STALB: A Spatio-Temporal Domain Autonomous Load Balancing Routing Protocol

Yujie Song, Kai Jiang, *Student Member, IEEE*, Yue Cao, *Senior Member, IEEE*, Ruiting Zhou, *Member, IEEE*, Chakkaphong Suthaputchakun, and Yuan Zhuang, *Member, IEEE*

***Abstract*—Due to vehicle mobility, the topology of Vehicle Ad-hoc Networks (VANETs) may change dynamically. High mobility, limited bandwidth, and dynamic network topology pose challenges for communication in the Internet of Vehicles (IoVs). Literature works have attempted to promote efficient (e.g., lower end-to-end latency) message forwarding. However, due to the uncertain direction of message forwarding and vehicle mobility, they suffer from unreachable destinations and unstable connections. This paper explores the efficient method of message forwarding to alleviate network congestion in IoVs. We propose a Spatio-Temporal domain Autonomous Load Balancing (STALB) routing protocol. Specifically, STALB is a trajectory-based method for controlling the direction of message forwarding. STALB can significantly reduce the end-to-end latency and overload ratio, since it considers the local status of network relay devices (i.e., buffer score, congestion status) from the spatio-temporal domain. Then, we present a path reconstruction mechanism, which ensures that messages are forwarded to destinations within limited Time-To-live (TTL). Extensive simulation results show that STALB significantly outperforms other baseline methods (BSaW, TDOR, and TBHGR) regarding overhead ratio, average delivery latency, and average buffer time. Especially, the delivery rate of STALB can reach 99.9% under the sparse network scenario (4,500 messages), at least 0.7% higher than other baseline methods. Similarly, the average delivery delay of STALB is at least 84.31% lower than that of other baseline methods under the dense network scenario (18,000 messages).**

***Index Terms*—Trajectory-based, message forwarding, spatio-temporal domain, autonomous load balancing.**

Manuscript received 26 January 2022; revised 30 June 2022 and 5 September 2022; accepted 15 September 2022. Date of publication 20 September 2022; date of current version 7 March 2023. The research is supported in part by the Wuhan AI Innovation Program (2022010702040056), Suzhou Municipal Key Industrial Technology Innovation Program (SYG202123), and Hubei Province International Science and Technology Collaboration Program (2021EHB012). The associate editor coordinating the review of this article and approving it for publication was C. Avin. *(Corresponding author: Yue Cao.)*

Yujie Song, Kai Jiang, and Ruiting Zhou are with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430000, China (e-mail: y.song@whu.edu.cn; kai.jiang@whu.edu.cn; ruitingzhou@whu.edu.cn).

Yue Cao is with the School of Cyber Science and Engineering and the Suzhou Research Institute of Wuhan University, Wuhan University, Wuhan 430000, China (e-mail: yue.cao@whu.edu.cn).

Chakkaphong Suthaputchakun is with the Electrical and Computer Engineering Department, Bangkok University, Khlong Nueng 12120, Thailand (e-mail: chakkaphong.s@bu.ac.th).

Yuan Zhuang is with the State Key Laboratory of Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430000, China (e-mail: yuan.zhuang@whu.edu.cn).

Digital Object Identifier 10.1109/TNSM.2022.3208025

I. Introduction

ALONG with the recent advances of Internet of Things (IoT) technologies and the popularity of vehicles on the roads, the traditional technology-driven transportation system is evolving into a more powerful data-driven intelligent era. As an essential paradigm in the 5G networks, the Internet of Vehicles (IoVs) has emerged as a reliable technology to provide transport services [1]. These services, especially innovative applications, such as telematics services, vehicle-road coordination, fleet operation, etc., have promoted the path towards intelligent transportation, as well as enhanced driving safety and travel comfort for drivers and passengers [2].

The implementation of above applications intensely depends on the innovation of Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) technologies [3], [4], [5]. Indeed, the network topology may change dynamically due to vehicle mobility. The unstable topology of Vehicle Ad-hoc Networks (VANETs) inevitably causes communication interruption and resources exhaustion. The latency has indirectly reflected the efficiency of message forwarding, while message forwarding aims to ensure the message reaches its destination. Conventional routing algorithms (e.g., DSRC [6], OSPF [7]) cannot fully ensure the quality of the network information, as they always occupy a large amount of communication and computing resources to maintain routing tables. Meanwhile, both high mobility and limited bandwidth pose inevitable challenges for the message forwarding in VANETs [8], [9], [10]. Besides, the heavy network load is a common problem all routing algorithms face [11]. The suboptimal communication resource allocation leads to unbalanced network load, causing messages congestion, long end-to-end latency, and line collapse.

Stemming from the recent literature [12], [13], some methods have promoted efficient message forwarding by controlling the number of message copies and optimizing message forwarding strategy. Generally, due to random vehicle mobility, the message may not be delivered within Time-To-Live (TTL) through store-carry-forward in Delay-Tolerant Networks (DTN), namely message destination unreachable [14]. Therefore, the trajectory-based routing method is considered to alleviate the problems of heavy network load, message destination unreachable, and low message delivery ratio by controlling the direction of message forwarding. Two main trajectory-based routing methods

rely on geographic trajectory and mathematical trajectory, respectively. The former is based on geographic information (location, direction of movement, destination, etc.), and the latter generates the trajectory according to the mathematical formula. Nevertheless, despite some trajectory-based methods [15], [16], [17], [18], [19] being employed to improve message forwarding efficiency, they have not adequately considered network conditions in line with trajectory. Moreover, these methods can not predict the future local status of network relay devices (i.e., buffer score, congestion status), and can't adapt to network change.

Above trajectory-based forwarding modes often have trapped in message unreachable destinations and unstable links. To compensate the network failure through V2V manner, the core network was introduced as a message forwarding medium. For example, concerning the scenario where 5G core network and IoVs are included, the message is forwarded to Road Side Units (RSUs) via V2I and then forwarded through the core network (e.g., the networking between network relay devices or RSUs). We utilize the flexible networking features of V2I to access, and gain the large bandwidth and high stability of the core network. Moreover, we accelerate message delivery at a low cost by controlling the number of copies. Then, we consider the trajectory fitting degree,[1] as the message forwarding control condition. Furthermore, the predicted network status is also be considered for load balancing. Ultimately, based on these concerns, we tackle the message forwarding problems by proposing a Spatio-Temporal domain Autonomous Load Balancing (STALB) routing protocol. The main contributions can be summarized as follows:

1) Several recent works have successfully highlighted the important role of encoding, congestion control, and space distance. However, most of these works do not pay much attention to solve problem via geometric manner, e.g., using trajectory-based method, the local status, and the direction of message forwarding from both time and space dimensions. Therefore, we premeditate an extended message delivery model of SDN, and define the spatio-temporal domain parameters related to this model. The trajectory-based method is applied for controlling the direction of message forwarding in STALB, which reduces the hops of messages. Furthermore, STALB can significantly reduce end-to-end latency and network load ratio (overhead ratio), considering the local status of network relay devices from the spatio-temporal domain.
2) Because STALB dynamically selects the next-hop network relay device for messages, some messages may fall into the problem of unreachable destination. To ensure each message can be successfully forwarded to its destination, the trapped message should reconstruct the reference trajectory path. Therefore, we present a path reconstruction mechanism to address the problem of message unreachable destinations within finite latency, considering the message destinations and forwarding directions. This mechanism will be triggered in following situations: (i) the candidate set of network relay devices is empty. (ii) the message is geographically closed to its destination.

[1]The trajectory fitting degree is to evaluate the degree to which the message fits the reference trajectory during the forwarding process.

TABLE I
RELATED WORKS

V2I Enabled Protocols	Single indicator	[20] [21] [22] [23] [24]
	Multi-indicators	[25] [26] [27] [28]
Core Network Protocols	Load balancing network	[17] [29] [30] [31]
	AI-assisted network	[32] [33] [34]
	Mobile backbone network	[35] [36] [37]

The remainder of this paper is organized as follows. Section II describes the related work. Section III displays a detailed system model and formulates the problem. Section IV shows the algorithm flow of STALB. Furthermore, the experiments are conducted in Section V. Finally, Section VI concludes this paper.

II. RELATED WORK

This section reviews the related work and classifies them in Table I. Based on this, we further explain the motivation for our work.

A. V2I Enabled Protocols

In recent years, extensive studies have focused on solving the access problem in IoVs [20], [21], [22], [23], [24]. Specifically, Sharma and Awasthi [20] exploited a dedicated short-range communication based vehicular communication model, and proposed an adaptive priority based data service scheduling method for efficient information dissemination in the heterogeneous traffic environment. To enable time-slotted multi-hop pipelining for structured data dissemination, Zhao *et al.* [21] tackled the structured bulk data dissemination problem by considering the out-of-order transmission and bursty encoding mechanisms. Furthermore, Liu *et al.* [22] focused on data services with consideration of the time constraint of data dissemination and data freshness, respectively. In [23], Akhtar *et al.* proposed a point-cloud multi-cast architecture based on V2I communication and employed a quadtree point-cloud source encoder with bitrate elasticity, which matches with an adaptive random network coding and maximizes the throughput under the delay constraint. In [24], Pyun *et al.* investigated the collisions caused by multiple vehicles access a single RSU, and proposed a contention-based channel access procedure in the V2I communication.

Further, based on the aforementioned works, the multi-indicators are introduced into V2I since the limited number of indicators will inefficiently deliver in complex network [25], [26], [27], [28]. Liu *et al.* [25] considers the distance between vehicles, throughput, and traffic density. The model uses Shannon's theorem, speed, and distance, introducing background noise to deduce the amount of communication between vehicles and infrastructure in a limited time. Guo *et al.* [26] proposed a dual graph coloring-based interference management scheme (DGCIM), solving the tricky problem of uplink and downlink interference of resource

sharing in V2I. In [27], Wu *et al.* proposed an analytical framework based on stochastic geometry to characterize the uplink V2I transmission performances, considering the average vehicle throughput, outage probability, and average spatial throughput. Furthermore, to maximize V2V throughput under the constraints of minimizing V2I throughput requirements, He *et al.* [28] proposed a resource allocation scheme for the uplink communication network, considering transmit powers, channel gains, and transmitting interference.

