On Vehicular Ad-Hoc Networks With Full-Duplex Radios: An End-to-End Delay Perspective

Momiao Zhou, *Member, IEEE*, Lei Liu, *Member, IEEE*, Yanshi Sun, *Member, IEEE*, Kan Wang, *Member, IEEE*, Mianxiong Dong, *Member, IEEE*, Mohammed Atiquzzaman, *Senior Member, IEEE*, and Schahram Dustdar, *Fellow, IEEE*

***Abstract*— The aim of this paper is to present a groundwork on the delay-minimized routing problem in a vehicular ad-hoc network (VANET) where some of the vehicles are equipped with full-duplex (FD) radios. We first give the generalized delay calculation model for a multi-hop path, and prove that the Dijkstra algorithm is unable to get the delay-minimized routing path from source to destination. Then we propose two routing methods: graph-based method and deep reinforcement learning (DRL)-based method. In the graph-based method, the network topology is reformulated as an equivalent graph and then an evolved-Dijkstra algorithm is proposed. In the DRL-based method, the deep Q network (DQN) is employed to learn the shortest end-to-end path, wherein the delay is modeled as the rewards for routing actions. The graph-based method can achieve the exact minimum end-to-end delay, while the DRL-based method is more feasible due to its acceptable complexity. Finally, extensive simulations demonstrate that the DRL-based approach with proper hyper-parameters can achieve near minimum end-to-end delay, and the achieved delay has a notably decline as the number of FD nodes increases.**

***Index Terms*— Routing, vehicular ad-hoc network (VANET), full-duplex, deep Q network (DQN).**

I. Introduction

WIRELESS devices have long been half-duplex (HD) enabled, where transmitting and receiving are performed in the time division manner. However, the breakthrough of the self-interference cancellation (SIC) technique in recent years has made it possible to enable co-time co-channel full-duplex (FD) operation at wireless devices [1], [2].

Manuscript received 11 August 2022; revised 2 March 2023; accepted 8 May 2023. Date of publication 5 July 2023; date of current version 4 October 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62001152 and Grant 62001357 and in part by the Fundamental Research Funds for the Central Universities under Grant JZ2022HGTB0344. The Associate Editor for this article was S. Garg. *(Corresponding author: Lei Liu.)*

Momiao Zhou and Yanshi Sun are with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China (e-mail: mmzhou@hfut.edu.cn; sys@hfut.edu.cn).

Lei Liu is with the Guangzhou Institute of Technology, Xidian University, Guangzhou 510555, China, and also with the Xidian Guangzhou Institute of Technology, Guangzhou 510555, China (e-mail: tianjiaoliulei@163.com).

Kan Wang is with the Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China (e-mail: wangkan@xaut.edu.cn).

Mianxiong Dong is with the Department of Information and Electronic Engineering, Muroran Institute of Technology, Muroran 050-8585, Japan (e-mail: mxdong@mmm.muroran-it.ac.jp).

Mohammed Atiquzzaman is with the School of Computer Science, University of Oklahoma, Norman, OK 73019 USA (e-mail: atiq@ou.edu).

Schahram Dustdar is with the Distributed Systems Group, TU Wien, 1040 Vienna, Austria (e-mail: dustdar@dsg.tuwien.ac.at).

Digital Object Identifier 10.1109/TITS.2023.3279322

FD radio has the capability of transmitting and receiving at the same time on the same frequency, thereby offering twice the time efficiency of the HD radio. In this regard, FD radio is supposed to be applicable to vehicular communications where the safety message delivery with ultra-low latency is of the highest priority [3].

A promising use case of FD-enabled vehicles is to serve as relays in two-hop vehicle-to-vehicle (V2V) communications as their receiving and forwarding processes can run simultaneously. Then the end-to-end delay can at most be reduced by half w.r.t. the legacy hop-by-hop delivery, which would finally benefit the driving safety of intelligent transportation systems. Such potential has been confirmed by some existing works. For example, a full-duplex relaying (FDR) system for use in platooning was introduced in [4], and simulation results demonstrated the significant performance gain with FDR over classical half-duplex relaying in terms of packet delivery ratio and physical-layer latency. In [5], an FD store-carry-forward scheme for intermittently connected vehicles was proposed, where the FD capability of the relay was exploited to extend the effective communication time with the target vehicle.

The existing works only consider a given two-hop V2V path with a fixed relay node. However, multi-hop (more than two hops) vehicular communication is indispensable to assist cooperative driving in future intelligent transportation systems, and there may be multiple feasible paths from source to destination. Thus, the routing efficiency is of great significance. In legacy HD-based vehicular networks, aiming at getting the delay-minimized routing path from source to destination, the Dijkstra algorithm is the most efficient approach [6], [7]. However, as mentioned above, if there are FD vehicles serving as relays, their receiving and forwarding processes can run simultaneously. Then the delay of a given end-to-end path is no longer the cumulative delay of all the hops. We wonder how does such difference affect the delay-minimized routing issues, which is exactly what motivates our work. This paper, extending preliminary results reported in the conference paper [8], primarily studies the delay-minimized routing problem in a vehicular ad-hoc network (VANET) with some of the vehicles being FD enabled. The main contributions of this paper are as follows.

- Based on the vehicle pattern (HD or FD) and the one-hop delay between pairwise vehicles, we establish the FD weighted graph, and present the generalized delay

calculation model for multi-hop paths in the FD weighted graph.
- We prove that the Dijkstra algorithm is unable to get the shortest (i.e. delay-minimized) source-to-destination path due to the fact that subpaths of shortest paths may not be shortest paths in the FD weighted graph.
- We propose two methods to solve the shortest-path problem in the FD weighted graph. The graph-based method can achieve the exact minimum end-to-end delay, while the deep reinforcement learning (DRL)-based method is more feasible due to its acceptable complexity.
- We conduct simulations to show that the DRL-based approach with proper hyper-parameters can achieve near minimum end-to-end delay, and the achieved delay has a notably decline as the number of FD nodes increases.

Compared with the conference version, this paper has the enhancements as follows. We prove in theory that the FD weighted graphs do not have the optimal-substructure property; we propose the DRL-based method to solve the shortest-path problem; we make complexity analysis to our proposed methods; we give more simulation results and discussions on our proposed methods.

The rest of this paper is organized as follows. Section II reviews the related work. Section III describes the system model. Section IV introduces the problem that we want to address. Sections V and VI present our proposed graph-based routing method and the DRL-based routing method, respectively. Section VII presents simulation-based performance analysis of our proposed methods. Finally, Section VIII concludes this paper.

II. Related Work

A. Development of FD Radios

For a long time, the greatest challenge of the FD technology was the terrible self-interference. Fortunately, the state-of-the-art SIC technique has achieved more than 120-dB SIC capability by together using propagation cancellation, analog cancellation and digital cancellation [9]. With increasingly-perfect SIC capability, FD radio has been favoured by cellular networks, wireless local area networks (WLAN) and cognitive networks, etc. In [10], authors showed that the sum-rate of FD-enabled small cell networks can benefit from the introduction of FD self-backhauling and the execution of adaptive power allocation. In [11], authors developed an analytical model of an FD medium access control (MAC) protocol for voice over WLANs, and it was verified that when a number of data connections are involved, FD communications can improve the network capacity significantly. In cognitive radio systems shown in [12], FD technique was employed by secondary users to simultaneously sense and access the vacant spectrum, which efficiently improves the spectrum utilization.

In spite of the harsh channel propagation environment w.r.t. the other type of networks, the technical feasibility of FD radios in VANETs is optimistic due to the fact that vehicular onboard units are good candidates to host high-end transceivers with unlimited power supply and large processing capacity [13]. Thus, some theoretical studies have been conducted. In [14], FD-enabled road side units (RSUs) that can assist V2V communications was considered, and it was shown that there is a great bit error rate (BER) reduction with the proposed SIC and beamforming method. Reference [15] considered a dual-hop V2V communication system equipped with FDRs in the millimeter-wave networks. The performance of this network was studied in a dense multi-lane highway considering cooperative best vehicular relay selection strategy. In [16], an enhanced carrier sense multiple access/collision avoidance (CSMA/CA) protocol was analyzed for vehicular networks, which improves the cooperative awareness messages (CAM) timeliness and reliability by leveraging FD transceivers on board. Reference [17] investigated the use of Long Term Evolution (LTE) direct communication among vehicles equipped with FD radios for the beaconing service. Compared to the use of LTE with the infrastructure, it was shown that the direct mode with FD avoids using the downlink and reduces the occupation of the uplink resource.

B. Routing in VANETs

The one-hop communication range at the carrier of 5.9 GHz is greatly limited by the complex outdoor environment. As such, multi-hop routing is required in VANETs. Traditional VANET is just one kind of ad-hoc networks, and the solutions for some theoretical problems, e.g., shortest path from source to destination, are still unchanged. The main concern of related works was the pragmatic routing protocols in the case that no vehicle knows the complete topology of the network.

