) learns the best route to the corresponding destination by evaluating the reward (feedback) from the cloud (in the case of V2I communications) or from the communication partner (in the case of V2V communications). As shown in Fig.1, an ordinary vehicle could receive a reward from a base station (BS) or receive a discounted reward from an edge vehicle. In the former case the vehicle connects with the BS using cellular interface, and in the latter case the vehicle connects to the BS through the edge vehicle using mmWave/IEEE 802.11p interface. Based on the reward, the vehicle chooses the best action between two possible next hops, specifically, the BS or the edge vehicle.

The route selection approach for traffic-intensive applications is as follows. By selecting a vehicle edge node/base station as the next hop (action), each agent could receive a feedback from the next hop and evaluate the goodness of the action. The reward is only allocated by the base station (BS), and is transmitted to each agent with a discounted value. The learning for the delay-sensitive applications uses a similar learning approach but with a difference in the allocation of the reward. While the learning of the traffic-intensive applications allocates the reward based on the throughput each route can provide, the learning of delay-sensitive routes allocates reward based on the expected transmission delay.

B. Fuzzy logic-based vehicle edge selection

We use a fuzzy logic-based approach to evaluate each vehicle is whether suitable for being an edge node or not.

¹In the following, “edge node” and “vehicle edge node” are used interchangeably; “non-edge node” denotes a vehicle that is not selected as an edge node.

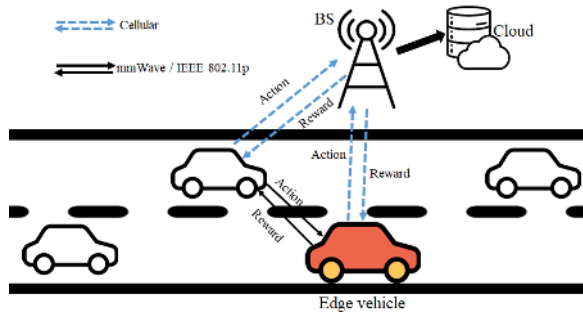


Fig. 1. Learning based on end-edge-cloud collaboration.

The suitability value is calculated by considering the vehicle velocity, vehicle distribution, and the connectivity between vehicles. The suitability value is calculated as follows. First, three different metrics, namely, stability metric, leadership metric, and connectivity metric, are defined. Then, we convert these metrics to fuzzy values using fuzzy membership functions, and then apply some predefined rules to calculate the fuzzy value for the suitability level. In the final step, the fuzzy value for the suitability is converted to a numerical value [33]. The suitability value is calculated for each vehicle within the range of $\frac{1}{2}R$ where R is the one-hop communication distance of IEEE 802.11p. If a node finds itself having the largest suitability value, the node declares itself as an edge node. Here, “edge node” works a leader to manage a group of vehicles and provide gateway supports for the ordinary vehicles in the group.

1) *First step – definition of three factors:* The stability metric, leadership metric, and connectivity metric factor are calculated based on the information in the *hello* messages received from neighbors.

Stability Metric (SM): Stability metric of node x is calculated as follows.

$$SM(x) = 1 - \frac{||v(x)| - \text{avg}_{y \in N_x} |v(y)||}{\max_{y \in N_x} |v(y)|} \quad (1)$$

where a higher value means a higher stability. Since *hello* messages are exchanged between neighbors, each vehicle can calculate its neighbors’ SM. Here, $\text{avg}_{y \in N_x} |v(y)|$ is predicted from the information attached in the *hello* messages. *SM* is updated periodically (one second interval) based on a weighted exponential moving average with a smoothing factor of 0.7. The value of smoothing factor is determined based on our experience [33].

Leadership Metric (LM): Leadership metric is calculated as follows.

$$LM(x) = \min \left(1, \frac{c(x)}{\text{Number of neighbor vehicles}} \right) \quad (2)$$

where $c(x)$ denotes the number of vehicles traveling to the same direction as the node x in its neighbors. The higher the number, the higher chance the vehicle is elected as an edge node. Here, the number of vehicles in one-hop region is acquired from the information attached in the *hello* messages. *LM* is updated periodically (one second interval) based on a

weighted exponential moving average with a smooth factor of 0.7.

Connectivity Metric (CM): Connectivity metric can be calculated in two ways depending on the available information. First, we can use the ratio of “the number of *hello* messages received from all one-hop neighbors” to “the number of *hello* messages sent by all one-hop neighbors” as

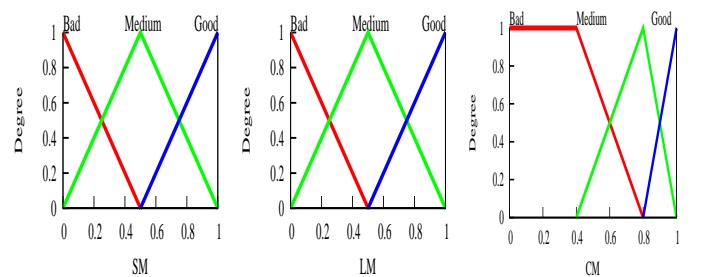
$$CM(x) = \frac{\text{Num of hellos received from all NBs}}{\text{Num of hellos sent by all NBs}}. \quad (3)$$

Note that “the number of *hello* messages sent by all one-hop neighbors” can be calculated by observing the sequence number of received *hello* messages since each *hello* message is identified with a unique sequence number which is incremented by a predefined value for each *hello* period. The other way is to use the antenna height to show the connectivity metric as a vehicle with a higher antenna always can provide a better connectivity to the neighbor vehicles. In that case, *CM* is calculated as

$$CM(x) = \frac{h(x)}{\max_{y \in N_x} h(y)}. \quad (4)$$

where $h(x)$ is the antenna height of node x .

2) *Second step – fuzzification and fuzzy rules:* Fuzzy logic is used to evaluate whether a vehicle is suitable for working as an edge node or not. This evaluation should be conducted as soon as possible in order to satisfy strict QoS requirements of vehicular IoT applications. Therefore, considering computational complexity, we use triangular or trapezoidal membership functions instead of non-linear membership functions, such as Gaussian membership functions that require more computational resources in fuzzy reasoning. The fuzzy membership functions are defined as shown in Fig. 2. The linguistic variables of the three metric are defined as {Good, Medium, Bad}.


