ELITE: An Intelligent Digital Twin-Based Hierarchical Routing Scheme for Softwarized Vehicular Networks

Liang Zhao, *Member, IEEE*, Zhenguo Bi, Ammar Hawbani, Keping Yu, *Member, IEEE*, Yan Zhang, *Fellow, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

Abstract—Software-Defined Vehicular Network (SDVN) is a networking architecture that can provide centralized control for vehicular networks. However, the design for routing policies in SDVNs is generally influenced by several limitations, such as frequent topological changes, complex service requests, and long model training time. Intelligent Digital Twin-based Software-Defined Vehicular Networks (IDT-SDVN) can overcome these weaknesses and maximize the advantages of the conventional SDVN architecture by enabling the controller to construct virtual network spaces and provide virtual instances of corresponding physical objects within the Digital Twin (DT). In this paper, we propose a junction-based hierarchical routing scheme in IDT-SDVN, namely, intelligent digital twin hierarchical (ELITE) routing. The proposed scheme is conducted in four phases: policy training and generation in the virtual network, and deployment and relay selection in physical networks. First, the policy learning phase employs several parallel agents in DT networks and derives multiple single-target policies. Second, the generation phase combines the learned policies and generates new policies based on complex communication requirements. Third, the deployment phase selects the most suitable generated policy according to the real-time network status and message types. A road path is calculated by the controller based on the selected policy and then sent to the requester vehicle. Finally, the relay selection phase is utilized to determine relay vehicles in a hop-by-hop process along the selected path. Simulation results demonstrate that ELITE achieves substantial improvements in terms of packet delivery ratio, end-to-end delay, and communication overhead compared with its counterparts.

Index Terms—Fuzzy logic, hierarchical routing, intelligent digital twin networks, reinforcement learning, software-defined vehicular networks

1 Introduction

Wireless connectivity empowers vehicles participating in data exchange and promotes the establishment of Vehicular Networks (VNs) in vehicular environments [1]. In VNs, vehicles are enabled to perceive the environment outside the visible range via vehicle-to-vehicle communication, which can improve driving safely, enhance traffic efficiency, provide information services, and support autonomous driving [2], [3]. In general, all these applications depend on stable and efficient information sharing among vehicles. Nevertheless, limited by the transmission capacity of vehicles, successful data transmission requires relay vehicles to forward packets to the destination vehicle in a hop-by-hop process. Since the link quality between adjacent vehicles substantially influences communication performance, designing the right next-hop vehicle selection scheme is vital to improve the overall network quality of multi-hop networking, in terms of delivery ratio, latency, and load balancing [4], [5], [6].

Among all characteristics of modern transportation, dynamic vehicle movement and uneven distribution are the main reasons for intermittent connectivity, thus becoming the key challenges for relay selection in multi-hop networking. Software-Defined Vehicular Network (SDVN) is a networking paradigm to overcome these weaknesses by enabling the global information available [7], [8]. The central controller of SDVN has the ability to calculate the optimal routes from sources to destinations online based on spatiotemporal information, thereby microscopically ensuring reliable links and macroscopically avoiding local optima and data congestion [9], [10]. However, in large-scale VNs, calculating routing paths entirely in the controller may occupy massive up-down link bandwidth and computational resources. Moreover, the

- *Liang Zhao and Zhenguo Bi are with the School of Computer Science, Shenyang Aerospace University, Shenyang 110136, China. E-mail: lzhao@sau.edu.cn, tfxp294@163.com.*
- *Ammar Hawbani is with the University of Science and Technology of China, Hefei, Anhui 230052, China. E-mail: anmande@ustc.edu.cn.*
- *Keping Yu is with the Graduate School of Science and Engineering, Hosei University, Chiyoda, Tokyo 184-8584, Japan, and also with the RIKEN Center for Advanced Intelligence Project, RIKEN, Chuo, Tokyo 103-0027, Japan. E-mail: keping.yu@ieee.org.*
- *Yan Zhang is with the University of Oslo, 0315 Oslo, Norway. E-mail: yanzhang@ifi.uio.no.*
- *Mohsen Guizani is with Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi 51133, UAE. E-mail: mguizani@ieee.org.*

Manuscript received 12 Nov. 2021; revised 26 Apr. 2022; accepted 24 May 2022. Date of publication 31 May 2022; date of current version 4 Aug. 2023.
This work was supported in part by the funds of Department of Science and Technology of Liaoning Province, in part by the Science Foundation of Liaoning Province under Grant 2020-MS-237, in part by the Liaoning Provincial Department of Education Science Foundation under Grant JYT19052, in part by the Digit Fujian Internet-of-Things Laboratory of Environmental Monitoring Research Fund (Fujian Normal University) under Grant 202001, in part by the National Natural Science Foundation of China under Grant 61872086, and in part by the Science Foundation of Liaoning Province (Key Projects).
(Corresponding authors: Zhenguo Bi and Ammar Hawbani.)
Digital Object Identifier no. 10.1109/TMC.2022.3179254

1536-1233 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.

preselected routing paths may contain broken links due to rapid vehicular mobility.

As vehicle distribution is subject to road topology, the presence of buildings and obstacles may decrease the signal intensity, especially in urban settings with numerous roadside buildings. By taking road topology into consideration, road-based routing can evaluate road status and reduce the risk of packets entering a segment with high-scale link disconnections [11], [12], [13], [14]. Based on the global knowledge of the road network, the central controller can find the routing path from the perspective of the road or junction within a limited time period. After that, relay vehicles can select the next hop along the planned path accordingly. In road-based hierarchical routing, the routing path is not defined as a complete path that consists of relay vehicles from the source to the destination, but rather, as a junction sequence from the start junction to the target junction.

In recent years, research efforts have focused on routing scheme designs for both distributed and centralized routing. Nevertheless, optimizing models online and validating them before application to adapt to changing traffic scenarios and communication requirements is still challenging [15]. A promising way to overcome this challenge is by opening up a specified space for preliminary model training and processing. As a potential cure, Digital Twin (DT) [16] can be utilized to create multi-scale digital mappings to target entities. Recent trends in the combination of autonomous driving and DT have led to a proliferation of studies that driving assistance [17], [18], [19], vehicular edge computing [20], [21], and traffic management [22], [23]. A typical application of DT in vehicular networks is proposed in [24], which introduces a new networking architecture, named Intelligent Digital Twin-based Software-Defined Vehicular Networks (IDT-SDVN). It constructs multiple virtual Digital Twin Networks (DTNs) to provide distinct functions for the real-world SDVN, such as prediction, model training, and verification. Based on the DTN space, it is promising to yield reliable schemes to meet the above considerations and improve the performance of the whole vehicular network. *The main motivations* of this paper are summarized as follows.

(1) Due to a lack of knowledge about future traffic situations when training a model, conventional pre-trained model-based routing policies suffer from low adaptability and insensitivity to the dynamics of network status. IDT-SDVN can handle the dynamic topology changing of vehicles better over SDVN and VN due to its virtualization and prediction abilities.

(2) Existing routing models generally pay attention to traffic-related or topological factors (inter-vehicle distance [25], relative velocity [26], density [27], etc.). However, the influence of these factors on the performance in certain routing metrics such as packet loss rate and end-to-end delay is still ambiguous. Thus, directly quantifying and utilizing routing metrics while training models is a more feasible solution.

(3) There are various types of messages with different transmission requirements in VNs. For example, collision avoidance messages are time-constraint, while multimedia messages are delay-tolerant, which might occupy massive bandwidth. However, few studies have been investigated in terms of adaptively deploying routing policies towards complex message types to provide personalized services.

To mitigate the aforementioned problems, we propose a junction-based hierarchical routing scheme in IDT-SDVN, known as intelligent digital twin hierarchical (ELITE) routing, which involves two aspects in system modeling. First, we train a group of multidimensional routing tables (policies) by parallel learning in DTNs and experience fusion. The table value indicates the tendency of a packet to be forwarded to each junction for a certain transmission requirement. Second, as these policies are generated aiming to meet diverse communication requirements, the controller maintains a state-action table that records the best policy choice for each combination of (network situation, message type). Once receiving a routing request, the controller determines which policy is the best choice for this transmission and deploys it. Data packets are forwarded along the path calculated based on the deployed policy. All the efforts made in this paper are focused on providing the most reliable multi-hop data transmission by jointly considering various routing metrics and communication types. *The main contributions* of this paper are summarized as follows.

(1) In IDT-SDVN, a new routing scheme, ELITE, is introduced with the advantages of (i) realizing adaptive routing policy generation for various communication requirements in VNs, and (ii) intelligently switching policies regarding the instant network status and message types.

(2) We propose a routing policy generation approach with parallel learning and experience fusion. In the learning process, we develop a DTN space and deploy a group of virtual agents in it to learn multiple single-target policies. In addition, the fusion process is responsible for combining these learning results and generating available policies for various service requirements. This approach allows automatic routing policy generation.

(3) We introduce a policy application mechanism that includes policy selection and road path planning in the IDT-SDVN controller, and road-based relay vehicle selection in distributed vehicular networks. By adopting this mechanism, the controller can reduce the impact of frequent topological changes in vehicular networks and decrease the time consumption in policy switching. Simulation results show that the proposed mechanism brings great performance in packet delivery ratio, average end-to-end delay, etc.

The rest of this paper is organized as follows. In Section 2, we discuss the state-of-the-arts and related works. Section 3 introduces a new IDT-SDVN-based routing policy generation scheme and outlines the system model. In Section 4, we describe the proposed scheme in detail. Section 5 gives the simulation scenario and evaluation results. Finally, in Section 6, we summarize the proposed scheme and end the paper by discussing future works.

2 Related Work

In this section, we discuss and summarize the existing routing schemes in varying scenarios (i.e., distributed and centralized) as well as using different technologies (i.e., fuzzy logic and reinforcement learning). Besides, we also introduce some common experience fusion methods.

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.

(1) *Existing Routing Schemes in Vehicular Ad hoc Network (VANET).* VANET plays an important role in Intelligent Transportation Systems (ITS). Routing decisions in VANETs are made by distributed vehicles according to the temporary information within one hop. On-road vehicles dynamically share/exchange basic information (e.g., position and velocity) with neighbors through periodic beacons. During packet propagation, intermediate vehicles select the next forwarder based on applied schemes without any additional information. Thus, this type of routing solution always accompanies potential risks of limited available and outdated data separately caused by perception ability and high mobility, thereby leading to link disconnection, data congestion, and local optimization.

A few contributions have been made in link evaluation [28], vehicular dynamic prediction [29], and bus trajectory utilization [30] to meet various demands of Quality of Service (QoS) in VANETs. The scheme in [28] aims to identify routing paths based on five metrics such as link capacity and connectivity. It uses the Strength Pareto Evolutionary Algorithm to optimize these metrics and proposes a path-finding algorithm for data routing. A Broadcast scheme based on the Prediction of Dynamics (BPD) [29] is proposed which conducts two novel metrics separately targeted to broadcast delay and dissemination efficiency. In [30], the Bus Trajectory-based Street Centric (BTSC) routing scheme utilizes buses as the main relay to deliver packets. It considers both bus trajectory and packet delivery between two buses to increase the transmission ratio and decrease delay.

(2) *Existing Routing Schemes in Software-Defined Vehicular Networks (SDVNs).* Benefited from splitting networking control from data forwarding services, SDVN controllers are enabled for global information gathering, network resource management, and centralized routing decision. Based on these characteristics, in SDVNs, a lot of efforts have been made in strategy design for load balancing, low latency control, security, etc. However, routing of SDVNs always faces the problems, including computation consumption, frequent bandwidth occupation, and relatively long delay.

The majority of existing schemes in SDVN are based on centralized link evaluation. In such schemes [25], [31], [32], global network information are utilized in path scheduling with the great advantages against classical distributed routing. The scheme proposed consists of both centralized and distributed routing methods in SDVN [33]. The authors formulates multi-hop data delivery in uncertain vehicular network conditions as a minimum cost capacitated flow problem with the constraints of routing, timing, and capacity. By finding the optimal solution, a feasible path for forwarding packets is outputted. In [10], four different strategies are proposed for different transmission demands on various metrics. The SDN controller is responsible for global routing scheduling by evaluating the encounter probability between buses.