Generally, the unicast routing protocols for VANETs can be clssified into topology based, position based, and cluster based. The topology-based protocols establish routing tables at each vehicle based on the topology information, which can be further categorized as being either proactive or reactive. Representative proactive protocols are Optimized Link State Routing (OLSR), Destination Sequenced Distance Vector (DSDV) and their enhancements [18], [19]. Representative reactive protocols are Ad-hoc On-Demand Distance Vector (AODV), Dynamic Source Routing (DSR) and their enhancements [20], [21]. Position-based protocols directly use geographical position of vehicles when selecting the best path to forward data. Representative position-based protocols are Greedy Perimeter Stateless Routing (GPSR) and its enhancements [22]. Cluster-based protocols partition the network into clusters, and the hierarchical routing (intra-cluster and inter-cluster) strategy is applied. Representative cluster-based protocols are Cluster Based Routing (CBR) and its enhancements [23]. A common underlying assumption of all the protocols is that the vehicles are only HD enabled.

It must be stated that our overriding concern in this paper is the possibly-changed shortest-path problem when FD radios can be equipped at vehicles. We assume that the complete topology of the network is available.

III. System Model

A. Network and Graph Models

As shown in Fig. 1(a), we consider a road segment where K vehicles move in the same direction with similar velocities. Some of the vehicles are equipped with FD radios, while the others are only equipped with HD radios. An ad-hoc network

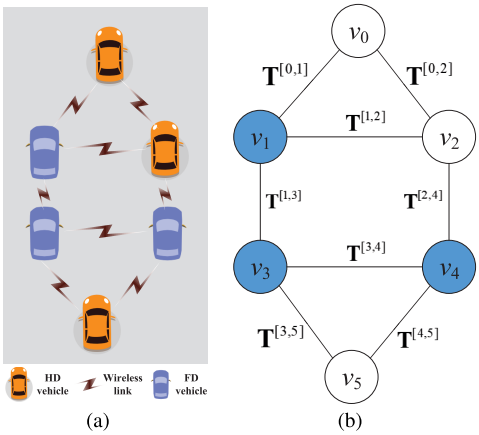


Fig. 1. Network and graph models. (a) Example of network topology. (b) FD weighted graph.

is established to provide inter-vehicle communications, where every vehicle can be the source, destination or relay node. If one vehicle sends data to another, then the total delay consists of the transmission delay and propagation delay. The network can be modeled as a weighted graph shown in Fig. 1(b). The solid vertices and hollow vertices represent FD vehicles and HD vehicles respectively; the edges represent the one-hop links. Given two adjacent vertices v_i and v_j, the weight of edge (v_i, v_j) is denoted as $\mathbf{T}^{[i,j]} = (T_{\mathrm{t}}^{[i,j]}, T_{\mathrm{p}}^{[i,j]})$, where $T_{\mathrm{t}}^{[i,j]}$ and $T_{\mathrm{p}}^{[i,j]}$ represent the transmission delay and the propagation delay respectively. $T_{\mathrm{t}}^{[i,j]}$ can be calculated by $T_{\mathrm{t}}^{[i,j]} = \frac{L}{R^{[i,j]}}$, where L is the data length and $R^{[i,j]}$ is the transmission rate of link (v_i, v_j). $T_{\mathrm{p}}^{[i,j]}$ can be calculated by $T_{\mathrm{p}}^{[i,j]} = \frac{D^{[i,j]}}{c}$, where $D^{[i,j]}$ is the distance from v_i to v_j, and c is the beam propagation velocity. It should be noted that the propagation delay is distance determined, and then there is $T_{\mathrm{p}}^{[i,j]} = T_{\mathrm{p}}^{[j,i]}$. Additionally, we assume that link (v_i, v_j) and (v_j, v_i) have the symmetric transmission rate, and thus there is $T_{\mathrm{t}}^{[i,j]} = T_{\mathrm{t}}^{[j,i]}$. In this regard, the graph is an undirected graph, hereafter referred to as FD weighted graph.

B. Delay Calculation Model

Legacy HD relay nodes perform the receiving and forwarding in a time division manner. When FD relaying is enabled, the receiving and forwarding can be performed simultaneously, which would result in the reduction of the end-to-end delay. We denote a path where all the relay nodes are FD enabled as complete FD relaying (C-FDR) path. Then the following theorem can be demonstrated.

Theorem 1: For an N-hop C-FDR path $\mathcal{P}$, where all the nodes (i.e. source, relays and destination) are sequentially indexed by $\mathcal{N} = \{0, 1, \ldots, N\}$, the end-to-end delay is

$$T_{\mathcal{P}} = \max_{i=\{1,2,\ldots,N\}} T_{\mathrm{t}}^{[\mathcal{N}(i),\mathcal{N}(i+1)]} + \sum_{i=1}^{N} T_{\mathrm{p}}^{[\mathcal{N}(i),\mathcal{N}(i+1)]}. \quad (1)$$

Proof: We graphically present the path in Fig. 2. Note that it does not matter whether the source/destination is FD enabled since they only utilize one of the receiving and transmitting

Fig. 2. N-hop path with complete FD relaying.

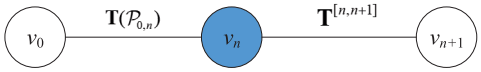


Fig. 3. Aggregated path from v_0 to v_{n+1} via v_n.

circuits. The theorem is proved by induction on N. When $N = 1$, node v_1 is the destination. This is an extreme case that there is no relay node, and then the total delay is just $T_{\mathrm{t}}^{[0,1]} + T_{\mathrm{p}}^{[0,1]}$. Obviously, equation (1) holds at $N = 1$. Then the inductive step is for $N = n$ $(n \geq 1)$, and according to (1) we assume that the end-to-end delay is

$$T_{\mathcal{P}_{0,n}} = \underbrace{\max_{i=\{1,\ldots,n\}} T_{\mathrm{t}}^{[\mathcal{N}(i),\mathcal{N}(i+1)]}}_{T_{\mathrm{t}}(\mathcal{P}_{0,n})} + \underbrace{\sum_{i=1}^{n} T_{\mathrm{p}}^{[\mathcal{N}(i),\mathcal{N}(i+1)]}}_{T_{\mathrm{p}}(\mathcal{P}_{0,n})}. \quad (2)$$

It should be noted that FD relaying does not impact the hop-by-hop propagation of data. Therefore, the total propagation delay is still the sum of the n-hop propagation delays, which is just $T_{\mathrm{p}}(\mathcal{P}_{0,n})$ in (2). Then it follows that $T_{\mathrm{t}}(\mathcal{P}_{0,n})$ in (2) is the total transmission delay.

If it comes to $N = n + 1$, the subpath from v_0 to v_n can be equivalently aggregated into one hop with the weight of $\mathbf{T}(\mathcal{P}_{0,n}) = (T_{\mathrm{t}}(\mathcal{P}_{0,n}), T_{\mathrm{p}}(\mathcal{P}_{0,n}))$, and then the path is shown as Fig. 3. After receiving the first packet from node v_0, the FD node v_n starts the forwarding operation. Node v_0, meanwhile, does not stop the first-hop's transmission. Due to time overlapping of the two hops, the total transmission delay can be approximately expressed as

$$\begin{aligned} T_{\mathrm{t}}(\mathcal{P}_{0,n+1}) &= \max\{T_{\mathrm{t}}(\mathcal{P}_{0,n}), T_{\mathrm{t}}^{[n,n+1]}\} \\ &= \max_{i=\{1,\ldots,n+1\}} T_{\mathrm{t}}^{[\mathcal{N}(i),\mathcal{N}(i+1)]}. \end{aligned} \quad (3)$$

On the other hand, the total propagation delay definitely increases by $T_{\mathrm{p}}^{[n,n+1]}$. Thus we have

$$\begin{aligned} T_{\mathcal{P}_{0,n+1}} &= T_{\mathrm{t}}(\mathcal{P}_{0,n+1}) + T_{\mathrm{p}}(\mathcal{P}_{0,n}) + T_{\mathrm{p}}^{[n,n+1]} \\ &= \max_{i=\{1,\ldots,n+1\}} T_{\mathrm{t}}^{[\mathcal{N}(i),\mathcal{N}(i+1)]} + \sum_{i=1}^{n+1} T_{\mathrm{p}}^{[\mathcal{N}(i),\mathcal{N}(i+1)]}, \end{aligned} \quad (4)$$

which completes the proof. ■

Remark 1: Theorem 1 indicates that the delay of a C-FDR path is subject to the slowest transmission rate of all the hops. In practice, however, C-FDR is always unavailable, i.e., FD relays and HD relays coexist within a given path. On this condition, we can deem the path as the connection of successive C-FDR subpaths. Then the end-to-end delay of the path can be obtained by adding up the delays of the C-FDR subpaths. Fig. 4 gives an example of a path which is the connection of two successive C-FDR subpaths, i.e., $\{v_0, v_1, v_2\}$ and $\{v_2, v_3, v_4, v_5\}$.