 Fig. 2. Fuzzy membership functions (left: *SM*, middle: *LM*, right: *CM*).

The fuzzy rule is defined in Table I.

Each node calculates the rank (a suitability value for being an edge node) of each neighbor based on the IF/THEN rules as defined in Table I. The linguistic variables for the rank are defined as {Perfect, Good, Acceptable, Unpreferable, Bad, VeryBad}. In Table I, Rule 1 is expressed as follows:

IF *Velocity* is Slow, *Leadership* is High, and *Connectivity* is Good **THEN** *Rank* is Perfect.

Note that multiple rules could be applied for the same fuzzy value. Here, we use the Min-Max method to combine the results from multiple rules.

TABLE I
FUZZY RULES

	Stability	Leadership	Connectivity	Rank
Rule1	Good	Good	Good	Perfect
Rule2	Good	Good	Medium	Good
Rule3	Good	Good	Bad	Unpreferable
Rule4	Good	Medium	Good	Good
Rule5	Good	Medium	Medium	Acceptable
Rule6	Good	Medium	Bad	Bad
Rule7	Good	Bad	Good	Unpreferable
Rule8	Good	Bad	Medium	Bad
Rule9	Good	Bad	Bad	VeryBad
Rule10	Medium	Good	Good	Good
Rule11	Medium	Good	Medium	Acceptable
Rule12	Medium	Good	Bad	Bad
Rule13	Medium	Medium	Good	Acceptable
Rule14	Medium	Medium	Medium	Unpreferable
Rule15	Medium	Medium	Bad	Bad
Rule16	Medium	Bad	Good	Bad
Rule17	Medium	Bad	Medium	Bad
Rule18	Medium	Bad	Bad	VeryBad
Rule19	Bad	Good	Good	Unpreferable
Rule20	Bad	Good	Medium	Bad
Rule21	Bad	Good	Bad	VeryBad
Rule22	Bad	Medium	Good	Bad
Rule23	Bad	Medium	Medium	Bad
Rule24	Bad	Medium	Bad	VeryBad
Rule25	Bad	Bad	Good	Bad
Rule26	Bad	Bad	Medium	VeryBad
Rule27	Bad	Bad	Bad	VeryBad

3) *Last step – defuzzification*: Fig. 3 shows the output membership function that is used to convert from a fuzzy value to a numerical value. The process of conversion is called defuzzification. In this work, the center of gravity (COG) method is used for the defuzzification.

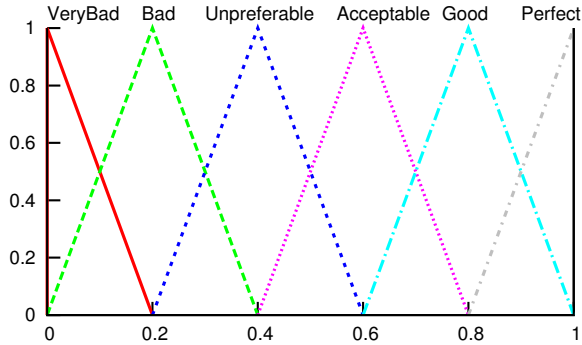


Fig. 3. Output membership function.

C. Q-learning based end-edge-cloud collaboration for traffic-intensive applications

Here, BS is connected to the cloud, and the vehicles are connected to the cloud through the BS. Therefore, the BS is treated as the cloud in our learning model for simplicity. The selected edge vehicles are the edges. The ordinary vehicles are the end nodes. Our aim is to conduct an intelligent collaboration within the end-edge-cloud architecture. The cloud and edge nodes give suggestion to the end nodes to select the next hop

node for information transmission. Note that the information could be sensor data or the data required for task offloading, which means that finding a next hop node is mandatory for communications as well as computing.

1) *Q-learning model*: The Q-learning model is defined as follows. Vehicles are the agents, and the actions are the pairs of a possible communication type and the next hop node for the packet forwarding. The possible actions at each node would be the set of its one-hop neighbors including base station. The BS is responsible for sending back a reward for each action the vehicle executed. The reward will be further transmitted with a discount by other edge vehicles. Each vehicle adjusts own behavior based on the feedback from the BS. The information exchange between agents is done with *hello* messages. Each node maintains a Q-Table where each Q-value shows the value for choosing m as the next hop to the RSU.

2) *Update of Q-Table*: The state is expressed by a pair of {destination, current node}, and the action is determined by {communication type, next hop}. In case of using the cellular communications, the next hop would be BS, and in the case of IEEE 802.11p, the next hop node would be a neighbor node. Each node has to maintain a Q-value for each triple of a destination, the communication type, and an one-hop neighbor. Upon reception of each *hello* message, the Q-Table is updated. The initial value for each Q-value is 0. Each vehicle maintains a Q-value to the BS and each vehicle in two-hop region. Considering the change of neighbors with the vehicle movement, we release the corresponding Q-Table space of old neighbors when necessary for the purpose of maintaining information about new neighbors. The proposed scheme does not maintain route for each possible destination considering the size of Q-Table. For finding a route to other nodes, the proposed scheme uses a hierarchical routing approach where different levels of gateway nodes exist. Note that for each vehicle, there is at least one neighbor would be an edge node that is working as a gateway node and responsible for finding a route to any other vehicle. The BS also performs a duty of gateway. After reception of a *hello* message from node m , node c updates the corresponding Q-value as

$$Q_c(d, t, m) \leftarrow \alpha \times LQ(c, m) \times \{R + \gamma \times \max_{y \in NB_m} Q_m(d, t, y)\} + (1 - \alpha) \times Q_c(d, t, m). \quad (5)$$

where d and t are the destination node and communication type, respectively. $LQ(c, m)$ is the link quality value between node c and m , which is expressed by the *hello* reception ratio between two nodes. NB_m denotes the one-hop neighbor set of node m . Here, the learning rate α and discount rate γ are set to 0.8 and 0.9, respectively, based on our experience [34]. The reward R_m is calculated as

$$R = \begin{cases} \bar{R}, & \text{if } m \text{ is base station and } c \text{ is an edge node} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where $\bar{R} \in [0, 1]$ is allocated by the BS according to the number of vehicles connected to the BS. The base station will set the reward according the vehicle density. If the density is high, the cloud will give a high reward to the corresponding

edge candidate. The reward is further discounted according to the number of hops. The exploration is achieved by exchanging *hello* messages among neighbors. Therefore, for a route selection, each node always can choose the pair of node and communication type showing the highest Q -value.