(3) *Fuzzy Logic-based Routing in VNs.* Fuzzy Logic empowers agents to process uncertain classification and decision-making problems by imitating the human brain. In VNs, through the concept of fuzzy logic, the vague relationship between traffic features and routing performance on various metrics can be well explained. Thus, a few routing schemes have been proposed by adopting fuzzy modeling and processing, to realize several functions such as hop-by-hop link evaluation and relay selection in VANETs or SDVNs.

Vehicular Environment Fuzzy Router (VEFR) [27] uses a fuzzy logic system to evaluate links from the perspectives of roads and vehicles. Road segments with high density and vehicles with similar speeds are selected for packet delivery. Fuzzy Logic-based Directional Greedy Forwarding (FL-DGR) [34] finds a trade-off among various metrics (e.g., position, link quality, and throughput) while next-hop node selection. In [35], Intersection Routing based on Fuzzy Multi-Factor Decision (IRFMFD) is proposed by integrating multiple features for geographic routing. It selects static nodes at intersections and builds multi-hop links between two static nodes by fuzzy logic based on factors like distance and relative velocity. However, the frequent usage of fuzzy inference must lead to excessive computational consumption, thereby decreasing the network efficiency.

(4) *Reinforcement Learning-based Routing in VNs.* In Reinforcement Learning (RL)-based routing schemes, the entire VN is treated as the environment, agents are the on-road vehicles, SDVN controller, or virtual agents flowing among vehicles. Reward/experience accumulation originates from real data routing or network awareness. One of the greatest advantages of RL-based routing is that agents have no need to achieve prior knowledge of the environment. Existing studies demonstrate that RL-based routing well adapts to high dynamic VNs because of its ability to react to transient changes in network situations.

QGrid [36], designed as a RL-based hierarchical routing scheme, is proposed by considering grid connectivity and vehicular movement. It divides the whole map into many small grids and uses Q-learning to find the transmission propriety between every two adjacent grids through periodical exploration of the environment. During routing, packets are delivered hop-by-hop along the grid sequence calculated by the learned Q-table. In [14], rewards and discount factors are determined according to routing situation and historical traffic flow. The authors propose a RL-based multi-hop route evaluation method [37]. As the vehicles exchanging hello messages, a table is learned as recording values for selecting next-hop vehicles for Road-Side Units (RSUs). However, the convergence of RL algorithms relies on extensive exploration of the environment, online training must lead to substantial bandwidth occupation and even failure routes.

(5) *Experience Fusion Methods.* Data fusion has been widely used in the fields of target recognition and detection. Based on certain rules, fusion mechanisms can analyze and integrate data from multiple different sensors or agents, and further provide more reasonable evaluation and decision. Fused data can reflect the information from each data source, as well as the information that cannot be provided by a single source. Common data fusion methods include voting, Kalman filter, Bayesian method, and fuzzy logic.

Many studies tend to implement multi-sensor data fusion to improve data reliability and integrity in wireless sensor networks. In [38], the optimal fusion rule selection problem of distributed decision fusion in wireless sensor networks is discussed. It uses the likelihood ratio test under the Neyman-Pearson sense to derive the optimal fusion rule and applies the fusion process to produce the global decision.

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.

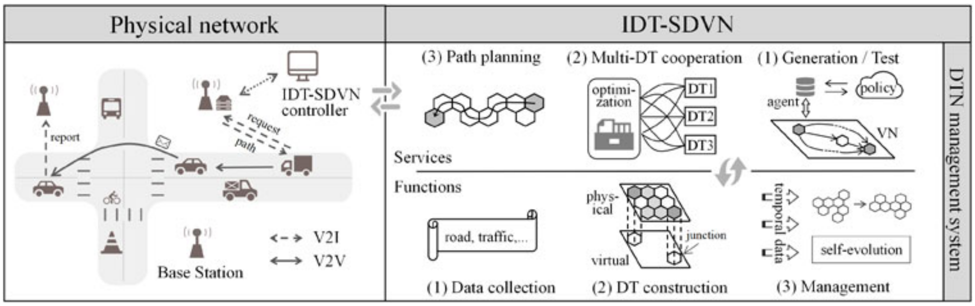


Fig. 1. The structure of hierarchical routing in IDT-SDVN.

Efforts in [39] aim to develop a real-time road condition system that strictly relies on comprehensive and accurate data. It designs a three-layer Back Propagation structure for the fusion of floating car data and fixed detector data to improve the accuracy of traffic speed estimation. A fine-grained traffic flow prediction method [40] is proposed to forecast the flow of different types of vehicles. By fusing multi-source traffic data, such as vehicle detector data, remote microwave sensors data, and toll collection data, vehicle types are well identified.

Compared with other fusion methods, through fuzzy fusion, data with uncertain concepts or vague descriptions from different sources can be properly combined into single-valued evaluation indicators. In RL-based vehicular networks, considering the differences in learning rates among agents, experience accumulation is incomplete and time-varying. Therefore, it is promising to apply the fuzzy system to multi-agent experience fusion as the relationship among learned policies can be well expressed and distinguished.

(6) *Summary.* Through the previous studies, we conclude that (i) the global view and computational power of SDVN can provide additional functions that distributed systems are not capable of, and also facilitate the realization of DT in VNs. (ii) DTNs provide platforms for verifying functions that are risky or hard to execute in physical worlds. Some key challenges of existing schemes, that is, membership function settings in FLR and sparse reward in RLR, have the potential to be optimized through IDT-SDVN with low cost. Benefited from the characteristics such as parallel with physical networks and strong spatiotemporal scalability, IDT-SDVN can provide multiple virtual twin networks that cover past, present, and future, thereby realizing intelligent routing optimization in VNs.

3 System Overview

In this paper, considering the real-time demand of making routing decisions and the difference in the transmission requirement of various types of messages in vehicular networks, by utilizing the advantages of the DT concept, we propose ELITE with the target to realize stable and efficient inter-vehicle communication in the IDT-SDVN. Before detailed introducing ELITE, in this section, we describe the basic preliminaries in three parts. First, we discuss the road-based hierarchical routing structure. Our proposed ELITE is implemented under this structure and mainly targets the first level, that is, road path planning for packet transmission. Then, a comprehensive outline of ELITE is given. Finally, we show the routing table mentioned in ELITE.

3.1 Routing Structure

In this paper, we adopt a road-based hierarchical routing structure for IDT-SDVN. Compared with other centralized and location-based structures, it has advantages including, (1) decreasing computation time and resource consumption, and (2) reducing the impact of roadside buildings on signal strength. Moreover, by utilizing DT, SDVN is empowered with various functions, including online routing policy learning, testing, verification, and optimization.

As presented in the left of Fig. 1, the hierarchical routing structure of IDT-SDVN is expressed in five steps. First, the source node sends request messages to the IDT-SDVN controller before packet transmission. Second, a routing path is calculated by the IDT-SDVN controller and sent to the requester. Note that the path here denotes a junction sequence. Then, packets are forwarded along the path hop-by-hop based on the relay node selection strategy which will be described later. Finally, the packet receiver reports the essential routing information, such as round delay and hop count, to the controller for further policy optimization.

The right of Fig. 1 illustrates how IDT-SDVN operates to optimize routing in the physical world. As shown, the SDVN controller develops a virtual presentation of the physical space based on the digital map and collected traffic data. The DT side runs in parallel with the physical vehicular network. IDT-SDVN provides multiple services for VNs as described at the top of Fig. 1, including (1) policy verification or new policy learning; (2) multi-DT collaborative optimization; (3) generating a reliable routing decision model or path scheduling strategy.

3.2 Routing Scheme Outline

Under the hierarchical routing structure, determining road paths with efficient and stable packet transmission ability is a crucial point. In this paper, we focus on the policy designing of road paths planning (also known as intermediate junctions selection) and propose a road-level routing policy generation and application scheme, ELITE. As shown in Fig. 2, our proposed ELITE consists of three processes, Single-target Policy Learning (SPL), Experience Fusion and new policy Generation (EFG), and Application in Physical Networks (APN).

In SPL, we set up multiple DTNs by duplicating the physical vehicular network, and correspondingly deploy virtual agents to them. Each agent acts as the data center of a DTN and imitates real inter-vehicle communication together with it. All DTN-agent pairs operate in parallel to learn policies that can maximize the performance for one

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.

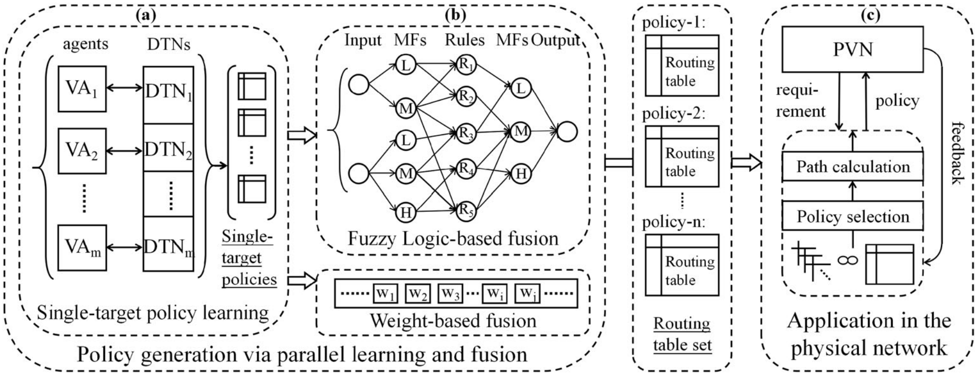


Fig. 2. The proposed ELITE routing scheme in IDT-SDVN. (a) The single-target policy learning process in DTNs. (b) The experience fusion mechanism which includes two ways: wight-based fusion and fuzzy logic-based fusion. Multiple available policies, presented as a routing table set, is generated after steps (a) and (b). (c) The policy application mechanism that aims to deploy the optimal one among all generated policies to the physical side of IDT-SDVN.

metric (e.g., delivery ratio, end-to-end delay, and hop counts).

EFG provides a fusion mechanism which includes two parts: weight-based fusion and fuzzy logic-based fusion. The inputs are single-target policies obtained in SPL, while the outputs are new policies that satisfy various service requirements (e.g., high reachability-first and low delay-first). In this case, EFG can infer more policies by providing more fusion rules, ensuring comprehensive and flexible routing support.

In APN, we design an adaptive routing policy deployment mechanism by employing the Q-learning algorithm. The controller learns from real vehicular communication and maintains a state-action table that records the priority of the generated policies under various network load situations and message types. Besides, a road-based relay vehicle selection strategy is proposed. Adopting this scheme is beneficial because it can intelligently select and apply the optimal policy in response to the instantaneous information from physical networks.

3.3 Routing Table

As illustrated above, in this paper, the proposed routing scheme is policy-based, and policies are presented in the form of a routing table set that contains n routing tables $\mathbb{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n\}$. A routing table, that is, a policy in this paper, is organized as a three-dimensional matrix $\mathcal{T}_x = \{T_1, T_2, \ldots, T_m\}$, where m is the number of junctions in the physical network. T_i is formulated in Eq. (1), where each entry $T_i : t_{j,k}$ indicates the priority of the current junction J_i forwarding packets to its adjacent junction J_k, with the target junction being J_j. q is the number of adjacent junctions of J_i

$$T_i = \begin{bmatrix} t_{1,1} & \cdots & t_{1,q} \\ \vdots & \ddots & \vdots \\ t_{m,1} & \cdots & t_{m,q} \end{bmatrix} \tag{1}$$

There are three types of table sets, all of which are organized as $\mathbb{T}$. More details are as follows.

(1) $\mathbb{Q} = \{Q_{PRR}, Q_{AD}, Q_{HC}, Q_{RC}\}$ denotes the single-target policy set learned by n parallel agents in DTNs. An entry $Q_x(i,j,k)$ corresponds to $\mathcal{T}_x : T_i : t_{j,k}$.

