Hierarchical traffic light-aware routing via fuzzy reinforcement learning in software-defined vehicular networks

Mohammad Naderi (mohammad.naderi@srbiau.ac.ir)
Islamic Azad university
Khorshid Mahdaee
Islamic Azad university
Parisa Rahmani
Islamic Azad university

Research Article

Keywords: Vehicular ad hoc networks, Software-defined networks, Intersection-base routing, Traffic lights, Fuzzy logic, Reinforcement learning

Posted Date: September 1st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1985034/v1

Hierarchical traffic light-aware routing via fuzzy reinforcement learning in software-defined vehicular networks

Mohammad Naderi [a,1], Khorshid Mahdaee [a], Parisa Rahmani [b]

[a] Department of Computer Engineering, Science and Research Branch, Islamic Azad university, Tehran, Iran

[b] Department of Computer Engineering, Pardis Branch, Islamic Azad university, Pardis, Iran

Abstract

Lack of a fully vehicular topology view and restricted vehicles' movement to streets with the time-varying traffic light conditions have caused drastic gaps in the traditional vehicular routing protocols. A hierarchical traffic light-aware routing scheme called HIFS is proposed in this paper using fuzzy reinforcement learning and software-defined network (SDN) to fill these gaps. At the first level of our HIFS scheme, a utility-based intersections selection policy is presented using fuzzy logic that jointly considers delay estimation, curve distance, and predicted of moving vehicles towards intersections. Then, a fuzzy logic-based path selection policy is proposed to choose the paths with highest flexibility against the intermittent connectivity and increased traffic loads. Residual bandwidth, Euclidean distance, angular orientation, and congestion are considered inputs of the fuzzy logic system. Meanwhile, traffic light states and nodes' information are used to tune the output fuzzy membership functions via reinforcement learning algorithm. The efficiency of our scheme in controlling ambiguity and uncertainty of the vehicular environment is confirmed through simulations in various vehicle densities and different traffic lights duration. Simulation results show our HIFS scheme's superiority over the state-of-the-art methods in terms of delivery ratio, average delay, path length, and routing overhead.

Keywords: Vehicular ad hoc networks, Software-defined networks, Intersection-base routing, Traffic lights, Fuzzy logic, Reinforcement learning

1. Introduction

Vehicular networks (VNs) have emerged as a promising technology in the future of Intelligent Transportation Systems (ITS), intending to improve road safety and develop vehicular entertainment applications [1-2]. Road safety applications, including traffic violation warnings, lane change alerts, and pre-crash warnings, aim to make driving easier and reduce casualties. Entertainment-related information would also provide passengers or drivers with entertainment, such as internet access, video streaming or gaming [1-5]. By providing a variety of communications, such as vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), or vehicle-to-everything (V2X), these applications can be shared with the surroundings [2], [6]. Sharing applications among vehicles via an efficient and reliable routing method has become an important research area in vehicular networks. In recent decades, numerous vehicular routing protocols and strategies [5-10] have been proposed in VNs that can be classified into location-based and topology-based routing categories based on the route formation strategy [5]. Lately, position-based routing protocols have attracted much attention among researchers

[1]* Corresponding author

E-mail addresses: mohammad.naderi@srbiau.ac.ir (M. Naderi), mahdaee@gmail.com (Kh. Mahdaee), rahmani@pardisiau.ac.ir (P. Rahmani)

and communities due to their applicability in vehicular networks [5]. However, intermittent connectivity caused by high dynamics of the vehicular topology, bandwidth limitations, traffic congestion, and obstacles can lead to communication bottlenecks and compromise location-based routing protocols' efficiency. That is why intersection-based routing protocols have been widely proposed to reduce the effects of the above-discussed challenges in vehicular networks [9]. Among the state-of-the-art intersection-based routing schemes, full path traffic-aware, local-based intersection selection, and broadcasting control packets (CP) are the most commonly employed methods for real-time road evaluations and choosing intersections [8-9]. However, limited knowledge of vehicular topology and additional costs such as increased latency and routing overhead may lead to sub-optimal traditional intersection-based routing performance. Some of the mentioned challenges are likely to be reduced by integrating new technological paradigms such as software-defined networking into vehicular intersection-based routing. SDN is a promising technology that offers an efficient network management solution by separating the control plane from the data plane [2], [6], [11-12]. Combined software-defined networks into vehicular networks (SDVNs) can provide programmability and access to entire network information. Therefore, the routing decisions can be made based on the global information obtained from the vehicular environments and road network topology.

1.1. Motivation

Significant challenges of traditional vehicular routing, including local maxima and congestion caused by limited knowledge in routing decisions, can be reduced through global visibility provided by SDVNs. However, intermittent connectivity, interface heterogeneity, and increasing demand for scalability and reliability, besides road constraints such as time-varying traffic lights, are common barriers to optimal SDVNs routing performance. Therefore, making routing decisions in SDN-enabled vehicular networks due to road restrictions and application requirements such as end-to-end latency is still challenging. Various routing criteria to tackle these challenges have been presented in SDVNs using different techniques and computational methods [13-22]. Among these techniques, fuzzy logic schemes such as [16] can make routing decisions more efficient as a user-friendly method closer to human thinking by processing approximate data using non-numeric linguistic variables. However, the use of the traditional fuzzy type-1 due to the possibility of conflicting criteria, in addition to the uncertainty and ambiguity of the vehicular environment, may lead to sub-optimal routing decisions. Accordingly, a new paradigm should be developed to cope with vehicular environment uncertainties. Fuzzy logic with the ability of tuning fuzzy membership functions can deal with ambiguities resulting in more suitable performance [23-24]. Adaptivity of membership functions can handle linguistic and numerical uncertainties associated with traditional type-1 fuzzy [24]. Motivated by dealing with the uncertainties of vehicular environments and considering the traffic light's effect, this paper presented a fuzzy-based hierarchical routing scheme with tuning output membership functions based on SARSA reinforcement learning.

1.2. Main contributions

- A hierarchical routing scheme is proposed in SDN-enabled vehicular networks to make decisions efficiently. Initially, a fuzzy logic-based intersections selection policy is provided by jointly considering estimated latency, predicted of moving vehicles towards candidate intersections, and curve distance. Based on the fuzzy results, we

proposed a utility-based scheme to dynamically select a sequence of intersections, which can improve the data packet transmission efficiency.

- A utility-based relay node selection policy is proposed via fuzzy logic with flexibility against network topology changes and increased traffic loads. The HIFS scheme considers multiple fuzzy metrics, including residual bandwidth, angular orientation, Euclidean distance, and load capacity.
- The output fuzzy membership functions are tuned by SARSA reinforcement learning to handle the impacts of uncertainties arising from the dynamic vehicular topology and road restricted to the different traffic lights conditions. The efficiency of adaptively tuning fuzzy membership functions depending on macroscopic and microscopic vehicular environment aspects has been proven through simulation. Our HIFS scheme has improved network performance in terms of packet delivery ratio, average delay, path length, and routing overhead than the state-of-the-art routing schemes.

1.3. Organization

The rest of this paper is organized as follows. A summary over the state-of-the-art intersection-based routing protocols is laid out in Section 2. Section 3 presents the details of our routing scheme. In Section 4, the simulation settings and the results are discussed. Finally, concluding remarks and our future work are given in Section 5.

2. Related work

In this section, some recent efforts are surveyed in the state-of-the-art intersection-based routing in two categories: 1) traditional intersection-based protocols in vehicular networks; 2) integration of software-defined networks into the intersection-based routing in vehicular networks. Details of the surveyed papers in each section are stated below.

2.1. Traditional intersection-based routing in VNs

Some of the intersection-based routing protocols using fuzzy logic or traffic light-aware are discussed below, along with their similarity to our scheme. In [25], Ding et al. to consider traffic light's effects on routing performance, have proposed a traffic light-aware routing scheme called TLRC. The next road section in this scheme was selected according to the density and vehicle distribution. A greedy strategy was used to choose the next-hop nodes in this scheme. Chang et al. [26] suggested a shortest path-based traffic light aware routing scheme called STAR. Data packet forwarding in the road sections is based on the traffic lights statutes (e.g., green lights have higher priority for forwarding data packets). The next-hop node between two intersections was chosen using the greedy method. In [27], Xia et al. presented a greedy traffic light and queue-aware routing method called GTLQR, which used street connectivity in various traffic light conditions to select the best intersection. Then it would consider the channel condition, distance to destination, and queueing delay in prioritizing neighbor nodes for selecting the next-hop nodes. Zhou et al. [28] proposed a multiple intersection selection routing called MISR, which modeled the intersection selection as an optimization problem. It selected the intersection with minimum delay and highest connectivity by considering traffic light conditions. In addition, the instability coefficient by combining the neighbor's progress towards the destination and the speed difference between neighbors were used to select the next-hop nodes. Cao et al. [29] suggested a fuzzy logic intersection-based

routing scheme called IRFMFD, in which intersections would be selected based on the number of vehicles and link lifetime obtained from two-hop neighbors' information. The next-hop node selection in the IFRMFD scheme is a fuzzy-based method considering vehicle density, distance, and relative speed of vehicles. Also, Cao et al. [30] presented an intersection-based routing scheme with a fuzzy multi-factor called IRFMF. Density, number of lanes, and traffic flow were considered inputs of the fuzzy system in selecting the intersections. This method used the limited greedy strategy by considering the number of contacts between vehicles for the next-hop selection. He et al. [31] proposed an intersection-based traffic-aware routing via fuzzy Q-learning called ITAR-FQ. ITAR-FQ scheme used a weighted cost function composed of density, road latency, and Manhattan distance in its intersection selection mechanism. Also, link lifetime, link quality, Euclidean distance, and bandwidth via a fuzzy logic system were utilized to choose the next-hop nodes. Debnath et al. [32] proposed a fuzzy logic scheme for inter-vehicle network routing. Intersections selection in this method depends on the number of vehicles, the distance between two intersections, and the average speed of vehicles moving between two intersections. The communication link expiration time and communication quality factor simultaneously selected the next-hop nodes based on the fuzzy logic.

2.2. Intersection-based routing with the aid of SDNs

In recent years, numerous intersection-based multi-path or single-path routing schemes have been proposed in the software-defined vehicular networks via various techniques such as computational intelligence, and position estimation methods [2,13-20]. Some of them are being studied in the following. Zhao et al. [13] proposed Penicillium reproduction-based Online Learning Adaptive Routing method called POLAR. A proper routing method (i.e., ADOV and DSR) would be selected in this scheme based on the information obtained from the current traffic conditions. The geographical area was divided into multiple grids to facilitate real-time information processing. Furthermore, Penicillium Reproduction Algorithm (PRA) algorithm with optimization abilities would enhance the learning process's efficiency. Oubbati et al. [14] proposed a three-tier architecture called SEARCH to develop the awareness of road conditions and select the best-traveled path for vehicles. Dijkstra's algorithm was used to assess shorter time routes to get to any specific destination based on the journey times. The journey time was computed according to the density of vehicles, obstacles, casualties on the roads, and the average velocity of vehicles. Noorani and Seno [15] proposed SDN and Fog computing-based Switchable Routing called SFSR to select the most suitable path for data routing. A weighted cost function by combining Euclidean distance, route length, route delay, traffic congestion, and stability parameters is used to select the best routing paths. Zhao et al. [16] proposed an SDN-based fuzzy logic routing scheme that divided urban regions into several areas. A fuzzy-based method with several features, including mixed distribution, one-way connectivity, and valid distance, was employed for selecting the most suitable areas. In addition, a reinforcement learning algorithm was used to adapt the area selection policy. The most stable paths were also selected through fuzzy logic considering movement direction, speed, and distance. Abbas et al. [17] developed a hybrid routing strategy for SDVNs that divides road segments into different sections. High-reliability paths were selected considering the number of neighbors, link connections, and node distributions. The controller also would utilize a mechanism to deal with link failures. Hello messages interval was adjusted depending on the vehicle velocity. Venkatramana et al. [18] proposed an SDN-enabled connectivity-aware geographical routing called SCGRP, in which the shortest vehicular route would be computed using OSM spatial

data. A traffic value threshold on the roads in this method was used to provide high path connectivity. The route between the source and destination nodes was computed using the distance, vehicle density, and vehicle speed. Kong and Zhang [19] proposed an ant colony routing algorithm for SDVNs, in which a pair of exploring ants would find the best routing paths for data transmission using pheromones. The intersection was computed considering the density and distance between the source and destination, and then next-hop nodes were selected greedily. Gao et al. [20] proposed a hierarchical load balancing scheme called HRLB for software-defined vehicular networks. In this method, the geographical area was divided into multiple small grids based on the geographical location. A sequence of grids was selected based on the probability of transition and real-time density of these grids for transmission of data packets. Two paths with the least cost would be selected considering the load balancing and node utility among the grids. The weighted cost function was obtained as a combined function of route length, the number of vehicles, distance from adjacent nodes, and route load. The utility of nodes was calculated based on the remaining buffer size and distance of nodes to the destination.