However, the above studies mainly focus on encoding, transfer slot efficiency, congestion control, resource allocation, or congestion avoidance, but seldomly consider network-layer technologies in message delivery. Besides, the nontrivial issues (e.g., broadcast storm, lack of message transmit strategy) still exist in message forwarding, degrading message transmission efficiency and network load balancing dramatically.

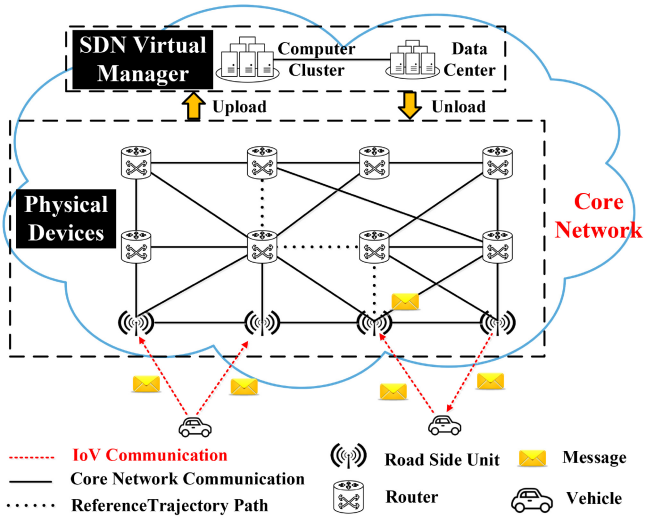

Fig. 1. Multi-copies Message Forwarding Network Architecture in IoVs.

B. Core Network Protocols

To cope with the challenges mentioned above, Software Defined Network (SDN) is introduced to improve routing management, message forwarding, routing algorithm unloading, data co-processing, and network load balancing [17], [29], [30], [31], [32], [33], [34], [35], [36], [37]. Specifically, Zhang *et al.* [17] considered the backpropagation mechanism and proposed a curve-based greedy routing algorithm to forward packets along trajectories. This mechanism can select multiple trajectories to ensure the accurate arrival of messages. Latif *et al.* [29] proposed an optimized and load-balancing path for an inter-domain communication system, which provides strong scalability and programmability for distributing the network traffic load. Similarly, according to the characteristics of heterogeneous traffic types, Trestian *et al.* [30] proposed a traffic dynamic balancing strategy based on openflow to achieve network balancing of the core network. Cui *et al.* [31] proposed a load balancing strategy based on response time for multiple SDN controllers, realizing real-time response by exchanging network relay devices' control.

Meanwhile, to balance the latency and packets delivery ratio, a source routing based flow instantiation scheme was proposed. This scheme delivers flow rules to each node with limited communication cost, providing a caching mechanism [32]. Furthermore, Artificial Intelligence (AI) assisted routing strategies were proposed to deliver data more autonomously and flexibly [33], [34]. In [33], machine learning-aided load balancing routing scheme considers the queue utilization. They predict the network traffic status through machine learning and then manage the queue according to predict outcomes. Casas-Velasco *et al.* [34] is a reinforcement learning routing algorithm based on the SDN framework for improving data delivery efficiency, considering the link-state information with the dynamical traffic.

In addition, a backbone network consisting of mobile nodes is an interesting perspective. Chaib *et al.* [35] proposed a Bus-based routing protocol for data delivery. The periodical routine of buses builds the backbone network for reducing the impact of random mobility of vehicles on data delivery. Wang *et al.* [36] presented a software-defined BusNet and formulated the problem of delay-constrained routing, considering bandwidth efficiency and delivery ratio. A distance-weighted back-pressure dynamic routing protocol is proposed in [37]. The Lyapunov drift theory and buffer queues are considered to maintain network stability on the buffer queues.

C. Motivation

Unfortunately, the aforementioned works only focus on the distance between network relay devices and the trajectory in selecting intermediate nodes, but seldomly (*1*) in addition to enabling V2V manner, we also study the impact of core network to improve the efficiency of message delivery. (*2*) however, the core network only considers the hops of messages, the distance between two network relay devices, end-to-end delay, and buffers. It does not consider the trajectory-based method, the local status, and the direction of message forwarding from both time and space dimensions. Different from them, this paper investigates message forwarding with consideration of the local status of network relay devices, time-varying message buffer time, the closeness of reference trajectory on spatio-temporal domains, to improve message forwarding efficiency while ensuring the delivery ratio.

III. System Model

This section describes the system model from three aspects: network architecture, message delivery model, and mathematical model.

A. Network Architecture

We premeditate an extended message transmission architecture of SDN with a virtual manager, network relay devices, messages, and various vehicles, as shown in Fig. 1. The set of network relay devices,[2] involving routers and RSUs, is

[2]Since RSUs perform the same function as routers in the system, RSUs can be regarded as network relay devices for simplifying the running script configuration.

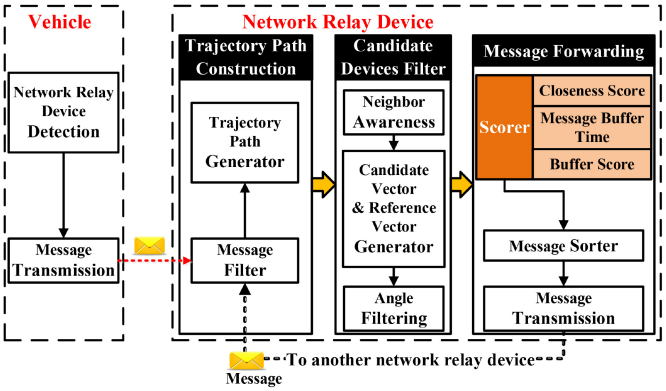

Fig. 2. Multi-copies Message Forwarding Model in IoVs.

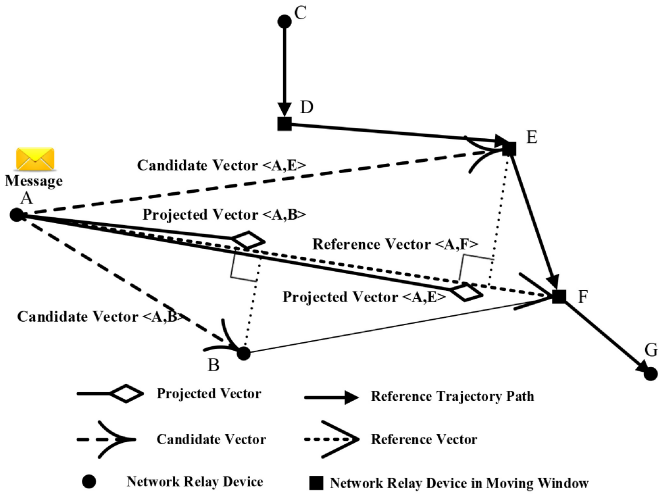

Fig. 3. An example of closeness score.

denoted as $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$. These network relay devices are interconnected to form the core network for providing the message forwarding. Moreover, the SDN virtual manager, composed of computer clusters and data centers, is responsible for generating reference trajectory paths[3] for each message. Notably, the network relay devices are deployed randomly in the digital map, and connected to each other through fixed links. The vehicles (denoted as $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$) are randomly scattered on digitally mapped streets. Without loss of generality, we assume that each vehicle can capture its movement information (current location, moving direction, speed, etc.) by the equipped GPS. Next, messages are generated with dedicated destinations from vehicles with certain TTL. Here, let $\mathcal{M} = \{m_1, m_2, \ldots, m_n\}$ denotes the index set of n messages generated by vehicles, and $\mathcal{S}_m$ denotes the size of message. Besides, the network relay devices and vehicles are endowed with sufficient buffer capacity, such that the messages can be stored without loss due to buffer overflow. Both of them have compatible communication interfaces to establish bi-directional communication.

B. Message Forwarding Model

The transmission of a single message can be considered as a unicast application. Specifically, a network relay device receives messages from different vehicles and the other network relay devices in proximity, determines the correct trajectory path, and ultimately routes the message to the appropriate destination application. This work mainly considers the message forwarding through the core network, with V2V as complementary way for access, whereas that focus through V2V manner can be referred to our previous works [10], [11], [12], [13] but is out of technical scope of this paper.

As shown in Fig. 2, when the vehicle is in the communication range of RSU, the vehicle and RSU establish a link for transmitting messages. At first, the vehicle determines whether to forward message. The RSU receives messages and reports network status (e.g., links status, device status, and destinations of messages) to the SDN virtual manager. The SDN virtual manager will then restore all network status in IoVs. Furthermore, according to the local status of network relay devices, the SDN virtual manager will create a reference trajectory path for the message and restore it into the message's header. After that, the network relay device scores the neighboring devices, considering the reference trajectory path, device's location, buffer capacity, and message buffer time. The next-hop network relay device is selected to perform forwarding with the highest score. Finally, the network relay device will sort all messages and transmit them according to smallest TTL.

[3]The reference trajectory path is a collection of network relay devices that a message may go through.

C. Definitions

1) Spatio Domain Parameters:

Definition 1 (Closeness score): It measures the network relay device's proximity to the reference trajectory path, denoted by $ct_{m_i}^{r_i}$.

$$ct_{m_i}^{r_i} = \frac{\left(cv_{m_i}^x \cdot ov_{m_i}^x + cv_{m_i}^y \cdot ov_{m_i}^y\right)}{\sqrt{\left(ov_{m_i}^x\right)^2 + \left(ov_{m_i}^y\right)^2}}, \tag{1}$$

where $ov_{m_i}^x$ and $ov_{m_i}^y$ are the *X* and *Y* axis coordinates values of the reference vector (ov_{m_i}), respectively. Also, $cv_{m_i}^x$ and $cv_{m_i}^y$ are the *X* and *Y* axis coordinates values of the candidate vector (cv_{m_i}), respectively. The source node is the network relay device (r_i), and the destination nodes in vectors are the candidate devices.