(2) $\mathbb{Q}^{norm} = \{Q^{norm}_{PRR}, Q^{norm}_{AD}, Q^{norm}_{HC}, Q^{norm}_{RC}\}$ denotes the normalized policy set which originates from $\mathbb{Q}$. $Q^{norm}_x(i,j,k)$ corresponds to $\mathcal{T}_x : T_i : t_{j,k}$.

(3) $\mathbb{G} = \{G_{bas}, G_{HRF}, G_{LDF}, G_{LBF}\}$ denotes the policy set that will be deployed to physical networks. $G_x(i,j,k)$ corresponds to $\mathcal{T}_x : T_i : t_{j,k}$.

A detailed description of all notations used in this paper is presented in Table 1.

4 Proposed ELITE Routing Scheme in IDT-SDVN

ELITE is proposed by the following three processes. First, we employ the Parallel Reinforcement Learning method in DTNs to learn multiple single-target routing policies. Then, two fusion techniques are introduced to combine those learning results and further derive new policies toward diverse message types. Finally, we apply the DTN-generated policies in the physical side of IDT-SDVN.

4.1 Single-Target Policy Learning

This section mainly discusses the PRL-based single-target policy learning approach. In the following subsections, we

TABLE 1
Notations

Notation	Description
$\mathbb{J}$	$\mathbb{J} = \{J_1, \ldots, J_m\}$. J_i is a junction in the map, with the identification i. N_i is the set of J's adjacent junctions, $q = len(N_i)$.
$\mathbb{T}$	$\mathbb{Q}$, $\mathbb{Q}^{norm}$, and $\mathbb{G}$ are all organized in the form of $\mathbb{T}$, $\mathbb{Q}$ is the single-target policy set learned by DTNs, $\mathbb{Q}^{norm}$ is the normalized single-target policy set, $\mathbb{G}$ contains policies that will be deployed.
$R(i,j,k)$	One-step reward from J_i to J_j via $J_k \in N_i$.
$\mathcal{L}_{i,j}$	A junction sequence between J_i and J_j. Generally obtained by querying policies in $\mathbb{Q}$ or $\mathbb{G}$.
$u_{i,j}$	A vector in Q_x that bound by J_i and J_j. $u_{i,j} = (Q_x(i,j,p)), \forall J_p \in N_i$.
$v^{i,j,k}$	A vector that contains all values in $\mathbb{Q}$ with the same constraint $J_i - J_j - J_k$. $v^{i,j,k} = (Q^{norm}_l(i,j,k)), \forall Q^{norm}_l \in \mathbb{Q}^{norm}$
V_i	A vehicle with the identification number i.

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.

first introduce the parallel learning objectives and then illustrate the SPL approach and its implementation in agent-DTN pairs in detail.

4.1.1 Routing Metric

In the DTN space, each virtual agent is assigned a specified learning objective, namely obtaining the optimal performance of one essential routing metric for vehicular communication by interacting with its corresponding DTN. The following metrics are taken into consideration:

(1) *Packet Delivery Ratio (PDR)*. This metric is defined as the ratio of the number of successfully received packets to the total number of generated packets in a certain period.

(2) *Average Delay (AD)*. It represents the average interval between data generation and reception, including the time caused by inter-vehicle transmission and packet-carrying at relay vehicles.

(3) *Hop Count (HC)*. It is considered a basic measurement of the distance in vehicular networks, obtained by accumulating the number of hops during successful packet delivery.

(4) *Routing Cost (RC)*. This metric is expressed as the number of control messages (e.g., beacon messages, and discovery and report messages) required between the generation and reception of a packet.

4.1.2 Parallel Learning in Digital Twin Networks

Here, we introduce three aspects of the SPL approach. *Parallel learning structure* demonstrates how to develop a learning space in the IDT-SDVN controller. *Virtual vehicular communication* describes the way of realizing routing decisions in DTNs. *Experience accumulation* provides the policy maintenance and update approach.

(1) *Parallel learning structure*. The first thing to be noted in parallel learning is DT, which is responsible for providing an internal virtual representation of the physical vehicular network in the IDT-SDVN controller. In practice, we assume that the central controller can set up a group of DTNs that (i) entirely emulate real road network topology, including junctions, roadside buildings, weather and so forth; and (ii) contain traffic flow information within a certain future period obtained by accurate prediction algorithms.

Based on the developed DTN space, we deploy a group of virtual agents on it to construct the parallel learning structure (as shown in Fig. 3), a DTN provides a virtual vehicular communication environment and an agent works as the data center on it. There are four pairs of DTN-agents developed in this paper, labeled *PDR*, *AD*, *HC*, and *RC*. All DTN-agent pairs work in parallel. Each DTN-agent pair targets a certain performance objective, that is, pursuing the best performance of a single metric by imitating virtual data transmission and centralized routing decisions. Agents keep accumulating routing experience and record them in the form of a policy. The policy learning process can be treated as a parallel implementation of the Q-learning algorithm in digital networks.

We take junctions and connected roads as states and actions of Q-learning, respectively. Because two adjacent junctions are connected by at least one road, in this paper, actions can also be explained as the adjacent junctions of the current junction. In this case, the Q-table maintained by virtual agents is defined as a three-dimensional matrix whose set is denoted by $\mathbb{Q} = \{Q_{PDR}, Q_{AD}, Q_{HC}, Q_{RC}\}$.

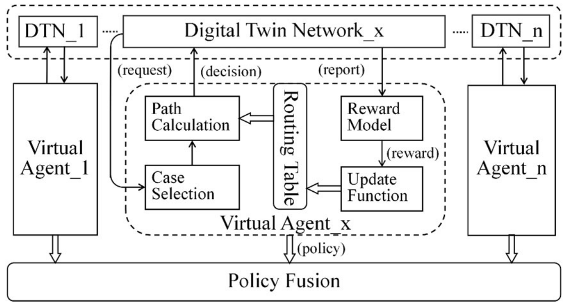


Fig. 3. The structure of single-target policy learning in parallel IDT-agent pairs.

(2) *Virtual vehicular communication*. As shown in Fig. 3, once virtual agent VA_x receives a request message from its corresponding DT network DTN_x, it first executes the junction selection algorithm, which consists of two steps: (i) a Case Selection procedure that determines the junction selection mode, and (ii) a Path Calculation procedure that selects intermediate junctions according to the assigned mode. The junction selection algorithm in parallel learners takes as inputs the source junction J_s and the destination junction J_d, as illustrated in Algorithm 1. It starts at J_s and aims to construct a road path from J_s to J_d. There are three cases mentioned while selecting the next junction:

Algorithm 1. Junction Selection in Parallel Learners

Input: Source junction J_s and destination junction J_d
Output: Junction list $\mathcal{L}_{sd}$
1: Set $c \leftarrow -1$ # the current junction
2: Set $\mathcal{L}_{sd} \leftarrow [\,]$ # a list of junctions
3: Randomly select three cases by the probability of 50%, 20%, and 30%
4: **while** $J_d \notin \mathcal{L}_{cn}$ **do**
5:   **if** Case1 **then**
6:     # exploitation mode
7:     $r \leftarrow \max Q(c, d, n), \forall J_n \in N_c$
8:   **else if** Case2 **then**
9:     # greedy mode
10:     Get the distance between c and $\forall J_n \in N_c$
11:     $r \leftarrow$ the adjacent junction closest to J_d
12:   **else**
13:     # exploration mode
14:     $r \leftarrow$ a random neighbor of c
15:   **end if**
16:   $c \leftarrow r$
17:   Add c into $\mathcal{L}_{sd}$
18: **end while**
19: **return** $\mathcal{L}_{sd}$

Case 1, Exploitation mode. In this mode, the agent queries its maintained routing table and chooses the next junction J_r based on the optimal policy, which is described in Lines 5-7 in Algorithm 1 correspondingly. J_c is the current junction and N_c indicates the set of its adjacent junctions.

Case 2, Greedy mode. In this mode, the agent calculates the junction sequence entirely based on location information. Each junction that appears in the path is the geographically

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.

optimal choice (i.e., closest to the target junction) of its previous junction, as shown in Lines 8-11 in Algorithm 1.

Case 3, Exploration mode. In this mode, the agent randomly selects intermediate junctions from the source junction to the target junction,as expressed in Lines 12-14 in Algorithm 1. Benefited from exploration, the experience of state-action pairs that are non-optimal will also be accumulated, and finally utilized in the fusion process.

Once J_d is achieved, the path planning process ends and returns $\mathcal{L}_{sd}$, which records the junction sequence from J_s to J_d. After that, the virtual agent sends a response back to the corresponding DTN. Subsequently, packets are forwarded hop by hop based on the road-based greedy routing strategy described in Section 4.3.2 (2) toward the destination vehicle along the received path.

(3) *Experience accumulation*. The foremost part of this parallel learning process is routing table maintenance. As a road sequence only brings two results: data reception or loss, the agents are responsible for accumulating successful experience and, if necessary, weakening the path that leads to packet loss. Once data transmission is complete in a DTN, the virtual agent updates its maintained routing table by

$$Q(c,d,a) \mathrel{+}= \alpha[R + \gamma(\max_h Q(a,d,h) - Q(c,d,a))], \tag{2}$$

where c, d, and a represent the current junction, the target junction, and the next junction, respectively. α is the learning rate, set to 0.9, and γ is the discount factor, set to 0.1. $\max_h Q(a,d,h)$ denotes the maximal expected reward. For a certain junction sequence $\mathcal{L}_{sd}$, the update process is executed in reverse, as presented in Algorithm 2. For each pair of adjacent junctions in $\mathcal{L}_{sd}$, this algorithm first calls the Reward Model to calculate the immediate reward R and then updates the corresponding table entry according to Eq. (2). Finally, it ends at $c = J_s$ and returns the updated table.

The routing table learned in a virtual agent is named a policy, as it can be used to guide the agent to plan road paths for data transmission. All virtual agents learn policies in the same way, but differ when it comes to the reward because they pursue different routing performance goals. As follows, we separately explain the settings of reward in this paper, including R_{PDR}, R_{AD}, R_{HC}, and R_{RC}.

R_{PDR} is expressed as the packet delivery ratio-related reward. It indicates the immediate feedback after executing action a in state (c,d). In this paper, because only paths that realize successful delivery are reported to the agent for policy update, we determine the reward as a constant $R_{PDR}(c,d,a) = 1$ to highlight these paths.

R_{AD} is expressed as the average delay-related reward, calculated by

$$R_{AD}(c,d,a) = \left(1 - \frac{AD_{ca}}{AD_{sd}}\right) \cdot \left(1 - \frac{AD_{sd}}{mAD_{sd}}\right), \tag{3}$$

where AD_{sd} is the delay of packet transmission on path $\mathcal{L}_{sd}$, and mAD_{sd} is the recorded longest delay. AD_{ca} represents the time spent on road L_{ca}.

R_{HC} is expressed as the hop count-related reward, calculated by

$$R_{HC}(c,d,a) = \exp\left(-\frac{HC_{ca} \cdot \mathcal{C}}{l_{ca}}\right), \tag{4}$$

where HC_{ca} is the hop count on road L_{ca} and $l_{c,a}$ denotes the road length. $\mathcal{C}$ is the vehicle communication radius.

R_{RC} is expressed as the routing cost-related reward, calculated by

$$R_{RC}(c,d,a) = \left(1 + \frac{RC_{ca}}{CC_{ca}}^{-1}\right), \tag{5}$$

where RC_{ca} is the number of control messages generated by relay vehicles on road L_{ca}, and CC_{ca} is the number of control messages generated by all vehicles during packet transmission.

In the end, the IDT-SDVN controller obtains the policy set $\mathbb{Q} = \{Q_{PDR}, Q_{AD}, Q_{HC}, Q_{RC}\}$, where Q_{PDR}, Q_{AD}, Q_{HC}, and Q_{RC} derive from the DTN-agent pairs labeled by PDR, AD, HC, and RC, respectively.

The main reason for implementing RL is to provide online and adaptive routing policy learning abilities for DTNs. To make $\mathbb{Q}$ adapt to dynamic network topology and situations, we repeatedly execute the routing path exploration process as discussed in Case 3 and conduct virtual inter-vehicle communication based on multiple settings, including varying distances between source vehicles and destination vehicles. Moreover, considering rewards are set according to routing results, values in $\mathbb{Q}$ are changing after each episode. However, an appropriate policy set should highlight the best choice when selecting intermediate roads in physical networks.