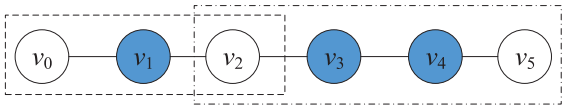


Fig. 4. Example of path with two successive C-FDR subpaths.

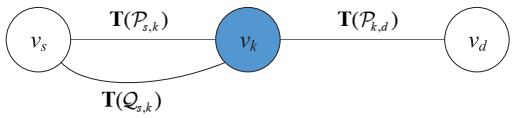


Fig. 5. Aggregated path from v_s to v_d via v_k.

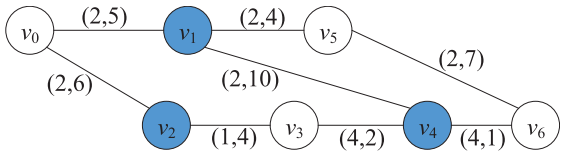


Fig. 6. An FD weighted graph without optimal substructure.

IV. Problem Statement

If there are multiple feasible paths from source to destination, then which path has the minimum end-to-end delay? Dijkstra's single-source shortest-path algorithm is the most representative approach to solve the delay-minimized routing problem [24,p.658]. The correctness of the Dijkstra algorithm relies on the optimal-substructure property of classic weighted graphs, i.e., subpaths of shortest paths are also shortest paths. Due to this property, the algorithm can repeatedly selects the vertex with the minimum shortest-path estimate until reaching the destination. With regard to the FD weighted graphs, however, we have the following astonishing proposition.

Proposition 1: ***(Subpaths of shortest paths may not be shortest paths in FD weighted graphs.)*** *In an FD weighted graph, let* $\mathcal{P}_{s,d} = \{\underbrace{v_s, \ldots, v_k}_{\mathcal{P}_{s,k}}, \ldots, v_d\}$ *be the shortest (i.e. delay-minimized) path from* v_s *to* v_d *where relay* v_k *is an FD node. Then* $\mathcal{P}_{s,k}$ *may not be the shortest path from* v_s *to* v_k.

Proof: First consider the case where $\mathcal{P}_{s,d}$ is a C-FDR path. Actually, $\mathcal{P}_{s,d}$ can equivalently be aggregated into two hops, i.e., $v_s \rightarrow v_k$ with the weight of $\mathbf{T}(\mathcal{P}_{s,k}) = (T_\mathrm{t}(\mathcal{P}_{s,k}), T_\mathrm{p}(\mathcal{P}_{s,k}))$ and $v_k \rightarrow v_d$ with the weight of $\mathbf{T}(\mathcal{P}_{k,d}) = (T_\mathrm{t}(\mathcal{P}_{k,d}), T_\mathrm{p}(\mathcal{P}_{k,d}))$. Then the end-to-end delay of $\mathcal{P}_{s,d}$ is

$$T_{\mathcal{P}_{s,d}} = \max\{T_\mathrm{t}(\mathcal{P}_{s,k}), T_\mathrm{t}(\mathcal{P}_{k,d})\} + T_\mathrm{p}(\mathcal{P}_{s,k}) + T_\mathrm{p}(\mathcal{P}_{k,d}). \quad (5)$$

Provided that $\mathcal{P}_{s,d}$ and another C-FDR path $\mathcal{Q}_{s,d}$ have different subpaths from v_s to v_k and common subpath from v_k to v_d (as shown in Fig. 5), then the end-to-end delay of $\mathcal{Q}_{s,d}$ can be given by

$$T_{\mathcal{Q}_{s,d}} = \max\{T_\mathrm{t}(\mathcal{Q}_{s,k}), T_\mathrm{t}(\mathcal{P}_{k,d})\} + T_\mathrm{p}(\mathcal{Q}_{s,k}) + T_\mathrm{p}(\mathcal{P}_{k,d}), \quad (6)$$

where $\mathcal{Q}_{s,k}$ is the subpath from v_s to v_k in $\mathcal{Q}_{s,d}$. Furthermore, the difference between the end-to-end delays of $\mathcal{P}_{s,k}$ and $\mathcal{Q}_{s,k}$ can be expressed as

$$\begin{aligned}\Delta T &= T_{\mathcal{P}_{s,k}} - T_{\mathcal{Q}_{s,k}} \\ &= T_\mathrm{t}(\mathcal{P}_{s,k}) - T_\mathrm{t}(\mathcal{Q}_{s,k}) + [T_\mathrm{p}(\mathcal{P}_{s,k}) - T_\mathrm{p}(\mathcal{Q}_{s,k})] \\ &\overset{(a)}{<} \underbrace{[T_\mathrm{t}(\mathcal{P}_{s,k}) + \max\{T_\mathrm{t}(\mathcal{Q}_{s,k}), T_\mathrm{t}(\mathcal{P}_{k,d})\}]}_{\alpha} \\ &\quad - \underbrace{[T_\mathrm{t}(\mathcal{Q}_{s,k}) + \max\{T_\mathrm{t}(\mathcal{P}_{s,k}), T_\mathrm{t}(\mathcal{P}_{k,d})\}]}_{\beta}, \end{aligned} \quad (7)$$

where inequality (a) comes from the fact that $T_{\mathcal{P}_{s,d}} < T_{\mathcal{Q}_{s,d}}$. In (7), if $T_\mathrm{t}(\mathcal{Q}_{s,k}) < \min\{T_\mathrm{t}(\mathcal{P}_{s,k}), T_\mathrm{t}(\mathcal{P}_{k,d})\}$, then $\alpha > \beta$. In such case, the polarity of ΔT cannot be judged from $\Delta T < \alpha - \beta$. If $\Delta T > 0$, it means $\mathcal{Q}_{s,k}$ is shorter than $\mathcal{P}_{s,k}$, which verifies the proposition.

Now consider the other case where $\mathcal{P}_{s,d}$ is a path with partial FD relaying. As is explained in Remark 1, $\mathcal{P}_{s,d}$ is the connection of successive C-FDR subpaths. The correctness of the proposition in the C-FDR case has been proved above, and thus the case of partial FD relaying surely meets the proposition. The proof is completed. ■

An example is shown in Fig. 6, where v_0 is assumed to be the source and v_6 be the destination. According to our proposed delay calculation model, $\{v_0, v_2, v_3, v_4, v_6\}$ is the shortest path from source to destination, while $\{v_0, v_1, v_4\}$ is the shortest path from v_0 to v_4, rather than $\{v_0, v_2, v_3, v_4\}$.

Proposition 1 indicates that the FD weighted graphs do not have the optimal-substructure property. Selecting shortest subpaths may not produce the shortest path. In this regard, the Dijkstra algorithm cannot be leveraged to calculate the delay-minimized routing solution for FD-based VANETs. In the following we will present two coping strategies.

V. Graph-Based Routing Method

This section presents a graph-based routing method for FD-based VANETs. First, the network topology is reformulated as an equivalent graph by decoupling all the FD-related links. The reformulated graph has the property that subpaths of a shortest path are also shortest paths. Then an evolved-Dijkstra algorithm is proposed, which can find the shortest routing path of the reformulated graph with low complexity.

A. Graph Reformulation

The FD weighted graph can be reformulated into an equivalent one with the following steps.

a) Within the FD weighted graph that has been established, find all the maximal FD-connected subgraphs. Here FD-connected subgraph refers to the connected subgraph which is composed of FD vertices only.
b) Extend each maximal FD-connected subgraph with all of its adjacent HD vertices and edges included.
c) For each extended subgraph $\mathcal{G}_m$, and each FD vertex $v_i \in \mathcal{G}_m$, find all the paths between every pairwise HD vertices in $\mathcal{G}_m$. The paths must pass v_i and not pass HD vertices except the source and destination.
d) Use a set $\mathbb{P}_i$ to accommodate the paths that have been found for each FD vertex v_i in the previous step, the cardinality of which is denoted as $\gamma^{[i]}$, i.e., $|\mathbb{P}_i| = \gamma^{[i]}$.
e) Split each FD vertex v_i into $\gamma^{[i]}$ FD vertices. Then reconnect adjacent vertices following the paths in $\mathbb{P}_i$,

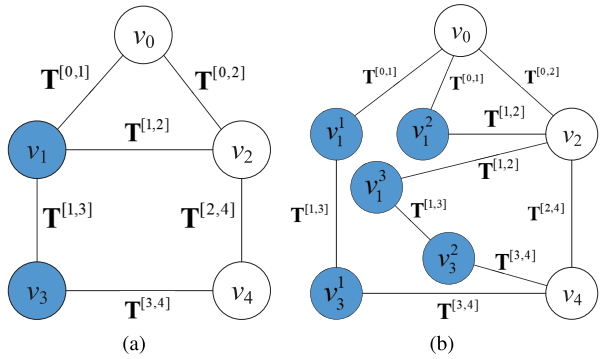


Fig. 7. Example of graph reformulation. (a) An extended maximal FD-connected subgraph. (b) Reformulation of the subgraph.