2.3. Research gap

- Transmission of CP packets at intersections such as ITAR-FQ [31] requires updating periodically, which increases latency and overhead, especially at high densities or high traffic loads.
- Routing decision-making based on the local knowledge or two-hop neighbor's information due to the restricted view of vehicular topology leads to nodes trapped into the local maxima problem resulting in degradation of data routing efficiency [25-32].
- Regardless of road constraints and traffic light conditions, routing efficiency would be restricted. The traditional routing methods, including TLRC [25], STAR [26], GTLQR [27], and MISR [28], considered the effect of traffic lights during the local selection of intersections. However, traffic light effects in SDN-enabled vehicular networks should be considered while selecting a sequence of intersections in the data routing process.
- Despite all advances gained by using the traditional fuzzy type-1 and other computational intelligence techniques in SDVNs routing schemes [13-20], coping with uncertainties and ambiguity of the vehicular environments is still a challenging task that needs to be addressed.

3. Proposed method

This paper proposes a fuzzy-based routing scheme in the software-defined vehicular network to address the shortfalls outlined in Section 2.3 so that the ambiguity and uncertainty of the vehicular environment by tuning membership functions can be handled. Details of our scheme are given below.

3.1. Network model

Our hierarchical proposed SDVN architecture includes a set of vehicles, wireless switches, roadside units (RSUs), base stations (BSs), and centralized SDN controller. The centralized SDN controller is at the top level of the network. At the bottom level, RSUs and BSs are deployed for providing fog services supported by OpenFlow protocol, processing, and storage

capabilities. The network road can be depicted as graph $\mathcal{G}_\mathcal{J}\langle\mathcal{V}_\mathcal{J},\mathcal{E}_\mathcal{J},\mathcal{U}_\mathcal{J}\rangle$, where $\mathcal{V}_\mathcal{J}$ is a set of all intersections, and $\mathcal{E}_\mathcal{J}$ is a set of connected road segments to intersections. If there is at least one road segment between the two intersections, they are said to be adjacent. Vehicular topology also can form a dynamic graph as $\mathcal{G}_\mathcal{v}\langle\mathcal{V}_\mathcal{v},\mathcal{E}_\mathcal{v},\mathcal{U}_\mathcal{v}\rangle$. $\mathcal{V}_\mathcal{v}$ represents a finite set of nodes, and $\mathcal{E}_\mathcal{v}$ denotes a set of asymmetric wireless links between neighboring vehicles. $\mathcal{U}_\mathcal{J}$ and $\mathcal{U}_\mathcal{v}$ are the utility of intersections and vehicle links, respectively, and are explained in Section 3.4. Using Global Positioning Systems (GPS), each node can inform of its positional information. The centralized SDN controller stores information, including congestion, residual bandwidth, location, distance, velocity, direction, etc., obtained from the local controllers. This information is gathered periodically from each vehicle using the southbound SDN interface. When a node is in the range of two local controllers, the controller with the shortest Hello's response time is taken as the nearest controller. The response time relies on the vehicle density and the length between the RSUs and vehicles. Our offered layered architecture is shown in Figure 1.

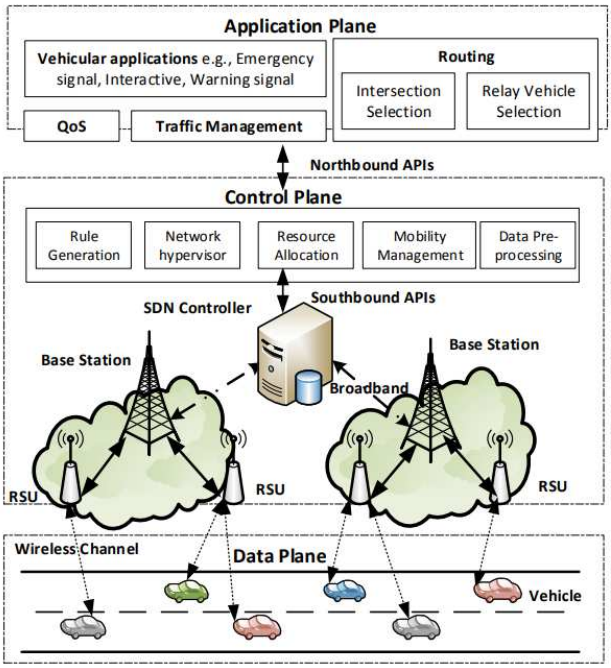


Figure 1. Proposed layered architecture

3.2. Neighbors discovery

Vehicles and roadside units can get their own one-hop neighbor's information by exchanging Hello packets periodically. The interval of broadcasting Hello messages can be adaptive [33] or fixed. This paper uses a fixed Hello interval ($\tau = 1$). Table 1 shows the format of Hello packet. By broadcasting Hello packets from vehicular nodes, RSUs can listen to these messages if they are within the vehicle's range and transmit the obtained information to the SDN controller. Hence, the centralized SDN controller views the entire vehicular network topology. Conversely, each RSU broadcasts ACK messages to notify vehicles of its presence by receiving Hello messages from the vehicles. If no information has been received within the pre-defined time, the local controllers notify the centralized SDN controller. Then both local and centralized SDN controllers update their information. In some cases, vehicular nodes can leave the current RSU's coverage area and join the coverage area of another controller. At this point, the previous controller (RSU) forwards a notification to the centralized SDN controller, waiting

for the flow table to be updated. The SDN controller fixes and updates the data routing path based on the new network topology and then sends back the flow table to the roadside units.

Table 1. Format of Hello packets

Field	Description
ID	**Vehicle's ID**
ts	**Sending time of Hello packet**
(dx_v, dy_v)	**Direction**
(x_v, y_v)	**Geographical location**
s_v	**Velocity**
CB	**Consumed Bandwidth**
R_c	**Remain Capacity of Buffer**
RWB	**Residual Bandwidth**

3.3. Proposed hierarchical SDN-enabled vehicular routing: HIFS

As shown in Figure 2, our HIFS scheme consists of intersection and forwarding vehicle selection strategies via fuzzy reinforcement learning algorithm. The details of these strategies and their steps are described below.

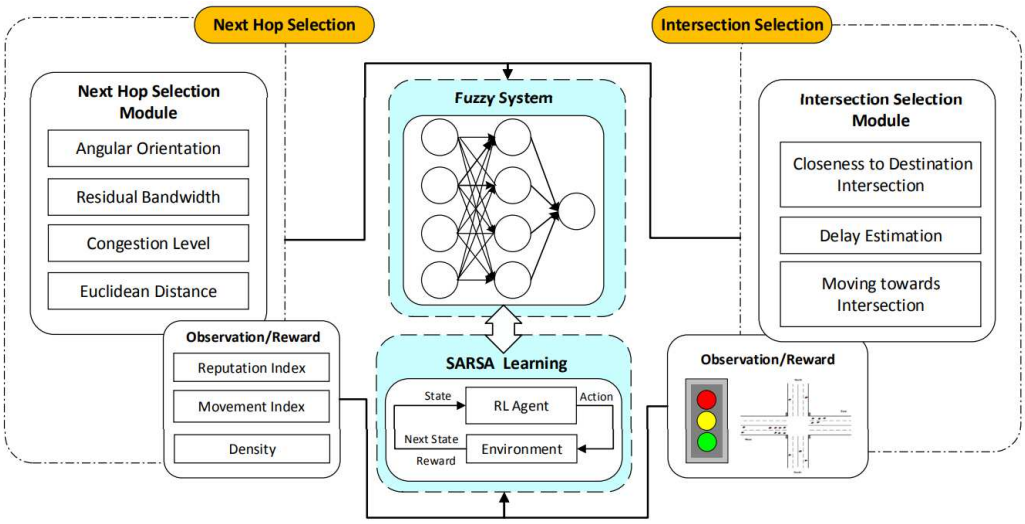


Figure 2. Proposed fuzzy SARSA learning strategy

3.3.1. Intersection selection strategy

Multi factors such as curve distance between intersections, predicting the number of vehicles moving towards intersections, and estimation latency between two intersections, via a fuzzy-based method, are used in the centralized SDN controller or local controllers to compute the candidate intersection's utility. Our utility-based intersection selection policy is described below.

3.3.1.1. Intersection selection criteria

1) Curve-Distance: The normalized curve distance to specify the most suitable candidate intersection in terms of closeness to the destination vehicle is calculated by Eq. (1).

$$Dist_{norm}(\mathcal{I}_m, \mathcal{I}_d) = \frac{Dist_{curve}(\mathcal{I}_m, \mathcal{I}_d)}{Dist_{curve}(\mathcal{I}_n, \mathcal{I}_d)} \quad (1)$$

where $Dist_{curve}(\mathcal{I}_n, \mathcal{I}_d)$ and $Dist_{curve}(\mathcal{I}_m, \mathcal{I}_d)$ are the curve distance of the current intersection $\mathcal{I}_n$ and candidate intersection $\mathcal{I}_m$ to destination intersection $\mathcal{I}_d$, respectively. Also, Eq. (2) [34] shows the curve distance between the two intersections.

$$Dist_{curve}(\mathcal{I}_k, \mathcal{I}_\ell) = |x_k - x_\ell| + |y_k - y_\ell| \tag{2}$$

The distance of the candidate destination intersection from the destination node and vehicle movement angle, as a weight function, is used to determine the destination intersection as Eq. (3) [35].

$$w_{(\mathcal{I}_\ell)} = \omega \times \left(1 - \frac{Dist_{(\mathcal{I}_\ell, v_d)}}{c}\right) + (1 - \omega) \times \cos\left(\overrightarrow{dV}_{(v_d)}, \overrightarrow{pV}_{(\mathcal{I}_\ell, v_d)}\right) \tag{3}$$

where

$$cos\left(\overrightarrow{dV}_{(v_d)}, \overrightarrow{pV}_{(\mathcal{I}_\ell, v_d)}\right) = \frac{a_{dV} \times a_{pV} + b_{dV} \times b_{pV}}{\sqrt{a_{dV}^2 + b_{dV}^2} \times \sqrt{a_{pV}^2 + b_{pV}^2}} \tag{4}$$

where $Dist_{(\mathcal{I}_\ell, v_d)}$ is the distance between candidate destination intersection $\mathcal{I}_\ell$ and destination v_d. c shows a normalization factor. $\overrightarrow{dV}_{(v_d)}$ as $(a_{dV}, b_{dV}) = (v_x t, v_y t)$ denotes the displacement vector of vehicle v_d at time period Δt. Also, $\overrightarrow{pV}_{(\mathcal{I}_\ell, v_d)}$ as $(a_{pV}, b_{pV}) = (x_{\mathcal{I}_\ell} - x_{v_d}, y_{\mathcal{I}_\ell} - y_{v_d})$ indicates the position vector forming at v_d and finishing at $\mathcal{I}_\ell$. ω is a weighted factor in the range of $(0,1)$. The highest weight of $w_{(\mathcal{I}_\ell)}$ refers to the selected destination intersection.