Algorithm 2. Routing Policy Update in Parallel Learners

Input: A junction list $\mathcal{L}_{succ}$
Output: An updated routing table

```
d ← L_succ[-1]
for i ∈ [len(L_succ) - 1, ..., 1] do
    c ← L_succ[i - 1]
    a ← L_succ[i]
    Calculate R by R = 1, Eq. (3), (4), or (5)
    Q(c,d,a) = Q(c,d,a) + α[R + γ(max_h Q(a,d,h)
                   -Q(s,d,a))]
end for
return Q
```

4.2 Experience Fusion

In DTNs, virtual agents learn multiple routing policies in parallel, and each policy aims to realize the perfect performance of one routing metric. However, it is quite likely that the routing path with the best performance in a certain metric may be the one with the worst performance in other metrics. This situation may reduce the service quality of the whole network. In fact, transmitting a certain type of message always has more than one metric requirement. For example, collision avoidance messages broadcasting needs both a high delivery rate and low delay. Thus, in this section, we introduce a policy fusion mechanism to generate new policies that adhere to various service requirements and describe two executed fusion techniques in detail. Separating policy generation into these two steps, i.e., learning

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.

and fusion, have the following advantages. (i) It allows freedom combinations of the learning results derived from DTNs, therefore, it is not necessary to develop more DTNs with the expansion of requirement types. (ii) A single objective makes the learning process in a DTN easy to execute, and the fusion process is periodically activated, therefore, the computation resource occupation is reduced.

For ease of calculation, input single-target policies are normalized bylines first. Then, based on the standard data, we carry out the fusion mechanism in two ways: weight-based fusion and fuzzy logic-based fusion. The reason for utilizing fuzzy logic is that it can solve problems with fuzzy concepts such as "perform *well* on PDR" or "perform *badly* on PDR". After fusion, the controller can obtain available routing policies for deployment. In this paper, we present four types of generated policies, which will be discussed later, according to potential communication requirements.

4.2.1 Preprocessing

Normally, values recorded in $\mathbb{Q}$ distribute in a large range due to the difference in learning rate between parallel agents. To unify the evaluation criteria and easily carry out fusion mechanisms, all entries in the learned policies should be normalized into the range (0,1]. Bound by the current junction J_i and the target junction J_j, the value of J_k (an adjacent junction of J_i) in the table Q_x is normalized by

$$Q_x^{norm}(i,j,k) = \min\left(1, \frac{\log\left(1 + Q_x(i,j,k)\right)}{\log\left(maq_x(i,j)\right)}\right), \tag{6}$$

where

$$\begin{aligned} maq_x(i,j) &= \max v_{i,j} \\ &= \max Q_x(i,j,p), \forall J_p \in N_i, \end{aligned} \tag{7}$$

is the maximum value among all adjacent junctions of J_i. N_i denotes the set of adjacent junctions of J_i. By implementing normalization upon $\mathbb{Q}$, we can obtain the normalized policy set $\mathbb{Q}^{norm} = \{Q_{PDR}^{norm}, Q_{AD}^{norm}, Q_{HC}^{norm}, Q_{RC}^{norm}\}$

4.2.2 Weight-Based Fusion

One type of policy fusion mechanism in this paper is implemented in the form of weight-based fusion, described as an average of parallel agents' independently learned policies. The output is a new policy G_{bas} in which an entry is calculated by

$$G_{bas}(i,j,k) = \frac{1}{n} \cdot \sum Q_x^{norm}(i,j,k), \forall Q_x^{norm} \in Q^{norm}, \tag{8}$$

where $Q_x^{norm}(i,j,k)$ denotes an entry in the normalized table Q_x^{norm}. J_i, J_j, and J_k represent the current junction, the destination junction, and an adjacent junction of J_i, respectively. By traversing all entries in $\mathbb{Q}^{norm}$, we finally obtain the first type of generated routing policy, the Basis Policy (BP).

Similarly, fuzzy logic-based fusion, which will be presented later, also aims to derive heterogeneous routing policies by integrating the accumulated experience of partial or all parallel agents. In practice, instead of employing a weighted average, the fuzzy logic-based fusion mechanism implements policy generation by designing various fuzzy/fusion rules.

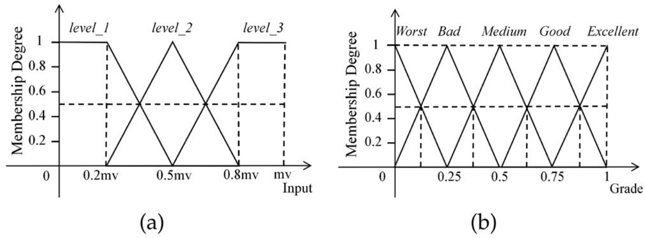


Fig. 4. (a) Input membership and (b) output membership functions.

4.2.3 Fuzzy Logic-Based Fusion

The fuzzy logic-based fusion mechanism is responsible for generating various routing policies according to potential communication requirements. It is an implementation of fuzzy logic in data fusion, including preprocessing, fuzzification, fuzzy inference, and defuzzification. In this way, based on the input normalized policy set, the fuzzy logic-based fusion mechanism generates new available routing policies that will be applied later in physical vehicular networks.

Algorithm 3 elaborates on the process of executing fuzzy fusion in four steps: (i) determining the processing vector $v^{i,j,k}$ by querying $\mathbb{Q}^{norm}$ (Line 2); (ii) calculating fuzzy values for each item in $v^{i,j,k}$ (Lines 4-5); (iii) mapping fuzzy values to fuzzy output based on the Min-Max fuzzy inference method and the predefined fuzzy rule table F_t (Line 7); and (iv) converting the fuzzy output to a crisp value (Line 9). After $\sum m \cdot N_i, \forall J_i \in \mathbb{J}$ cycles, a new policy G_t is obtained. In addition, more policies can be generated by expanding the rule base.

Algorithm 3. Fuzzy Logic-Based Policy Fusion

Input: A normalized policy set $\mathbb{Q}^{norm}$, and a rule table F_t
Output: A new policy G_t
1: **for** $J_i \in \mathbb{J}$, $J_j \in \mathbb{J}$, $J_k \in N_i$ **do**
2: $v^{i,j,k} \leftarrow (Q_l^{norm}(i,j,k)), \forall Q_l^{norm} \in \mathbb{Q}^{norm}$
3: # fuzzification
4: **Classify** all $e \in v^{i,j,k}$ based on fuzzy sets
5: **Get** the grade of membership for each $e \in v^{i,j,k}$ based on Fig. 4a
6: # fuzzy inference
7: **Reasoning** by F_t-based Min-Max Inference
8: # defuzzification
9: **Compute** $G_t(i,j,k)$ based on Fig. 4b by CoG
10: **end for**
11: **return** G_t

(1) *Fuzzification*. This process converts a numerical value to a fuzzy value based on predefined membership functions. Here, we define the four inputs as linguistic variables, $\mathcal{R}$ labeled by PRR, $\mathcal{D}$ labeled by AD, $\mathcal{H}$ labeled by HC, and $\mathcal{S}$ labeled by RC. Each variable has three sets of measurement items: $T(\mathcal{R}) = \{Poor, Medium, Good\}$, $T(\mathcal{D}) = \{Short, Middle, Long\}$, $T(\mathcal{H}) = \{Bad, Medium, Good\}$, and $T(\mathcal{S}) = \{Low, Medium, High\}$. Each item is modeled by a fuzzy set and map with crisp input values based on membership functions. The fuzzy membership functions demonstrate to

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.

which degree an input belongs to its corresponding fuzzy sets.

The method of designing fuzzy input membership functions is shown in Fig. 4a. The triangle membership function is used since it is easy to model and performs adequately for the proposed fusion mechanism. In practice, for a processing vector $u^{i,j,k}$, we first determine the maximum value $mv = max(v^{i,j,k})$. Then, the intersections of membership curves and the x-axis are separately set to $(0.8mv, 0)$, $(0.5mv, 0)$, and $(0.2mv, 0)$. Accordingly, the membership functions can be easily determined.

(2) *Rule-based fuzzy inference*. After calculating the fuzzy values of the four inputs, we utilize the predefined IF-THEN rule table to map these fuzzy values to fuzzy output which is denoted by the Grade. The Grade represents the evaluation of roads in terms of packet delivery ability. The linguistic variable of the Grade is defined as the combination of $\{Worst, Bad, Medium, Good, Excellent\}$, and the corresponding membership functions are shown in Fig. 4b.

Most existing rule-based fuzzy inference algorithms assume an accurate knowledge of fuzzy questions. It allows setting intuitively optimal rules based on expert experience. In this paper, considering the potential lack of understanding of the relation between routing metrics and request types, we introduce a numerical approach while designing fuzzy rules. A fuzzy rule table in this paper includes $3 \times 3 \times 3$ rules, and each rule consists of three antecedent items and a consequent item. Each antecedent item is assigned to two parameters, weight (W) and score (S). W indicates the influence of an antecedent item on the consequence, $\sum W = 1$. S is determined according to the belonged fuzzy set. In detail, W is set by experience, and the three fuzzy sets $level\_1$, $level\_2$, and $level\_3$ (shown in Fig. 4a) correspond to $S = 1/3$, $S = 2/3$, and $S = 1$, respectively. While designing fuzzy rules one by one, we first calculate g by

$$g = W_1 \cdot S_1 + W_2 \cdot S_2 + W_3 \cdot S_3, \tag{9}$$

where g ranges in [0,1], and it is a numerical representation of the consequent. According to the calculated g value, we can determine the consequent by

$$Grade = \begin{cases} Worst & 0 \leq g < 0.2 \\ Bad & 0.2 \leq g < 0.4 \\ Medium & 0.4 \leq g < 0.6 \\ Good & 0.6 \leq g < 0.8 \\ Excellent & 0.8 \leq g \leq 1 \end{cases} \tag{10}$$

In this way, a fuzzy rule table can be defined. The Mamdani's Min-Max fuzzy inference method is utilized to derive fuzzy outputs. In this method, the fuzzy operator first takes the minimum value among antecedent items in each activated rule as the weight of the consequent item. Then, as multiple activated rules may have the same consequence, we select the maximum value of the consequence as the final output membership when combining these rules.

(3) *Defuzzification*. The process of producing a numerical output based on inference results and predefined output membership functions is called "defuzzification". The Center of Gravity method is employed to defuzzify the fuzzy output, as it equips smooth output control and is widely used in practice. We set the crisp output obtained through defuzzification as an entry in the newly generated routing policy.

In this paper, we propose three alternate routing policies by the fuzzy logic method as above: High Reachability First (HRF), Low Delay First (LDF), and Load Balancing First (LBF). Details of the proposed policies are as follows.

HRF. Intuitively, the one with the highest predictive packet delivery rate among all candidate roads is preferred, as it has the greatest probability of forwarding packets to the destination. However, high reachability may always be accompanied by long end-to-end delay and excessive bandwidth consumption, which are caused by following the sub-optimal path selection in distance and frequent rescheduling. Thus, we define HRF as the combination of three metrics with weights of $\{PDR : 0.5, AD : 0.25, RC : 0.25\}$.

LDF. The main purpose of LDF is to provide stable and efficient service. Shorter delays will bring about better performance on time-constraint services. Moreover, considering the carry-and-forward strategy, LDF can save sufficient time for rescheduling once the last transmission fails. Nevertheless, blindly pursuing a short delay may lower the packet reachability. Considering these factors, LDF is defined as $\{AD : 0.5, HC : 0.25, PDR : 0.25\}$.

LBF. Considering the constraints of bandwidth and computing capacity, a large number of packets flowing into a junction or road must result in data congestion, and correspondingly degrade the service quality. In the high-level load network, LBF is treated as a viable solution to ensure global load balance as well as providing stable data transmission. By taking overhead into consideration, the LBF can evaluate the real-time road load. Moreover, a low hop count and short delay also have the advantage of reducing the number of detection messages during data transmission. Thus, LBF is defined as $\{RC : 0.5, HC : 0.25, AD : 0.25\}$.