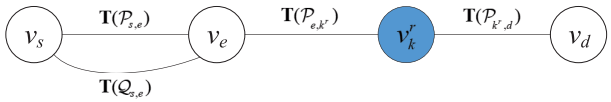


Fig. 8. Aggregated path from v_s to v_d via v_e and v_k^r.

which must finally ensure that every FD vertex belongs to one path only.

Fig. 7 shows an example of the extended maximal FD-connected subgraph and its equivalent reformulation. It can be observed that v_1 was split into v_1^1, v_1^2 and v_1^3; v_3 was split into v_3^1 and v_3^2. Fig. 7(b) decouples three paths of Fig. 7(a), i.e., $\{v_0, v_1, v_3, v_4\}$, $\{v_0, v_1, v_2\}$ and $\{v_2, v_1, v_3, v_4\}$. With regard to the reformulated graphs, we have the following proposition.

Proposition 2: ***(Subpaths of shortest paths are shortest paths in reformulated graphs.)*** *In the reformulated graph, let* $\mathcal{P}_{s,d} = \{\underbrace{v_s, \ldots, v_k}_{\mathcal{P}_{s,k}}, \ldots, v_d\}$ *be the shortest path from* v_s *to* v_d *where* v_k *is one of the relays. Then* $\mathcal{P}_{s,k}$ *is the shortest path from* v_s *to* v_k.

Proof: Two cases should be considered. First, if v_k is an HD vertex, $\mathcal{P}_{s,d}$ can equivalently be aggregated into two hops, i.e., $v_s \rightarrow v_k$ with the weight of $\mathbf{T}(\mathcal{P}_{s,k}) = (T_\mathrm{t}(\mathcal{P}_{s,k}), T_\mathrm{p}(\mathcal{P}_{s,k}))$ and $v_k \rightarrow v_d$ with the weight of $\mathbf{T}(\mathcal{P}_{k,d}) = (T_\mathrm{t}(\mathcal{P}_{k,d}), T_\mathrm{p}(\mathcal{P}_{k,d}))$. Then the end-to-end delay of $\mathcal{P}_{s,d}$ is

$$T_{\mathcal{P}_{s,d}} = \underbrace{T_\mathrm{t}(\mathcal{P}_{s,k}) + T_\mathrm{p}(\mathcal{P}_{s,k})}_{T_{\mathcal{P}_{s,k}}} + \underbrace{T_\mathrm{t}(\mathcal{P}_{k,d}) + T_\mathrm{p}(\mathcal{P}_{k,d})}_{T_{\mathcal{P}_{k,d}}}. \quad (8)$$

Provided that $\mathcal{P}_{s,d}$ and another path $\mathcal{Q}_{s,d}$ have different subpaths from v_s to v_k and common subpath from v_k to v_d, then the end-to-end delay of $\mathcal{Q}_{s,d}$ is

$$T_{\mathcal{Q}_{s,d}} = \underbrace{T_\mathrm{t}(\mathcal{Q}_{s,k}) + T_\mathrm{p}(\mathcal{Q}_{s,k})}_{T_{\mathcal{Q}_{s,k}}} + \underbrace{T_\mathrm{t}(\mathcal{P}_{k,d}) + T_\mathrm{p}(\mathcal{P}_{k,d})}_{T_{\mathcal{P}_{k,d}}}, \quad (9)$$

where $\mathcal{Q}_{s,k}$ is the subpath from v_s to v_k in $\mathcal{Q}_{s,d}$. Since $\mathcal{P}_{s,d}$ is the shortest path, there is $T_{\mathcal{P}_{s,d}} < T_{\mathcal{Q}_{s,d}}$. Then from (8) and (9), $T_{\mathcal{P}_{s,k}} < T_{\mathcal{Q}_{s,k}}$ can be derived, i.e., $\mathcal{P}_{s,k}$ is the shortest path from v_s to v_k.

Now consider the other case where v_k is an FD vertex. v_k is actually a vertex after splitting, which could be more precisely denoted as v_k^r to avoid confusion. In $\mathcal{P}_{s,d}$, we assume that v_e is the nearest HD vertex in front of v_k^r. The aggregated path is shown in Fig. 8. Then according to the proof of the first case, the subpath $\mathcal{P}_{s,e}$ must be the shortest path from v_s to v_e. Furthermore, it should be noted that after the graph reformulation, the degree of each FD vertex does not exceed 2. Therefore, the subpath $\mathcal{P}_{e,k^r}$ is the only path from v_e to v_k^r. Provided that $\mathcal{P}_{s,d}$ and another path $\mathcal{Q}_{s,d}$ have different subpaths from v_s to v_e and common subpath from v_e to v_d, then we have $T_{\mathcal{P}_{s,k^r}} = T_{\mathcal{P}_{s,e}} + T_{\mathcal{P}_{e,k^r}}$ and $T_{\mathcal{Q}_{s,k^r}} = T_{\mathcal{Q}_{s,e}} + T_{\mathcal{P}_{e,k^r}}$, where $T_{\mathcal{P}_{s,k^r}}$, $T_{\mathcal{Q}_{s,k^r}}$, $T_{\mathcal{P}_{s,e}}$, $T_{\mathcal{Q}_{s,e}}$ and $T_{\mathcal{P}_{e,k^r}}$ respectively stand for the end-to-end delay of subpath $\mathcal{P}_{s,k^r}$, $\mathcal{Q}_{s,k^r}$, $\mathcal{P}_{s,e}$, $\mathcal{Q}_{s,e}$ and $\mathcal{P}_{e,k^r}$. It is obvious that $T_{\mathcal{P}_{s,k^r}} < T_{\mathcal{Q}_{s,k^r}}$ holds due to $T_{\mathcal{P}_{s,e}} < T_{\mathcal{Q}_{s,e}}$, i.e., $\mathcal{P}_{s,k^r}$ is the shortest path from v_s to v_k^r. The proof is completed. ■

Proposition 2 indicates that the reformulated graphs have the optimal-substructure property, where the Dijkstra algorithm is assuredly applicable. However, since the number of vertices and edges both expand during reformulation, the computation complexity of the Dijkstra algorithm would inevitably have an increase. We will then present an evolved-Dijkstra algorithm dedicated for the reformulated graphs, with which the computation complexity can be greatly reduced.

B. Evolved-Dijkstra Algorithm

The Dijkstra algorithm calculates the delay from the source to each visited vertex. The evolved-Dijkstra algorithm is designed based on the fact that the degree of each FD vertex does not exceed 2 in the reformulated graphs. When we visit an FD vertex along whichever route, there is only one next-hop vertex at most. Therefore, we can skip the delay calculation at this FD vertex (if it is not the destination) and straightly visit its next-hop vertex. The skipping repeats until an HD vertex is visited. That is, the delay calculation and updating are only performed at HD relays and the destination. Specifically, the algorithm is given as Algorithm 1.

Algorithm 1 Evolved-Dijkstra Algorithm

Input $\widehat{\mathcal{G}} = (\widehat{\mathcal{V}}, \widehat{\mathcal{E}})$, $\mathcal{V}_h$, v_s and v_d.
Initialize $v^* = v_s$, $\tau[v_s] = 0$, $\tau[v_k] = \infty$, $\forall v_k \in \mathcal{V}_h \backslash \{v_s\}$.
while $v_d \in \mathcal{V}_h$ **do**
  Delete the vertex $u = \arg\min_{v_k \in \mathcal{V}_h} \{\tau[v_k]\}$ from $\mathcal{V}_h$.
  if $u \neq v_d$ **then**
    for each adjacent vertex of u **do**
      Visit the vertex and update v^*.
      while v^* is an FD vertex **do**
        Visit its next-hop vertex and update v^*.
      end while
      if $v^* \in \mathcal{V}_h$ **then**
        $\tau[v^*] = \min\{\tau[v^*], \tau[u] + T[uv^*]\}$.
        Store the shortest path from v_s to v^*.
      end if
    end for
  end if
end while
Output $\tau[v_d]$ and the shortest path from v_s to v_d.

In Algorithm 1, $\widehat{\mathcal{G}} = (\widehat{\mathcal{V}}, \widehat{\mathcal{E}})$ is the reformulated graph, where $\widehat{\mathcal{V}}$ and $\widehat{\mathcal{E}}$ stand for the vertices and edges respectively. $\mathcal{V}_h \subseteq \widehat{\mathcal{V}}$ is the collection of all the HD vertices. Here the source v_s and destination v_d are both counted as HD vertices since they only utilize one of the receiving and transmitting circuits. v^* represents the vertex being visited currently. $T[uv^*]$ is the total delay from u to v^* via C-FDR.