In case of V2V communication routes, the reward is updated as

$$R = \begin{cases} 1, & \text{if } m \text{ is an edge node} \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

The reward is set to 1 here in order to utilize decentralized communications as far as possible. The decentralized communication approaches include IEEE 802.11p and mmWave. Since the mmWave communication is only possible in a line-of-sight communication link, we only use mmWave for the communications between an edge vehicle node and its non-edge neighbors. For this type of communication pairs, if both mmWave and IEEE 802.11p are available, then the mmWave is used.

Each edge vehicle attaches its own cost (the corresponding Q -value) to its neighbors in the *hello* message. Upon reception of a *hello* message from a neighbor vehicle, the vehicle could update the corresponding cost in the case of using the neighbor as the next hop based on the Q -learning algorithm. The exploration of Q -learning is achieved by the periodical *hello* message exchange. Therefore, each agent is aware of the best action based on the Q -values.

D. Q -learning based packet forwarding for delay-sensitive unicast applications

The delay-sensitive applications could be used to send the information required for the collaboration between vehicles, which could enable collaborative autonomous driving. The data transfer can be either conducted by up to 2-hop V2V communications or cellular communications. Here the reward is calculated based on the delay. For V2V communications, the reward is sent by each vehicle to its neighbors. The selected edge vehicle could further transmit the discounted reward to its neighbors, but the reward will not be disseminated to more than two hops. This is achieved by only the edge vehicle nodes attaching the Q -Table entries of one-hop neighbors to the *hello* messages.

The reward from the BS through cellular interface is calculated as

$$\hat{R} = \min\left(1, \frac{D_{th}}{\frac{S_{pkt} * N_{ue}}{BW_{ul}} + \frac{S_{pkt} * N_{ue}}{BW_{dl}} + D_{bs}}\right) \quad (8)$$

where D_{th} , S_{pkt} , N_{ue} , BW_{ul} , BW_{dl} , D_{bs} are the delay requirement, packet size, number of user devices, uplink bandwidth, down link bandwidth, and processing delay at the base station. If the BS is able to satisfy the incoming request, \hat{R} is set as 1, and otherwise set to a smaller value. The processing delay includes all the times required for the scheduling, queueing, and computing except the propagation delay.

For the path using IEEE 802.11p communications, the reward is received from a neighbor node. In this case, the

reward from a vehicle is calculated as

$$\hat{R} = \min\left(1, \frac{D_{th}}{\frac{S_{pkt}}{BW_{11p} * HRR} + D_{11p}}\right) \quad (9)$$

where BW_{11p} is the bandwidth of IEEE 802.11p (27 Mbps) and HRR is *hello* reception ration between two neighbors. D_{11p} is the processing delay at each vehicle, which includes all the contention delay, the retransmission delay due to packet collisions, and computing delay. Here, the discount rate is set as 0.5 in order to avoid the use of 2-hop transmissions as far as possible. Note that the parameters of Eq.(8) and Eq.(9) should be tuned based on the corresponding hardware and environment, which is beyond the scope of this paper.

E. Data dissemination for delay-sensitive broadcast applications

Data dissemination for delay-sensitive broadcast applications should be conducted through IEEE 802.11p V2V communications as it could be difficult to use cellular communications to detect all the intended receivers. We propose a multi-hop broadcast approach based on the proposed edge architecture. As shown in Fig.4, the broadcast messages are forwarded by the edge nodes. The ordinary nodes do not rebroadcast the messages, which can significantly reduce the redundant forwarding. It is also possible to confirm the packet delivery status of a broadcast packet by sending the acknowledge message (ACK) from the edge nodes, which makes the retransmission of a broadcast message possible and therefore ensure a high packet dissemination ratio. The forwarding algorithm is also simple as follow. If the current node is an edge node, the node just rebroadcasts the packet, and otherwise just receives the packet without further forwarding.

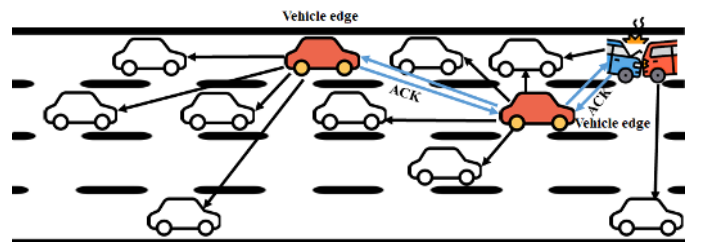


Fig. 4. Edge-based multi-hop broadcast.

IV. SIMULATION RESULTS

To evaluate the performance of the proposed scheme, we conducted extensive computer simulations using ns-2.34 [35] (see Table II). We used a freeway road having three lanes in each direction, which was generated using the same approach as [34]. Each vehicle had three different types of wireless interfaces, namely, cellular, IEEE 802.11p, and mmWave. For IEEE 802.11p, we used Nakagami model to include a realistic fading environment, and the parameters were the same as [34]. The average transmission range of IEEE 802.11p was set as 250 m. The proposed protocol was compared with “Without edge” (without edge computing), “Random edge” (random edge selection), and “Edge without preemptive” (edge

without preemptive route change). In “Without edge”, each node only uses the cellular interface for communications. “Random edge” collects some nodes randomly to conduct data caching. “Edge without preemptive” employs the same approach as the proposed scheme for the edge selection but does not conduct efficient route change between different wireless interfaces. Three different types of applications were considered in the simulations, namely, traffic-intensive applications, delay-sensitive unicast applications, and delay-sensitive broadcast applications. In the following simulation result figures, each error bar shows the 95% confidence interval of the corresponding data.