As a result, together with the weight-based fusion method, we can obtain four policies for certain service requirements, i.e., BP, HRF, LDF, and LBF, denoted by $\mathbb{G} = \{G_{bas}, G_{HRF}, G_{LDF}, G_{LBF}\}$. All these policies are maintained by the controller and intelligently deployed to the physical vehicular networks, according to the method mentioned in the next section.

4.3 Application in Physical Side of IDT-SDVN

In this section, we present the way of applying generated policies in the physical side of IDT-SDVN, which aims to optimize policy deployment and provide valid route planning. The policy application mechanism in ELITE consists of three essential parts. (i) A routing policy set $\mathbb{G}$ which is used to calculate road paths. (ii) A state-action table that is established to select and deploy the optimal policy in $\mathbb{G}$ to the physical network. (iii) A road-based routing strategy that is responsible for determining relay nodes hop-by-hop along road paths. In what follows, we first discuss how to intelligently select and deploy the optimal one among all policies to the physical network. Then, we investigate the road-based hierarchical routing strategy. Finally, adaptive refreshing of the policy deployment mechanism is presented in detail.

4.3.1 State-Action Table-Based Policy Deployment

The policy deployment mechanism is implemented based on a state-action table. It helps the controller to provide the

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.

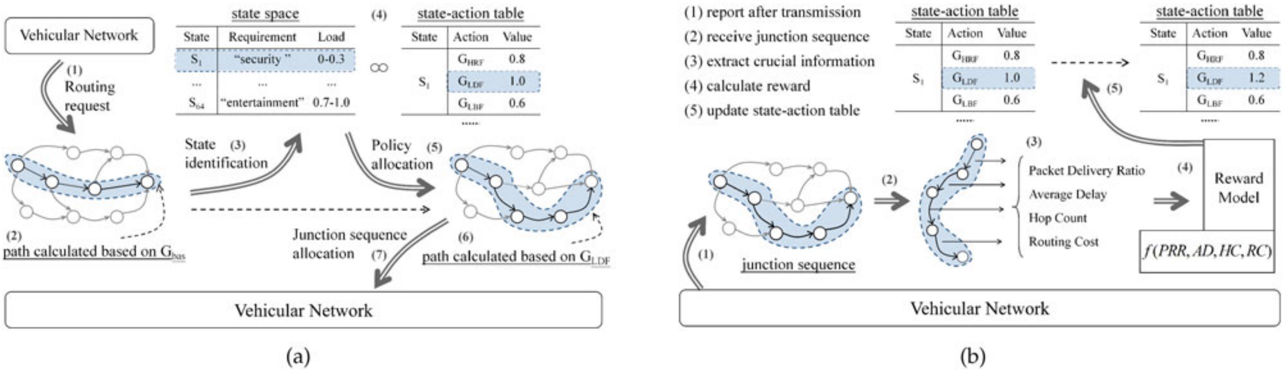


Fig. 5. (a) State-action table aided routing policy deployment and road path planning. (b) The structure of state-action table maintenance.

optimal policy selection according to the real-time network load. As presented at the top of Fig. 5a, a value in the state-action table is defined as $P(state, action)$ representing the expected rewards of taking $action$ from $state$. The state space is determined on the basis of the service requirement type and the network load, and the action space consists of three policies G_{HRF}, G_{LDF}, and G_{LBF}. In this paper, there are four types of requirements, including security (accident forewarning, reverse overtaking warning, *etc*.), efficiency (speed guide, congestion alert, *etc*.), information services (differential data sharing, floating car data collection, *etc*.), and entertainment (multimedia equipment interconnection, *etc*.). There are also three network load divisions, including [0,0.3), [0.3,0.7], and (0.7,1.0]. Thereby, the size of the state space is 4^3.

Fig. 5a illustrates the whole procedure of policy deployment in the IDT-SDVN controller. Once the central controller receives a routing request from vehicles, it first determines the road path $\mathcal{L}_{bas}$ for this request based on G_{bas}, as shown in steps (1) and (2), and then calculates the road load by

$$load_{bas} = \min\left(\frac{L^{bas}_{avg}}{Bw}, 1\right), \tag{11}$$

where L^{bas}_{avg} is the average buffer usage, formulated by $L^{bas}_{avg} = L^{bas}_{sum} nv_{bas}$. nv_{bas} and L^{bas}_{sum} separately indicate the number of vehicles and the total occupied buffer length of all vehicles on the path $\mathcal{L}_{bas}$. Bw is the channel bandwidth. Step (3) determines the current state based on the requirement type and the calculated path load. Then, it is easy to obtain the most suitable policy for application by querying the state-action table. As demonstrated in step (4), we assume that the current state is S_1; then, G_{LDF} is selected because it has the maximum value. After that, the controller calculates a junction sequence $\mathcal{L}_{LDF}$ via G_{LDF} and sends it to the requester, as presented in steps (5)-(7).

4.3.2 Road-Based Hierarchical Routing Strategy

The proposed road-based hierarchical routing strategy is a combination of a junction selection strategy in the controller and a relay vehicle selection strategy in distributed vehicular networks.

(1) *Junction selection*. Here we introduce how to utilize the deployed policy $G_{op} \in \mathbb{G}$ to calculate a junction sequence $\mathcal{L}_{sd}$, thereby determining the road path from the source junction J_s to the target junction J_d. In $\mathcal{L}_{sd}$, each item J_c is the best choice of its previous item J_p, obtained by

$$c = \max_{\forall J_n \in N_p} G_{op}(p, d, n), \tag{12}$$

where c, p, d, and n are the IDs of J_c, J_p, J_d, and J_n, respectively. After the computation, $\mathcal{L}_{sd}$ is sent to the routing requester in the form of a reply message.

(2) *Relay vehicle selection*. Once receiving the reply message sent from the controller, the requester inserts the assigned road path into the header of packets, and all packet receivers select the next forwarder along this path based on the road-based greedy routing strategy. In this case, packets are delivered hop-by-hop along the path until achieving the destination vehicle.

We define V_c as the current vehicle, which is located within the coverage of the current junction J_c. Here, V_d is the destination vehicle in the target junction J_d, and J_n is the next junction of J_c recorded in the junction sequence $\mathcal{L}$. According to the designed relay vehicle selection strategy, there are two cases we need to consider when selecting the relay vehicle on the road.

Case 1, meeting the greedy condition: If there are more than one V_c's neighbor vehicles located closer to J_n/V_d, the current vehicle V_c will select the neighbor that is the closest to J_n/V_d.

Case 2, violating the greedy condition: If a void appears, that is, there is no V_c's neighbor vehicle located nearer to J_n/V_d, V_c will select the neighbor that is nearest to the current junction J_c. This is because data packets inside junctions are given more forwarding opportunities from vehicle aggregation.

4.3.3 State-Action Table Maintenance

A stable state-action table is a key foundation to provide efficient and accurate policy deployment. The iterations for the table are performed upon reception of report messages. As shown in Fig. 5b, in VNs, the packet receiver or failure node reports routing results to the central controller after once packet delivery (step (1)). Then, the controller updates the corresponding item in the state-action table according to the entire transmission path (step (2)) and path information (step (3)). One-step update of each $(state, action)$ pair is defined by

$$P(state, action) += R'(state, action). \tag{13}$$

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.

TABLE 2
Initialization of the State-Action Table

Message type	Load	Value		
		G_{HRF}	G_{LDF}	G_{LBF}
Security	0-1.0	1	0	0
Efficiency	0-1.0	1	1	0
Information service	0-0.3	1	0	0
	0.3-1.0	0	0	1
Entertainment	0-1.0	0	0	1

We define reward R' in light of the routing performance of the deployed policy (step (4)). The four metrics introduced in Section 4.1.1 are taken for the evaluation of the policy deployment mechanism. In this case, we formulate the reward function as

$$\begin{aligned} R' &= f(PRR, AD, HC, RC) \\ &= \alpha \cdot g(AD_{sd}) + \beta \cdot q(HC_{sd}) + \gamma \cdot l(RC_{sd}), \end{aligned} \tag{14}$$

where

$$g(AD_{sd}) = \sum_{J_c \in L_{sd}} R_{AD}(c, d, x_c)/n_{sd} \tag{15}$$

$$q(HC_{sd}) = \sum_{J_c \in L_{sd}} R_{HC}(c, d, x_c)/n_{sd} \tag{16}$$

$$l(RC_{sd}) = \sum_{J_a \in L_{sd}} R_{RC}(c, d, x_c)/n_{sd} \tag{17}$$

denote the average delay-related, hop count-related, and routing cost-related characteristics of path $\mathcal{L}_{sd}$, respectively. J_c is a junction on $\mathcal{L}_{sd}$, and x_c indicates the next junction of J_c. n_{sd} is the number of road segment contained in $\mathcal{L}_{sd}$. Note that $\mathcal{L}_{sd}$ is calculated based on the deployed policy $action$. Details of the variables are presented in Section 4.1.2 (3). α, β, and γ indicate the weights whose crisp values are determined by $action$. For illustration, LDF is defined by $\{AD: 0.5, HC: 0.25, PDR: 0.25\}$ in the fuzzy fusion process (discussed at the end of Section 4.2). Thus, three parameters are $\alpha = 0.5, \beta = 0.25, \gamma = 0$. After the controller receives a path that is determined under state S_1 and policy G_{LDF}, it updates item (S_1, G_{LDF}) in the state-action table by

$$R'_{LDF} = 0.25 \cdot AD_{sa} + 0.25 \cdot HC_{sa} \tag{18}$$

$$P(S_1, G_{LDF}) = P(S_1, G_{LDF}) + R'_{LDF} \tag{19}$$

5 Performance Evaluation

This section is devoted to the performance evaluation of our proposed intELligent dIgital Twin hiErarchical (ELITE) routing. First, we introduce the simulation environment, including basic parameter settings, evaluation metrics, varying scenarios, and representative schemes for comparison. Second, experimental simulations are conducted based on these settings for performance evaluation.

5.1 Simulation Environment

5.1.1 Basic Settings

We downloaded the simulation map from OpenStreetMap and generated traffic flows by Simulation of Urban Mobility (SUMO). The map is a real urban scenario in Shandong province, China of an area of 1.5 km x 1.5 km with 12 junctions and 26 roads. Accordingly, in our simulations, the routing table maintained by the IDT-SDVN controller contains a rough number of 12 x 4 x 11 entries. There are 50 to 250 vehicles per kilometer road and all vehicles have random origins and destinations. The simulation is conducted strictly following the IEEE 802.11p standard. While routing, the number of messages generated per vehicle varies from 1 to 5 packets per second. In the policy application phase, we initialize the state-action table randomly, as shown in Table 2. More simulation environment parameters can be found in Table 3. The source code of our simulation is available online at the link: https://github.com/NetworkCommunication/ELITE-zg.

TABLE 3
Simulation Parameters

Parameter	Value
Network size	1500m×1500 m
Number of junctions	12
Number of vehicles	350, 700, 1050, 1400, 1750
S-D distance	0-375 m, -750 m, -1125 m, -1500 m
Average speed	0∼23 m/s
Communication range	300 m
Generation rate	1∼5 packets/second
Beacon interval	1 s
Data packet size	1024 Bytes for data packets, 256 Bytes for control packets
MAC protocol	IEEE 802.11p

5.1.2 Evaluation Metrics and Scenarios

We introduce the following performance metrics to evaluate the simulation results.