An example of closeness score is shown in Fig. 3, where the projection vector $pv_{m_i}<A,B>$ denotes the projection of candidate vector $cv_{m_i}<A,B>$ on the reference vector $ov_{m_i}<A,F>$. Furthermore, the closeness score of network relay device *B* (denoted as $ct_{m_i}^{r_B}$) is the ratio of projection vector length $L_{pv_{m_i}<A,B>}$ and the reference vector length $L_{ov_{m_i}<A,F>}$. Similarly, the closeness score of network relay device *F* (denoted as $ct_{m_i}^{r_F}$) is obtained. As the closeness score $ct_{m_i}^{r_F}$ is larger than $ct_{m_i}^{r_B}$, network relay device *F* can

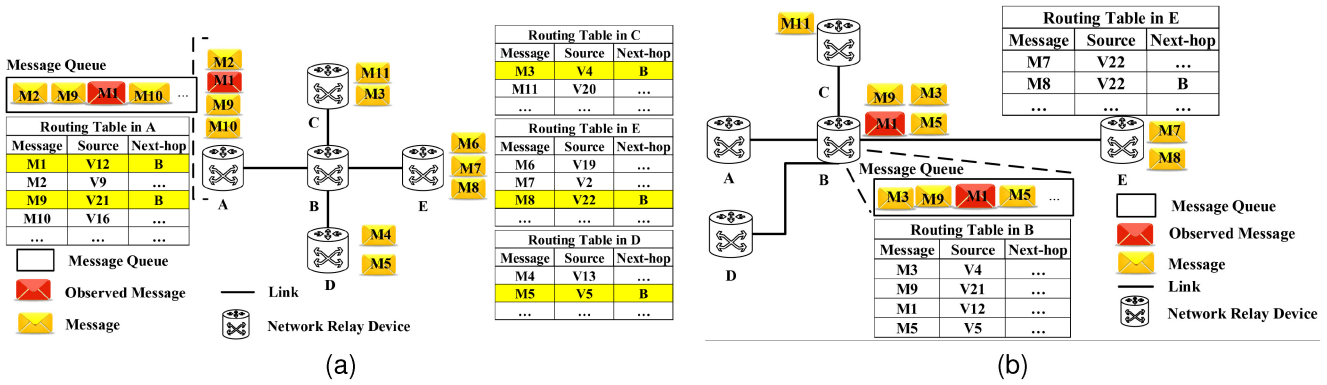

Fig. 4. An example of calculating message buffer time dt_m^r in period t. (a) Message m_1 is in the current network relay device A. (b) Message m_1 is in the network relay device B.

be considered more suitable than network relay device B in trajectory proximity.[4]

Definition 2 (Buffer score): It represents the ratio of available buffer size to the total buffer size in a network relay device, denoted by θ.

$$\theta_t = \frac{\kappa}{\kappa + \nu}, \tag{2}$$

where κ and ν are the size of network relay device's available and occupied buffer size in period t, respectively.

2) Temporal Domain Parameters:

Definition 3 (Message buffer time): It is the summation of buffer time of message m_i in period t and period[5] $t+1$, denoted as $dt_{m_i}^{r_i}$.

$$dt_{m_i}^{r_i} = dt_{m_i}^{r_i,t} + dt_{m_i}^{r_i,t+1}, \tag{3a}$$

$$dt_{m_i}^{r_i,t} = \sum_{i=1}^{N} \frac{S_{m_i}}{T_{m_i}}, \tag{3b}$$

where t and $t+1$ are consecutive time periods. $dt_{m_i}^{r_i,t}$ and $dt_{m_i}^{r_i,t+1}$ are buffer time of m_i in period t and $t+1$, respectively. Specifically, $dt_{m_i}^{r_i,t}$ denotes the buffer time for forwarding message m_i after sorting it in the current network relay device during period t. $dt_{m_i}^{r_i,t+1}$ denotes the buffer time for forwarding message m_i after sorting it in the candidate network relay device [6] in period $t+1$. However, it is more complex to calculate the value of $dt_{m_i}^{r_i,t+1}$ than $dt_{m_i}^{r_i,t}$. In detail, the value of $dt_{m_i}^{r_i,t}$ depends on the number of messages, transmission speed of links, and the order of m_i in the current network relay device. Whereas, the value of $dt_{m_i}^{r_i,t+1}$ is affected by more factors in period $t+1$ as following:

- The number of messages that neighbor network relay device r_j receives before message m_i arrives.
- The order of m_i in neighbor network relay device r_j's sorted forwarding list.
- The link transmission speed between network relay device r_i and r_j.

[4]The trajectory proximity indicates the relationship between the position of a network relay device and the reference trajectory of a message.

[5]The simulation time is divided into n slots. The period t represents the current slot (t-th slot), and the period $t+1$ is the next slot.

[6]The candidate network relay device is one of network relay devices filtered by Algorithm 3 mentioned later.

An example of calculating message buffer time dt_m^r is shown in Fig. 4. Each network relay device maintains a routing table with source of messages, next-hop device, and a message queue for storing the order. In period t, messages (m_2, m_9, m_{10}) and an observed message m_1 are sorted in the message queue of network relay device A (denoted by r_A). The message buffer time of observed message (denoted by $dt_{m_1}^{r_A,t}$) can be calculated according to Eq. (3b). Then, the proposed STALB will predict the other part buffer time $dt_{m_1}^{r_A,t+1}$. In Fig. 4b, the message queue of network relay device B (denoted by r_B) contains the messages m_1, m_3, m_5, and m_9. Attentively, although the message m_8 will be transmitted to r_B, it will not affect the buffer time of message m_1. Messages in queue will arrive before m_1 in r_B, and will be sorted by smallest TTL. Thus, $dt_{m_1}^{r_A,t+1}$ can be calculated as well as $dt_{m_1}^{r_A,t}$.

Definition 4 (Buffer prediction score): It represents the prediction value of buffer score in period $t+1$, denoted by $\overline{\theta_t}$.

$$\overline{\theta_t} = \theta_t + \rho(\theta_{t-1} - \theta_t), \tag{4}$$

where θ_t is the buffer score of network relay device in period t, and $\overline{\theta_t}$ is the prediction value of buffer score in period $t+1$. Besides, ρ denotes the inertia coefficient, controlling the increase/decrease status of message forwarding when forecasting. Attentively, the first $\overline{\theta_t}$ can be calculated only in the second period ($t=2$) because the value of t is eternally greater than 0. In detail, θ_{t-1} is the buffer time score in last period ($t-1$) while the current period is t. To ensure that $t-1>0$, the value of t is at least 2, which is the second period ($t=2$). According to Eq. (4), the predictive value of θ_t can be dynamically adjusted through real-time network traffic. In addition, the predicted value $\overline{\theta_t}$ will affect the composition of candidate network relay device set. Furthermore, there are two functions of predicted value: (*1*) to judge the network relay device's status. (*2*) to adjust the inertia coefficient according to Eq. (9).

3) Preferred Index:

Definition 5 (Preferred index): It indicates the comprehensive level of the network relay device r_i for m_i, denoted by $rs_{m_i}^{r_i}$.

$$rs_{m_i}^{r_i} = \alpha \cdot ct_{m_i}^{r_i} + \frac{\beta}{dt_{m_i}^{r_i}} + \gamma \cdot \theta_t, \tag{5}$$

TABLE II
NOTATIONS AND SYMBOLS

Notations	Explanation
$\mathcal{M}$	Messages set
$\mathcal{R}$	Network relay devices set
r	the network relay device
$\mathcal{C}$	Vehicles set
$\mathcal{S}_m$	Message size
RT_m	Reference trajectory path of m_i
CR	Candidate set of network relay devices
MW	Moving windows for generating the reference vector
pv	The projection vector of m_i
cv	The candidate vector of m_i
ov	The reference vector of m_i
L	The length of vector
ct_m^r	The closeness score of message in network relay device r
κ	The size of unused cache space of network relay device
ν	The size of used cache space of network relay device r
dt_m^r	The buffer time score for the candidate network relay device r
$dt_m^{r,t}$	The time that message m stays in the current network relay device
$dt_m^{r,t+1}$	The time that message m stays in the candidate network relay device
θ	The buffer score of network relay device
$\overline{\theta}$	Prediction value of buffer score
$\overline{\overline{\theta}}$	Adjusted value of buffer score
Γ	Mean square error of network relay device
ρ	Inertia coefficient for controlling increase/decrease of messages
$\overline{\rho}$	Adjusted coefficient of inertia
δ_{ov}^{cv}	The angle between the candidate vector and reference vector
$\Upsilon_{i,j}$	The ratio of total number of network relay devices (i) to the number of vehicles (j)

where α, β, γ are 0.2, 0.6, 0.2, respectively. Precisely, the sorting strategy greatly affects the latency of messages in the actual transmission process. So the message buffer time $dt_{m_i}^{r_i}$ is set to 0.6. Besides, the closeness score $ct_{m_i}^{r_i}$ and buffer score θ_t are considered equally important. Then, their values are set to 0.2. The preferred index $rs_{m_i}^{r_i}$ is a quantitative value of the candidate network relay device's status. So, a higher preferred index represents a better status of r_i will be. Therefore, the network relay device with the highest $rs_{m_i}^{r_i}$ will be selected for the message forwarding.

4) Inertia Coefficient Adjustment: To predict the message congestion degree accurately in period $t+1$, the value of the inertia coefficient should be adjusted.

Definition 6 (Inertia coefficient): It represents the increase/decrease rate of total number of message in a short period, denoted by ρ. The parameter can predict the buffer score $\overline{\theta_t}$.