2) Predicting movement of vehicles towards intersections: Vehicles distributed sparsely at intersections or road segments have lower probability of finding a suitable relay node for sending the data packets. Based on the estimated movement of vehicles, this paper employed an evaluation index to predict the number of nodes entering or approaching the candidate intersections. Accordingly, moving vehicles towards or getting away from candidate intersections is computed as Eq. (5) [36].

$$Movement\ (M) = \frac{\sqrt{(x_{cur} - x_m)^2 + (y_{cur} - y_m)^2}}{\sqrt{\left(x_{future} - x_m\right)^2 + \left(y_{future} - y_m\right)^2}} \tag{5}$$

where (x_m, y_m) is the position of candidate intersection $\mathcal{I}_m$, and (x_{cur}, y_{cur}) shows the current position of vehicle v_i. By assuming the current speed of the vehicle as s_{cur}, the predicted position (x_{future}, y_{future}) of vehicle v_i is computed by Eq. (6) [36].

$$\begin{aligned} x_{future} &= x_{cur} + s_{cur}.t \sin\xi \\ y_{future} &= y_{cur} + s_{cur}.t \cos\xi \end{aligned} \tag{6}$$

where, by considering the previous position of vehicle v_i as (x_{prev}, y_{prev}), the angular orientation ξ can be calculated by Eq. (7).

$$\xi = \tan^{-1}\left(\frac{y_{cur} - y_{prev}}{x_{cur} - x_{prev}}\right) \tag{7}$$

The centralized SDN controller or local controllers have set a predefined threshold value ε to calculate the number of vehicles approaching candidate intersections on the road segments. If

M is higher than ε with a value of 0.65, the vehicle moves towards the intersection; otherwise, it moves away from it. The number of vehicles approaching the candidate intersection can be predicted as:

$$MTI_{norm}(\mathcal{I}_m) = \frac{N_e(\mathcal{I}_m)}{\sum_{t=1}^{q} N_t(\mathcal{I}_m)} \tag{8}$$

where $\sum_{t=1}^{q} N_t(\mathcal{I}_m)$ indicates the total number of vehicles at all road segments q leading to intersection $\mathcal{I}_m$. Also, $N_e(\mathcal{I}_m)$ shows the number of vehicles entering/approaching intersection $\mathcal{I}_m$ at road segment e. Based on Eq. (8), the road segment with a higher value of MTI_{norm} has a higher chance of forwarding data packets successfully. An exponentially weighted moving average (EWMA) is used by Eq. (9) to ensure that MTI_{norm} is not affected by the sudden changes. The coefficient α is set to 0.75 based on the simulation results.

$$MTI_{norm(t)}(\mathcal{I}_m) \leftarrow (1-\alpha) \times MTI_{norm(t-1)}(\mathcal{I}_m) + \alpha \times MTI_{norm(t)}(\mathcal{I}_m) \tag{9}$$

3) Delay estimation: The estimated delay between the two intersections can be computed based on the vehicle link's connectivity. The carry-and-forward mechanism is used if there is no link in the vicinity of the packet carrying vehicle. In this case, delay of data packets depends on the velocity of the carrying vehicle and the length of the traveled road segment by that vehicle. Also, if the road segment is connected, the delay induced to data packets relies on the hop counts passed at the road segment. Therefore, the estimated delay between the two intersections $\mathcal{I}_n$, and $\mathcal{I}_m$, can be computed as:

$$D_{ete}(\mathcal{I}_n, \mathcal{I}_m) = D_{hop}(\mathcal{I}_n, \mathcal{I}_m) + D_{cf}(\mathcal{I}_n, \mathcal{I}_m) \tag{10}$$

where $D_{cf}(\mathcal{I}_n, \mathcal{I}_m)$ is obtained as ${s_v}/{l^m_{r_{mn}}}$. s_v is the velocity of vehicle v_i, and $l^m_{\mathcal{I}_n,\mathcal{I}_m}$ indicates the length of the road segment between the two intersections in which data packets must be carried by vehicle. Also, $D_{hop}(\mathcal{I}_n, \mathcal{I}_m)$ is the summation of the buffer delay $d^q_{(v_i,v_j)}$, transmission delay $d^t_{(v_i,v_j)}$, and the propagation delay $d^p_{(v_i,v_j)}$ of vehicles forming the data transmission path $(v_i, v_j) \in \mathcal{V}_v$ between the two intersections $\mathcal{I}_n, \mathcal{I}_m \in \mathcal{V}_\mathcal{I}$ and obtained as Eq. (11) [37].

$$D_{hop}(\mathcal{I}_n, \mathcal{I}_m) = \sum_{(v_i,v_j)\in\{\mathcal{I}_m,\mathcal{V}_v,\mathcal{I}_n\}} D(v_i, v_j) = \sum_{(v_i,v_j)\in\{\mathcal{I}_m,\mathcal{V}_v,\mathcal{I}_n\}} \left(d^q_{(v_i,v_j)} + d^t_{(v_i,v_j)} + d^p_{(v_i,v_j)}\right) \tag{11}$$

where $d^t_{(v_i,v_j)} = \frac{l}{\varphi}$, and $d^p_{(v_i,v_j)} = \frac{d_{(v_i,v_j)}}{\gamma}$. l shows the packet size per bit, φ is the bandwidth in bits per second, $d_{(v_i,v_j)}$ denotes the Euclidean distance between v_i and v_j, and γ is the propagation speed in meters per second. Also, $d^q_{(v_i,v_j)}$ is computed by considering the buffer of nodes as $\mathrm{M/M/1}$ model with a Poisson distribution of input λ and exponential variable of output μ as Eq. (12).

$$d^q_{(v_i,v_j)} = \frac{1}{\mu - \lambda} \tag{12}$$

Therefore, based on the collected information from the vehicles, the centralized SDN controller can compute the latency $D_{ete}(\mathcal{I}_n, \mathcal{I}_m)$ for each road segment r_{mn}. In the computation of the delay, an exponentially weighted average method with coefficient $\alpha = 0.75$ is used as:

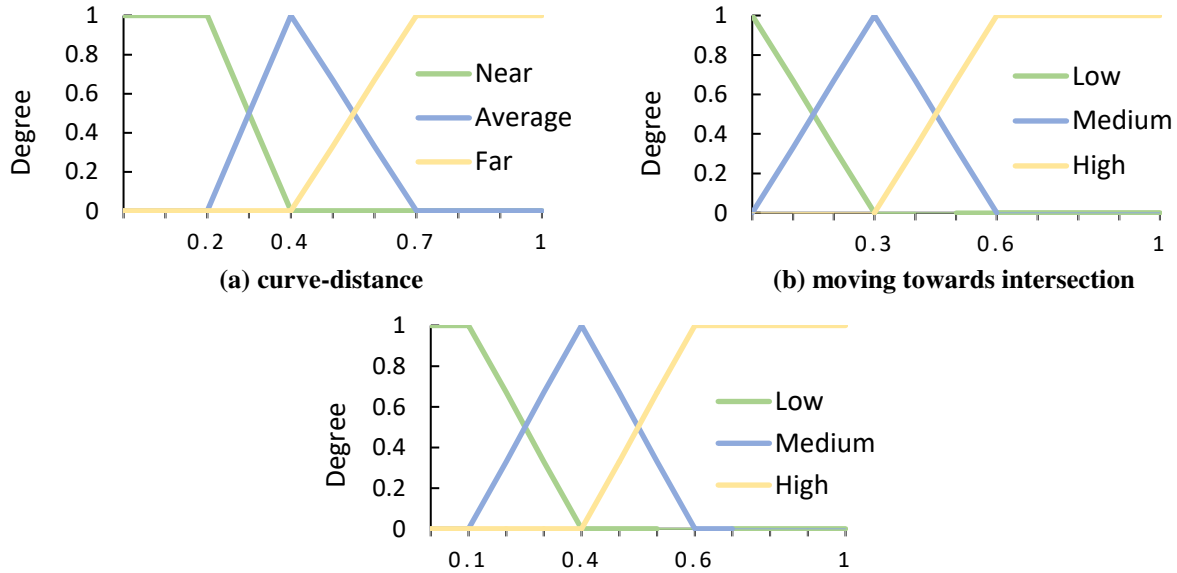
$$D_{ete(t)}(\mathcal{I}_n, \mathcal{I}_m) \leftarrow (1-\alpha) \times D_{ete(t-1)}(\mathcal{I}_n, \mathcal{I}_m) + \alpha \times D_{ete(t)}(\mathcal{I}_n, \mathcal{I}_m) \quad (13)$$

Finally, $D_{ete}(\mathcal{I}_n, \mathcal{I}_m)$ using Eq. (14) is normalized to (0,1).

$$0 \leq D_{norm}(\mathcal{I}_n, \mathcal{I}_m) = \frac{D_{ete}(\mathcal{I}_n, \mathcal{I}_m)}{max\{D_{ete}(\mathcal{I}_n, \mathcal{I}_k), .., D_{ete}(\mathcal{I}_n, \mathcal{I}_m)\}} \leq 1 \quad (14)$$

3.3.1.2. Fuzzy system design for intersection selection strategy

Fuzzification, fuzzy inference, and defuzzification are the main parts of a fuzzy inference system (FIS) [38]. In the fuzzification process, a set of inputs crisp values are converted to the corresponding fuzzy set. This process depends on the membership functions defined separately for each input. Fuzzy membership functions describe the degrees of a crisp value belonging to linguistic variables. The most commonly used membership functions are Gaussian, triangular, and trapezoidal. In this paper, triangular and trapezoidal membership functions are used. Obtaining the final fuzzy values, mapping these values to predefined IF/THEN rules, and combining rules are the main responsibilities of fuzzy inference. Finally, defuzzification converts the final fuzzy value to a crisp one. In our fuzzification process, crisp values of the curve distance of candidate intersections to destination intersection, moving vehicles towards intersections, and the estimated delay between two intersections are converted to the linguistic variables $\{Near, Average, Far\}$, $\{Low, Medium, High\}$, and $\{Low, Medium, High\}$, respectively, as illustrated in Figure 3.



(a) curve-distance **(b) moving towards intersection**

(c) estimated delay

Figure 3. Input membership functions for intersection selection strategy

The ranking of the intersections according to the fuzzy rule base (i.e., IF/THEN rules) is shown in Table 2. In a rule, the IF part is called antecedent, and THEN part is called the consequent. Each rule combines input variables and obtains fuzzy decision output based on the linguistic variables as $\{Very\ Low\ (VL), Low(L), Medium(M), High(H), Very\ High(VH)\}$. For example, if the curve distance is far, the latency is high, and the number of vehicles moving towards the intersection is low, the candidate intersection has a much lower chance of being selected in the data routing path. Finally, the fuzzy output is converted to a numerical value using the output membership function depicted in Figure 4 and the Last of Maxima (LOM) defuzzification method. The crisp value in the LOM method is the largest element of the fuzzy

value. Here LOM shows the utility of candidate intersections to be selected. A higher value of LOM would indicate a better intersection.