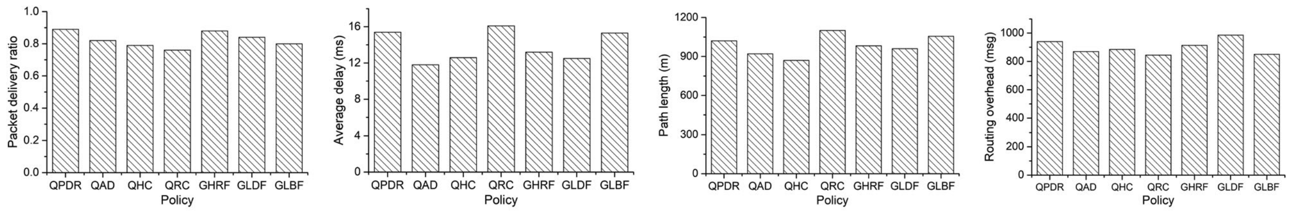
(1) *Packet delivery ratio (PDR)* and (2) *average delay (AD)* are two key metrics that can directly reflect network stability and communication quality in VNs. Further details on these two metrics are described in Section 4.1.1.

(3) *Path length (PL)* indicates the packet transmission distance from generators (source) to receivers (destination), calculated by the sum of the air-line distance between every two adjacent relays.

(4) *Routing overhead (RO)* is defined as the average number of routing control messages generated by relay vehicles for constructing a routing path. The routing control messages include beacons for neighbor detection and lightweight packets for routing requests, path discovery, and essential information reports.

(5) *Control overhead (CO)* is expressed as the average number of network control messages generated per second for maintaining routing tables in case of no data routing. The network control messages include vehicle beacons and vehicular information report messages, and especially in ELITE, the virtual packets for policy training in DTNs are also considered. Generally, generated network control messages allow basic information sharing between vehicles and RSUs and benefit to online routing model updating, but can cause abundant computation and bandwidth resource consumption. In this paper, CO acts as a vital performance

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c) (d)

Fig. 6. Performance comparison among policies in $\mathbb{Q}$ and $\mathbb{G}$ in terms of (a) PDR, (b) AD, (c) PL, and (d) RO.

metric to evaluate the essential resource consumption for realizing schemes/policies in both virtual and physical sides of IDT-SDVNs.

To comprehensively evaluate the performance of ELITE, extensive experimental simulations are performed under three scenarios with different parameter constraints.

(1) *Number of vehicles*. To demonstrate the impact of different vehicle densities on ELITE, we set up five groups of experiments with a different number of on-road vehicles from 350 to 1750, as presented in Table 3.

(2) *Average distance from Sources to Destinations (S-D distance)*. To evaluate the performance of ELITE in both short-distance and long-distance data transmission, in simulations, the distance between each pair of source and destination is uniformly distributed in (0 m-375 m), (375 m-750 m), (750 m-1125 m), and (1125 m-1500 m).

(3) *Number of packets*. By dynamically changing the packet generation rate, ELITE is evaluated under various network load situations. With 1050 vehicles on the map, we set the total number of generated packets varying from 1000 to 5000 per second and carry out five groups of simulation experiments accordingly.

5.1.3 Schemes for Comparison

We compare our proposed ELITE scheme with three different algorithms, Q-learning-based Cognitive Routing (QCR) [9], Intersection-based V2X routing via Q-learning (IV2XQ) [14], and Intersection Gateway and Connectivity based Routing (IGCR) [41]. We also consider two scenarios that ELITE with the application of RSU and QCR with the application of DT to present a more comprehensive comparison and evaluation.

(1) QCR is an intelligent routing scheme switching scheme in Software-Defined Cognitive Network for IoV, in which the Q-learning method is utilized to learn the optimal strategy in varying traffic scenarios. It enabled adaptive strategy switching between Greedy Perimeter Stateless Routing (GPSR) and Ad hoc On-Demand Distance Vector Routing (AODV) according to the sensed network status.

(2) IV2XQ is a hierarchical routing scheme in road-based vehicular networks. It consists of two parts: a multidimensional Q-table that is established based on historical traffic flow and implemented to select road paths, and an improved greedy strategy to select relay vehicles.

(3) IGCR is a distributed routing scheme of VANETs. During routing, traffic-aware routing metrics such as node mobility and traffic density are considered for route discovery with high connectivity, while node position, velocity, and direction are used to select relay nodes.

(4) In ELITE-RSU, RSUs are considered and responsible for data transmission at junctions. When the packet sender reaches a junction, it will forward packets to the nearby RSU. This RSU selects one vehicle which is the geographically closest to the next target junction or destination vehicle among its connected vehicles as the next relay node.

(5) In QCR-DT, the Q-table training process is online and executed in the DTN, in which the DTN has the same network setup as the physical network. During real data routing, decisions are made entirely based on the Q-table rather than the epsilon-greedy policy.

5.2 Performance Comparison Among Single-Target Policies and Generated Policies

In this section, we show a comprehensive comparison among single-target policies in $\mathbb{Q}$ and generated policies in $\mathbb{G}$ in terms of *PDR*, *AD*, *PL*, and *RO* in Figs. 6a, 6b, 6c, and 6(d). Policies are evaluated under the scenario 1050 vehicles on the map with the S-D distance and packets generation rate evenly ranging in 0$\sim$1500 m and 1$\sim$5 packet/second, respectively.

As shown in Fig. 6a, the PDR of utilizing Q_{PDR} and G_{HRF} performs better than other policies. This is mainly because, in these policies, road paths with a high probability of successful packet delivery have been highlighted during the exploration and experience accumulation process in DTNs. Different from this, rewards designed for training Q_{AD}, Q_{HC}, and Q_{RC} only consider average end-to-end delay, hop count, and cost on a road, respectively. Similarly in Figs. 6b, 6c, and 6(d), packet delivery abilities of roads in terms of different metrics are well distinguished by $\mathbb{Q}$ and $\mathbb{G}$. It is worth illustrating that although routing based on $\mathbb{G}$ sometimes performs worse than $\mathbb{Q}$ in certain metrics, it shows better comprehensive performance as policies in $\mathbb{G}$ tend to find a trade-off among multiple metrics for satisfying diverse service requirements. Q_{RC} presents the longest delay and routing path due to the selection of geographically non-optimal roads in path scheduling. However, it can reduce the number of generated control messages during routing significantly. In practice, G_{LBF} is always used in scenarios with high network load such as multimedia data transmission to ensure the load balancing and guarantee service quality.

As discussed above, it is obvious that implementing DT in VNs is feasible and effective. This point can be demonstrated in two aspects. First, compared with conventional online model learning mechanisms, policy learning in DTNs alleviates the impact of network performance caused by frequent exploration processes in the physical world significantly (about 20% packet loss as shown in Fig. 6a).

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.

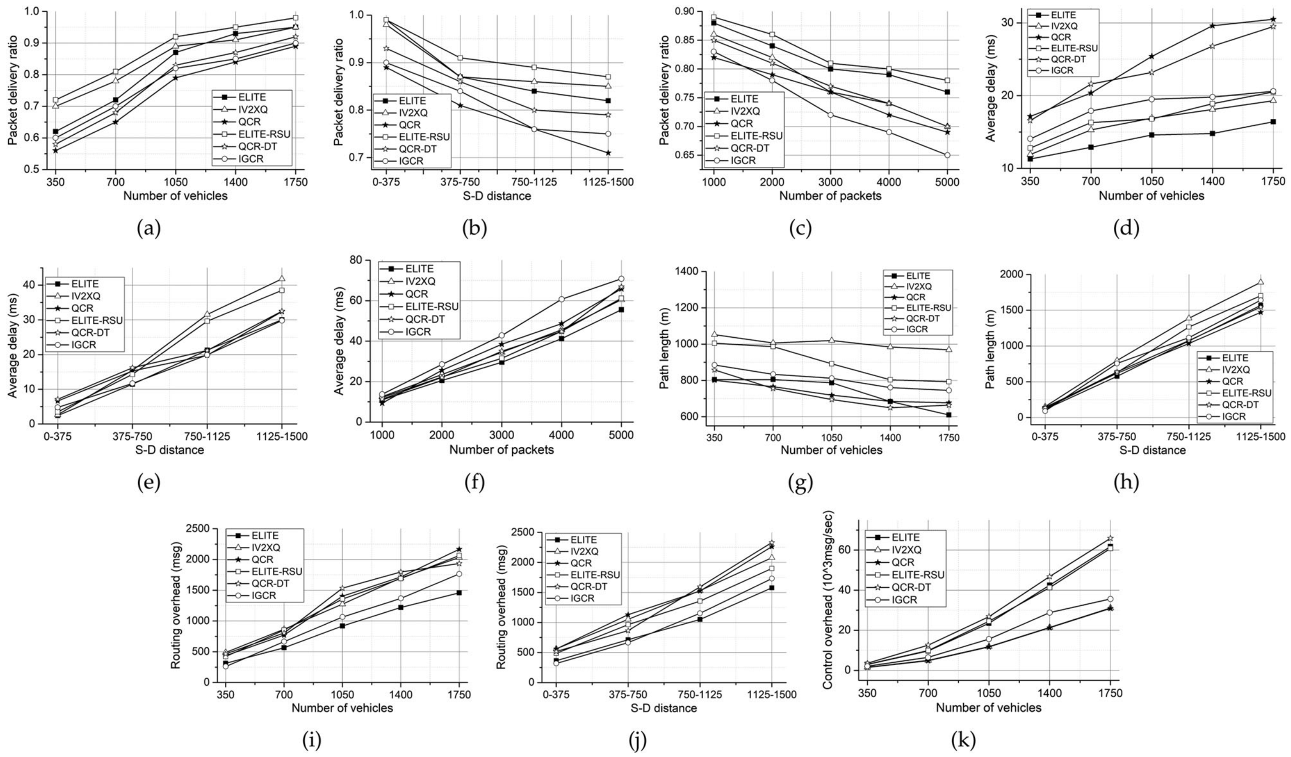


Fig. 7. Packet delivery ratio under (a) various number of vehicles, (b) various S-D distance, and (c) various number of packets. Average end-to-end delay under (d) various number of vehicles, (e) various S-D distance, and (f) various number of packets. Path length under (g) various number of vehicles and (h) various S-D distance. Routing overhead under (i) various number of vehicles and (j) various S-D distance. (k) Control overhead under various number of vehicles.

Policies that adapt to current network situations can be learned by DTN-agent pairs and used in physical networks for routing discovery. Second, single-target policies are scalable and easy to train. By expanding $\mathbb{G}$, generated policy set shows great potential in satisfying the diverse changing scenarios and temporal service requirements. More comprehensive evaluations of ELITE and further verification are presented in the next section.

5.3 Simulation Performance Evaluation for ELITE

5.3.1 Evaluation of PDR

(1) *Varying number of vehicles*. Fig. 7a illustrates the *packet delivery ratio* for varying vehicle densities with the average S-D distance uniformly ranging from 0 m to 1500 m. This constraint is also applied in the evaluations of other metrics. As shown, with an increasing number of vehicles, the PDR of all the six schemes shows an upward trend. This tendency occurs because, in low-density scenarios, packets are more likely to be trapped in *dead zones* that have no valid next-hop candidates, while in high-density scenarios, this dilemma is greatly improved. In our proposed ELITE, road paths have been extensively evaluated in parallel digital twin networks before being deployed. Thus, it shows acceptable performance in PDR and keeps an increasing trend, especially when the number of vehicles exceeds 1400, the PDR is greater than 90%. As shown, IV2XQ has the best performance when the number of vehicles is less than 1400, which mainly benefits from it utilizing RSUs at junctions to forward packets; therefore, even at junctions with sparse vehicle distribution, packets can also be successfully transmitted. By applying RSUs at junctions, ELITE-RSU can increase the PDR under all vehicle densities significantly. Packets reaching junctions can be directly forwarded to relays located at the next intermediate road. Compared with QCR, QCR-DT achieves about 10% performance improvement as its routing policy selection strategy has been pre-verified in DTNs before applying in physical networks.