In the initial stage of the system, ρ is set to 0.5. Furthermore, ρ is defined to reflect network traffic fluctuation, which enables the network relay device to predict the buffer score more accurately. Based on this, the adjusted buffer score can be expressed by $\overline{\overline{\theta_t}}$.

$$\overline{\overline{\theta_t}} = \theta_t + \overline{\rho}(\theta_{t-1} - \theta_t), \tag{6}$$

where $\overline{\rho}$ is the adjusted value of ρ, which reflects the fluctuation of network traffic. Besides, $\overline{\theta_t}$ is the predicted value of θ_t in period $t+1$. To dynamically adjust the value of inertia coefficient ρ, the system needs to find the difference between the actual buffer score and the predicted value $\overline{\theta_t}$. Since the adjusted buffer score should be close to the actual buffer score, we assume that the adjusted buffer score $\overline{\overline{\theta_t}}$ is equal to the actual buffer score θ_{t+1} in period $t+1$. After calculating Eq. (4) and Eq. (6), the relationship between $\overline{\theta_t}$ and ρ is established. The system needs an equation to describe the degree of network fluctuations for adjusting ρ. Generally, mean squared error is the average squared difference between the predicted and actual values in statistics, representing data fluctuations. Therefore, mean square error equation is appropriately applied here to characterize the network fluctuations.

Definition 7 (Mean square error): It represents the fluctuation degree of the buffer occupancy of network relay device, denoted by Γ.

$$\Gamma = \overline{\overline{\theta_t}} - \overline{\theta_t} = \frac{1}{n-2}\sum_{t=3}^{n}\left(\overline{\theta_{t-1}} - \theta_t\right)^2, \tag{7}$$

$$\overline{\theta_{t-1}} = \theta_{t-1} + \rho(\theta_{t-2} + \theta_{t-1}), \tag{8}$$

where n is the upper limit of periods. Attentively, the first $\overline{\theta_t}$ can be calculated only in the second period ($t=2$) (the reason for $t=2$ can refer to Eq. (4)). Similarly, to get the Eq. (8), we substitute $t-1$ into t in Eq. (4). $\overline{\theta_{t-1}}$ can be obtained only when $t-2>0$, which is meaning that t is at least equal to 3 ($t \geq 3$). Therefore, we can calculate the mean square error from the third period ($t=3$). Meanwhile, to ensure that the denominator of Eq. (7) is greater than 0, we change $n-3$ to $n-2$.

In summary, the inertia coefficient adjustment Eq. (9) is obtained with the following form by incorporating Eq. (4), Eq. (6) with Eq. (7):

$$\overline{\rho} = \frac{1}{(n-2)(\theta_{t-1} - \theta_t)}\sum_{t=3}^{n}\left(\overline{\theta_{t-1}} - \theta_t\right)^2 + \rho, \tag{9}$$

where $\overline{\rho}$ is the adjusted value of ρ. The Eq. (9) is an adaptive adjustment equation, which is used to adjust the inertia coefficient after being affected by network fluctuations.

D. Problem Formulation

Firstly, the destination of message m_i is defined upon generation. According to the destination of m_i, r_i finds a reference trajectory path $RT_{m_i} = \{r_i, r_j, \ldots, r_k\}$ by Dijkstra algorithm[7] in the core network [16]. Furthermore, the next-hop network relay device is selected for m_i based on its local status (e.g., cache score θ_t in Eq. (2)), link status (e.g., message buffer time $dt_{m_i}^{r_i}$ in Eq. (3a)), and the closeness score $ct_{m_i}^{r_i}$. The main purpose of proposed STALB is to find out the maximum overall score of candidate network relay device according to Eq. (5) while satisfying some specific constraints. Therefore, the objective function of preferred index $rs_{m_i}^{r_i}$ is formulated as follows:

$$\max_{r_i, m_i} \ \alpha \cdot ct_{m_i}^{r_i} + \frac{\beta}{dt_{m_i}^{r_i}} + \gamma \cdot \theta_t$$

[7]Dijkstra algorithm is an algorithm for finding the shortest paths between nodes in a graph.

Algorithm 1: V2I Multi-Copy Communication

Input: Neighbor Set *RS*, Messages Set $\mathcal{M}$
Output: Optimal Order Messages Set *priorityMC*

for *neighbor node* r_j in *RS* **do**
  if r_j *is in congestion* **then**
    continue;
  end
  for *message* m_i in $\mathcal{M}$ **do**
    if r_j *has received this message* m_i **then**
      continue;
    end
    if r_j *is transmitting message* **then**
      continue;
    end
    nrofCopies ← Get the number of copies of messages m_i;
    if $nrofCopies > 1$ **then**
      Add message *m* to the send queue *SQ*;
    end
  end
end
priorityMC ← Sort *SQ* based on the messages sorting policy;
Return *priorityMC*;

$$s.t. \begin{cases} C1: & 0 \le ct_{m_i}^{r_i} \le 1 \\ C2: & 0 < dt_{m_i}^{r_i} \\ C3: & 0 \le \theta_t \le 1 \\ C4: & 0 \le \alpha, \beta, \gamma \le 1 \\ C5: & \alpha + \beta + \gamma = 1, \end{cases} \tag{10}$$

the meaning of the above constraints is as follows:

- *C1* ensures that the closeness score is reasonable.
- *C2* guarantees that the denominator of the objective function is not 0.
- *C3* is the limitation of buffer size of network relay device.
- *C4* and *C5* jointly restrict the value of $ct_{m_i}^{r_i}$, $dt_{m_i}^{r_i}$, and θ_t to a limited range.

Then, r_i sorts and transmits all messages according to smallest TTL. Next, the occupied buffer space of the network relay device and the sequence of messages will change with the message forwarding. Therefore, the local status of the network relay device change dynamically, and needs to be updated along. Finally, once the message m_i is successfully transmitted in network relay device r_i, r_i will update its local status, for example, mean square error Γ, inertia coefficient ρ and buffer score θ, etc.

IV. Spatio-Temporal Domain Empowered Messages Transmission

This section elaborates on the main algorithms of STALB in detail, including multi-copy communication, trajectory construction and reconstruction, filtering and forwarding, and periodic network status update.

Algorithm 2: Trajectory Path Construction

Input: Current Message Set $\mathcal{M}_t$, Last Message Set $\mathcal{M}_{t-1}$, Constructed Set $\mathcal{CS}$

Remove the common messages of $\mathcal{M}_t$ and $\mathcal{M}_{t-1}$;
for *message* m_i in $\mathcal{M}_t$ **do**
  if m_i in $\mathcal{CS}$ **then**
    Remove m_i from $\mathcal{M}_t$ and $\mathcal{M}_{t-1}$;
  end
end
for *message* m_i in $\mathcal{M}_t$ **do**
  Get the destination r_t and source r_s from message header;
  Path *p* ← new Path();
  mp ←Get the connection graph of network relay devices;
  p ←Find the shortest path according to Dijkstra algorithm (*mp*);
  Sets *p* to the message m_i header;
end

A. Multi-Copy Communication

Generally, the number of message copies is determined by the network density [38]. The multi-copy communication scheme can control message copies' numbers in each node and formulate specific forwarding rules. On the other hand, the advances of V2I access technologies enable vehicles to establish connections and forward messages with RSU when in proximity. STALB follows the Source SaW [13] for V2I communication. The network relay device can find the next-hop device according to Algorithm 4. Each message will be forwarded to its destination through the fixed link.

B. Trajectory Path Constructing

The RSU receiving messages from vehicles will determine the reference trajectory path according to the decision of the SDN virtual manager. Specifically, the SDN virtual manager receives the data of local status (e.g., buffer space, link speed) from RSUs, and calculates reference trajectory paths for messages.

The processes of trajectory path constructing are shown in Algorithm 2. Firstly, the delivered message set $\mathcal{DM}$ has been obtained, and stores the message of which has been delivered. Then, STALB removes the messages with the same message IDs (a unique number for each message) in the current message set $\mathcal{M}_t$ and last message set $\mathcal{M}_{t-1}$, since the messages in $\mathcal{M}_{t-1}$ had constructed the reference trajectory path. If the remaining message m_i in $\mathcal{M}_t$ is also in the message constructed set $\mathcal{CS}$, a set stores messages that have been constructed the reference trajectory path, and the message m_i is removed. Next, the source and destination are obtained from each message, and Dijkstra algorithm [39] are employed for calculating a reference trajectory path *p*. Finally, set *p* to the header of message m_i.

Algorithm 3: Candidate Set Filtering

Input: Reference Trajectory Path RT_{m_i}, Neighbour Set *NR*
Output: Candidate Set *CR*

Moving Window *MW* $\leftarrow$ Get the three sequential network relay devices closest to the current network relay device in RT_{m_i};
$ov_{m_i} \leftarrow$ *MW*;
for *network relay device* r_i in *NR* **do**
  cv $\leftarrow$ The current network relay device and r_i are formed;
  if $\delta_{ov}^{cv} \leq \frac{\pi}{2}$ **then**
    Add r_i to *CR*;
  end
end
Return *CR*;

Algorithm 4: Message Forwarding

Input: Candidate Set *CR*, Message m_i, Reference Vector ov_{m_i}, Message Set $\mathcal{M}$
Output: Network Relay Devices Score Set *RS*

$CT_{m_i} \leftarrow$ Create a set for closeness score;
$DT_{m_i} \leftarrow$ Create a set for message buffer time;
$FBS_{m_i} \leftarrow$ Create a set for buffer score;
for r_i in *CR* **do**
  $ct_{m_i}^{r_i} \leftarrow$ Calculate the closeness score of according to Eq. (1);
  Add $ct_{m_i}^{r_i}$ to CT_{m_i};
end
Sort $\mathcal{M}$ by smallest TTL;
$dt_{m_i}^{r_i,t} \leftarrow$ Calculate the buffer time that m_i in $\mathcal{M}$;
for r_i in *CR* **do**
  NR $\leftarrow$ The neighbor network relay device set of r_i;
  AM $\leftarrow$ Create a set to store the messages from the neighbor network relay devices;
  for r_k in *NR* **do**
    Sort the messages $\mathcal{M}$ of r_k by smallest TTL;
    AM $\leftarrow$ Add the messages that r_k can transmit to r_i before message m_i arrives at r_i;
  end
  Sort *AM* by smallest TTL;
  $dt_{m_i}^{r_i,t+1} \leftarrow$ Calculate the message buffer time of m_i in *AM*;
  $dt_{m_i}^{r_i} = dt_{m_i}^{r_i,t} + dt_{m_i}^{r_i,t+1}$;
  $\theta_t \leftarrow$ Calculate buffer score according to Eq. (2).
  $DT_{m_i} \leftarrow$ Add $dt_{m_i}^{r_i}$;
  $FBS_{m_i} \leftarrow$ Add θ_t;
end
for r_i in *CR* **do**
  $RS_{r_i} \leftarrow \alpha * CT_{r_i} + \beta * DT_{r_i} + \gamma * FBS_{r_i}$;
end
Return *RS*;

C. Filtering and Forwarding

1) Candidate Set Filtering: It filters network relay devices which are inadequate for message forwarding.