C. Complexity Analysis

First, for steps a)-b) of the graph reformulation in Section V, use the Depth First Search (DFS) method to traverse the adjacency matrix of the FD vertices, and then all the maximal FD-connected subgraphs and their extended subgraphs can be obtained [24,p.603]. The number of FD vertices can be expressed as $|\mathcal{V}| - |\mathcal{V}_h|$, where $\mathcal{V}$ stands for the vertices of the graph before reformulation. As such, the complexity is $\mathcal{O}((|\mathcal{V}|-|\mathcal{V}_h|)|\mathcal{V}|)$. Provided that the largest extended subgraph has M vertices, then for steps c)-e) of the graph reformulation, the complexity does not exceed $\mathcal{O}(LM!)$, where L is the number of extended subgraphs. Finally, Algorithm 1 traverses the adjacency matrix of the HD vertices in the reformulated graph, which has the complexity of $\mathcal{O}(|\mathcal{V}_h||\widehat{\mathcal{V}}|)$. The above analysis shows that the running speed of the graph-based routing method is mainly limited by the size of $M!$. If the FD nodes are sparsely distributed, then $M!$ can be an acceptable value; otherwise, steps c)-e) has quite high complexity. That is to say, the graph-based routing method is not practical in some cases, despite its correctness in acquiring the minimum end-to-end delay.

VI. DRL-Based Routing Method

As one of the machine learning approaches, DRL embeds the strong fitting capability of deep learning into reinforcement learning so that the agent can achieve powerful decision-making ability even in a high-dimensional input environment [26], [27]. Since a complex routing problem is just like an enormous decision-making game, we believe that DRL has the potential to address it with low complexity. In this section we attempt to give a DRL-based method to solve the shortest routing problem for FD-based VANETs. First, a data packet that needs to be delivered can act as the agent. The agent learn the environment through gaining rewards when it moves from any vertex to another in network. Here we give some definitions that will be used.

A. State Space

The state refers to the location of the agent. If the agent is located at vertex v_i, then its current state is v_i. Since there are K vehicles in the network, the state space can be denoted as $\mathcal{S} = \{v_1, v_2, \ldots, v_K\}$.

B. Action Space

The action determines the state transition of the agent. If the agent moves from state v_i to state v_j, then we directly use v_j to represent this action. Note that v_j is also one of the states. Therefore, the same as the state space, the action space can be denoted as $\mathcal{A} = \{v_1, v_2, \ldots, v_K\}$. A typical Markov decision process (MDP) of the agent is $s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \cdots$, with $s_t \in \mathcal{S}$, $a_t \in \mathcal{A}$ and $s_{t+1} = a_t$.

1) Design of Reward Function: The reward is used to evaluate the influence (positive, negative or none) of an action on the search of the shortest path. As such, the reward is an important hyper-parameter that should be designed for each state-action pair. In this paper, we propose the reward function as follows.

Given $s_t = v_i$, $s_{t+1} = a_t = v_j$ ($\forall v_i \in \mathcal{S}, v_j \in \mathcal{A}$), different cases should be considered. First, if v_j is not adjacent to v_i in the FD weighted graph, then $v_i \to v_j$ is not a feasible hop in practice. It is similar to the case that the agent hits walls in the maze problem [25]. Therefore, the reward should be a great negative value, i.e., $r_{i,j} = -R_1$.

If v_j is adjacent to v_i, then the reward should be related to the one-hop delay of $v_i \to v_j$. Since delay is undesired in communication, a larger delay should be given a smaller reward. Then the reward function can be given by

$$r_{i,j} = -(T_{\mathrm{t}}^{[i,j]} + T_{\mathrm{p}}^{[i,j]}) + \delta_{i,j}, \tag{10}$$

where $\delta_{i,j}$ is the additional reward apart from the impact of the $v_i \to v_j$ delay. Specifically, $\delta_{i,j}$ is expressed as

$$\delta_{i,j} = \begin{cases} R_2, & \text{C1} \\ \dfrac{\sum\limits_{v_x \in \mathcal{C}_j \setminus \{v_i\}} \min\{T_{\mathrm{t}}^{[j,x]}, T_{\mathrm{t}}^{[i,j]}\}}{|\mathcal{C}_j| - 1}, & \text{C2} \\ 0, & \text{C3}. \end{cases} \tag{11}$$

In (11), C1 represents the case that v_j is just the destination, and then a great positive value R_2 is given as the additional reward. C2 represents the case that v_j is not the destination but is an FD vertex. $\mathcal{C}_j$ is the collection of vertices that are adjacent to v_j. Due to the fact that v_j is capable of reducing the delay by $\min\{T_{\mathrm{t}}^{[j,x]}, T_{\mathrm{t}}^{[i,j]}\}$ via FD radio in the path $v_i \to v_j \to v_x$ ($\forall v_x \in \mathcal{C}_j \setminus \{v_i\}$), $\delta_{i,j}$ for C2 is the average delay reduction with $\forall v_x$. Finally, C3 stands for all the other cases, and they have no influence on path searching.

The reward can only estimate the immediate return of an action. Q value is the extension of the reward, which also considers the long-term feedback of an action. For a state-action pair (s_t, a_t), its Q value can be given by

$$Q(s_t, a_t) = r_{s_t,a_t} + \gamma \max_{a \in \mathcal{A}} Q(s_{t+1}, a), \tag{12}$$

where $\gamma < 1$ is the discount factor of the maximal future rewards. Then the goal of the agent is to learn a policy that can get the maximum cumulative Q values from source to destination.

2) DQN-Based Algorithm: DQN is an off-policy DRL model that can get the optimal policy with substantial training episodes [28]. During each training episode, the agent iteratively uses the ε-greedy strategy to perform state transition and update Q values until reaching the destination or the number of iterations exceeds a certain limitation. Specifically, to balance exploration and exploitation, the agent selects the next-step action with the largest Q value with probability $1 - \varepsilon$ and selects a random next-step action with probability ε. Usually,

Algorithm 2 DQN-Based Routing Algorithm

Input state space $\mathcal{S}$, action space $\mathcal{A}$, source state $v_s \in \mathcal{S}$, destination state $v_d \in \mathcal{S}$, reward matrix $\mathbf{R}_{K\times K} = [r_{i,j}]$.
Initialize the routing path $\mathcal{P} = \{v_s\}$.
Initialize the evaluation network with random weights $\boldsymbol{\omega}$.
Initialize the target network with weights $\boldsymbol{\omega}' = \boldsymbol{\omega}$.
for episode=1:$M_{\max}$ **do**
  Set $t = 1$, $s_t = v_s$.
  while $s_t \neq v_d$ and $t < t_{\max}$ **do**
    Get the vector of Q values for s_t in the evaluation network.
    Select an action a_t with the ε-greedy strategy.
    Store $(s_t, a_t, r_{s_t,a_t}, s_{t+1})$ in the ERU.
    Sample random mini-batch transitions $(s_\sigma, a_\sigma, r_{s_\sigma,a_\sigma}, s_{\sigma+1})$.
    Update target Q values of all samples (s_σ, a_σ) with (13).
    Update $\boldsymbol{\omega}$ in the evaluation network using the SGD algorithm.
    if mode$(t, M_{\text{cp}}) = 0$ **then**
      Reset $\boldsymbol{\omega}' = \boldsymbol{\omega}$.
    end if
    Set $t = t + 1$.
  end while
end for
Reset $t = 1$, $s_t = v_s$.
while $s_t \neq v_d$ **do**
  Get the vector of Q values for s_t in the evaluation network.
  Select the action $a_t = \arg\max_{a\in\mathcal{A}} Q(s_t, a)$.
  Add $s_{t+1} = a_t$ to $\mathcal{P}$.
  Set $t = t + 1$.
end while
Output $\mathcal{P}$.

ε linearly decays from 1 to 0 as the training episode grows. Then we will show how to update Q values after the ε-greedy transition.

DQN employs the deep neural network (DNN) to produce a function $f(s_t, a_t, \boldsymbol{\omega})$ that approximates $Q(s_t, a_t)$, with which the curse of dimensionality can be avoided [29]. $\boldsymbol{\omega}$ in $f(s_t, a_t, \boldsymbol{\omega})$ is the weight matrix of the DNN. The input of the DNN is s_t, and the output is a vector of Q values for all the next-step actions of s_t. Given a fixed DNN structure, then $\boldsymbol{\omega}$ is the only parameter that needs to be updated through training. To achieve this, two neural networks with the same structure are established, i.e., the evaluation network and the target network, in which the former is the main network to train $\boldsymbol{\omega}$ and the latter provides target Q values to the main network. The weights of the target network are copied from the evaluation network every M_{cp} steps of ε-greedy transition. In DQN, there is an experience replay unit (ERU) that stores each ε-greedy transition in memory with the format of $(s_t, a_t, r_{s_t,a_t}, s_{t+1})$. Once the ERU has updates, random mini-batch of transitions $(s_\sigma, a_\sigma, r_{s_\sigma,a_\sigma}, s_{\sigma+1})$ are sampled and delivered to both of the two networks. Then the target Q values are given by

$$Q(s_\sigma, a_\sigma) = \begin{cases} r_{s_\sigma,a_\sigma}, & s_{\sigma+1} = v_d \\ r_{s_\sigma,a_\sigma} + \gamma \max\limits_{a\in\mathcal{A}} q(s_{\sigma+1}, a), & s_{\sigma+1} \neq v_d \end{cases} \tag{13}$$

where $q(s_{\sigma+1}, a)$ is the Q value of $(s_{\sigma+1}, a)$ produced by the target network; v_d is the destination vertex. With the target $Q(s_\sigma, a_\sigma)$, the following loss function is minimized in the evaluation network using the stochastic gradient descent (SGD) algorithm:

$$L(\boldsymbol{\omega}) = \frac{1}{K} \sum_{a_\sigma\in\mathcal{A}} (Q(s_\sigma, a_\sigma) - f(s_\sigma, a_\sigma, \boldsymbol{\omega}))^2. \tag{14}$$

In this way, $\boldsymbol{\omega}$ can be updated.