Table 2. Rule base for ranking candidate intersections

Rule	IF			THEN	Rule	IF			THEN
	$Dist_{norm}$	D_{norm}	MTI_{norm}	Rank		$Dist_{norm}$	D_{norm}	MTI_{norm}	Rank
1.	Near	Low	Low	Medium	**15.**	Average	Medium	High	Medium
2.	Near	Low	Medium	High	**16.**	Average	High	Low	Low
3.	Near	Low	High	Very High	**17.**	Average	High	Medium	Low
4.	Near	Medium	Low	Medium	**18.**	Average	High	High	Low
5.	Near	Medium	Medium	Medium	**19.**	Far	Low	Low	Low
6.	Near	Medium	High	High	**20.**	Far	Low	Medium	Low
7.	Near	High	Low	Low	**21.**	Far	Low	High	Medium
8.	Near	High	Medium	Low	**22.**	Far	Medium	Low	Low
9.	Near	High	High	Medium	**23.**	Far	Medium	Medium	Low
10.	Average	Low	Low	Low	**24.**	Far	Medium	High	Medium
11.	Average	Low	Medium	Medium	**25.**	Far	High	Low	Very Low
12.	Average	Low	High	High	**26.**	Far	High	Medium	Low
13.	Average	Medium	Low	Low	**27.**	Far	High	High	Low
14.	Average	Medium	Medium	Medium					

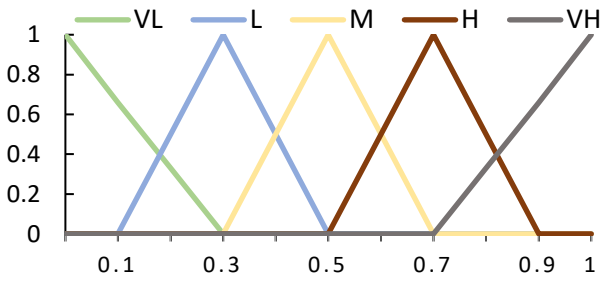


Figure 4. Output membership function for ranking candidate intersections

3.3.2. Forwarding node selection strategy

This section describes selecting the relay nodes that make up the data transmission path from source to destination. The considered criteria and fuzzy system processing are described below.

3.3.2.1. Forwarding vehicle selection criteria

1) Load capacity: In this paper, to mitigate the effect of nodes buffer overflow in the formed data routing path, the size of the remaining buffer is used as a criterion for the next-hop node selection. The residual buffer size of a node similar to [39] is computed by Eq. (15).

$$L_c(v_i) = \frac{R_c(v_i)}{T_c(v_i)} \tag{15}$$

where R_c and T_c indicate the current buffer size, and the maximum buffer size of node v_i at time t, respectively. The EWMA method with coefficient $\alpha = 0.75$ as Eq. (16) is employed in load capacity calculation.

$$L_{c(t)}(v_i) \leftarrow (1-\alpha) \times L_{c(t-1)}(v_i) + \alpha \times L_{c(t)}(v_i) \tag{16}$$

2) Angular orientation: In our strategy to select the next-hop node, the angular orientation is computed using Eq. (17) [36].

$$\theta(v_i, v_j) = \tan^{-1}\left(\frac{y_m - y_i}{x_m - x_i}\right) - \tan^{-1}\left(\frac{y_j - y_i}{x_j - x_i}\right) \quad (17)$$

where (x_m, y_m) is the location of intersection $\mathcal{I}_m$, (x_i, y_i) denotes the position of forwarding vehicle v_i, and (x_j, y_j) shows the position of neighboring node v_j. Assigning higher priority to a neighbor with a minimum angle caused higher link stability between vehicles. By Eq. (18), the angular orientation is normalized in the range of (0,1).

$$0 \leq \theta_{v_i,v_j}^{norm} = \frac{\theta(v_i, v_j)}{max\{\theta(v_i, v_j), \dots, \theta(v_i, v_n)\}} \leq 1 \quad (18)$$

where $\theta(v_i, v_j)$ shows the angular orientation between vehicle v_i and neighbor v_j, and $max\{\theta(v_i, v_j), \dots, \theta(v_i, v_n)\}$ is the maximum angles in the one-hop neighbors of vehicle v_i.

3) Euclidean distance: Normalized Euclidean distance for selecting the next-hop node is considered in this paper as Eq. (19).

$$0 \leq Dist_{norm}(v_i, v_j) = \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{R} \leq 1 \quad (19)$$

where(x_i, y_i), and (x_j, y_j) are the position of the current vehicle v_i and its neighbor node v_j, respectively, and R denotes the communication range.

4) Residual bandwidth: RBW_i is one of the routing metrics applied to the fuzzy system in the next-hop selection policy in our scheme, which is computed by Eq. (20) [34].

$$RBW_i = \frac{C_{rate} - \delta}{C_{rate}} \quad (20)$$

where C_{rate} is the channel data rate, and δ denotes the total data generation rate computed as the sum of the bandwidth consumed by forwarding vehicle and its neighbors, including the MAC layer overhead, ACKs, and retransmissions. The EWMA method is used to consider the effect of traffic changes as follows:

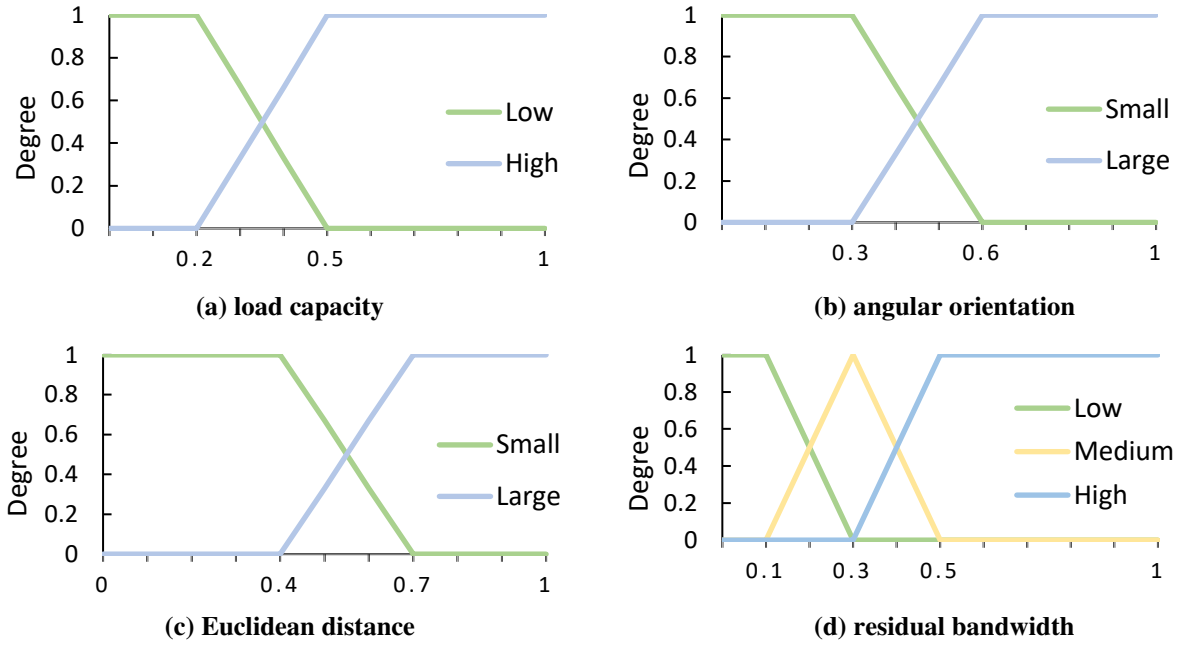
$$\overline{\delta_t} = (1 - \alpha) \times \overline{\delta_{t-1}} + \alpha \times \delta_t \quad (21)$$

where $\overline{\delta_t}$ is the average data generation rate according to the data generation rates at time t and $t - 1$, respectively, and a is the configuration parameter set to 0.4.

3.3.2.2.Fuzzy system design for forwarding vehicles selection strategy

Our forwarding vehicle selection scheme is a TSK-FLS method with a four-input/one-output zero-order and 24 rules. In the fuzzification process, four quantifiable input parameters (e.g., crisp values), including load capacity, angular orientation, Euclidean distance, and residual bandwidth, are converted to the linguistic variables and expressed by $\{Low, High\}$, $\{Small, Large\}$, $\{Small, Large\}$, and $\{Low, Medium, High\}$, respectively, as shown in Figure 5. In addition, the fuzzy rule base (i.e., IF/THEN rules) in Table 3 is defined for ranking the one-hop neighboring vehicles. The linguistic variables are denoted as $\{Very\ Low\ (VL), Low(L), Medium(M), High(H), Very\ High(VH)\}$. Finally, Figure 6 shows the

output membership function in the defuzzification process to get the crisp values. The defuzzification process in this level, similar to the first level, used the LOM method.



(a) load capacity **(b) angular orientation**

(c) Euclidean distance **(d) residual bandwidth**

Figure 5. Input membership functions in the forwarding node selection strategy

Table 3. Rule base in the forwarding node selection strategy

Rule	IF				THEN	Rule	IF				THEN
	$Dist_{norm}$	L_c	RBW_i	$\theta^{norm}_{v_i,v_j}$	Rank		$Dist_{norm}$	L_c	RBW_i	$\theta^{norm}_{v_i,v_j}$	Rank
1.	Small	Low	Low	Small	Low	**13.**	Small	Low	Low	Large	Low
2.	Small	High	Low	Small	Low	**14.**	Small	High	Low	Large	Vey Low
3.	Small	Low	Medium	Small	Medium	**15.**	Small	Low	Medium	Large	Low
4.	Small	High	Medium	Small	Low	**16.**	Small	High	Medium	Large	Low
5.	Small	Low	High	Small	Medium	**17.**	Small	Low	High	Large	Medium
6.	Small	High	High	Small	Low	**18.**	Small	High	High	Large	Low
7.	Large	Low	Low	Small	Medium	**19.**	Large	Low	Low	Large	Low
8.	Large	High	Low	Small	Low	**20.**	Large	High	Low	Large	Low
9.	Large	Low	Medium	Small	High	**21.**	Large	Low	Medium	Large	Medium
10.	Large	High	Medium	Small	Low	**22.**	Large	High	Medium	Large	Low
11.	Large	Low	High	Small	Very High	**23.**	Large	Low	High	Large	High
12.	Large	High	High	Small	Medium	**24.**	Large	High	High	Large	Low

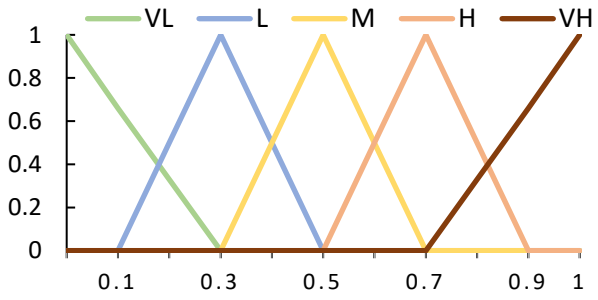


Figure 6. Output membership function in the forwarding node selection strategy

3.3.3. Tuning fuzzy membership functions

In our scheme, SARSA learning algorithm [40] is employed for tuning the consequent fuzzy output membership functions of intersection and forwarding node selection strategies. SARSA learning allows agents to learn online policy while interacting with their environments and is defined by quintuple $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$. By considering the whole network as the environment, centralized SDN controller and RSUs are defined as our agents due to their ability to compute data routing paths. The learning task is finding the best parameters for output fuzzy membership functions in both strategies. It can be done by observing the current state s_t of the environment and taking action a_t based on their own policies $\pi_{\mathcal{I}}: \mathcal{S}_{\mathcal{I}} \rightarrow \mathcal{A}_{\mathcal{I}}$ and $\pi_{\mathcal{V}}: \mathcal{S}_{\mathcal{V}} \rightarrow \mathcal{A}_{\mathcal{V}}$. State-space, action-space, and immediate rewards in our policies are defined below.

State-space: Each combination of antecedent parts in the intersection and forwarding vehicle selection policies is considered the agent's state and shown as $\mathcal{S}_{\mathcal{I}} = \{s_{1_{\mathcal{I}}}, s_{2_{\mathcal{I}}}, \dots, s_{k_{\mathcal{I}}}\}$, and $\mathcal{S}_{\mathcal{V}} = \{s_{1_{\mathcal{V}}}, s_{2_{\mathcal{V}}}, \dots, s_{l_{\mathcal{V}}}\}$, respectively.