Firstly, STALB finds the nearest network relay device r_i in the reference trajectory path (RT_m) of message m_i. Then, the subsequent devices (r_{i+1}, r_{i+2}) beyond r_i in RT_m can be obtained. Next, the devices (r_i, r_{i+1} and r_{i+2}) form the moving window *MW* (a container for generating the reference vector). The current network relay device r_c and r_{i+2} are selected as the endpoints of the reference vector $ov =< r_c, r_{i+2} >$. After that, the neighbor network relay device r_k and r_c constitute the candidate vector $cv =< r_c, r_k >$, where k is the index number in the neighbor network relay devices set *NR*. The angle between *cv* and *ov* is represented as δ_{ov}^{cv}. If $\delta_{ov}^{cv} \leq \frac{\pi}{2}$, r_k will be put into the candidate set *CR* of network relay devices. More details of filtering the candidate set *CR* are summarized in Algorithm 3.

2) Message Forwarding: STALB scores the local status of network relay devices from the filtered candidate set, and judges the potential capabilities for transmitting messages. Notably, the score (preferred index $rs_{m_i}^{r_i}$) consists of closeness score $ct_{m_i}^{r_i}$, message buffer time $dt_{m_i}^{r_i}$, and buffer score θ_t. In order to store calculation results of $ct_{m_i}^{r_i}$, $dt_{m_i}^{r_i}$, and θ_t, some sets (CT_{m_i}, DT_{m_i}, and FBS_{m_i}) are created, respectively. The calculation processes for the corresponding components are exhibited in detail as follows:

1) Closeness score $ct_{m_i}^{r_i}$ can be calculated according to Eq. (1), which is the projection length of candidate vector $cv =< r_c, r_k >$ on reference vector $ov =< r_c, r_{i+2} >$.
2) Message buffer time $dt_{m_i}^{r_i}$ can be divided into two different parts, denoted as $dt_{m_i}^{r_i,t}$ and $dt_{m_i}^{r_i,t+1}$.
   (i) To calculate $dt_{m_i}^{r_i,t}$ in network relay device r_c, messages set $\mathcal{M}$ is obtained to sort by smallest TTL. Then, according to the link speed, message size, and position of message m_i in sorted queue, $dt_{m_i}^{r_i,t}$ can be calculated.
   (ii) The key point of calculating $dt_{m_i}^{r_i,t+1}$ is to obtain message set $\mathcal{M}_{r_k}$ of neighbor device r_k in period $t+1$. So, STALB obtains the message of reaching r_k, and filters out message of which arrival time is greater than $dt_{m_i}^{r_i,t}$. This is because the message before m_i reaches r_k will affect the sort position of m_i. Furthermore, the messages that arrive r_k before m_i in period $t+1$ are stored in a list. Then, STALB sorts the list by smallest TTL, and calculates $dt_{m_i}^{r_i,t+1}$ of m_i until m_i is ready to transmit. After that, $dt_{m_i}^{r_i,t}$ and $dt_{m_i}^{r_i,t+1}$ are added up to obtain $dt_{m_i}^{r_i}$.
3) The buffer score θ_t can be calculated according to Eq. (2).

Finally, the weighting summation of the above three parameters ($ct_{m_i}^{r_i}$, $dt_{m_i}^{r_i}$, and θ_t) is preferred index $rs_{m_i}^{r_i}$.

D. Local Network Status Updating

The local status reflects the forwarding performance of the network relay device. It consists of congestion status, buffer score, buffer prediction score, and inertia coefficient. If the local status is inaccurate due to delayed routing information

Algorithm 5: Local Network Status Updating

Input: Update Flag $\mathcal{F}$, The Local Network Status ID t, Congestion Status η, Prediction Congestion Status $\overline{\eta}$, Buffeer Score θ_t, Status Threshold ω, Messages $\mathcal{M}$

```
if F = 1 then
    if t > 2 then
        θ̄_t ← Predict the buffer score according to Eq. (4);
        if θ̄_t > ω then
            η̄ ← 1;
        else
            η̄ ← 0;
        end
        Γ ← Calculate the mean square error according to Eq. (7);
        M ← The latest messages;
    else
        η, η̄ ← False
    end
    Update t, M, θ_t, θ̄_t, η, η̄, Γ;
end
```

exchange, the forwarding latency and hop count will increase correspondingly. Furthermore, newly received messages can change the local status of network relay device. Therefore, updating the local status plays an important role in reducing message delivery latency, buffering time, and network load ratio.

As shown in Algorithm 5, if the value of update flag $\mathcal{F}$ is equal to 1, the local status will be updated after finishing the message forwarding. In the first two local statuses (t<2), the local status and the predicted local status are both set as 0. Next, because the size of initial messages is small and the network load is low, the network relay device can forward messages quickly. So, the congestion status will not occur in the first two statuses. Whereas, if $t > 2$, the buffer prediction score $\overline{\theta_t}$ will be predicted according to Eq. (4). Then, the prediction status of congestion $\overline{\eta}$ is defined according to the status threshold ω. If $\overline{\theta_t} > \omega$, $\overline{\eta}$ will set to be 1. Otherwise, $\overline{\eta}$ will set to be 0. Next, the mean square error gets updated according to Eq. (7). Finally, STALB updates the local status of network relay devices (e.g., status ID t, messages $\mathcal{M}$, buffer score θ_t, prediction buffer score θ_t, congestion status η, mean square error Γ) after obtaining the current messages set $\mathcal{M}$.

E. Path Reconstruction

When message forwarding suffers from local maximum problem (in Fig. 3, refers to transmission from *A* to *E*, then *E* to *A* due to mutually selecting as the next-hop device), a path reconstruction mechanism is presented to compensate for this potential disadvantage. On the other hand, if messages are forwarded to the suboptimal network relay devices, the forwarding latency and hop count will increase correspondingly. This is because the suboptimal network relay devices may forward messages to the network relay device that deviates from the reference trajectory path. Therefore, the path reconstruction mechanism employs the shortest path to transmit messages in next three hops, triggered in two ways: (i) the set of candidate network relay devices is empty. (ii) the current network relay device is located very close to the destination (the moving window *MW* includes the destination). Whenever the above two triggers occur, the path reconstruction mechanism generates a shortest path as a new reference trajectory path. Then, the message is transmitted to three nodes according to the reference trajectory path.

TABLE III
SIMULATION PARAMETERS

Parameters	Values
Simulation area	$4500 \times 3400m^2$
Simulation time	$18000s$
Number of repetition for each run	10
Number of nodes	100
Velocity of vehicles	$18 \sim 54kmph$
Transmission in vehicles	$500k$
Transmission in core network	$1M$
Transmission range	$200m$
Packet size	$1MB$
Number of packets	$3600 \sim 18000$

V. PERFORMANCE EVALUATION

This section first describes the simulator for simulations and configuration. Then, extensive simulations are conducted to illustrate the performance of STALB.

A. Simulation Setup

The evaluations are based on Opportunistic Network Environment (ONE). Realistic road data from Helsinki city with a $4500 \times 3400m^2$ area serves as the scenario for our simulation experiment, similar to [15], [16]. This public dataset has been collected in ONE. In real urban road data, street density is diverse among different regions, and the information density transmitted by vehicles running on the street is also different. We consider the simulation time which is set with 18000*s*. To be in a sparse network environment with benchmark algorithms (BSaW [13], TDOR [16]) mentioned later, 80 network relay devices and 20 vehicles exist in this system. Furthermore, the moving speed of vehicles is randomly chosen from $[18 \sim 54]km/h$. Following the configuration of network relay device in [40], and [15] for opportunistic routing in sparse networks, the core network communication technique is set with 200*m* transmission range and 1 *Mbit/s* bandwidth. Similar to [16], the vehicle communication technique is set with 200*m* transmission range and 500 *kbit/s* bandwidth, considering as the midding power WiFi technology. The buffer size of all nodes is 1G, including vehicles and network relay devices.

Messages are randomly generated from the vehicles, and the destination of messages is randomly set to any network relay device. This article only considers the transmission capacity of the core network, so the vehicle only communicates with the network relay devices. Messages are randomly generated

at all vehicles in different network scenarios (the message generation interval is between 1s and 5s, and is an integer), with 60 minutes TTL and 1M size. There are 10 copies of each message, which is calculated by 10% of the total number of network relay devices and vehicles (100), according to [38].

For performance comparison, the following DTN routing protocols are evaluated:

- *BSaW [13]:* Each node transfers half the number of message copies to another node.
- *TDOR [16]:* An opportunistic routing protocol based on trajectory-driven policy. The vehicle carrying the message will choose the vehicle whose trajectory is closer to the destination as the relay forwarding node.
- *TBHGR [15]:* This scheme also considers the copies of messages, which limits the network load.

The main results of the 10 runs show performance with the following parameters:

- *Delivery Ratio:* It is the ratio of messages successfully delivered to the total number of messages generated, denoted as ψ.

$$\psi = \frac{|M_d|}{|M_g|}, \tag{11}$$

  where $|M_d|$ represents the number of messages delivered to the destinations. $|M_g|$ is the number of generated messages.
- *Overhead Ratio:* It is the ratio between the number of relayed messages (excluding the delivered messages) and the number of delivered messages, denoted as $\mathcal{O}$.

$$\mathcal{O} = \frac{|M_r| - |M_d|}{|M_d|}, \tag{12}$$

  where $|M_r|$ is the number of relayed messages.
- *Average Delivery Latency:* The average time it takes for a message to be successfully forwarded from source to destination, denoted as $T_{d,avg}$.

$$T_{d,avg} = \frac{1}{|M_g|} \sum_{i=1}^{|M_g|} \left(t_{d,i} - t_{s,i}\right), \tag{13}$$

  where $t_{d,i}$ represents the time that a message is received by its destination. $t_{s,i}$ is the generated time.
- *Average Buffer Time:* The average time a message is held in a network relay device after it has been received, denoted as $T_{c,avg}$.

$$T_{c,avg} = \frac{1}{M_r} \sum_{i=1}^{|M_r|} \left(t_{t,i} - t_{r,i}\right), \tag{14}$$

  where $t_{t,i} - t_{r,i}$ represents the time a message is cached in a network relay device.