TABLE II
SIMULATION ENVIRONMENT

Topology	Freeway, 2000m, 6 lanes
Number of nodes	400
Maximum velocity	100 km/h
Mobility generator	Ref. [34]
MAC	IEEE 802.11p MAC
Data rate	27 Mbps (unicast), 3 Mbps (broadcast)
Fading model	Nakagami Model
Simulation time	1000 s

A. Performance for traffic-intensive applications

For traffic-intensive applications, we evaluated the communication performance between vehicles and the cloud. The number of cellular base station was 1, which means that the communication path to the cloud must go through this base station. For simplicity, we used a down link traffic where the data were sent from the cloud to all the vehicles and data were cached at the base station. Note that, in the simulation topology, the base station was “cloud”, and the selected vehicle edge nodes were “edges”. Therefore, “end-edge-cloud collaboration” was simulated by the collaboration among non-edge vehicles, edge vehicles, and the base station.

Fig.5 shows the TCP throughput for various numbers of receivers where the cellular bandwidth was 500 Mbps. The maximum vehicle velocity was 100 Km/h. We can observe that “Without edge” approach fails to provide an acceptable throughput since all the vehicles use cellular communications and therefore the resource allocated to each vehicle is small. “Random edge” achieves better performance by conducting data caching at randomly selected edge nodes. However, due to the inefficiency of the random edge selection, the performance improvement is limited. By choosing the best nodes for the edge nodes based on a fuzzy logic algorithm, the proposed scheme shows the largest throughput. “Edge without preemptive” uses fixed edge node and changes a route only when the link becomes unavailable, resulting in difficulty of adapting to the change of network environments. The advantage of the proposed protocol over “Edge without preemptive” explains that it is promising to conduct a route change by preemptively finding a better route. The proposed scheme is able to preemptively change to a better route by using the Q-learning approach to explore all the possible paths based on the exchange of periodical *hello* messages.

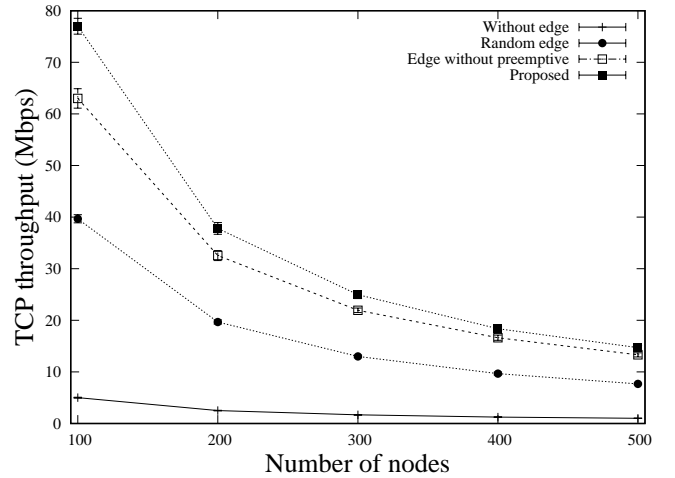


Fig. 5. TCP throughput for various numbers of receivers.

Fig.6 shows the TCP throughput for various maximum vehicle velocities. The number of vehicle was 300. The performance of “Random edge” is affected by the vehicle velocity significantly, which shows the importance of using an efficient edge selection algorithm. The proposed scheme can find stable edge nodes by taking into account the velocity factor, vehicle distribution, and the signal quality between vehicles for the edge selection. This ensures that the proposed scheme could achieve the best performance for different vehicle velocities.

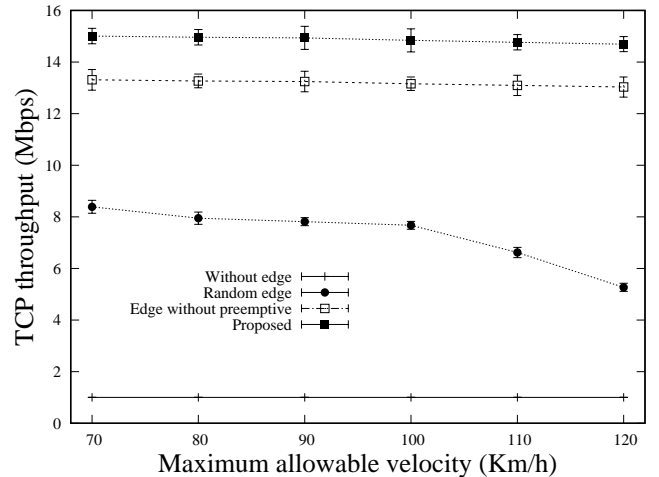


Fig. 6. TCP throughput for various vehicle velocities.

The effect of the available cellular bandwidth on the protocol performance is shown in Fig.7. The number of vehicle was 300, and the maximum vehicle velocity was 100 Km/h. In “Without edge”, all the receiver nodes directly get the data from the cloud by using cellular communications, which results in an inefficient use of cellular resources. Therefore, the throughput improvement with the increase of cellular bandwidth is not notable. Other approaches could utilize the cellular resources more efficiently by conducting edge computing at edge nodes. However, the performance of “Random edge” is still unsatisfactory due to the blindness of the random edge selection. “Edge without preemptive” performs

better than “Random edge” by selecting better and more stable edge nodes. The proposed scheme can further improve “Edge without preemptive” by finding the best route based on Q-learning and switching between different communication interfaces.

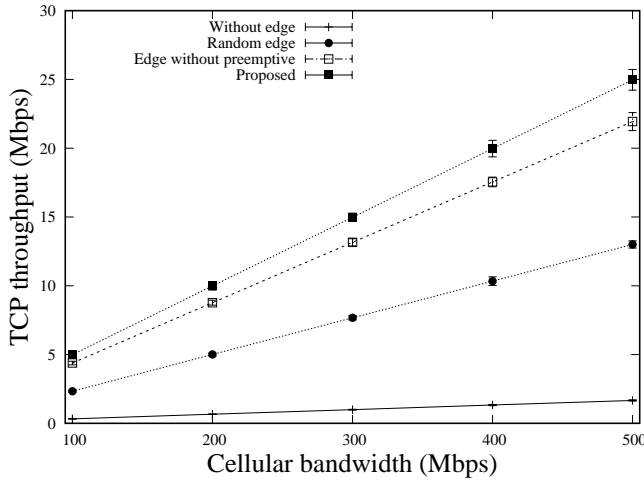


Fig. 7. TCP throughput for various cellular bandwidths.