Action space: Two sets of actions, $a_i(t) \in \mathcal{A}_{\mathcal{V}}$, and $a_m(t) \in \mathcal{A}_{\mathcal{I}}$, are defined for tuning consequent fuzzy membership functions in the intersection and forwarding vehicle selection policies and displayed as $\mathcal{A}_{\mathcal{I}} = \{a_{1_{\mathcal{I}}}, a_{2_{\mathcal{I}}}, \dots, a_{k_{\mathcal{I}}}\}$ and $\mathcal{A}_{\mathcal{V}} = \{a_{1_{\mathcal{V}}}, a_{\mathcal{V}}, \dots, a_{l_{\mathcal{V}}}\}$. The possible discrete actions in both cases consist of 11 singleton functions {0.0,0.1,0.2,0.3,0.4,0,5,0.6,07,0.8,0.9,1.0}.

Reward: As a utility, agents will receive the rewards $r_t^{\mathcal{I}_{n,m}}$ (e.g., intersection selection reward) and $r_t^{\mathcal{V}_{i,j}}$ (e.g., relay node selection reward), and transition to new states. Then, the agents will update their current policies, $\pi_{\mathcal{I}}$ and $\pi_{\mathcal{V}}$, according to the reward received to perform the optimal action. The following reward functions in both levels are explained.

i. **Intersection reward:** The reward function $r_t^{\mathcal{I}_{n,m}}: \mathcal{S}_{\mathcal{I}} \times \mathcal{A}_m \mapsto \mathbb{R}$ for tuning the fuzzy output membership function of the intersection selection policy is obtained as Eq. (22).

$$r_t^{\mathcal{I}_{n,m}} = \log_2\left(\frac{R_G \times R_c}{R_D} + 0.01\right) \tag{22}$$

where R_G is the ratio of the green time to the intersection cycle time, R_c shows the number of passing vehicles at the intersection in the green light duration, and R_D indicates the total number of vehicles stopped on all the roads at candidate intersection $\mathcal{I}_m$ and computed as Eq. (23) [41]. If the vehicle's velocity is less than v_{min}, it is assumed that the vehicle is waiting. D_{lk} indicates the number of waiting vehicles at road l and lane k.

$$R_D = \sum_{l=1}^{l} \sum_{k=1}^{k} D_{lk} \tag{23}$$

ii. **Vehicle reward:** For the fuzzy output membership function in the forwarding vehicle selection strategy, the reward $r_t^{\mathcal{V}_{i,j}}: \mathcal{S}_{\mathcal{V}} \times \mathcal{A}_{\mathcal{V}} \mapsto \mathbb{R}$ is calculated as:

$$r_t^{v_{i,j}} = \log_2\left(\frac{s_{v_j}(t, t-\Delta t) \times n_{v_j}(t)}{c+1} + 0.01\right) \tag{24}$$

where $s_{v_i}(t, t-\Delta t)$ is the stability of the neighboring vehicle v_j in the transmission range R and computed based on [42], t is the current time, and Δt is the previous time

interval of sending Hello message. $n_{v_i}(t)$ denotes the local density of vehicle v_j, divided by the optimal local density found in [43]. Finally, c as a reputation index shows the number of times a node between the two intersections is selected as a next-hop node. The reward function is divided by one to avoid ambiguity.

Each rule R_i in the fuzzy SARA strategy is depicted as **if** x_1 is L_{i1}, x_2 is L_{i2}, and x_n is L_{in}, **then** (a_{i1} is $Q(\mathrm{i},1), \dots, a_{im}$ is $Q(i,m)$). Here, $L_i = L_{in} \times \dots \times L_{i2} \times L_{i1}$ is a fuzzy set of the ith rule. m shows the number of possible discrete actions that are the same for both levels. a_{ij} denotes the jth candidate action, and $Q(i,j)$ is the Q-value of the jth action in the ith rule. In the intersection selection strategy, x_1, x_2, and x_3 are the distance from the candidate intersections to the destination intersection, the delay estimated, and moving vehicles towards intersections, respectively. Conversely, load capacity, Euclidean distance, angular orientation, and residual bandwidth are considered as x_1, x_2, and x_3, respectively, in the forwarding vehicle selection strategy. In both fuzzy SARSA levels in rule i, action a_{ij} is chosen based on the softmax action selection rule as Eq. (25) [40].

$$P(a_{ij}) = \frac{exp(\psi_i Q(i,j)/\delta)}{\sum_{k=1}^{m} exp(\psi_i Q(i,k)/\delta)} \tag{25}$$

The fire strength ψ_i of each rule i is calculated as Eq. (26). In addition, δ is a positive variable called temperature.

$$\psi_i = MIN\left(\mu_{x_1 \, is \, L_{i1}}(x_1), \dots, \mu_{x_n \, is \, L_{in}}(x_n)\right) \tag{26}$$

The rule with highest fire strength value is considered as output in both levels of fuzzy systems. The Q table will be updated according to Eq. (27) at the predefined intervals.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left(r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right) \tag{27}$$

where α is the learning rate, and γ depicts the discount factor. Algorithm 1 shows the process of the fuzzy SARSA learning algorithm in tuning output fuzzy membership functions of the intersection selection and forwarding vehicle selection strategies.

Algorithm 1: **Fuzzy SARSA learning**

Initialize $Q(s_t, a_t) = 0$ for all s_t and a_t
Initialize α and γ
for every time step t **do**
  Initialize s_t as an antecedent part of the fuzzy system
  Calculate the firing strength of each rule using the MIN operator
  Choose the R_i rule to run $\left(\mu(R_i) \leftarrow max_{j=1,\dots,z}\mu(R_j)\right)$
  while a_t is not a terminal state, **do**
    Take action a_{ij} using modified Boltzman action selection
    Observe the reward r_t and next state s_{t+1}
    Update $Q(s_t, a_t)$ value using Eq. (27)
    Current state $s_t \leftarrow s_{t+1}$
    Current action $a_t \leftarrow a_{t+1}$
  end while
end for

3.3.4. Proposed SDN-enabled vehicular data routing

Since the end-to-end routing performance depends on all intersections and relays constituting data transmission paths, their utilities based on the descriptions in Section 3.3 are considered

in our routing scheme. Let $u_{\mathcal{I}_n,\mathcal{I}_m}$ be the utility of candidate intersection $\mathcal{I}_m$ from intersection $\mathcal{I}_n$, and u_{v_i,v_j} is the link utility between two vehicles v_i, v_j. The utilities of formed routes consisting of sequence intersections and relays are defined as Eq. (28).

$$\mathcal{U}_{(\mathcal{I}_s,\mathcal{I}_d)} = \sum_{\mathcal{I}_n,\mathcal{I}_m \in \{\mathcal{I}_s,\mathcal{V}_\mathcal{I},\mathcal{I}_d\}} u_{\mathcal{I}_n,\mathcal{I}_m}, \qquad \mathcal{U}_{(v_s,v_d)} = \sum_{v_i,v_j \in \{v_s,\mathcal{V}_v,v_d\}} u_{v_i,v_j} \tag{28}$$

where $\mathcal{U}_{(\mathcal{I}_s,\mathcal{I}_d)}$, and $\mathcal{U}_{(v_s,v_d)}$ show the utilities of the sequence intersections and forwarding vehicles in the data transmission path. Due to the different combinations of intersections and vehicular nodes, the centralized SDN controller can form k various paths $\mathcal{P}_k \in \langle \mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_k \rangle$ consisting of a sequence of intersections $\langle \mathcal{I}_s, .., \mathcal{I}_n, \mathcal{I}_m .., \mathcal{I}_d \rangle$. In addition, different routes $\mathcal{R}_r \in \langle \mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_r \rangle$ consisting of various forwarding vehicles $\langle v_s, \dots, v_i, v_j, \dots, v_d \rangle$ can be formed in the selected intersections. Therefore, to have high end-to-end routing performance, the best sequence of intersections and forwarding vehicles among all possible paths found must form the data routing path. First, a sequence intersection with the maximum utility using Eq. (29) is selected among all intersection sequences from the source to the destination intersection.

$$\max_{\mathcal{P}_k \in \langle \mathcal{P}_1,\mathcal{P}_2,\dots,\mathcal{P}_k \rangle} \mathcal{U}_{\mathcal{P}_k \langle \mathcal{I}_s,..,\mathcal{I}_n,\mathcal{I}_m..,\mathcal{I}_d \rangle} \tag{29}$$

After determining the intersections with maximum utility, our scheme in the selected intersection sequence has chosen the route consisting of vehicle links with maximum utility according to Eq. (30).

$$\max_{\mathcal{R}_r \in \langle \mathcal{R}_1,\mathcal{R}_2,\dots,\mathcal{R}_k \rangle} \mathcal{U}_{\mathcal{R}_r \langle v_s,\dots,v_i,v_j,\dots,v_d \rangle} \tag{30}$$

Finally, to ensure data deliver successfully from source to destination, several constraints are considered as follows:

$$\varsigma_{(v_i,v_j)} + \varsigma_{(v_j,v_i)} \leq 1, \qquad \forall\, v_i \in \mathcal{V}_v,\ v_j \in N(v_i) \tag{31a}$$

$$\chi_{(\mathcal{I}_n,\mathcal{I}_m)} + \chi_{(\mathcal{I}_m,\mathcal{I}_n)} \leq 1, \qquad \forall\, \mathcal{I}_n \in \mathcal{V}_\mathcal{I}, \mathcal{I}_m \in N(\mathcal{I}_n) \tag{31b}$$

$$dist_{curve}^{\mathcal{I}_m,\mathcal{I}_d} \leq dist_{curve}^{\mathcal{I}_n,\mathcal{I}_d}, \qquad \forall \langle \mathcal{I}_n, \mathcal{I}_m, \mathcal{I}_d \rangle \in \mathcal{V}_\mathcal{I} \tag{32}$$

$$dist_{estimate}^{(v_i,v_j)} \leq R, \qquad \forall\, v_i \in \mathcal{V}_v,\ v_j \in N(v_i) \tag{33}$$

$$D_{ete}(v_s, v_d) = \sum_{(v_i,v_j) \in \{v_s,\mathcal{V}_v,v_d\}} \left(\left(\frac{1}{\mu - \lambda}\right) + \frac{l}{\varphi} + \frac{d_{(v_i,v_j)}}{\gamma} \right) \leq \Delta, \quad \forall\, v_i \in \mathcal{V}_v,\ v_j \in N(v_i) \tag{34}$$

- **Constraint 1:** The following two constraints are considered to avoid nodes trapped in the routing loop:
  i. **(31a):** The variable $\varsigma_{(v_i,v_j)}$ is set to one if data is sent from node v_i to v_j; otherwise, it is set to zero.
  ii. **(31b):** If a forwarding vehicle is traveled from $\mathcal{I}_m$ to $\mathcal{I}_n$, $\chi_{(\mathcal{I}_m,\mathcal{I}_n)}$ is equal to one; otherwise equals zero.
- **Constraint 2 (32):** To ensure the progress of data packets toward the destination, the curve distance of candidate intersections to the destination intersection must be less than the curve distance of the current intersection to the destination intersection.

- **Constraint 3 (33):** Let $dist_{estimate}^{(v_i,v_j)}$ is defined as the estimated distance between vehicles v_i and v_j and R be the transmission radius between two neighbor's vehicles. $dist_{estimate}^{(v_i,v_j)}$ must be less than R; otherwise, the data forwarding will fail.
- **Constraint 4 (34):** The total end-to-end delay of the application should not exceed the delay constraint Δ. $\mathcal{V}_v$ is a set of relay vehicles in the data transmission path. To find out the definitions of contained parameters in Eq. (34), please see Eq. (11).

The best routing path is computed in the centralized SDN or RSUs at time t through Dijkstra algorithm. Assuming that packet k is sent from source $v_s(k)$ to destination $v_d(k)$, Algorithm 2 summarized our SDN-enabled data routing process. Given the local density, two possibilities are considered in the routing processing: 1) the source node is within the transmission range of RSU, and 2) the source node is outside the RSU's communication range. In the first case, $v_s(k)$ forwards a path request message directly to the RSU (line 3). It includes source and destination IDs, road ID, data packet size, and delay constraint of application. Upon receiving the route request packets, controller perform the routing algorithm to specify the most suitable path between the source-destination nodes (lines 5-6).