B. The Impact of Total Number of Messages

In this section, we consider the impact of different message densities on the efficiency of STALB and benchmark methods (TDOR, BSaW, and TBHGR). The message density can be set by different message generation interval. We change the number of messages from 3,600 to 18,000. In other words, the generating interval of message is changed from 5s to 1s in this system while the simulation time is fixed at 18,000s. Then, the results are displayed in Fig. 5.

1) Delivery Ratio: Fig. 5a shows the delivery ratio versus the number of messages. It can be observed that for three message forwarding methods (STALB, BSaW, and TDOR), the delivery ratio of each method maintains stability with the increase of the number of messages. Obviously, the proposed STALB was the upper limit of all benchmark methods, which was approximately the optimal solution. From the perspective of trend changes, different from the above methods, the delivery ratio of TBHGR had obviously downtrend. In the case of dense network scenario, i.e., 18,000 messages, the delivery ratio of TBHGR was less than 86%.

The above variance mainly affected by the different strategies of message forwarding. The geographical information-based methods for Points-Of-Interest (POI), TBHGR and TDOR, performed poorly in general network scenarios that the POI of each location is the same value. Additionally, the destinations of messages are scattered with high probability on the digital map. On the contrary, the destination of a vehicle is unique. In the worst case, the vehicle does not encounter a suitable relay vehicle to deliver messages. So, some messages cannot be delivered to their destination within a limited time. Different from the other benchmark methods, a trajectory-based method, STALB, prohibits message forwarding in the opposite direction of destinations. In detail, to guarantee the right forwarding direction, STALB filters out the network relay devices with a poorly direction of message forwarding —— the angle between the candidate vector and the reference vector is within the scope of $(\frac{\pi}{2}, \pi]$. Moreover, when all the candidate network relay devices cannot satisfy the above conditions, the path reconstruction mechanism generates a newly shortest path as the reference trajectory path for each message to its destination. Therefore, STALB controls the direction of message forwarding and ensures messages can be delivered to their destinations.

2) Overhead Ratio: It is seen from Fig. 5b that the experiment compared the overhead ratio of different message forwarding methods in terms of various number of messages. Both STALB and BSaW maintained a very low overhead that the mean values were 7.72 and 12.92, respectively. However, although the overhead ratio of TDOR and TBHGR both shown a downward trend with the increase of total number of messages, their overhead ratio were permanently higher than STALB. Especially, when the number of messages was 18,000, the overhead ratio of TDOR and TBHGR was even higher than STALB by 93.57% and 156.96%, respectively.

In STALB, the number of message copies can be controlled by message forwarding strategy, which naturally causes limited number of message copies in the core network. Furthermore, the number of message copies increases linearly in the core network because each vehicle merely forwards one message copy at a slot. Thanks to the relatively constant number of connections with the network relay devices, the slope of linear increase in the number of message copies keeps stable. Moreover, the total number of message hops affects the overhead ratio. The reference trajectory maintains the direction of

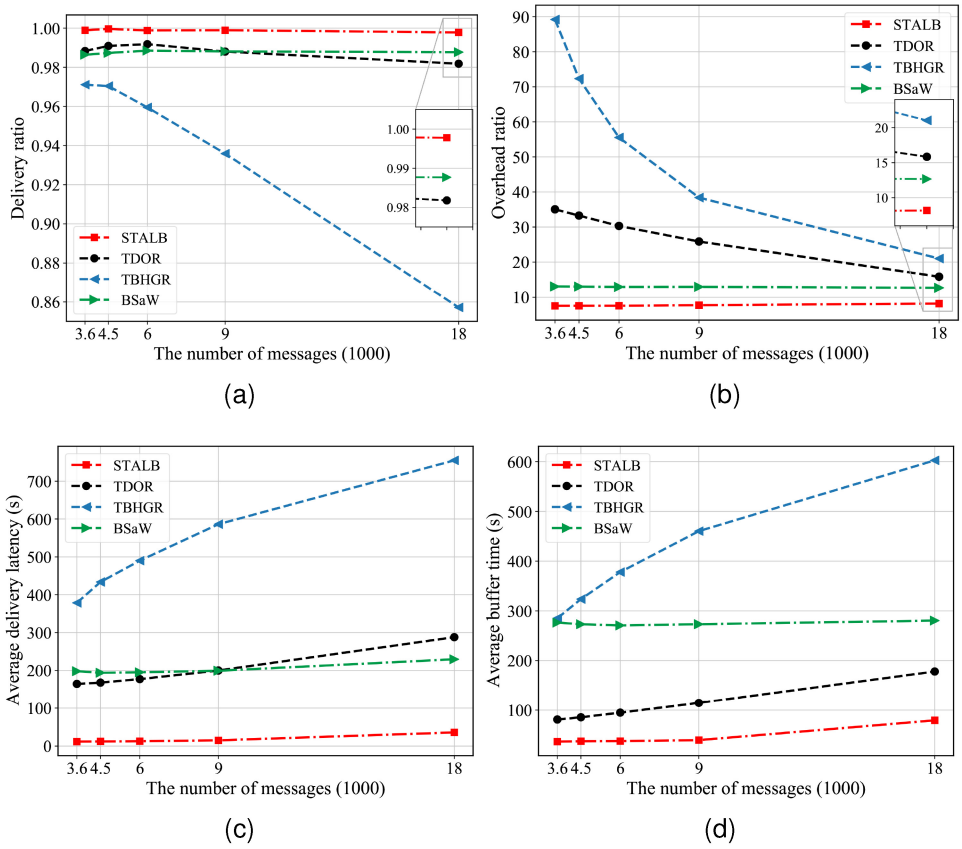

Fig. 5. (a) Delivery ratio versus the number of messages. (b) Overhead ratio versus the number of messages. (c) Average delivery latency versus the number of messages. (d) Average buffer time versus the number of messages.

message forwarding, reducing the number of hops for each message copies. Based on the above reasons, STALB shows the bottom line of overhead ratio. Although TDOR, TBHGR, and BSaW can control the number of message copies as well as STALB, vehicles equipped with these protocols forward half the number of message copies each time. The number of message copies increases exponentially in the VANETs. When the number of messages is low, i.e., 3,600 messages, in TBHGR, TDOR, and BSaW, vehicles can promptly forward messages to each other at the beginning of simulation, causing the number of message copies to soaring sharply (exponential increase). Nevertheless, in dense network scenarios, i.e., 18,000 messages, numerous messages are waiting for being forwarded because of the limited bandwidth (500k) and vehicle mobility (unstable connections). The increasing trend of the number of message copies is slower than that in the sparse network scenarios, i.e., 3600 messages. Therefore, the overhead ratio in TBHGR and TDOR shows a downward trend as the number of messages increases. Different from TBHGR and TDOR, considering the same value of POI, BSaW is an optimal solution for forwarding message randomly in VANETs. The overhead ratio in BSaW is much lower than that in TBHGR and TDOR.

3) Average Delivery Latency: As illustrated in Fig. 5c, it demonstrated that the average delivery latency of four methods all increase with the number of messages increasing. Attentively, the increase of the average delivery latency in STALB and BSaW was not obvious, while that in TDOR and TBHGR had increased significantly. The average delivery latency of STALB and BSaW stabilized around 12s and 200s, respectively. Specially, the average delivery latency of STALB is at least 87.49%, 84.31%, and 95.23% lower than that TDOR, BSaW, and TBHGR.

Obviously, as the number of messages increases, more copies of messages exist in the network, which greatly increases the buffer time of the message. As expected, the trajectory-based method, STALB, considers the closeness of reference trajectory for controlling the direction of message forwarding, which can optimize the hop-counts of messages. To predictably votes the best next-hop device (a network relay device), the local status are considered from spatio-temporal domain. Therefore, STALB showed a lower average delivery latency. Unfortunately, the other benchmark methods cannot completely guarantee the direction of message forwarding because of the vehicle mobility. In detail, the vehicle may move in the opposite direction of the message destination, which causes the efficiency of BSaW to degenerate to DD [10]. To address the efficiency degradation, the geographical location and relay vehicle collection filtering are considered by TDOR and TBHGR. Although the candidate vehicles can be controlled, vehicles may carry the messages and move to suboptimal direction within a short time (limited slots). In addition, the list of delivered messages is updated slowly between vehicles, causing extra delivery time —— the waiting time for updating that list has been additionally increased. As a result, the average delivery latency has increased in BSaW, TDOR, and TBHGR.