B. Performance for delay-sensitive unicast applications

We generated UDP traffics to evaluate the performance of the proposed scheme for delay-sensitive unicast applications. UDP packet size was 512 bytes, and the data rate for each traffic flow was set as 1 packet per second. The cellular bandwidths for uplink and downlink were 250 Mbps and 500 Mbps, respectively. The processing delay at the base station was set as 50 ms. Fig.8 shows the end-to-end delay for various numbers of traffic flows. The delay of “Without edge” is the largest as the use of pure cellular transmissions always encounter the problem of insufficient bandwidths when then number of traffic flows is large. This is because all the traffic flows go through the base station, which is inefficient for some communication pairs that are close to each other geographically. “Random edge” shows a better outcome by conducting some IEEE 802.11p communications through randomly selected edge nodes without going through the base station. However, the random selection of edge node is unsatisfactory in terms of MAC layer contention efficiency. Since the proposed scheme could select the best edge nodes for selection, it can achieve the lowest delay, which is a significant improvement especially when the number of traffic flows is large. The difference between “edge w/o preemptive” and the proposed scheme shows that it is difficult to achieve a good performance if the route is not changed until the route is disconnected. It is important to switch efficiently between the cellular communications and IEEE 802.11p according to the usage ratio of cellular spectrum. This requires an adaptive algorithm that can evolve by efficiently perceiving the environments.

We also conducted evaluations for different processing latencies at the base station where the number of traffic flows was 300. As shown in Fig.9, with the increase of the

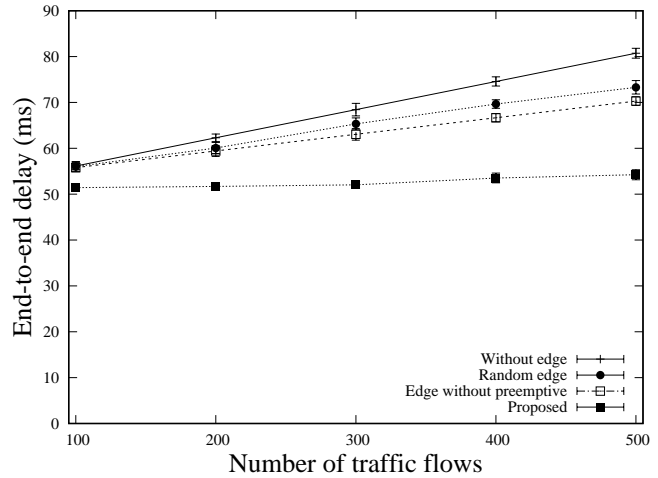


Fig. 8. End-to-end delay for various vehicle densities.

processing delay at the base station, the advantage of the proposed scheme becomes more significant. This is because it becomes important to use other types of communication approaches to support the cellular communications in order to provide a low delay for all the communication pairs. The proposed scheme always can achieve a low delay by using IEEE 802.11p communications for some communication pairs. By using the Q-learning algorithm, the proposed scheme can find the best communication interface and the corresponding route for each communication pair, resulting in the lowest delay.

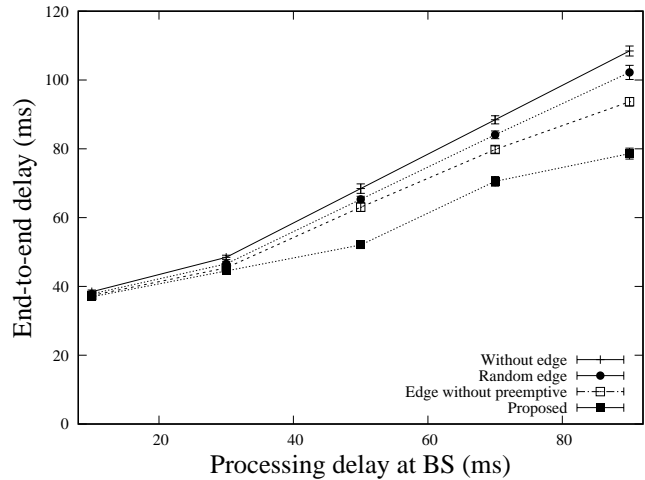


Fig. 9. End-to-end delay for various processing latencies at the base station.

Fig.10 shows the end-to-end delay for various vehicle velocities. The number of traffic flows was 100. We can observe that the delay of “Without edge” and “Random edge” increases as the vehicle velocity becomes faster. This explains the importance of selecting stable edge nodes for the data caching. By considering the vehicle velocity and vehicle distribution in the edge node selection, the proposed scheme achieves a stable latency for different vehicle velocities. The preemptive route change approach also contributes to the short delay by finding a better route before the current route is broken.

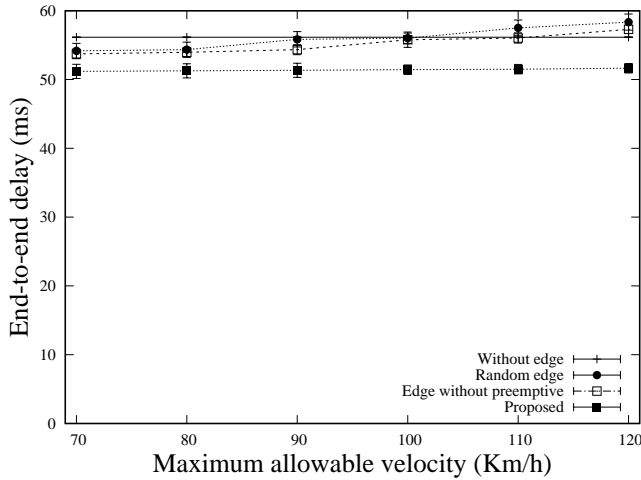


Fig. 10. End-to-end delay for various vehicle velocities.