(2) *Varying S-D distance*. Fig. 7b shows the changes of *packet delivery ratio* under varying S-D distances with the number of vehicles in the network being 1500. As the transmission distance expands, the PDR of all schemes decreases, because long transmission needs frequent relay node switches, which may accumulate the risks of packet loss. Similar to the scenes in varying vehicle densities, when the S-D distance exceeds a certain threshold (750 m), the performance of our proposed ELITE in PDR is worse than that of IV2XQ. This can be explained in two aspects: First, ELITE pays attention to the whole network performance and aims to provide routing support for diversified requirements rather than a certain metric. Second, IV2XQ utilizes RSUs to forward packets at junctions, and this process can substantially improve the final packet delivery ratio. In ELITE-RSU, RSUs at junctions empower the routing paths with stronger connectivity, thereby optimizing the routing performance of ELITE in terms of PDR. Moreover, this improvement expands with S-D distance as the curves shown in Fig. 7b. All schemes completely or partially implement greedy selection. However, while making relay selection decisions, ELITE and IV2XQ pre-calculate the junction sequence that has the optimal network condition, while QCR and IGCR entirely relies on topological information. Thus, both ELITE

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.

and IV2XQ show better performance in PDR than QCR and IGCR.

(3) *Varying number of packets*. Fig. 7c shows the *packet delivery ratio* versus network load. As the number of generated packets increases, the PDR values of all schemes decreases. This result occurs because of the increasing risk of network congestion. Under all network load conditions, the proposed ELITE and ELITE-RSU perform better than others. This can be explained from two perspectives. (i) Routing along the paths calculated based on the policy Q_{RC} generates fewer routing control messages. (ii) ELITE and ELITE-RSU also implement an intelligent policy application mechanism to switch policies adaptively according to real-time network load and message type. In this way, alternative routing paths are considered for certain data transmissions to keep the global load balance. Benefited from pre-verification, QCR-DT performs better than QCR. However, due to the performance bottleneck of GPSR and AODV, this improvement is weakened in the case of more than 3000 packets in the network.

5.3.2 Evaluation of AD

(1) *Varying number of vehicles*. Fig. 7d shows the effect of vehicle density on the *average end-to-end delay*. For all schemes, the AD increases with the vehicle density because during the next-hop relay selection process, packet receivers must deal with more one-hop information, thereby increasing the calculation delay. Compared with IV2XQ, QCR, and IGCR, ELITE achieves the best performance in AD under all vehicle densities for the following reasons. First, before policy deployment, a certain policy has been generated with the target of low delay. Second, during the path planning process, the controller maintains a routing table set for all times and is enabled to calculate road paths by only querying those tables. Third, during the routing process, only position information is used to select the next-hop vehicle. As a result, the AD is considerably reduced. Among all comparisons, QCR has the longest delay, especially when the number of vehicles increases beyond 1050, the average delay increases sharply, as there exists long time consumption caused by frequent plane topology computation (in GPSR) and route discovery (in AODV and IGCR). It is worth discussing that the AD of ELITE-RSU is nearly double of ELITE and similar to IV2XQ. This is mainly because a large number of packets sent to the same RSU may lead to temporary data congestion and further long delay. Moreover, in scenarios with high vehicular density, packets generally can leave a junction within one or two hops, thereby leading to a shorter transmission delay compared with forwarding packets by RSUs.

(2) *Varying S-D distance*. The performance comparison for evaluating the *average end-to-end delay* over the S-D distance is shown in Fig. 7e. Results express that the AD of all schemes grows as the S-D distance gets longer. The main reason is that a long S-D distance leads to more hops, and further increases computation time, transmission time, and sometimes queue time. Compared with IV2VQ, QCR, and IGCR, ELITE maintains a shorter delay at different transmission distances. The key reason is that ELITE directly targets routing metrics, including AD, while training models. Nevertheless, IV2XQ, QCR, and IGCR only consider some basic road network-related attributes, such as vehicle density and speed. In ELITE, benefited from the hierarchical routing structure, the packet holder far from its destination can perceive the network status beyond one hop to some extent, rather than select relays with limited network information.

(3) *Varying number of packets*. Fig. 7f shows the *average end-to-end delay* of these schemes as the network load situation varies. As the number of packets flowing in the network increases, the AD climbs up because the increment in the risk of data congestion cause longer queuing time during hop-by-hop transmission. Compared with other routing methods without the consideration of routing cost, the proposed ELITE and ELITE-RSU achieve a low AD benefited by the load balancing mechanism. The routing of IGCR is restricted by roads but pays more attention to common features such as vehicular density. This must cause severe data congestion on certain roads and further long delay or even packet loss.

5.3.3 Evaluation of PL

(1) *Varying number of vehicles*. Fig. 7g shows the relationships between the average routing *path length* and the number of vehicles. The PL of all schemes decreases as the number of vehicles increases. In fact, the PL mainly depends on the S-D distance, hop count, and road path length (in road-based routing). On the one hand, once position-based greedy routing strategies are used, the one-hop transmission distance most likely increases with the number of neighbor vehicles around packet holders, thereby decreasing the hop count from sources to destinations. On the other hand, in ELITE, the increment of vehicle density at junctions decreases search times caused by Case 2 in Section 4.3.2 (2), therefore making the next junction easier to reach with a few hops. In IV2XQ and ELITE-RSU, forwarding packets by RSUs at junctions markedly increases the hop count. Moreover, the application of alternative paths in IV2XQ also leads to a long road path. The PL of QCR and QCR-DT is shorter than others and remains relatively low. The main reason is that packet transmission is position-based rather than constrained by tortuous road paths. Conversely, IGCR performs worst among RSU-free schemes due to the less consideration of hop count-related features such as inter-vehicle distance while selecting relay vehicles.

(2) *Varying S-D distance*. Fig. 7h analyzes how the S-D distance impacts the average routing *path length*. As shown, there are several observations to be explained. (i) The PL becomes longer for all schemes as the S-D distance increases. (ii) When the S-D distance exceeds 750, IV2XQ, IGCR, ELITE, and ELITE-RSU have a longer PL compared with QCR and QCR-DT, as they must forward packets along roads, while QCR and QCR-DT completely depend on the position information for packet delivery. (iii) Nevertheless, ELITE alleviates this restriction because of the consideration of hop count while generating junction selection policies. Therefore, the average path length of ELITE is shorter than that of the compared road-based routing schemes.

5.3.4 Evaluation of RO

(1) *Varying number of vehicles*. Fig. 7i demonstrates the results of evaluating the *routing overhead* for these policies/schemes

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.

under varying numbers of vehicles. As the number of vehicles increases, the *RO* of IV2XQ and QCR increases rapidly, while that of ELITE remains at a relatively low level for the following reasons. First, unlik IV2XQ, QCR, and IGCR, ELITE takes the number of generated control messages as a crucial metric while training policies. This means that once congestion occurs, the controller can plan road paths based on G_{LBF} for low-priority messages to balance the global overhead. In addition, in IV2XQ, QCR, and IGCR, vehicles need to frequently exchange information messages with their neighbors or RSUs for route discovery while routing. These processes result in high communication overhead.

(2) *Varying S-D distance*. Fig. 7j shows the *RO* of the three routing schemes under different S-D distances. It is obvious that for all comparisons, the *RO* increases with increasing S-D distance. This is mainly because the longer paths there are to be established, the more links there are to be maintained, resulting in an increase in the number of routing control messages for exchanging essential information between adjacent vehicles/RSUs. Compared with IV2XQ, QCR, and IGCR, the proposed ELITE achieves better performance in *RO* since it trains a load control policy to balance the global overhead. IV2XQ and QCR show similar performance in terms of *RO*. Although IV2XQ utilizes alternative routing paths for load balancing, it needs additional information exchanges between RSUs and their neighboring vehicles at junctions. Therefore, it leads to a rise in the number of routing control messages. In QCR, vehicles should periodically broadcast hello messages to maintain the neighbor list and generate massive information messages to discover routes in some cases. In IGCR, vehicles at junctions should interact with gateway fixed nodes for routing discovery frequently, thereby leading to extra *RO*. Due to the increasing number of junctions contained in the routing path, the curve of IGCR rises with the S-D distance dramatically.

5.3.5 Evaluation of CO

Fig. 7k shows the *control overhead* against the increase in the number of vehicles. The *CO* increases for all schemes with the vehicle density increasing, since both the controller/server and vehicles have more tasks to execute due to the growth of the number of communication requests and network control messages. As shown, the *CO* of IV2XQ, QCR, and IGCR is similar and lower than that of our proposed ELITE. The results can be explained in two aspects. First, vehicles under both these schemes should periodically broadcast beacons and send location-report messages to base stations or RSUs. Second, in ELITE, the central controller develops a group of DTNS for single-target policy learning, and this learning process is continuous and parallel. That means a large number of virtual communication processes are executed by each agent-DTN pair to output a satisfactory routing policy. This situation also occurs in QCR-DT. As the settings in Table 2, about 10$\sim$80 virtual messages are generated in each DTN per second. Therefore, the *CO* of ELITE, ELITE-RSU, and QCR-DT climbs up dramatically as the vehicle density increases, and then keeps at a high level. Nevertheless, it is essential to explain that even the utilization of DT accompanied by extensive resource cost, the performance of ELITE in terms of *PDR*, *AD*, *PL*, and *RO* have gained great improvement. Therefore, the extra resource consumption in the policy generation process in ELITE is acceptable.

6 Conclusion

This paper presents the ELITE routing scheme in IDT-SDVN that aims to provide a road-based hierarchical routing path planning method for frequent topological changes and complex service requests in vehicular networks. First, a group of DTN-agent pairs is deployed in the IDT-SDVN controller to learn single-target policies. Second, fuzzy inference is responsible for combining the learned results into new policies. Then, we investigate an adaptive policy deployment mechanism to select the most suitable policy according to service requirements and traffic situations. Finally, during transmission, a road path is planned based on the deployed policy, and packets are greedily forwarded along the path hop-by-hop.

The principal theoretical implication of this study is that IDT-SDVN has the advantages of realizing intelligent generation and adaptive iteration of routing policies. Simulation results show that in multi-hop data transmission in vehicular networks, our proposed scheme achieves considerable improvement in the packet delivery ratio, end-to-end delay, and routing path length while maintaining network load in an appropriate range compared with existing routing schemes. For the future, by optimizing the IDT-SDVN structure in terms of resource savings, more virtual network spaces can be developed in the central controller, ensuring faster learning speed, encompassing more routing metrics, and covering more types of services. In addition, developing traffic prediction technologies and thereby establishing future topology-based DTNs is also a crucial issue to expand the functions of IDT-SDVN.

References

[1] G. Karagiannis *et al.*, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Commun. Surv. Tut.*, vol. 13, no. 4, pp. 584–616, Oct.–Dec. 2011.

[2] H. H. Jeong, Y. C. Shen, J. P. Jeong, and T. T. Oh, "A comprehensive survey on vehicular networking for safe and efficient driving in smart transportation: A focus on systems, protocols, and applications," *Veh. Commun.*, vol. 31, 2021, Art. no. 100349.

[3] B. Chaurasia, W. Manjoro, and M. Dhakar, "Traffic congestion identification and reduction," *Wireless Pers. Commun.*, vol. 114, pp. 1267–1286, Sep. 2020.

[4] A. Hawbani *et al.*, "Zone probabilistic routing for wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 3, pp. 728–741, Mar. 2019.

[5] H. I. Abbasi, R. C. Voicu, J. A. Copeland, and Y. Chang, "Towards fast and reliable multihop routing in vanets," *IEEE Trans. Mobile Comput.*, vol. 19, no. 10, pp. 2461–2474, Oct. 2020.

[6] M. Sepulcre and J. Gozalvez, "Heterogeneous V2V communications in multi-link and multi-rat vehicular networks," *IEEE Trans. Mobile Comput.*, vol. 20, no. 1, pp. 162–173, Jan. 2021.

[7] L. Zhao, J. Li, A. Al-Dubai, A. Y. Zomaya, G. Min, and A. Hawbani, "Routing schemes in software-defined vehicular networks: Design, open issues and challenges," *IEEE Intell. Transp. Syst. Mag.*, vol. 13, no. 4, pp. 217–226, Winter 2021.

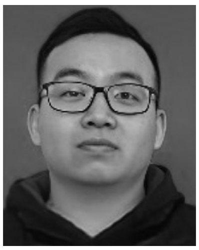
[8] L. Zhao, T. Zheng, M. Lin, A. Hawbani, J. Shang, and C. Fan, "SPIDER: A social computing inspired predictive routing scheme for softwarized vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, early access, Oct. 29, 2021, doi: 10.1109/TITS.2021.3122438.