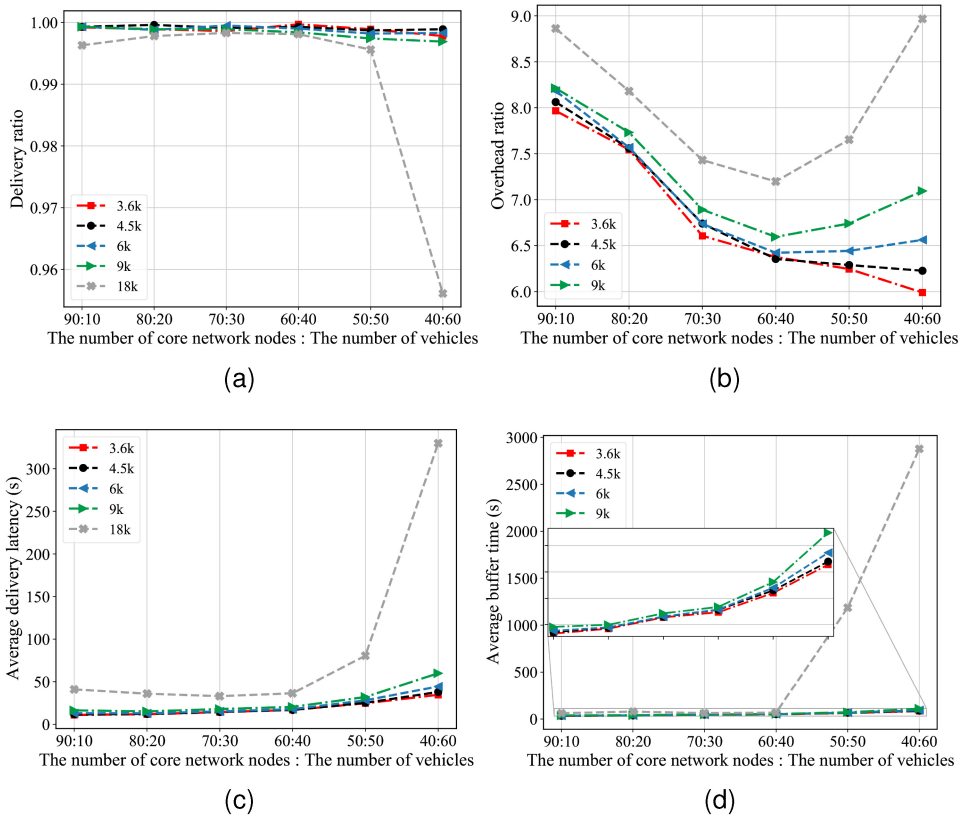

Fig. 6. (a) Delivery ratio versus the ratio of node's number. (b) Overhead ratio versus the ratio of node's number. (c) Average delivery latency versus the ratio of node's number. (d) Average buffer time versus the ratio of node's number.

4) Average Buffer Time: The line chart Fig. 5d has demonstrated the average buffer time variation with the increase of the number of messages. It can be found that STALB had the lowest average buffer time, i.e., 36.94*s*, while that in BSaW, TDOR, and TBHGR were 270.80*s*, 81.19*s*, and 285.27*s*, respectively. Compared with TDOR, BSaW, and TBHGR in the dense network scenario, i.e., 18,000*s*, the average buffer time of STALB is lower than 55.33%, 71.64%, and 86.80%, respectively. The mainly reason for lower average buffer time in STALB is the ability to quickly update the list of delivered messages in the core network. In detail, those messages stored in this list will be calculated as the buffer time (waiting time between receiving and deleting). The speed of updating the list of delivered messages determines the time horizon —— updating the list quickly avail reducing the buffer time. Whereas, the other benchmark methods update their list of delivered messages slowly because of the randomly mobility of vehicles. It is not facilitative to exchanging the list of delivered messages because some vehicles move to remote places, which causes a high average buffer time. The time for all vehicles to receive the list of delivered messages converges slowly.

C. The Impact of Ratio of Node's Numbers

In this section, the ratio of two types nodes (*i* network relay devices and *j* vehicles) was donated as $\Upsilon_{i,j}$. For example, there were 90 network relay devices and 10 vehicles, and its ratio was $\Upsilon_{90,10}$. Then, we changed $\Upsilon_{i,j}$ from $\Upsilon_{90,10}$ to $\Upsilon_{40,60}$, and the interval was 10, keeping the total number of network relay device and vehicles were 100. Without loss of generality, the other constraints didn't change.

1) Delivery Ratio: In Fig. 6a, STALB maintained a high delivery ratio in all network scenarios except the worst network scenario (18,000 messages, 40 network relay devices, and 60 vehicles). Although the delivery ratio in the network scenario ($\Upsilon_{40,60}$, 18,000 messages), 95.61%, was lower than the other network scenarios, it was acceptable that the delivery ratio closed to the other benchmarks (TDOR, BSaW, and TBHGR). In other network scenarios, the delivery ratio was at least 99.56%. The delivery ratio lightly decreased with the increase of total number of vehicles in most network scenarios. STALB based on trajectory path enables the messages to be forwarded along the trajectory path to their destinations. Moreover, a path reconstruction mechanism in STALB generates a newly shortest path as the reference trajectory path for each message to its destination. Then, the messages will follow the newly reference trajectory path. In the worst network scenario ($\Upsilon_{40,60}$, 18,000 messages), the limited number of links between network relay devices in the core network causes inevitable overhead. A small number of messages are buffered in the network relay device, waiting to be forwarded. Until the end of the simulation, there are still some messages that cannot be forward to their destinations. Fortunately, the number of messages waiting to be forwarded is small. Therefore, STALB can ensure the high delivery ratio.

2) Overhead Ratio: The Fig. 6b shows that the overhead ratio changed in different network scenarios (the number of

messages, the ratio of node's numbers $\Upsilon_{i,j}$). In a sparse network scenario, i.e., 3,600 messages, 4,500 messages, the overhead ratio decreased as the number of network relay devices decreased. However, in other network scenarios (6,000 messages, 9,000 messages, and 18,000 messages), when the ratio of node's numbers was $\Upsilon_{60,40}$, the overhead ratio was the minimum value (6.42, 6.60, and 7.20, respectively). In a non-sparse network scenario, the overhead ratio is the optimal solution when the ratio of network relay devices number to the number of vehicles is 1.5 ($\Upsilon_{60,40} = 6/4 = 1.5$). As the number of network relay devices was lower than 60 ($\Upsilon_{50,50}$, $\Upsilon_{40,60}$), the overhead ratio was increasing. The limited number of links between network relay devices constrain the efficiency of message forwarding, causing some messages in cache are waiting to be forwarded. As the number of network relay devices decreases, the number of message hops for a message to be forwarded to its destination also decreases, meaning that the number of relayed messages decreases. Therefore, the decreasing trend of overhead ratio is showed. In non-sparse network scenario, each message is forwarded to the network relay device at least once, so the number of relayed messages is linear increasing. Although the number of network relay devices is decreasing, its reduced value is limited. Overall, the overhead ratio is increasing in the above non-sparse network scenarios.

3) Average Delivery Latency: Fig. 6c demonstrates that the variation of average delivery latency in different $\Upsilon_{i,j}$. As the number of vehicles increased, the average delivery latency of STALB showed an upward trend. Especially, in dense network scenario ($\Upsilon_{40,60}$, 18,000 messages), the average delivery latency was the highest 329.91s. Fortunately, in most network scenarios, the average delivery delay of STALB was lower than 50s, and even the lowest value was 11.14s ($\Upsilon_{90,10}$, 3,600 messages). To reduce the average delivery latency, STALB controls the direction of message forwarding and provides a path reconstruction mechanism, which ensures limited number of message hops. The above method enables messages to be forwarded to their destinations as soon as possible. On the other hand, the large number of network relay devices means that the core network can carry more messages, supporting the fast message forwarding. As the number of network relay devices decreases, the links in the core network also decreases. Numerous messages are waiting to be forwarded in the core network. In the extreme network scenarios (e.g., no network relay device exists), STALB degenerates into DD (Directer Delivery) [10].

4) Average Buffer Time: In Fig. 6d, the average buffer time was increasing as $\Upsilon_{i,j}$ decreased. Although the average buffer time increased with the decrease of number of network relay devices, its value is relatively stable from a macro perspective (mostly below 110s). This is because the messages can be delivered to their destinations quickly. When messages are forwarded to their destinations, their Ids will be recorded to a table in the network relay device. The table is exchanged periodically between network relay devices, and it will be obtained by vehicles when the connections are established. Then, the network relay devices and vehicles will delete the messages which in the exchanged table. Therefore, the lower average delivery latency STALB has, the lower average buffer time STALB obtains. However, when $\Upsilon_{i,j} \leq 1$ and the number of messages was 18,000, the average buffer time surged more than 1000s. In details, numerous messages are trapped in cache because the reduction of links in the core network, waiting to be forwarded, which causes the longer message buffer time. Furthermore, in STALB, vehicles only communicate with the network relay devices, so if a message is waiting for forwarding in the core network for a long time, another way to deliver it successfully is for the vehicle to deliver the message directly to the destination. Unfortunately, it will take a lot of time because of vehicle mobility.

To sum up, it can be found that STALB reduces the average delivery latency and overhead while ensuring a high average delivery ratio, but also achieve lower the average buffer time. Therefore, we demonstrate that STALB is effective under different network scenarios.

VI. Conclusion

This paper explored the efficient message forwarding method to avoid network congestion in IoVs. We first proposed a Spatio-Temporal domain Autonomous Load Balancing (STALB) routing protocol for network balancing. Specifically, STALB was a trajectory-based method for controlling the direction of message forwarding. STALB can significantly reduce end-to-end latency and network load ratio, since it considers the local status of network relay devices from the spatio-temporal domain. Then, we presented a path reconstruction mechanism, which ensured that messages are forwarded to destinations within finite latency. Extensive simulations were conducted to evaluate the performance of the proposed STALB. Extensive simulation results show that STALB significantly outperforms other baseline methods (BSaW, TDOR, and TBHGR) regarding overhead ratio, average delivery latency, and average buffer time. Especially, the delivery rate of STALB can reach 99.9% under the sparse network scenario (4,500 messages), at least 0.7% higher than other baseline methods. Similarly, the average delivery delay of STALB is at least 84.31% lower than that of other baseline methods under the dense network scenario (18,000 messages).

References

[1] B. Ji *et al.*, "Survey on the Internet of vehicles: Network architectures and applications," *IEEE Commun. Stand. Mag.*, vol. 4, no. 1, pp. 34–41, Mar. 2020.

[2] K. Jiang, C. Sun, H. Zhou, X. Li, M. Dong, and V. C. M. Leung, "Intelligence-empowered mobile edge computing: Framework, issues, implementation, and outlook," *IEEE Netw.*, vol. 35, no. 5, pp. 74–82, Sep./Oct. 2021.

[3] L. Wei, J. Cui, H. Zhong, Y. Xu, and L. Liu, "Proven secure tree-based authenticated key agreement for securing V2V and V2I communications in VANETs," *IEEE Trans. Mobile Comput.*, vol. 21, no. 9, pp. 3280–3297, Sep. 2022.