After $M_{\max}$ training episodes, the agent greedily selects action with the largest Q value until reaching the destination. Algorithm 2 shows the overall procedure of the DQN-based routing algorithm, where $\mathcal{P}$ is just the routing solution.

3) Complexity Analysis: Obviously, the complexity of the DQN-based algorithm mainly depends on the training stage, whose worst-case complexity can be expressed as $\mathcal{O}(M_{\max} t_{\max})$. $M_{\max}$ is the number of episodes, and $t_{\max}$ limits the number of actions in each episode. Generally, $M_{\max}$ and $t_{\max}$ are set linearly scaling with the size of the state space. We will show in the simulations that hundreds of episodes and tens of actions in each episode are enough to achieve near optimal solutions in a 10-node network. As such, the DRL-based method is more feasible in addressing the shortest routing problem for FD-based VANETs.

TABLE I
BASIC SYSTEM SETTINGS OF DQN

Parameter	Value
$t_{\max}$, M_{cp}	20, 200
Capacity of the ERU	1000
R_1, R_2	40, 25

VII. SIMULATION RESULTS AND DISCUSSIONS

In this section, MATLAB is used to conduct simulations to evaluate the performance of our proposed routing methods. The hardware configuration is 12th Generation Intel Core i5-1235U CPU and Samsung DDR4 16GB RAM. A four-lane road segment with 10 m width and 100 m length is considered. 10 vehicles establish a network and each vehicle randomly selects one of these four lanes. Within each lane, the time headway follows exponential distribution with the parameter of 1 [30]. The velocity of vehicles is 72 km/h. The communication range of vehicles is set to 20 m. The transmission rate of each link is randomly chosen from 1 Mbps, 2 Mbps and 4 Mbps in our simulations. Basic system settings of DQN are given in Table I.

A. Parametric Study

The DRL-based method has a group of tunable hyper-parameters. First, we will study the impact of several critical

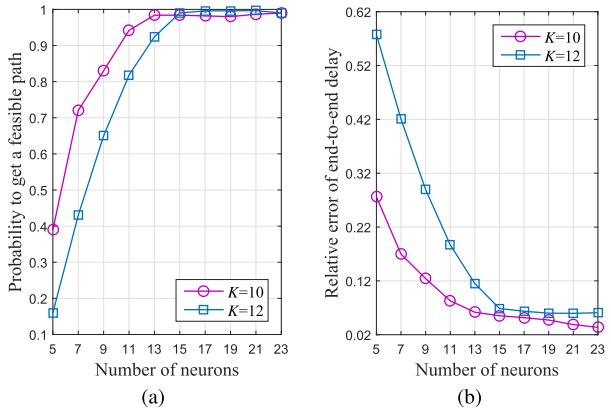


Fig. 9. The impact of the number of neurons. (a) Probability to get a feasible path vs. number of neurons. (b) Relative error of end-to-end delay vs. number of neurons.

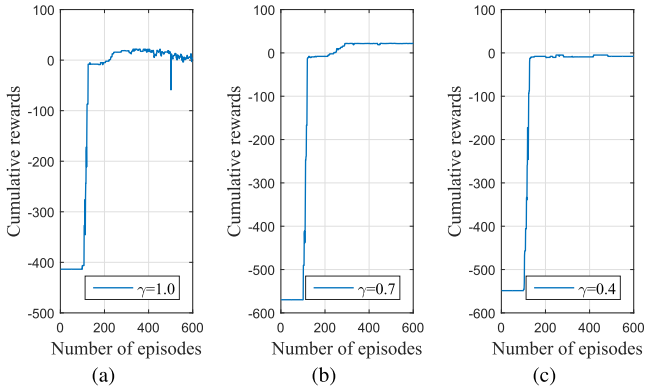


Fig. 10. Convergency impacted by γ. (a)$\gamma = 1.0$. (b)$\gamma = 0.7$. (c)$\gamma = 0.4$.

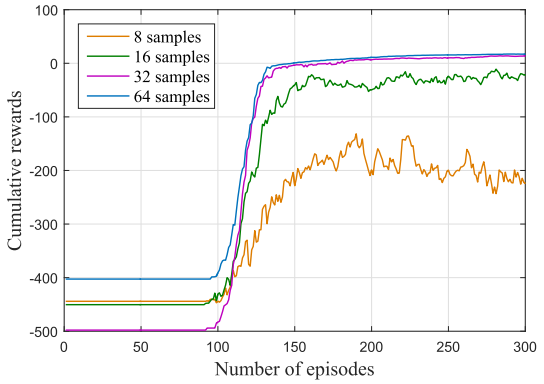


Fig. 11. Convergency impacted by size of mini-batch samples.

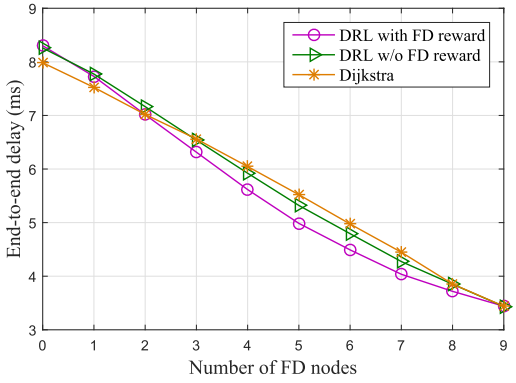


Fig. 12. Comparison of DRL performance with and without FD reward.

hyper-parameters on the performance of DRL and make an intuitive analysis to the optimal settings of them.

First, we study the impact of the number of neurons used in the hidden layer of DNN in Fig. 9. $\gamma = 0.7$ and $M_{\max} = 400$ are given. The size of the mini-batch samples is set to 64. Fig. 9(a) shows the probability that the DRL-based method can get a feasible path from source to destination with given number of neurons. The feasible path means that there is no disconnected hop within the path. We can see that the probability increases with the increase of the number of neurons. The reason is that the DNN with few neurons cannot provide high fitting accuracy of Q values, and therefore the agent inevitably hits the disconnected hops. Particularly, we also give the simulation results of the network with 12 vehicles (i.e., $K = 12$) for comparison. We find that the probability to get a feasible path approximates 1 when the number of neurons reaches 13 at $K = 10$ and 15 at $K = 12$. As such, it is better to set the number of neurons a bit greater than the number of vehicles to guarantee the robustness the DRL-based method.

Even if the DRL-based method gets a feasible path from source to destination, its accuracy is also impacted by the number of neurons. Fig. 9(b) shows the relative error of the DRL-based end-to-end delay with respect to the minimum end-to-end delay. As anticipated, the relative error declines with the increase of the number of neurons until converging to a fine level (0.03 for $K = 10$ and 0.06 for $K = 12$). Particularly, the relative error converges when the number of neurons reaches 13 at $K = 10$ and 15 at $K = 12$. As such, it can be stated that if the number of neurons is a bit greater than the number of vehicles, the DRL-based method can achieve the near minimum end-to-end delay.

Then in Fig. 10 we study the impact of the discount factor on the convergency of the DRL-based method. 15 neurons are given to the 10-vehicle network. The size of the mini-batch samples is set to 64. The cumulative reward versus the number of episodes is observed. It is shown that when $\gamma = 1.0$, the cumulative reward cannot get convergence within 600 episodes. This is because if $\gamma = 1.0$, the agent cares too much about the future return of an action. On the contrary, when $\gamma = 0.7$ or $\gamma = 0.4$, the cumulative reward can get convergence within 300 episodes. However, the cumulative reward after convergence at $\gamma = 0.4$ is -8.0011, while the cumulative reward after convergence at $\gamma = 0.7$ is 22.0398. This means that if the discount factor is too small, the agent cares little about the future return of an action and as a result the algorithm may fall into local optimum.