[9] C. Wang, L. Zhang, Z. Li, and C. Jiang, "SDCoR: Software defined cognitive routing for Internet of Vehicles," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3513–3520, Oct. 2018.

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.

[10] J. Wang, K. Liu, K. Xiao, X. Wang, Q. Han, and V. C. S. Lee, "Delay-constrained routing via heterogeneous vehicular communications in software defined busnet," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5957–5970, Jun. 2019.
[11] X. Zhang, X. Cao, L. Yan, and D. K. Sung, "A street-centric opportunistic routing protocol based on link correlation for urban VANETs," *IEEE Trans. Mobile Comput.*, vol. 15, no. 7, pp. 1586–1599, Jul. 2016.
[12] Y. R. B. Al-Mayouf *et al.*, "Real-time intersection-based segment aware routing algorithm for urban vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2125–2141, Jul. 2018.
[13] G. Sun, Y. Zhang, H. Yu, X. Du, and M. Guizani, "Intersection fog-based distributed routing for V2V communication in urban vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2409–2426, Jun. 2020.
[14] L. Luo, L. Sheng, H. Yu, and G. Sun, "Intersection-based V2X routing via reinforcement learning in vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 02, 2021, doi: 10.1109/TITS.2021.3053958.
[15] M. Camelo, M. Claeys, and S. Latré, "Parallel reinforcement learning with minimal communication overhead for IoT environments," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1387–1400, Feb. 2020.
[16] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 13 789–13 804, Sep. 2021.
[17] Z. Wang *et al.*, "A digital twin paradigm: Vehicle-to-cloud based advanced driver assistance systems," in *Proc. IEEE 91st Veh. Technol. Conf.*, 2020, pp. 1–6.
[18] P. Wang, N. Xu, W. Sun, G. Wang, and Y. Zhang, "Distributed incentives and digital twin for resource allocation in air-assisted Internet of Vehicles," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2021, pp. 1–6.
[19] K. Zhang, J. Cao, S. Maharjan, and Y. Zhang, "Digital twin empowered content caching in social-aware vehicular edge networks," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 1, pp. 239–251, Feb. 2022.
[20] X. Xu *et al.*, "Service offloading with deep Q-network for digital twinning empowered Internet of Vehicles in edge computing," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1414–1423, Feb. 2020.
[21] K. Zhang, J. Cao, and Y. Zhang, "Adaptive digital twin and multiagent deep reinforcement learning for vehicular edge computing and networks," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1405–1413, Feb. 2021.
[22] J. Liu, Y. Dong, Y. Liu, P. Li, S. Liu, and T. Wang, "Prediction study of the heavy vehicle driving state based on digital twin model," in *Proc. IEEE Int. Conf. Power Electron., Comput. Appl.*, 2021, pp. 789–797.
[23] C. Hu *et al.*, "A digital twin-assisted real-time traffic data prediction method for 5G-enabled Internet of Vehicles," *IEEE Trans. Ind. Informat.*, vol. 18, no. 4, pp. 2811–2819, Apr. 2022.
[24] L. Zhao, G. Han, Z. Li, and L. Shu, "Intelligent digital twin-based software-defined vehicular networks," *IEEE Netw.*, vol. 34, no. 5, pp. 178–184, Sep./Oct. 2020.
[25] Y. Gao, Z. Zhang, D. Zhao, Y. Zhang, and T. Luo, "A hierarchical routing scheme with load balancing in software defined vehicular ad hoc networks," *IEEE Access*, vol. 6, pp. 73 774–73 785, 2018.
[26] C. Ghorai, S. Shakhari, and I. Banerjee, "ASPEA-based multimetric routing protocol for intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6737–6747, Nov. 2020.
[27] A. Hawbani, E. Torbosh, X. Wang, P. Sincak, L. Zhao, and A. Al-Dubai, "Fuzzy-based distributed protocol for vehicle-to-vehicle communication," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 3, pp. 612–626, Mar. 2019.
[28] C. Ghorai, S. Shakhari, and I. Banerjee, "A spea-based multimetric routing protocol for intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6737–6747, Nov. 2021.
[29] X. M. Zhang, L. Yan, K. H. Chen, and D. K. Sung, "Fast, efficient broadcast schemes based on the prediction of dynamics in vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 531–542, Feb. 2020.
[30] G. Sun, Y. Zhang, D. Liao, H. Yu, X. Du, and M. Guizani, "Bus-trajectory-based street-centric routing for message delivery in urban vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7550–7563, Aug. 2018.
[31] K. Z. Ghafoor, L. Kong, D. B. Rawat, E. Hosseini, and A. S. Sadiq, "Quality of service aware routing protocol in software-defined Internet of Vehicles," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2817–2828, Apr. 2019.
[32] K. Mershad, "SURFER: A secure SDN-based routing protocol for Internet of Vehicles," *IEEE Internet Things J.*, vol. 8, no. 9, pp. 7407–7422, May 2021.
[33] K. L. K. Sudheera, M. Ma, and P. H. J. Chong, "Link stability based optimized routing framework for software defined vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2934–2945, Mar. 2019.
[34] O. Alzamzami and I. Mahgoub, "Fuzzy logic-based geographic routing for urban vehicular networks using link quality and achievable throughput estimations," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2289–2300, Jun. 2019.
[35] Z. Cao, B. N. Silva, M. Diyan, J. Li, and K. Han, "Intersection routing based on fuzzy multi-factor decision for vanets," *Appl. Sci.*, vol. 10, no. 18, 2020, Art. no. 6613.
[36] 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.
[37] C. Wu, T. Yoshinaga, Y. Ji, and Y. Zhang, "Computational intelligence inspired data delivery for vehicle-to-roadside communications," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12 038–12 048, Dec. 2018.
[38] M. A. Al-Jarrah, A. Al-Dweik, M. Kalil, and S. S. Ikki, "Decision fusion in distributed cooperative wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 797–811, Jan. 2019.
[39] J. Liu, J. Huang, R. Sun, H. Yu, and R. Xiao, "Data fusion for multi-source sensors using GA-PSO-BP neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 10, pp. 6583–6598, Oct. 2021.
[40] P. Wang, W. Hao, and Y. Jin, "Fine-grained traffic flow prediction of various vehicle types via fusion of multisource data and deep learning approaches," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6921–6930, Nov. 2021.
[41] K. N. Qureshi, M. Ahmed, G. Jeon, and F. Piccialli, "An enhanced multi-hop intersection-based geographical routing protocol for the internet of connected vehicles network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3850–3858, Jun. 2021.

Liang Zhao (Member, IEEE) received the PhD degree from the School of Computing, Edinburgh Napier University, in 2011. He is currently a professor with Shenyang Aerospace University, China. His research interests include ITS, VANET, WMN, and SDN. He has published more than 120 papers. He served as the chair of several international conferences and workshops, including 2022 IEEE BigDataSE (steering co-chair), 2021 IEEE TrustCom (program co-chair), 2019 IEEE IUCC (program co-chair), and 2018-2021 NGDN workshop (founder). He was the recipient of the best/outstanding paper awards with 2015 IEEE IUCC, 2020 IEEE ISPA, and 2013 ACM MoMM.



Zhenguo Bi received the BSc degree from Shenyang Aerospace University. He is currently working toward the master's degree with Shenyang Aerospace University, Shenyang, China. His research interests include ITS, VANET, and SDN.

Ammar Hawbani received the BS, MS, and PhD degrees in computer software and theory from University of Science and Technology of China, in 2009, 2012, and 2016, respectively. He is currently an associate professor of networking and communication algorithms with the School of Computer Science and Technology, University of Science and Technology of China. From 2016 to 2019, he worked as postdoctoral researcher with the School of Computer Science and Technology, University of Science and Technology of China. His research interests include IoT, WSNs, WBANs, WMNs, VANETs, and SDN letters.

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.

Keping Yu (Member, IEEE) received the ME and PhD degrees from the Graduate School of Global Information and Telecommunication Studies, Waseda University, Tokyo, Japan, in 2012 and 2016, respectively. He was a research associate, junior researcher, and researcher with the Global Information and Telecommunication Institute, Waseda University, from 2015 to 2019, 2019 to 2020, and 2020 to 2022, respectively. He is currently an associate professor with Hosei University and a visiting scientist with the RIKEN Center for Advanced Intelligence Project, Japan. He has hosted and participated in more than ten projects, is involved in many standardization activities organized by ITU-T and ICNRG of IRTF, and has contributed to ITU-T Standards Y.3071 and Supplement 35. He received the IEEE Outstanding Leadership Award from IEEE BigDataSE 2021, the Best Paper Award from IEEE Consumer Electronics Magazine Award 2022 (first Place Winner), IEEE ICFTIC 2021, ITU Kaleidoscope 2020, the Student Presentation Award from JSST 2014. He has authored 100+ publications including papers in prestigious journal/conferences such as the *IEEE Wireless Communications*, *IEEE Communications Magazine*, *IEEE Network Magazine*, *IEEE Internet of Things Journal*, *IEEE Transactions on Fuzzy Systems*, *IEEE Transactions on Industrial Informatics*, *IEEE Transactions on Intelligent Transportation Systems*, *IEEE Transactions on Vehicular Technology*, *IEEE Transactions on Mobile Computing*, *Journal of Biomedical and Health Informatics*, *IEEE Transactions on Reliability*, *IEEE Transactions on Communications*, *IEEE Transactions on Network and Service Management*, *IEEE Transactions on Instrumentation and Measurement*, *IEEE Transactions on Network Science and Engineering*, *IEEE Transactions on Green Communications and Networking*, *IEEE Transactions on Computational Social Systems*, *IEEE Consumer Electronics Magazine*, *IEEE Internet of Things Magazine*, ICC, GLOBECOM etc. He is also an associate editor of *IEEE Open Journal of Vehicular Technology*, *Journal of Intelligent Manufacturing*, *Journal of Circuits, Systems and Computers*. He has been a guest editor for more than 20 journals such as *IEEE Transactions on Computational Social Systems*. He served as general co-chair and publicity co-chair of the IEEE VTC2020-Spring 1st EBTSRA workshop, general co-chair of IEEE ICCC2020 2nd EBTSRA workshop, general co-chair of IEEE TrustCom2021 3nd EBTSRA workshop, session chair of IEEE ICCC2020, ITU Kaleidoscope 2016. His research interests include smart grids, information-centric networking, Internet of Things, artificial intelligence, blockchain, and information security.

Yan Zhang (Fellow, IEEE) is currently a full professor with the University of Oslo. He is also an editor of several IEEE publications, including *IEEE Communications Magazine*, *IEEE Network*, *IEEE Transactions on Green Communications and Networking*, *IEEE Communications Surveys and Tutorials*, and the *IEEE Internet of Things Journal*. His current research interests include next generation wireless networks leading to 5G and cyber physical systems. He is a fellow of IET, an elected member of Academia Europaea, an elected member of the Royal Norwegian Society of Sciences and Letters (DKNVS), and an elected member of the Norwegian Academy of Technological Sciences.

Mohsen Guizani (Fellow, IEEE) received the BS (with distinction), MS, and PhD degrees in electrical and computer engineering from Syracuse University, Syracuse, New York. He is currently a professor and the associate provost with Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, UAE. Previously, he worked in different institutions in the USA. His research interests include applied machine learning and artificial intelligence, Internet of Things (IoT), intelligent systems, smart city, and cybersecurity. He was listed as a clarivate analytics highly cited researcher in computer science, in 2019, 2020, and 2021, respectively. He has won several research awards including the "2015 IEEE Communications Society Best Survey Paper Award" as well 4 best paper awards from ICC and Globecom Conferences. He is the author of ten books and more than 800 publications. He is also the recipient of the 2017 *IEEE Communications Society Wireless Technical Committee (WTC) Recognition* Award, the 2018 AdHoc Technical Committee Recognition Award, and the 2019 *IEEE Communications and Information Security Technical Recognition (CISTC)* Award.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.

Authorized licensed use limited to: University of Minnesota. Downloaded on September 14,2025 at 04:52:44 UTC from IEEE Xplore. Restrictions apply.