[4] V. Vijayakumar and K. S. Joseph, "Adaptive load balancing schema for efficient data dissemination in vehicular ad-hoc network VANET," *Alexandria Eng. J.*, vol. 58, no. 4, pp. 1157–1166, Dec. 2019.

[5] J. Cui, L. Wei, J. Zhang, Y. Xu, and H. Zhong, "An efficient message-authentication scheme based on edge computing for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1621–1632, May 2019.

[6] W. Qi, B. Landfeldt, Q. Song, L. Guo, and A. Jamalipour, "Traffic differentiated clustering routing in DSRC and C-V2X hybrid vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7723–7734, Jul. 2020.
[7] A. A. Khan, M. Zafrullah, M. Hussain, and A. Ahmad, "Performance analysis of OSPF and hybrid networks," in *Proc. Int. Symp. Wireless Syst. Netw. (ISWSN)*, Nov. 2017, pp. 1–4.
[8] A. Alsarhan, Y. Kilani, A. Al-Dubai, A. Y. Zomaya, and A. Hussain, "Novel fuzzy and game theory based clustering and decision making for VANETs," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1568–1581, Feb. 2020.
[9] K. Liu, K. Xiao, P. Dai, V. C. Lee, S. Guo, and J. Cao, "Fog computing empowered data dissemination in software defined heterogeneous VANETs," *IEEE Trans. Mobile Comput.*, vol. 20, no. 11, pp. 3181–3193, Nov. 2021.
[10] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad-hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 477–486, Aug. 2002.
[11] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Dept. Comput. Sci., Duke Univ., Duhram, NC, USA, Rep. CS-2000-06, 2000.
[12] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 77–90, Feb. 2008.
[13] S. Thrasyvoulos, P. Konstantinos, and S. Cauligi, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM Workshop Delay-Tolerant Netw.*, Aug. 2005, pp. 252–259.
[14] S. Burleigh *et al.*, "Delay-tolerant networking: An approach to interplanetary Internet," *IEEE Commun. Mag.*, vol. 41, no. 6, pp. 128–136, Jun. 2003.
[15] Y. Cao, K. Wei, G. Min, J. Weng, X. Yang, and Z. Sun, "A geographic multicopy routing scheme for DTNs with heterogeneous mobility," *IEEE Syst. J.*, vol. 12, no. 1, pp. 790–801, Mar. 2018.
[16] Y. Cao *et al.*, "A trajectory-driven opportunistic routing protocol for VCPS," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 6, pp. 2628–2642, Dec. 2018.
[17] J. Zhang, Y.-P. Lin, M. Lin, P. Li, and S.-W. Zhou, "Curve-based greedy routing algorithm for sensor networks," in *Proc. Int. Conf. Netw. Mobile Comput.*, 2005, pp. 1125–1133.
[18] B. Nath and D. Niculescu, "Routing on a curve," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 155–160, Jan. 2003.
[19] H. Labiod, N. Ababneh, and M. G. de la Fuente, "An efficient scalable trajectory based forwarding scheme for VANETs," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Apr. 2010, pp. 600–606.
[20] A. Sharma and L. K. Awasthi, "AdPS: Adaptive priority scheduling for data services in heterogeneous vehicular networks," *Comput. Commun.*, vol. 159, no. 1, pp. 71–82, Jun. 2020.
[21] Z. Zhao, Z. Wang, G. Min, and Y. Cao, "Highly-efficient bulk data transfer for structured dissemination in wireless embedded network systems," *J. Syst. Archit.*, vol. 72, no. 4, pp. 19–28, Jan. 2017.
[22] K. Liu, V. C. S. Lee, J. K.-Y. Ng, J. Chen, and S. H. Son, "Temporal data dissemination in vehicular cyber–physical systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2419–2431, Dec. 2014.
[23] A. Akhtar *et al.*, "Low latency scalable point cloud communication in VANETs using V2I communication," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May. 2019, pp. 1–7.
[24] S.-Y. Pyun, H. Widiarti, Y.-J. Kwon, D.-H. Cho, and J.-W. Son, "TDMA-based channel access scheme for V2I communication system using smart antenna," in *Proc. IEEE Veh. Netw. Conf.*, Dec. 2010, pp. 209–214.
[25] X. Liu, Z. Xu, Y. Meng, W. Wang, J. Xie, and Y. Li, "An elastic-segment-based V2V/V2I cooperative strategy for throughput enhancement," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 5272–5283, May 2022.
[26] S. Guo, B.-J. Hu, and Q. Wen, "Joint resource allocation and power control for full-duplex V2I communication in high-density vehicular network," *IEEE Trans. Wireless Commun.*, early access, Jun. 3, 2022, doi: 10.1109/TWC.2022.3177199.
[27] P. Wu, L. Ding, Y. Wang, L. Li, H. Zheng, and J. Zhang, "Performance analysis for uplink transmission in user-centric ultra-dense V2I networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9342–9355, Sep. 2020.
[28] C. He, Q. Chen, C. Pan, X. Li, and F.-C. Zheng, "Resource allocation schemes based on coalition games for vehicular communications," *IEEE Commun. Lett.*, vol. 23, no. 12, pp. 2340–2343, Dec. 2019.
[29] Z. Latif *et al.*, "DOLPHIN: Dynamically optimized and load balanced path for inter-domain SDN communication," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 331–346, Mar. 2021.
[30] R. Trestian, K. Katrinis, and G.-M. Muntean, "OFLoad: An openflow-based dynamic load balancing strategy for datacenter networks," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 4, pp. 792–803, Dec. 2017.
[31] J. Cui, Q. Lu, H. Zhong, M. Tian, and L. Liu, "A load-balancing mechanism for distributed SDN control plane using response time," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 4, pp. 1197–1206, Dec. 2018.
[32] 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.
[33] H. Yao, X. Yuan, P. Zhang, J. Wang, C. Jiang, and M. Guizani, "Machine learning aided load balance routing scheme considering queue utilization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7987–7999, Aug. 2019.
[34] D. M. Casas-Velasco, O. M. C. Rendon, and N. L. S. da Fonseca, "Intelligent routing based on reinforcement learning for software-defined networking," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 870–881, Mar. 2021.
[35] N. Chaib, O. S. Oubbati, M. L. Bensaad, A. Lakas, P. Lorenz, and A. Jamalipour, "BRT: Bus-based routing technique in urban vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4550–4562, Nov. 2020.
[36] 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.
[37] R. Han, Q. Guan, F. R. Yu, J. Shi, and F. Ji, "Congestion and position aware dynamic routing for the Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 16082–16094, Dec. 2020.
[38] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *Proc. 5th Annu. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerComW)*, Mar. 2007, pp. 79–85.
[39] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959. [Online]. Available: https://doi.org/10.1007/BF01386390
[40] H. Zhou, K. Jiang, X. Liu, X. Li, and V. C. M. Leung, "Deep reinforcement learning for energy-efficient computation offloading in mobile edge computing," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1517–1530, Jan. 2022.

Yujie Song is currently pursuing the M.S. degree in cyberspace security with Wuhan University, China. His research interests include Internet of Vehicles and networking transmission.

Kai Jiang (Student Member, IEEE) received the M.S. degree from China Three Gorges University, China, in 2021. He is currently pursuing the Ph.D. degree in cyberspace security with Wuhan University, China. His research interests include mobile edge computing, deep reinforcement learning, and Internet of Vehicles.

Yue Cao (Senior Member, IEEE) received the Ph.D. degree from the Institute for Communication Systems formerly known as Centre for Communication Systems Research, University of Surrey, Guildford, U.K., in 2013. Further to his Ph.D. study, he had conducted a Research Fellow with the University of Surrey, and an Academic Faculty with Northumbria University, U.K., Lancaster University, U.K., and Beihang University, Beijing, China. He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University, Wuhan, China. His research interests include intelligent transport systems, including E-mobility, V2X, and edge computing.

Ruiting Zhou (Member, IEEE) received the Ph.D. degree from the Department of Computer Science, University of Calgary, Canada, in 2018. She has been an Associate Professor with the School of Cyber Science and Engineering, Wuhan University, since June 2018. Her research interests include cloud computing, machine learning and mobile network optimization. She has published research papers in top-tier computer science conferences and journals, including IEEE INFOCOM, ACM MobiHoc, ICDCS, IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, and IEEE TRANSACTIONS ON MOBILE COMPUTING. She also serves as a Reviewer for journals and international conferences, such as the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and IEEE/ACM IWQOS.

Chakkaphong Suthaputchakun received the B.Eng. degree in computer engineering from the King Mongkut's University of Technology Thonburi, Thailand, in 2002, the M.Sc. degree in electrical and computer engineering from the University of Massachusetts at Amherst, USA, in 2006, and the Ph.D. degree in electronic engineering from the University of Surrey, U.K., in 2014. Recently, he has been accepted for the SwiftV2X Project funded by Horizon 2020 Marie Skłodowska-Curie RISE from 2021 to 2025. He has also been granted by Hubei Provincial Science and Technology Department of China in Network Information Security Detection System Project. His research works have been published in more than 20 high-impact journals and conferences. He has experience in several international projects. He specializes in wireless sensor networks and the next generation wireless networks emphasizing on vehicle ad-hoc network, such as traffic light communication network.

Yuan Zhuang (Member, IEEE) received the bachelor's degree in information engineering from Southeast University, Nanjing, China, in 2008, the master's degree in microelectronics and solid-state electronics from Southeast University, Nanjing, in 2011, and the Ph.D. degree in geomatics engineering from the University of Calgary, Canada, in 2015. He is a Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, China. To date, he has co-authored over 70 academic papers and 11 patents and has received over 10 academic awards. His current research interests include multi-sensors integration, real-time location system, wireless positioning, Internet of Things, and machine learning for navigation applications. He is an Associate Editor of IEEE ACCESS, the Guest Editor of the IEEE INTERNET OF THINGS JOURNAL, *Satellite Navigaiton*, and *IEEE Access*, and a Reviewer of over 10 IEEE journals.