Algorithm 2: **SDN_enabled data routing**

```
1:  for every packet that v_i needs to forward, do
2:    if v_s is within the RSUs range, then
3:      v_s sends a route request to RSU
4:      if RSU has a route to v_d then
5:        RSU computes the utilities of intersections and relay vehicles
                    according to the fuzzy SARSA Algorithm in Section 3.3
6:        RSU computes the best path based on Eqs. (29), (30) and sends it to v_s
7:      else
8:        RSU sends the request to the centralized SDN controller
9:        SDN controller computes the utilities of intersections and relay vehicles
                    according to the fuzzy SARSA Algorithm in Section 3.3
10:       SDN controller computes the best path based on Eqs. (29), (30) and sends it to v_s
11:     end if
12:   else
13:     if v_d is a one hop neighbor of v_i, then
14:       v_i forwards the packet directly to v_d
15:     else if v_i has neighbor nodes then
16:       compute the utility of neighbor nodes based on Section 3.3
17:       v_i chooses the neighbor with maximum utility as the next hop node
18:     else
19:       v_i uses the carry and forward mechanism
20:     end if
21:   end if
22: end for
```

The request message is transmitted to the SDN controller if the destination is outside the local controller's communication range (line 8). SDN computes the transmission path, including a sequence of intersections and relay nodes. Then, the obtained route through the local controller in a replay packet is sent back to the source vehicle (lines 9-10). If the destination of packet k is in the neighbor table, packet k is delivered to it directly (lines 13-14). Finally, if the source node is not within the local controller's communication range, the forwarding node selects the best relay node obtained by fuzzy results (lines 16-17). If the forwarding vehicle has no neighbors within its communication range, it uses a carry-and-store mechanism. This process

will proceed until the carrier vehicle arrives at the appropriate neighbor vehicle or controller node located at the next intersection (line 19).

4. Performance analysis

The performance evaluation of our HIFS scheme is discussed in this section in various scenarios compared to the different state-of-the-art methods, including real-time evaluation of road conditions using CP packets (e.g., ITAR-FQ [31]), traffic light aware strategy (e.g., MISR [28]), and SDN-enabled routing scheme (e.g., GLS [16]). Details of these methods for selecting intersections and relay nodes are described in Section 2.

4.1. Simulation settings

Our scheme and other compared methods were simulated via NS-3.29 [44] as a network simulator and SUMO as a mobility generator tool [45] to generate realistic vehicle mobility. The simulated scenario describes real urban roads belonging to Los Angeles in USA, with an area of 3000 meters by 300 meters extracted from OpenStreetMap (OSM) (see Figure 7). The considered scenario includes the different number of lanes, intersections, and varied traffic light cycles. The car-following mobility model was used in SUMO for the movement of vehicles. In this model, the velocity of a node depends on the velocity of the vehicular node ahead of it. Before initiating the simulation, the vehicular nodes were distributed randomly. As the simulation starts, each vehicle moves on the roads with a minimum and maximum velocity of 10 km/h to 90 km/h. The Nakagami-m model was considered as a propagation model in the physical layer. RSUs as fixed nodes were located at intersections, and Open Flow protocol was installed on them. Vehicles and RSUs have multiple user interfaces, including LTE, WiMax, and IEEE 802.11p. The transmission radius was varied according to the interfaces. Using UDP, ten randomly source-destination pairs with the data packet sizes of 512 bytes and a generation rate of two packets per second as foreground traffic were produced by the Constant Bit Rate (CBR) traffic flows. The foreground traffic was generated 30 seconds after the simulation started to reduce the effect of transient changes. Also, the last generated foreground traffic was sent 50 seconds before the simulation ended. Background traffic, including ten random source-destination pairs with various data packet sizes, is also created to interfere with the foreground traffic. That is' why source-destination pairs between the foreground and background flows are overlapped to each other. It is assumed that each vehicle (e.g., source node or an intermediate node) can get the positional information of the destination with a query of a centralized administration unit such as RLSMP [46]. Each point of each routing scheme in all graphs is an average of 30 simulations, where the error bars indicate the 95% confidence intervals. Each simulation duration is set to 500 seconds. The tunable simulation parameters are detailed in Table 4.

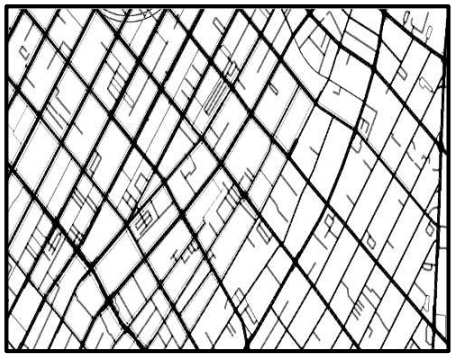


Figure 7. Los Angeles urban scenario imported to SUMO

Table 4. Simulation parameters

Simulation Parameter	Value
Number of Vehicles	50-500
Network Simulator	NS-3.29
Simulation Area	Urban, 3000 m×3000 m
Mobility Generator	SUMO
Wireless Technology	802.11p, LTE, WiMax
Transmission Range	300 m
OpenFlow Module	OFSwitch1.3
Channel Data Rate	6 Mbps
Velocity of Vehicles	10-90 km/h
Background/ Foreground Connection Paris	10
Simulation Duration	500 s
Traffic Type	Constant Bit Rate (CBR)
Queue Length	50 packets
Hello Interval (τ)	1 s
Size of Data Packets for Foreground Traffic	512 Byte
Size of Data Packets for Background Traffic	256-2048 Byte
Propagation Model	Nakagami-m
Foreground Packet Generation Rate	0.5 Packet/s
Background Traffic Load	0.1 Mbps
Learning Rate (α)	0.05
Discount Factor (γ)	0.95
Threshold (ε)	0.65
Delay Constraint (Δ)	1-2 s
Temperature (δ)	0.1
Q-Table Update Time	500 milliseconds
Stopped Vehicles (v_{min})	0.5 m/s
Number of Run	30

4.2. Performance metrics.

- **Packet Delivery Ratio (PDR)**: PDR is defined as the ratio of successfully received data packets at the destination vehicular node to the number of data packets sent at the source vehicle.
- **Average End to End Delay (AE2ED):** AE2ED depicted the average end-to-end delay of data packets sent from the source vehicle and received at the destination.
- **Path length (PL):** PL is computed as the sum of the curve distances of the vehicle links from the source to the destination node that make up the data transmission path.
- **Normalized Routing Overhead (NRO):** NRO is defined as the size of the control message sent to the size of data packets successfully received at the destination.

4.3. Evaluation scenarios

Two sets of tests were employed to assess the effectiveness of routing schemes; the following parameters in them are varied.

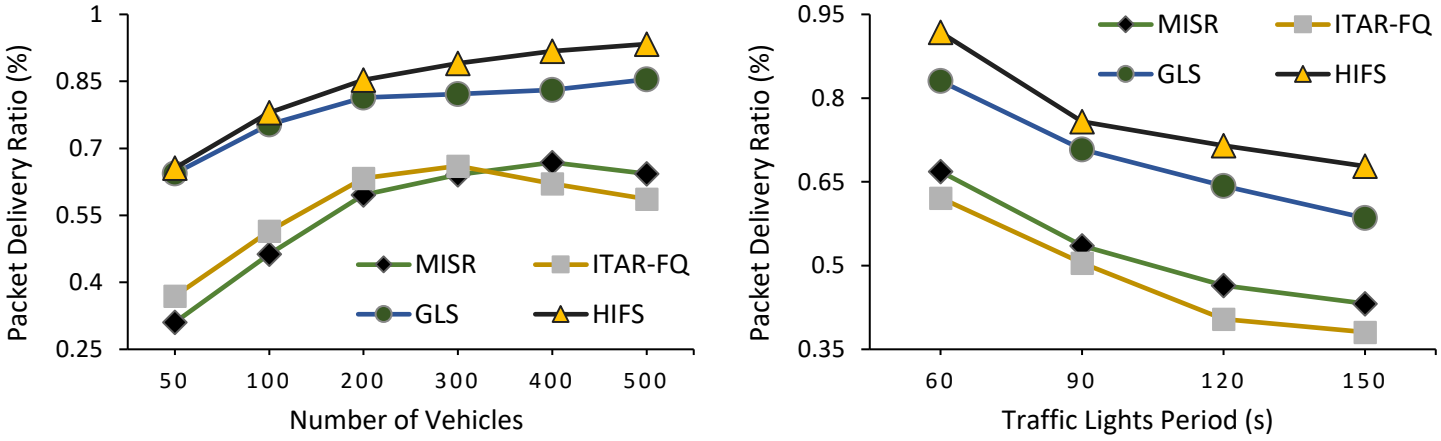
- **Vehicle density changes:** Vehicle density was changed from 50 to 500 vehicles. Traffic lights duration in this test is set to 60 seconds.
- **Different traffic light duration:** The traffic light duration varied from 60 seconds to 150 seconds. The time ratio of green and red traffic lights is 1:1. It means that the first

half of the traffic lights period equals the green light time, and the second half equals the red-light duration. The number of vehicles in this test is set to 400 vehicles.

4.4. Simulation results

4.4.1. Packet delivery ratio

The evaluated performance of the ITAR-FQ, MISR, and GLS methods compared to our HIFS scheme are shown in Figure 8 with respect to the packet delivery ratio for the different number of vehicles and various traffic lights duration. For various vehicle densities, as shown in Figure 8(a), the packet delivery ratio is increased in all compared methods at densities from 50 to 300. Then, from 300 to 500 vehicles, the GLS method and our HIFS scheme have slightly increased, while ITAR-FQ and MISR methods have declined. At low densities, our HIFS and GLS scheme reduced the likelihood of nodes trapped into the local maxima problem by considering the global perspective and using microscopic and macroscopic information. The MISR and ITAR-FQ schemes with limited visibility have led to sub-optimal routing decisions. Therefore, their difference and our scheme at low densities are significant. At higher densities, limited bandwidth and nodes competition caused the performance of all routing methods to be restricted. Our scheme has adapted routing decisions considering traffic light conditions, resulting in a higher packet delivery ratio than all compared methods. As shown in Figure 8(a), by increasing vehicle density, the performance improvement of our proposed HIFS scheme is more significant. Since the possibility of forming long queues of vehicles waiting behind red lights is vague at low densities, our HIFS and GLS method differences are negligible. Figure 8(b) shows the influence of various traffic lights duration on the performance of all compared schemes. As the traffic light duration increases, a declining behavior in the packet delivery ratio of all schemes can be seen in Figure 8(b). Lack of consideration of traffic light's effects when selecting intersections besides the afore-mentioned ITAR-FQ scheme's shortfalls has further reduced its performance. The GLS scheme also suffers from the lack of considering the traffic light's effects during the data routing process, resulting in reduced performance upon increasing traffic light duration. Our scheme can deal with dynamic vehicular environments bound to the road topology with different traffic light conditions by tuning the vehicular environment-dependent fuzzy membership functions. Hence, our HIFS scheme offered more suitable performance than all compared methods under various traffic light circumstances.