Furthermore, the impact of the size of mini-batch samples on the convergency of the DRL-based method is studied in Fig. 11. We set $\gamma = 0.7$. 15 neurons are given to the 10-vehicle network. The cumulative reward versus the number of episodes is observed. It is demonstrated that the cumulative reward cannot get convergence within 300 episodes when the size of mini-batch samples is 8 or 16. The reason is that too

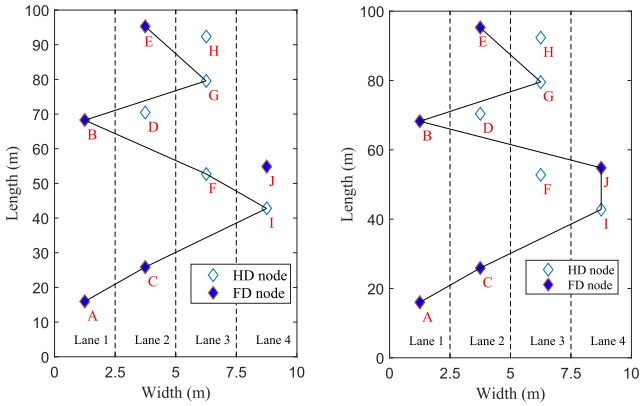


Fig. 13. Routing solutions for a given topology. (a) The Dijkstra algorithm. (b) Our proposed methods.

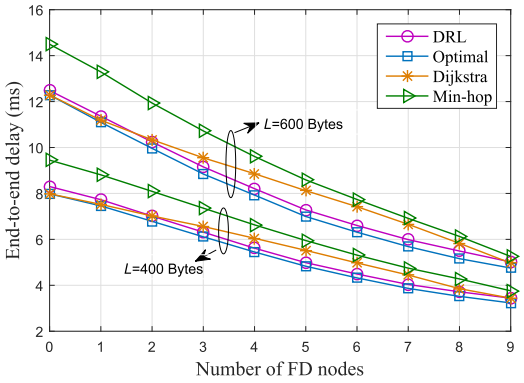


Fig. 14. End-to-end delay vs. number of FD nodes.

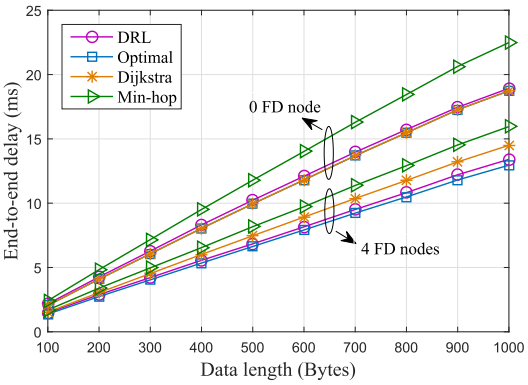


Fig. 15. End-to-end delay vs. data length.

few samples cannot provide enough labeled training data for the DNN. When the size of mini-batch samples is 32 or 64, the cumulative reward can get convergence quickly and their convergent cumulative rewards are very close.

Finally, we compare the DRL performance with and without FD reward in Fig. 12. Specifically, in the C2 case of Eqn. (11), we give an additional reward to FD vertices, which is just the "DRL with FD reward" in Fig. 12. If this additional reward is not given, then there is $\delta_{i,j} = 0$ in the C2 case of Eqn. (11), which is just the "DRL w/o FD reward" in Fig. 12. Obviously, DRL without FD reward can only achieve the comparable performance to the Dijkstra algorithm. This is because they both ignore the capability of FD radios. As such, it is verified that our proposed reward function in this paper is effective in modeling the capability of FD radios.

B. Routing Solution

Given a topology shown as Fig. 13, the routing solutions are simulated, with A and E being the source and destination respectively, and the data length being 400 Bytes. Fig. 13(a) shows the routing solution of the Dijkstra algorithm, i.e., A→C→I→F→B→E. The end-to-end delay of this path is 7.2 ms. Fig. 13(b) shows the routing solution of our proposed methods. In the DRL-based method, we determine the hyper-parameters empirically based on the above parametric study. Specifically, DNN we established has one hidden layer with 15 neurons; size of mini-batch samples is 64; γ is set to 0.7; $M_{\max}$ is set to 400. We can see that both of the graph-based method and the DRL-based method can get the optimal solution, i.e., A→C→I→J→B→E. The end-to-end delay of this path is only 6.4 ms.

C. Performance Comparison

To complete the performance analysis, we compare the end-to-end delays achieved by different routing methods (DRL, Dijkstra, Min-hop) with the exact minimum end-to-end delay (Optimal). "Min-hop" is a classic routing method which minimizes the number of hops from source to destination. "Optimal" is just the outcome of our proposed graph-based method. The DNN structure and hyper-parameters used in the DRL-based method are the same as the ones given in the previous subsection.

Fig. 14 shows the end-to-end delay averaged over 200 random topologies versus the number of FD nodes. It can be observed that for whichever method, the end-to-end delay declines with the increase of the number of FD nodes. Obviously it benefits from the improvement of the relay efficiency. Importantly, the DRL-based method achieves the very close performance to the optimal one, which verifies the feasibility of DRL for addressing the FD-based shortest-path problem. For given number of FD nodes, the min-hop method has the worst performance among all the methods due to the fact that less hops does not mean less delay. Moreover, the performance gap between the Dijkstra algorithm and the optimal one at intermediate number of FD nodes is bigger than that at small or great number of FD nodes. This is because when the number of FD nodes is comparable to HD nodes, the optimal routing chooses more FD relay nodes, while the Dijkstra algorithm chooses more HD relay nodes.

Fig. 15 shows the end-to-end delay averaged over 200 random topologies versus data length. We notice that for whichever routing method, the end-to-end delay increases linearly with the increase of data length. The reason is that the millisecond transmission delay scales as the data length, and the microsecond propagation delay can be overlooked. As anticipated, the DRL-based method can achieve the near minimum delay no matter whether there are FD nodes. Meanwhile, the Dijkstra algorithm can achieve the exact minimum

delay in the 0-FD-node case, but it loses efficacy in the 4-FD-node case. Particularly, as the data length increases, the minimum delay of 4-FD-node case grows much slower than that of 0-FD-node case. This exactly illustrates the superiority of FD radios in relaying large-sized inter-vehicle communication data.

VIII. Conclusion

The delay-minimized routing problem of the FD-based VANETs was studied. First, the FD weighted graph model and the delay calculation model were presented. The delay calculation model indicates that the delay of a C-FDR path is subject to the slowest transmission rate of all the hops. Then a proposition was proposed to show that the Dijkstra algorithm is unable to get the delay-minimized source-to-destination path in the FD weighted graph. To handle this, we proposed a graph-based method, which consists of network topology reformulation and the evolved-Dijkstra algorithm. In order to avoid extra-high complexity, we also proposed a DRL-based method, wherein DQN is employed to learn the shortest source-to-destination path. Simulations demonstrated that the DRL-based method can achieve an end-to-end delay which is close to the graph-based method. Future work is in progress to investigate FD-based routing protocols if each vehicle can only get partial topology, which would further enhance the study of FD-based VANETs.

References

[1] A. Sabharwal, P. Schniter, D. Guo, D. W. Bliss, S. Rangarajan, and R. Wichman, "In-band full-duplex wireless: Challenges and opportunities," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 9, pp. 1637–1652, Sep. 2014.

[2] M. Ma et al., "A prototype of co-frequency co-time full duplex networking," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 132–139, Feb. 2020.

[3] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2169–2182, Feb. 2023.

[4] M. S. Amjad, T. Hardes, M. Schettler, C. Sommer, and F. Dressler, "Using full duplex relaying to reduce physical layer latency in platooning," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2019, pp. 1–4.

[5] A. A. Siddig, A. S. Ibrahim, and M. H. Ismail, "Full-duplex store-carry-forward scheme for intermittently connected vehicular networks," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–6.

[6] F. Zhang, B. Jin, Z. Wang, H. Liu, J. Hu, and L. Zhang, "On geocasting over urban bus-based networks by mining trajectories," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1734–1747, Jun. 2016.

[7] V. T. N. Nha, S. Djahel, and J. Murphy, "A comparative study of vehicles' routing algorithms for route planning in smart cities," in *Proc. 1st Int. Workshop Veh. Traffic Manag. Smart Cities (VTM)*, Nov. 2012, pp. 1–6.

[8] M. Zhou, W. Xu, Z. Ding, and Y. Sun, "Delay-minimized routing for full-duplex vehicular ad-hoc Networks," in *Proc. IEEE 95st Veh. Technol. Conf.*, Jun. 2022, pp. 1–5.

[9] T. Zhang, C. Su, A. Najafi, and J. C. Rudell, "Wideband dual-injection path self-interference cancellation architecture for full-duplex transceivers," *IEEE J. Solid-State Circuits*, vol. 53, no. 6, pp. 1563–1576, Jun. 2018.

[10] K. Wang et al., "Interference alignment with adaptive power allocation in full-duplex-enabled small cell networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3010–3015, Mar. 2019.

[11] W. Kim, T. Song, T. Kim, H. Park, and S. Pack, "VoIP capacity analysis in full duplex WLANs," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11419–11424, Dec. 2017.