C. Performance for delay-sensitive broadcast applications

We also conducted simulations for delay-sensitive broadcast applications. The number of broadcast source nodes was 1, and all other vehicles were considered as the intended receivers. Since the broadcast communication through cellular interface is not realistic, here “Without edge” denotes the weighted p-persistence [33]. Fig.11 shows the packet dissemination ratio for various vehicle densities. Since “Without edge” does not use edge computing on packet dissemination, it results in a low packet dissemination ratio due to the redundant rebroadcast of data. The edge selection of “Random edge” also cannot achieve a satisfactory result. “Edge without preemptive” shows a higher packet dissemination ratio as compared with “Without edge” and “Random edge” by using a more efficient edge selection. Based on the efficient edge node selection and the edge-based retransmission scheme, the proposed scheme can provide the perfect packet dissemination ratio.

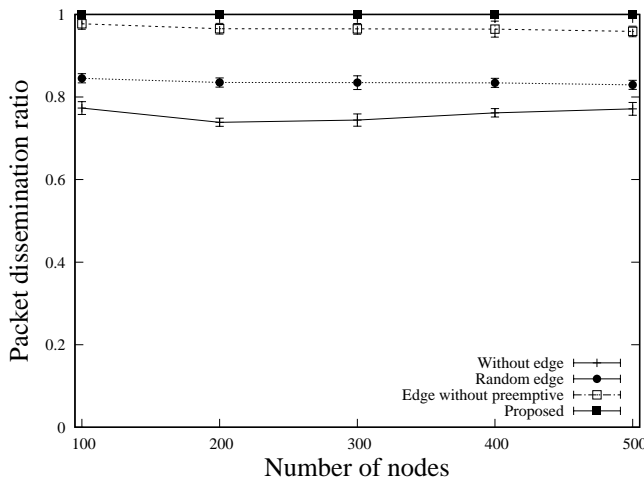


Fig. 11. Packet dissemination ratio for various vehicle densities.

As shown in Fig.12, the proposed scheme shows the lowest end-to-end delay. When the node density is high, the selection of edge node has more a remarkable impact on the delay. Since the proposed scheme considers vehicle velocity,

vehicle distribution, and the link quality between vehicles, the proposed scheme is able to find the best edges, and therefore satisfy the low latency requirement of delay-sensitive broadcast applications.

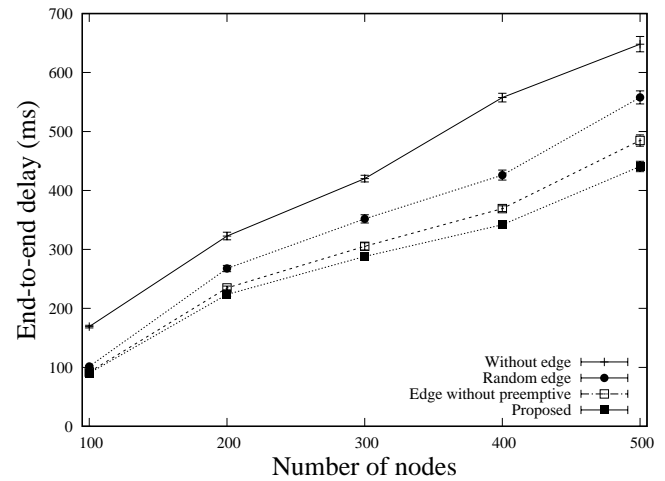


Fig. 12. End-to-end delay for various vehicle densities.

V. CONCLUSIONS

We discussed the problem of route selection in multi-access vehicular edge computing environment, and proposed a scheme based on a reinforcement learning approach. The proposed scheme employs a “proactive” approach to find communication routes based on periodical *hello* message exchange with low overhead, and conducts “preemptive” change of communication interfaces and routes to ensure a high performance in varying network environments. The proposed scheme uses different learning criteria for traffic-intensive applications and delay-sensitive applications in order to satisfy different QoS requirements. The simulation results show that the proposed approach can achieve a better performance than existing baselines.

ACKNOWLEDGMENT

This research was supported in part by JSPS KAKENHI Grant Numbers, 18KK0279 and 19H04093, and the Telecommunications Advanced Foundation.

REFERENCES

- [1] C. Wu, Z. Liu, D. Zhang, T. Yoshinaga, and Y. Ji, “Spatial Intelligence towards Trustworthy Vehicular IoT,” *IEEE Commun. Mag.*, vol.56, no.10, pp.22–27, Oct. 2018.
- [2] P. Liu, Y. Ding, and Tingting Fu, “Optimal ThrowBoxes assignment for big data multicast in VDTNs,” *Wireless Networks*, DOI:https://doi.org/10.1007/s11276-019-01974-z, March 2019.
- [3] T. Fu, C. Wang, and N. Cheng, “Deep Learning based Joint Optimization of Renewable Energy Storage and Routing in Vehicular Energy Network,” *IEEE Internet Things J.*, DOI: 10.1109/JIOT.2020.2966660, Early Access.
- [4] X. He, K. Wang, and W. Xu, “QoE-Driven Content-Centric Caching With Deep Reinforcement Learning in Edge-Enabled IoT,” *IEEE Comput. Intell. Mag.*, vol.14, no.4, pp.12–20, Nov. 2019.
- [5] X. He, K. Wang, H. Huang, T. Miyazaki, Y. Wang, and S. Guo, “Green Resource Allocation based on Deep Reinforcement Learning in Content-Centric IoT,” *IEEE Trans. Emerg. Topics Comput.*, DOI: 10.1109/TETC.2018.2805718, Early Access.