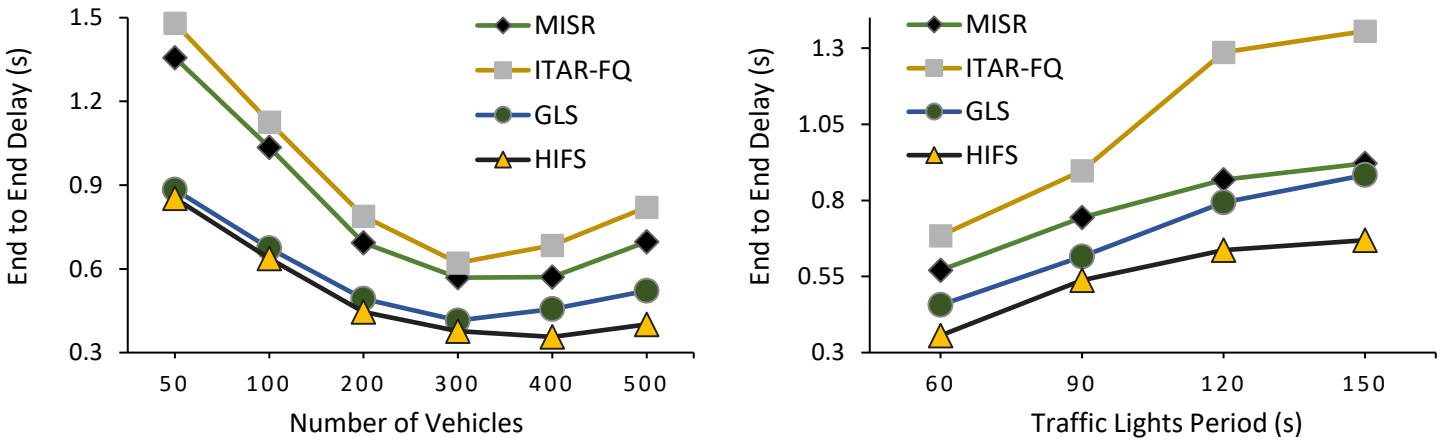


(a) different numbers of vehicles **(b) various traffic lights period**

Figure 8. Packet delivery ratio

4.4.2. Average end-to-end delay

The effect of vehicle densities and various traffic lights duration on the average end-to-end latency is shown in Figure 9. Figure 9(a) indicates the advantages of our HIFS proposed over other methods in terms of average delay as a function of vehicle density. The end-to-end delay has two different behaviors in various vehicle densities. Initially, it decreased for all compared schemes at densities of 50 to 300 vehicles, then increased at densities from 300 to 500. At low densities, trapping nodes into the local maxima problem has occurred more. Consequently, in a scattered environment, the end-to-end delay is more. Node's competition in the channel and passing more hop counts, besides higher congestion level, are the main reasons for the increased average delay at high vehicle densities (e.g., from 300 to 500 vehicles). Figure 9(a) shows that our HIFS scheme performs significantly better than the MISR and ITAR-FQ methods, especially at low densities. Moreover, our HIFS scheme has the minimum delay at high densities over the traditional intersection-based routing schemes for the following reasons: 1) Using buffering delay and limited bandwidth in the forwarding selection strategy; 2) Considering traffic light effects in the intersection selection strategy; 3) Having the global visibility of the vehicular environments. The ITAR-FQ method has increased hop counts by considering bidirectional vehicles and limited visibility in the next-hop node selection. Broadcasting of CP packets, especially at high densities, has further increased latency in the ITAR-FQ scheme. Greedily choosing a high-density path, increases the processing, transmission, and buffering delays due to high packet congestion and node competition. The GLS and MISR methods suffer from this problem. In addition, lack of consideration of the vehicles waiting behind red-light traffic has augmented the increased delay in the GLS scheme at high densities. Figure 9(b) compares end-to-end latency for all compared routing schemes at various traffic light durations. According to Figure 9(b), the latency increases in all routing methods by increasing the traffic lights period. The GLS and ITAR-FQ methods have caused further delays due to the increased unintended hop counts in the data routing process. It is due to the inability of these methods to avoid selecting intersections with long waiting queues of vehicles behind red traffic lights. The obtained results in Figure 9(b) show that the data received at the destination in our HIFS scheme has experienced lower delay than in other methods. This is because our HIFS scheme can avoid passing the paths with more hop counts by considering traffic light effects in selecting intersections.

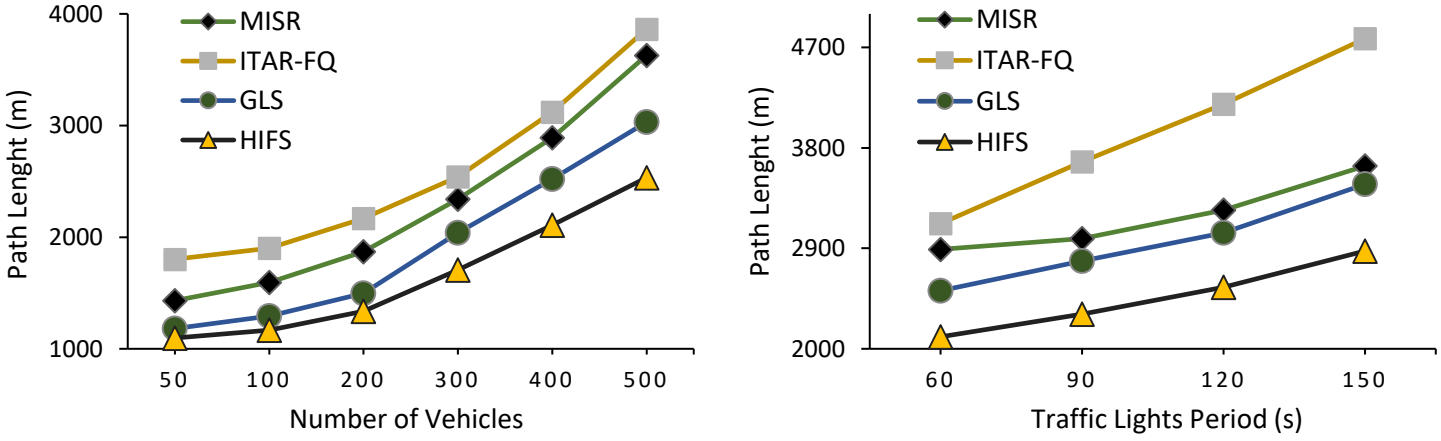


(a) different numbers of vehicles **(b) various traffic lights period**

Figure 9. End-to-end delay

4.4.3. Path length

Figure 10 depicts the path length for the different numbers of vehicles and various traffic lights duration in all compared schemes. Increasing the path length can be affected by increased hop counts. Increasing vehicle densities increases path length for all schemes (see Figure 10(a)). Because of the increase in density, forwarding vehicles are more likely to be closer. The ITAR-FQ method has caused a further increase in path length by considering densities in both directions when selecting the next-hop nodes. The MISR method also increased the path length, regardless of the distance of intersections in its intersection selection mechanism. The GLS and HIFS schemes, considering the global traffic information, have reduced the path length compared to the MISR and ITAR-FQ methods. The lack of consideration of traffic lights effects in the GLS method has led to selecting areas with a longer duration of remaining red lights traffic. Consequently, the path length traveled by data packets increased. The adaptability of our scheme in determining intersections by taking into account the impact of traffic lights has led to performing better than the GLS scheme. By increasing vehicle densities due to the creation of longer queues behind the red lights, the difference between our HIFS method and the GLS scheme has increased. The influence of various traffic light duration on the path length for the ITAR-FQ, MISR, GLS protocols, and our proposed HIFS scheme is assessed in Figure 10(b). Forming longer queues of vehicles waiting behind red traffic lights is the leading cause of increasing the traveled path length in the various traffic lights duration. The GLS scheme prioritized high-density areas without considering the traffic light effects, resulting in more path length than our method. To avoid jeopardizing the next-hop node selection process and interrupting network connectivity, the proposed method, in addition to choosing intersections with proper connections, bypasses the intersections with long queues formed behind red lights.



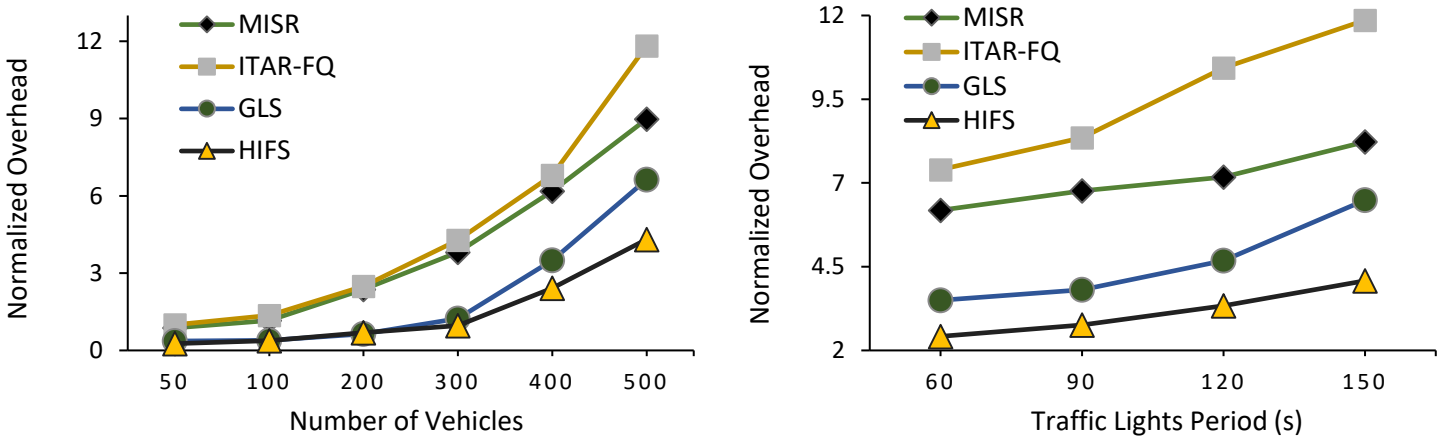
(a) different numbers of vehicles **(b) various traffic lights period**

Figure 10. Path length

4.4.4. Normalized routing overhead

Figure 11 shows the communication overhead for all compared schemes at different vehicle densities and various traffic lights duration. Figure 11(a) reveals that the communication overhead has increased by increasing vehicle densities. The increased demands for more communications have caused more control messages. The ITAR-FQ protocol has highest routing overhead due to the broadcasting of CP packets. The SDN-based methods resulted in less overhead due to the provided overall visibility. Because by developing routing sight and

specifying routes with higher stability, broadcasting new control packets requiring new path discovery is reduced. Our method's adaptability in dealing with uncertainty and ambiguity by considering vehicular environment attributes has reduced route failure and thus diminished requests to discover new routes compared to the GLS method. Figure 11(b) shows the effect of various traffic lights' duration on routing overhead. Since the number of vehicles is fixed, the increased overhead, in this case, is due to the reduction in the packet delivery ratio of routing schemes. The proposed method performs better than other methods, recalling that routing overhead relies on the packet delivery ratio.



(a) different numbers of vehicles **(b) various traffic lights period**

Figure 11. Normalized overhead

4.4.5. Discussion

The average performance enhancement of our HIFS scheme for all performance metrics compared to other methods in various densities and traffic lights duration are outlined in Tables 5 and 6. The improvement enhancement reported in Table 5 shows that our HIFS scheme performed better than the other compared methods in all performance metrics for various densities. Moreover, obtained gains shown in Table 6 reveal that the differences between our HIFS scheme with the ITAR-FQ and GLS methods have increased by increasing traffic light duration. Our HIFS scheme has dealt with the ambiguity and uncertainties of the vehicular environment by using the fuzzy reinforcement learning algorithm for tuning output membership functions based on the time-varying traffic light conditions. Therefore, besides proving the shortcomings of traditional intersection-based routing methods, our scheme outperformed all performance metrics compared to the SDN-enabled vehicular routing scheme.