[12] Y. Liao, L. Song, Z. Han, and Y. Li, "Full duplex cognitive radio: A new design paradigm for enhancing spectrum usage," *IEEE Commun. Mag.*, vol. 53, no. 5, pp. 138–145, May 2015.

[13] C. Campolo, A. Molinaro, A. O. Berthet, and A. Vinel, "Full-duplex radios for vehicular communications," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 182–189, Jun. 2017.

[14] A. S. Ibrahim, "Self-interference cancellation and beamforming in repeater-assisted full-duplex vehicular communication," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5.

[15] K. Eshteiwi, G. Kaddoum, K. Ben Fredj, E. Soujeri, and F. Gagnon, "Performance analysis of full-duplex vehicle relay-based selection in dense multi-lane highways," *IEEE Access*, vol. 7, pp. 61581–61595, 2019.

[16] A. Bazzi, C. Campolo, B. M. Masini, A. Molinaro, A. Zanella, and A. O. Berthet, "Enhancing cooperative driving in IEEE 802.11 vehicular networks through full-duplex radios," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2402–2416, Apr. 2018.

[17] A. Bazzi, B. M. Masini, and A. Zanella, "Performance analysis of V2V beaconing using LTE in direct mode with full duplex radios," *IEEE Wireless Commun. Lett.*, vol. 4, no. 6, pp. 685–688, Dec. 2015.

[18] M. Kadadha, H. Otrok, H. Barada, M. Al-Qutayri, and Y. Al-Hammadi, "A Stackelberg game for street-centric QoS-OLSR protocol in urban vehicular ad hoc networks," *Veh. Commun.*, vol. 13, pp. 64–77, Jul. 2018.

[19] F. Belamri, S. Boulfekhar, and D. Aissani, "A survey on QoS routing protocols in vehicular ad hoc network (VANET)," *Telecommun. Syst.*, vol. 78, no. 1, pp. 117–153, Sep. 2021.

[20] X. Shen, Y. Wu, Z. Xu, and X. Lin, "AODV-PNT: An improved version of AODV routing protocol with predicting node trend in VANET," in *Proc. 7th IEEE/Int. Conf. Adv. INFOCOMM Technol.*, Nov. 2014, pp. 91–97.

[21] R. M. Kumar and S. K. Routray, "Ant colony based dynamic source routing for VANET," in *Proc. 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT)*, Jul. 2016, pp. 279–282.

[22] N. Lyu, G. Song, B. Yang, and Y. Cheng, "QNGPSR: A Q-network enhanced geographic ad-hoc routing protocol based on GPSR," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–6.

[23] S. Sennan, S. Ramasubbareddy, S. Balasubramaniyam, A. Nayyar, C. A. Kerrache, and M. Bilal, "MADCR: Mobility aware dynamic clustering-based routing protocol in Internet of Vehicles," *China Commun.*, vol. 18, no. 7, pp. 69–85, Jul. 2021.

[24] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.

[25] D. Osmanković and S. Konjicija, "Implementation of Q—Learning algorithm for solving maze problem," in *Proc. 34th Int. Conv. (MIPRO)*, May 2021, pp. 1619–1622.

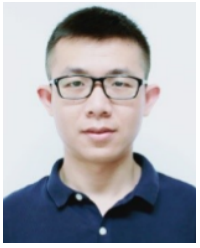
[26] L. Liu, J. Feng, X. Mu, Q. Pei, D. Lan, and M. Xiao, "Asynchronous deep reinforcement learning for collaborative task computing and on-demand resource allocation in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, early access, Mar. 6, 2023, doi: 10.1109/TITS.2023.3249745.

[27] J. Li, R. Wang, and K. Wang, "Service function chaining in industrial Internet of Things with edge intelligence: A natural actor-critic approach," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 491–502, Jan. 2023.

[28] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7450, pp. 529–533, 2015.

[29] Y. Ju et al., "Joint secure offloading and resource allocation for vehicular edge computing network: A multi-agent deep reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 5555–5569, May 2023.

[30] R. J. Cowan, "Useful headway models," *Transp. Res.*, vol. 9, no. 6, pp. 371–375, Dec. 1975.

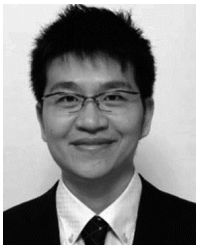


Momiao Zhou (Member, IEEE) received the B.E. and Ph.D. degrees in communication engineering from Xidian University, Xi'an, China, in 2012 and 2018, respectively. He is currently with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China. His research interests include the Internet of Vehicles, wireless heterogeneous networks, and autonomous driving.

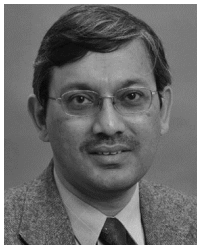
Lei Liu (Member, IEEE) received the B.Eng. degree in electronic information engineering from Zhengzhou University, Zhengzhou, China, in 2010, and the M.Sc. and Ph.D. degrees in communication and information systems from Xidian University, Xi'an, China, in 2013 and 2019, respectively. From 2013 to 2015, he was a subsidiary of China Electronics Corporation. From 2018 to 2019, he was supported by the China Scholarship Council to be a Visiting Ph.D. Student with the University of Oslo, Oslo, Norway. He is currently a Lecturer with the State Key Laboratory of Integrated Service Networks, Xidian University, and he is also with the Xidian Guangzhou Institute of Technology, China. His research interests include vehicular ad hoc networks, intelligent transportation, mobile edge computing, and the Internet of Things.

Yanshi Sun (Member, IEEE) received the B.Eng. and Ph.D. degrees from the University of Science and Technology of China in 2016 and 2021, respectively. Since December 2021, he has been with the Hefei University of Technology as a Professor. His research interests include stochastic geometry, non-orthogonal multiple access systems, unmanned aerial vehicles, vehicular communications, and satellite communications.

Kan Wang (Member, IEEE) received the Ph.D. degree in military communications from the State Key Laboratory of ISN, Xidian University, Xi'an, China, in 2016. Since March 2017, he has been with the School of Computer Science and Engineering, Xi'an University of Technology, Xi'an. His research interests include wireless resource allocation, network slicing, edge computing, and satellite networks.



Mianxiong Dong (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science and engineering from The University of Aizu, Japan. He was a JSPS Research Fellow with the School of Computer Science and Engineering, The University of Aizu, and a Visiting Scholar with the BBCR Group, University of Waterloo, Canada, supported by JSPS Excellent Young Researcher Overseas Visit Program, from April 2010 to August 2011. He is currently the Vice President and a Professor with the Muroran Institute of Technology, Japan. He was selected as a Foreigner Research Fellow (a total of three recipients all over Japan) by NEC C&C Foundation in 2011. He was a recipient of the 12th IEEE ComSoc Asia-Pacific Young Researcher Award in 2017, the Funai Research Award in 2018, the NISTEP Researcher in 2018 (one of only 11 people in Japan) in recognition of significant contributions in science and technology, the Young Scientist's Award from MEXT in 2021, the SUEMATSU-Yasuharu Award from IEICE in 2021, and the IEEE TCSC Middle Career Award in 2021. He is a Clarivate Analytics 2019 and 2021 Highly Cited Researcher (Web of Science) and a Foreign Fellow of EAJ.



Mohammed Atiquzzaman (Senior Member, IEEE) is currently a Professor in computer science with The University of Oklahoma. He teaches courses in data networks and computer architecture. His research interests include next generation computer networks, wireless and mobile networks, satellite networks, switching and routing, optical communications, and vehicular networks. His research works have been supported by the National Science Foundation, National Aeronautics and Space Administration, U.S. Air Force, Honeywell, and Cisco. He is currently the Editor-in-Chief of the *Journal of Network and Computer Applications*, the Founding Editor-in-Chief of the *Vehicular Communications* journal, and an Associate Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE TRANSACTIONS ON MOBILE COMPUTING, the *IEEE Communications Magazine*, the *International Journal of Sensor Networks*, the *International Journal of Communication Networks and Distributed Systems*, and the *Journal of Real-Time Image Processing*.

Schahram Dustdar (Fellow, IEEE) received the Ph.D. degree in business informatics from the University of Linz, Linz, Austria, in 1992. He is currently a Full Professor in computer science (informatics) with a focus on internet technologies heading the Distributed Systems Group, TU Wien, Wein, Austria. He has been the Chairperson of the Informatics Section of the Academia Europaea since December 2016. He has been a member of the IEEE Conference Activities Committee (CAC) since 2016, the Section Committee of Informatics of the Academia Europaea since 2015, and the Academia Europaea: The Academy of Europe, Informatics Section since 2013. He was a recipient of the ACM Distinguished Scientist Award in 2009 and the IBM Faculty Award in 2012. He is an Associate Editor of the IEEE TRANSACTIONS ON SERVICES COMPUTING, *ACM Transactions on the Web*, and *ACM Transactions on Internet Technology*. He is on the editorial board of IEEE.