- [6] N. Hassan, K. A. Yau, and C. Wu, "Edge Computing in 5G: A Review," *IEEE Access*, vol.7, pp.127276–127289, Aug. 2019.
- [7] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized Computation Offloading Performance in Virtual Edge Computing Systems via Deep Reinforcement Learning," *IEEE Internet Things J.*, vol.6, no.3, pp.4005–4018, June 2019.
- [8] Y. Wang, K. Wang, H. Huang, T. Miyazaki, and S. Guo, "Traffic and computation co-offloading with reinforcement learning in fog computing for industrial applications," *IEEE Trans. Industrial Informatics*, vol.15, no.2, pp.976–986, Feb. 2019.
- [9] J. Feng, Z. Liu, C. Wu, and Y. Ji, "Mobile Edge Computing for Internet of Vehicles: Offloading Framework and Job Scheduling," *IEEE Veh. Technol. Mag.*, vol.14, no.1, pp.28–36, March 2019.
- [10] Z. Su, Y. Hui, Q. Xu, T. Yang, J. Liu, and Y. Jia, "An Edge Caching Scheme to Distribute Content in Vehicular Networks," *IEEE Trans. Veh. Technol.*, vol.67, no.6, pp.5346–5356, June 2018.
- [11] L. Ale, N. Zhang, H. Wu, D. Chen, and T. Han, "Online Proactive Caching in Mobile Edge Computing Using Bidirectional Deep Recurrent Neural Network," *IEEE Internet Things J.*, vol.6, no.3, pp.5520–5530, June 2019.
- [12] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, and N. Ansari, "Deep Reinforcement Learning for Cooperative Content Caching in Vehicular Edge Computing and Networks," *IEEE Internet Things J.*, vol.7, no.1, pp.247–257, Jan. 2020.
- [13] J. Feng, Z. Liu, C. Wu, and Y. Ji, "AVE: Autonomous Vehicular Edge Computing Framework with ACO-Based Scheduling," *IEEE Trans. Veh. Technol.*, vol.66, no.12, pp.10660–10675, Dec. 2017.
- [14] Y. Wang, P. Lang, D. Tian, J. Zhou, X. Duan, Y. Cao, and D. Zhao, "A Game-based Computation Offloading Method in Vehicular Multi-access Edge Computing Networks," *IEEE Internet Things J.*, DOI: 10.1109/JIOT.2020.2972061, Early Access.
- [15] Y. Liu, S. Wang, Q. Zhao, S. Du, A. Zhou, X. Ma, and F. Yang, "Dependency-Aware Task Scheduling in Vehicular Edge Computing," *IEEE Internet Things J.*, DOI: 10.1109/JIOT.2020.2972041, Early Access.
- [16] X. Zhang, J. Zhang, Z. Liu, Q. Cui, X. Tao, and S. Wang, "MDP-based Task Offloading for Vehicular Edge Computing under Certain and Uncertain Transition Probabilities," *IEEE Trans. Veh. Technol.*, DOI: 10.1109/TVT.2020.2965159, Early Access.
- [17] L. T. Tan, 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.
- [18] Z. Ning, K. Zhang, X. Wang, M. S. Obaidat, L. Gu, X. Hu, B. Hu, Y. Guo, B. Sadoun, and R. Y. K. Kwok, "Joint Computing and Caching in 5G-Envisioned Internet of Vehicles: A Deep Reinforcement Learning-Based Traffic Control System," *IEEE Trans. Intell. Transp. Syst.*, DOI: 10.1109/TITS.2020.2970276, Early Access.
- [19] Y. Wu, and J. Zheng, "Modeling and Analysis of the Uplink Local Delay in MEC-based VANETs," *IEEE Trans. Veh. Technol.*, DOI: 10.1109/TVT.2020.2970551, Early Access.
- [20] S. Zhang, H. Luo, J. Li, W. Shi, X. S. Shen, "Hierarchical Soft Slicing to Meet Multi-Dimensional QoS Demand in Cache-Enabled Vehicular Networks," *IEEE Trans. Wireless Commun.*, DOI: 10.1109/TWC.2019.2962798, Early Access.
- [21] D. Wang, Q. Zhang, J. Liu, and D. Yao, "A Novel QoS-Awared Grid Routing Protocol in the Sensing Layer of Internet of Vehicles Based on Reinforcement Learning," *IEEE Access*, vol.7, pp.185730–185739, Dec. 2019.
- [22] S. Garg, K. Kaur, S. H. Ahmed, A. Bradai, G. Kaddoum, and M. Atiqzaman, "MobQoS: Mobility-Aware and QoS-Driven SDN Framework for Autonomous Vehicles," *IEEE Wireless Communications*, vol.26, no.4, pp.12–20, August 2019.
- [23] N. Kumar, S. Zeadally, and J. J. P. C. Rodrigues, "QoS-Aware Hierarchical Web Caching Scheme for Online Video Streaming Applications in Internet-Based Vehicular Ad Hoc Networks," *IEEE Trans. Ind. Electron.*, vol.62, no.12, pp.7892–7900, Dec. 2015.
- [24] H. Peng, Q. Ye, and X. Shen, "Spectrum Management for Multi-Access Edge Computing in Autonomous Vehicular Networks," *IEEE Trans. Intell. Transp. Syst.*, DOI: 10.1109/TITS.2019.2922656, Early Access.
- [25] P. Sun, N. AlJeri, and A. Boukerche, "DACON: A Novel Traffic Prediction and Data-Highway-Assisted Content Delivery Protocol for Intelligent Vehicular Networks," *IEEE Trans. Sustain. Comput.*, DOI: 10.1109/TSUSC.2020.2971628, Early Access.
- [26] F. Li, X. Song, H. Chen, X. Li, and Y. Wang, "Hierarchical Routing for Vehicular Ad Hoc Networks via Reinforcement Learning," *IEEE Trans. Veh. Technol.*, vol.68, no.2, pp.1852–1865, Feb. 2019.
- [27] C. Wu, X. Chen, T. Yoshinaga, Y. Ji, and Y. Zhang, "Integrating Licensed and Unlicensed Spectrum in Internet-of-Vehicles with Mobile Edge Computing," *IEEE Network*, vol.33, no.4, pp.48–53, July/August 2019.
- [28] L. Nkenyerere, L. Nkenyerere, S. M. R. Islam, C. A. Kerrache, M. Abdullah-Al-Wadud, and Atif Alamri, "Software Defined Network-Based Multi-Access Edge Framework for Vehicular Networks," *IEEE Access*, vol.8, pp.4220–4234, Dec. 2019.
- [29] C. Liu, G. Zhang, W. Guo, and R. He, "Kalman Prediction-Based Neighbor Discovery and Its Effect on Routing Protocol in Vehicular Ad Hoc Networks," *IEEE Trans. Intell. Transp. Syst.*, vol.21, no.1, pp.159–169, Jan. 2020.
- [30] S. Kuhlmoorgen, H. Lu, A. Festag, J. Kenney, S. Gemsheim, and G. Fettweis, "Evaluation of Congestion-Enabled Forwarding With Mixed Data Traffic in Vehicular Communications," *IEEE Trans. Intell. Transp. Syst.*, vol.21, no.1, pp.233–247, Jan. 2020.
- [31] H. Yang, K. Zheng, L. Zhao, and L. Hanzo, "Twin-Timescale Radio Resource Management for Ultra-Reliable and Low-Latency Vehicular Networks," *IEEE Trans. Veh. Technol.*, vol.69, no.1, pp.1023–1036, Jan. 2020.
- [32] Y. Liu, H. Zhang, K. Long, A. Nallanathan, and V. C. M. Leung, "Energy-Efficient Subchannel Matching and Power Allocation in NOMA Autonomous Driving Vehicular Networks," *IEEE Wireless Communications*, vol.26, no.4, pp.88–93, August 2019.
- [33] C. Wu, S. Ohzahata, and T. Kato, "VANET Broadcast Protocol Based on Fuzzy Logic and Lightweight Retransmission Mechanism," *IEICE Trans. Commun.*, vol.95-B, no.2, pp.415–425, Feb. 2012.
- [34] C. Wu, S. Ohzahata, and T. Kato, "Flexible, Portable, and Practicable Solution for Routing in VANETs: A Fuzzy Constraint Q-Learning Approach," *IEEE Trans. Veh. Technol.*, vol.62, no.9, Nov. 2013, pp.4251–4263.
- [35] The Network Simulator - ns-2, <http://www.isi.edu/nsnam/ns/>, Accessed on Sep. 29, 2019.