Table 5. Average performance gains of HIFS for different vehicle densities

Metric / Scheme	PDR (%)	AE2ED (%)	PL (%)	NRO (%)
MISR	51.42	37.59	27.67	61.37
ITAR-FQ	48.61	44.37	35.38	67.44
GLS	6.64	10.8	13.94	29.23

Table 6. Average performance gains of HIFS for various traffic lights duration

Metric / Scheme	PDR (%)	AE2ED (%)	PL (%)	NRO (%)
MISR	46.09	29.11	22.83	55.6
ITAR-FQ	60.65	47.91	37.56	66.92
GLS	10.92	19.97	16.7	31.87

5. Conclusion

A hierarchical SDN-enabled vehicular intersection-based routing scheme was proposed in this paper by complete accessibility to vehicular environment information. In our scheme, candidate intersections were initially prioritized by jointly considering fuzzy values of curve intersections distance, predicting the number of nodes moving towards candidate intersections, and estimating the delay between the two intersections. Fuzzy results chose a sequence of intersections with maximum utility from the source to the destination intersection. Then, in the selected sequence intersections, a relay sequence with maximum utility was chosen by considering multi fuzzy factors, including Euclidean distance, residual bandwidth, congestion, and angular orientation, to participate in the data forwarding process. A reinforcement learning algorithm at both levels was used to tune the routing policies depending on the time-varying conditions. At the intersection selection level, the effect of road constraints, including traffic lights, was employed to tune the output membership functions. Stability, reputation index, and density were also utilized to adjust the output membership function of the next-hop nodes selection policy. Extensive simulations were performed in an urban scenario using NS-3 and SUMO. The obtained results show that our proposed strategy has led to significant improvement in terms of all routing performance over the state-of-the-art schemes. Therefore, the effectiveness of considering road structures and traffic light conditions in our routing decisions was confirmed. Finally, our strategy can be extended to various urban scenarios in our near future work.

Declarations

*Funding: No funding was received
*Conflicts of interest/Competing interests: There is no conflict of interest
*Availability of data and material: 'Not applicable'
*Authors' contributions: All authors contributed equally to this manuscript
*Acknowledgements: 'Not applicable'
*Consent to participate: 'Not applicable'
*Consent for publication: 'Not applicable'
*Ethics approval: The paper is original, and any other publishing house is not considering it for publication. The paper reflects the author's own research and analysis in a truthful and complete manner. All sources used are properly disclosed (correct citation).

References

[1] F. Cunha, L Villas, A. Boukerche, G. Maia, A. Viana, R. A. F. Mini, and A. A. F. Loureiro, Data Communication in VANETs: Survey, Applications and Challenges, Ad hoc Networks, 44 (2016) 90-103.

[2] Md. M. Islam, M. T. R. Khan, M. M. Saad, and D. Kim, Software-defined vehicular network (SDVN): A survey on architecture and routing, Journal of Systems Architecture, (2020) 101961.

[3] M. Lee, T. Atkison, VANET Applications: Past, Present, and Future, Vehicular Communications, 28, (2021) 100310, 2021.

[4] NN. Qadri, M. Fleury, M. Altaf, and M. Ghanbari, Multi-source video streaming in a wireless vehicular ad hoc network, IET communications, 4 (11) (2010) 1300-1311.

[5] A. Srivastava, A. Prakash, and R. Tripathi, Location based routing protocols in VANET: Issues and existing solutions, Vehicular Communications, 23 (2020) 100231.

[6] O. S. Al-Heety Z. Zakaria, M. Ismail, Mohammed, M. Shakir, S. Alani, and H. Alsariera, A comprehensive survey: Benefits, Services, Recent works, Challenges, Security and Use cases for SDN-VANET, IEEE Access, 8 (2020) 91028-91047.

[7] M. Naderi, F. Zargari, V. Sadatpour, and M. Ghanbari, A 3-Parameter Routing Cost Function for Improving Opportunistic Routing Performance in VANETs, Wireless Personal Communications, 97 (1) (2017) 1-15.

[8] T. Darwish, K. A. Bakar, Traffic aware routing in vehicular ad hoc networks: Characteristics and challenges, Telecommunication Systems, 61 (3) (2016) 489-513.

[9] T. Darwish, K. A. Bakar, Lightweight Intersection-based Traffic Aware Routing in Urban Vehicular Networks, Computer Communications, 87 (2016) 60-75

[10] I. T-A-Halim, H. M-A-Fahmy, Prediction-based Protocols for Vehicular Ad Hoc Networks: Survey and Taxonomy, Computer Networks, 130 (2018) 34-50.

[11] W. B. Jaballah, M. Conti, and Ch. Lal, Software-Defined VANETs: Benefits, Challenges, and Future Directions, arXiv preprint arXiv:1904.04577, 2019.

[12] A. J. Kadhim, S. H. Seno, Energy-Efficient Multicast Routing Protocol based on SDN and Fog Computing for Vehicular Networks, Ad Hoc Networks, 85 (2019) 68-81.

[13] L. Zhao, W. Zhao, A. Hawbani, A. Y. Al-Dubai, G. Min, A. Y. Zomaya, and C. Gong, Novel Online Sequential Learning-Based Adaptive Routing for Edge Software-Defined Vehicular Networks, IEEE Transactions Wireless communications, 20 (5) (2021) 2991-3004.

[14] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, A. Baz, and H. Alhakami, SEARCH: An SDN-enabled Approach for Vehicle Path-Planning, IEEE Transactions on Vehicular Technology, 69 (12) (2020) 14523-14536.

[15] N. Noorani, S. A. H. Seno, SDN and fog computing-based switchable routing using path stability estimation for vehicular ad hoc networks, Peer-to-Peer Networking and Applications, 13 (3) (2020) 948-964.

[16] L. Zhao, Z. Bi, M. Lin, A. Hawbani, J. Shi, and Y. Guan, An intelligent fuzzy-based routing scheme for software-defined vehicular networks, Computer Networks, 187 (2021) 107837.

[17] M. T. Abbas, A. Muhammad, and W-C. Song, SD-IoV: SDN enabled routing for internet of vehicles in road-aware approach, Journal of Ambient Intelligence and Humanized Computing, 11 (3) (2020) 1265-1280.

[18] D. K. N. Venkatramana, Sh. B. Srikantaiah, and J. Moodabidri, SCGRP: Sdn-enabled connectivity-aware geographical routing protocol of VANETs for urban environment, IET Networks, 6 (5) (2017) 102–111.

[19] D. Kong, G. Zhang, Ant colony algorithm-based routing protocol in software defined vehicular networks, in: Proceedings of the 2020 the 4th International Conference on Innovation in Artificial Intelligence, ICIAI 2020, Association for Computing Machinery, New York, NY, USA, 2020, pp. 200–204.

[20] Y. Gao, Zh. Zhang, D. Zhao, Y. Zhang, and T. Luo, A Hierarchical Routing Scheme with Load Balancing in Software Defined Vehicular Ad Hoc Networks, IEEE Access, 6 (2018) 73774-73785.

[21] M.Chahal, S. Harit, Network selection and data dissemination in heterogeneous software-defined vehicular network, Computer Networks, 161 (2019) 32-44.

[22] M. Balta, İ. zçelik, A 3-stage fuzzy-decision tree model for traffic signal optimization in urban city via a SDN based VANET architecture, Future Generation Computer Systems, 104 (2020) 142-158.

[23] H. Hagras, Type-2 FLCs: A new generation of fuzzy controllers, IEEE Computational Intelligence Magazine, 2 (1) (2007) 30-43.

[24] M. R. Mardani, M. Ghanbari, Robust resource allocation scheme under channel uncertainties for LTE-A systems, Wireless Networks, 25 (3) (2018) 1313-1325.

[25] Q. Ding, B. Sun, and X. Zhang, A Traffic-Light-Aware Routing Protocol Based on Street Connectivity for Urban Vehicular Ad Hoc Networks, IEEE Communications Letters, 20 (8) (2016) 1635-1638.

[26] J-J. Chang, Y-H. Li, W. Liao, and I-CH. Chang, Intersection-based routing for urban vehicular communications with traffic-light considerations, IEEE Wireless Communications, 19 (1) (2012) 82-88.

[27] Y. Xia, X. Qin, B. Liu, and P. Zhang, A greedy traffic light and queue aware routing protocol for urban VANETs, China Communications, 15 (7) (2018) 77-87.

[28] Sh. Zhou, D. Li, Q. Tang, Y. Fu, Ch. Guo, and X. Chen, Multiple intersection selection routing protocol based on road section connectivity probability for urban VANETs, Computer Communications, 177 (2021) 255-264.

[29] Zh. Cao, B. N. Silva, M. Diyan, J. Li and K. Han, Intersection Routing Based on Fuzzy Multi-Factor Decision for VANETs, Applied Sciences, 10 (18) (2020) 6613.

[30] Zh. Cao, Z. Fan and J. Kim, Intersection-Based Routing with Fuzzy Multi-Factor Decision for VANETs, Applied Sciences, 11 (16) (2021) 7304.

[31] Y. He, R. He, and C. Yu, Intersection-based Traffic-Aware Routing with Fuzzy Q-learning for Urban VANETs, 2021 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), pp. 511-515, 2021.

[32] A. Debnath, H. Basumatary, M, Dhar, M. K. Debbarma, and B. K. Bhattacharyya, Fuzzy logic-based VANET routing method to increase the QoS by considering the dynamic nature of vehicles, computing, 103 (7) (2021) 1391-1415.

[33] M. Naderi, F. Zargari, and M. Ghanbari, Adaptive beacon broadcast in opportunistic routing for VANETs, Ad Hoc Networks, 86 (2019) 119-130.

[34] O. Alzamzami, I. Mahgoub, Link utility aware geographic routing for urban VANETs using two-hop neighbor information, Ad Hoc Networks, 106 (2020) 102213.

[35] L. Luo, L. Sheng, H. Yu, and G. Sun, Intersection-Based V2X Routing via Reinforcement Learning in Vehicular Ad Hoc Networks, IEEE Transactions on Intelligent Transportation Systems, 2021.

[36] A. Srivastava, A. Prakash, R. Tripathi, Fuzzy-based beaconless probabilistic broadcasting for information dissemination in urban VANET, Ad Hoc Networks, 108 (2020) 102285.

[37] T-T. Huynh, A-Vu. Dinh-Duc, and C-H. Tran, Delay-Constrained Energy-Efficient Cluster-based Multi-Hop Routing in Wireless Sensor Networks, Journal of Communications and Networks, 18 (4) (2016) 580-588.

[38] I. Tal, G. M. Muntean, Towards reasoning vehicles: A survey of fuzzy logic-based solutions in vehicular networks, ACM Computing Surveys (CSUR), 50 (6) (2017) 1–37.

[39] B-S. Kima, S. Ullahb, K. H. Kimc, B-S. Rohd, J-H. Hamd, and K-I. Kima, an enhanced geographical routing protocol based on multi criteria decision making method in mobile ad-hoc networks, Ad Hoc Networks, 103 (2020) 102157.

[40] V. Derhami, V. Majd, and M. Ahmadabadi, Fuzzy Sarsa learning and the proof of existence of its stationary points, Asian Journal of Control, 10 (5) (2008) 535–549.

[41] T. Wu, P. Zhou, K. Liu, Y. Yuan, X. Wang, H. Huang, and D. O. Wu, Multi-Agent Deep Reinforcement Learning for Urban Traffic Light Control in Vehicular Networks, IEEE Transactions on Vehicular Technology, 69 (8) (2020) 8243-8256.

[42] J. Rak, LLA: A New Anypath Routing Scheme Providing Long Path Lifetime in VANETs, IEEE communications letters, 18 (2) (2013) 281-284.

[43] S. Zemouri, S. Djahel, and J. Murphy, An Altruistic Prediction-Based Congestion Control for Strict Beaconing Requirements in Urban VANETs, IEEE Transactions on Systems, Man, and Cybernetics Systems, 49 (12) (2018) 2582-2597.

[44] B. Dupont, Improvements in VANET Simulator in NS-3, Masters Project, Department of Computer Science, Old Dominion University, 2011.

[45] D. Krajzewicz, G. Hertkorn, C. Rossel, and P. Wagner, SUMO (Simulation of Urban MObility), An open-source traffic simulation, in Proc 4th MESM, Sharjah, UAE, Sep. 2002, pp. 183–187.

[46] H. Saleet, O. Basir, R. Langar, and R. Boutaba, Region-Based Location-Service-Management Protocol for VANETs, IEEE Transactions on Vehicular Technology, 59 (2) (2009) 917-931.