Celimuge Wu (M'10–SM'18) received his ME degree from the Beijing Institute of Technology, China in 2006, and his PhD degree from The University of Electro-Communications, Japan in 2010. He is currently an associate professor with the Graduate School of Informatics and Engineering, The University of Electro-Communications. His current research interests include vehicular ad hoc networks, sensor networks, intelligent transport systems, IoT, and mobile cloud computing. He is/has been a TPC Co-Chair of Wireless Days 2019, ICT-DM 2018, a track Co-Chair of many international conferences including ICCCN 2019 and IEEE PIMRC 2016. He is/has been serving as an associate editor of IEEE Access, IEICE Transactions on Communications, International Journal of Distributed Sensor Networks, MDPI Sensors, and a guest editor of IEEE Transactions on Emerging Topics in Computational Intelligence, IEEE Computational Intelligence Magazine, ACM/Springer MONET etc.



Zhi Liu (M'14–SM'19) received the B.E., from the University of Science and Technology of China, China and Ph.D. degree in informatics in National Institute of Informatics. He is currently an Assistant Professor at Shizuoka University. His research interest includes video network transmission, vehicular networks and mobile edge computing. He has served as Guest Editor of journals including Wireless Communications and Mobile Computing and IEICE Transactions on communications. He has been serving as the chair for number of international conference and workshops. He is a senior member of IEEE.



Fuqiang Liu (M'09) received the Bachelor's degree from Tianjin University, Tianjin, China, in 1987 and the Ph.D. degree from China University of Mining and Technology, Xuzhou, China, in 1996. He is a Professor with the School of Electronics and Information Engineering, Tongji University, Shanghai, China, where he is currently the Director of the Broadband Wireless Communication and Multimedia Laboratory. He is a Guest Professor with the National Institute of Informatics, Tokyo, Japan. He has published more than 300 scientific papers and

nine books. His research mainly focuses on theories and technologies of broadband wireless communications (fifth-generation mobile communications and vehicular communications/dedicated short-range communications) and their applications in automotive and intelligent transportation systems. Dr. Liu has participated in numerous national research projects in China and has received research funding from the United States, Finland, the European Union, and Japan.



Tsutomu Yoshinaga (M) received the BE, ME, and DE degrees from Utsunomiya University in 1986, 1988, and 1997, respectively. From 1988 to July 2000, he was a research associate of the Faculty of Engineering, Utsunomiya University. He was also a visiting researcher at Electro-Technical Laboratory from 1997 to 1998. Since August 2000, he has been with the Graduate School of Information Systems, The University of ElectroCommunications, where he is now a professor. His research interests include computer architecture, interconnection networks, and network computing. He is a fellow of IEICE, and a member of

ACM, IEEE, and IPSJ.



Yusheng Ji (M'94–SM'18) received B.E., M.E., and D.E. degrees in electrical engineering from the University of Tokyo. She joined the National Center for Science Information Systems, Japan (NACSIS) in 1990. Currently, she is a Professor at the National Institute of Informatics (NII), and SOKENDAI (the Graduate University for Advanced Studies). Her research interests include network architecture, resource management, and quality of service provisioning in wired and wireless communication networks. She is/has been an Editor of IEEE Transactions on Vehicular Technology, a Symposium Co-chair of IEEE GLOBECOM 2012, 2014, and a Track Co-chair of IEEE VTC2016-Fall, VTC2017-Fall etc.

She is/has been an Editor of IEEE Transactions on Vehicular Technology, a Symposium Co-chair of IEEE GLOBECOM 2012, 2014, and a Track Co-chair of IEEE VTC2016-Fall, VTC2017-Fall etc.



Jie Li (M'94–SM'04) received the B.Eng. degree in computer science from Zhejiang University, Hangzhou, China, the M.Eng. degree in electronic engineering and communication systems from the China Academy of Posts and Telecommunications, Beijing, China, and the Dr.Eng. degree from the University of Electro-Communications, Tokyo, Japan. He is currently with Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai, China, where he is also a Chair Professor. He was a full Professor with the Department of

Computer Science, University of Tsukuba, Japan. He was a Visiting Professor with Yale University, USA, Inria Sophia Antipolis, and Inria Grenoble-Rhone-Alpes, France. His current research interests include big data and AI, cloud and Edge computing, machine learning and networking, network security, OS, and information system architectures. He has also served on the program committees for several international conferences. He is a Co-Chair of the IEEE Technical Community on Big Data, and the Founding Chair of the IEEE ComSoc Technical Committee on Big Data. He serves as an Associate Editor for many IEEE journals and TRANSACTIONS.