QoS-guaranteed Routing Protocol for Intelligent Transport System: A Hybrid Approach

Xiangyu Ren, Lin Cai, Pooria Seyed Eftetahi, Miguel Angel Verdi

***Abstract*—As one of the key applications in future vehicular networks, extended sensor sharing (ESS) requires stringent quality-of-service (QoS) for disseminating sensed information to multiple vehicles under high mobility. Ensuring QoS for ESS is challenging due to network dynamics. To address this issue, we propose a novel network architecture with the ability to tune network control functions via function modularization and decoupling control and data planes, where a protocol control agent (PCA) is designed to take in service requirements and network measurements and generate corresponding control functions. Based on the architecture, a QoS-guaranteed clustering and routing protocol (QCRP) is proposed to make routing decisions while adapting to network changes. Specifically, QCRP uses global network information for network clustering to ensure network connectivity and compute optimal routing paths to satisfy the QoS requirements. To reduce frequent topology updates while avoiding network disconnection, a predictive clustering algorithm is proposed to adjust the global topology update period according to network mobility prediction. Within each cluster, a QoS-guaranteed routing algorithm is adopted for route planning. In addition, QCRP enables re-routing at each relay vehicle based on local network observations to quickly respond to network topology changes caused by mobility. We conduct extensive simulations to evaluate the proposed routing protocol using traffic traces of different densities in a highway scenario and show our solution is the first to achieve end-to-end QoS guarantee for ESS applications in vehicular networks.**

***Index Terms*—QoS guaranteed routing, Vehicular networks, Extended sensor sharing**

I. Introduction

6G is anticipated to support many new applications where multiple end-users are involved in information exchange. For example, for real-time monitoring and control in the digital twin, avatar interactions in the metaverse, and enhanced vehicle-to-everything (V2X) communications in advanced driving [1], [2]. Extended sensor sharing (ESS) is a promising V2X application requiring frequent information dissemination in a coverage with guaranteed quality-of-service (QoS) [3]. By sharing the sensor information, vehicles can enhance the perception of the environment to improve the driving experience, road safety, and facilitate other V2X applications.

ESS is difficult to support due to the following reasons. First, ESS requires a wide range of QoS support for transmitting various types of data, e.g., it requires $3 \sim 100$ ms end-to-end (E2E) latency, $10 \sim 1000$ Mbps data rate, $90 \sim 99.99$ % reliability, and $10 \sim 100$ meters coverage [4]. Second, high mobility causes frequent changes in network topology and channel variations, which further introduces higher topology control overhead, longer E2E delay, and higher link failure probability [5], [6]. Third, sharing the same data among multiple end-users requires high resource efficiency. Due to the shared nature of the wireless medium, high interference impedes network performance [7], [8]. Last but not least, as shared information is required by multiple end users within a certain range, using unicast (one-to-one) service mode is uneconomical while broadcast (one-to-all) mode causes high information redundancy in the network [9].

Vehicles and roadside infrastructure involved in the ESS system naturally form a network. An efficient routing algorithm is desired to enable data sharing with guaranteed QoS for multiple end-users subscribed to the same ESS session. While there are few works proposed specifically for ESS applications, in the past decades, many routing solutions have been proposed for general V2X applications [10], [11]. A delay-aware grid-based geographic routing (DGGR) protocol was proposed in [12] to address the local maxima and data congestion issues in a distributed manner. The target region was first divided into grid zones, based on which, a road weight evaluation (RWE) algorithm was developed to determine the routing path among the grid zones. More recently, the software-defined vehicular network (SDVN) has been introduced, which integrates the concept of software-defined network (SDN) into vehicular networks for higher flexibility and programmability [13], [14]. Several routing algorithms were developed based on the SDVN architecture. [15] proposed a centralized cluster-based routing algorithm to trade-off between bandwidth cost and E2E delay. In [15], both vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications were considered to find the optimal routing strategy.

However, these solutions are insufficient to support ESS. First, a fully centralized control solution requires frequent topology updates to maintain performance leading to high control overhead. Second, centralized routing protocols cannot quickly respond to short-term network changes due to mobility. Third, it is very difficult if not possible for distributed routing solutions to guarantee QoS requirements due to a lack of global coordination.

To bridge the gap, we present a QoS-guaranteed clustering and routing protocol (QCRP) following the SET protocol architecture we proposed in our previous work [16]. A protocol control agent (PCA) is designed to leverage global and local network information to make control decisions[1]. It uses the global network information to find the optimal routing path to satisfy the E2E QoS requirements and the local network information to fine-tune control decisions to adapt to net-

[1] Thus, two types of PCA, global PCA (GPCA) and local PCA (LPCA) are defined in QCRP.

work dynamics. Moreover, topology control and re-routing are adopted to reduce control overhead and maintain network performance over time.

Specifically, we first compute the inter-vehicle reference distances based on the QoS requirements, which yield the upper and lower bounds for clustering and path calculation. Next, the network is clustered by the reference distance to ensure network connectivity. To mitigate the impact of network disconnection caused by network mobility, a proactive clustering algorithm based on the Long-Short-Term-Memory (LSTM) is adopted in QCRP. GPCA, i.e., the PCA with global network information, predicts the number of time slots before the network disconnection occurs, i.e., the inter-vehicle distance exceeds the reference distance upper bound, using historical vehicle mobility traces. GPCA proactively determines the global topology update period based on the prediction results. For route planning, we consider a hybrid usage of V2V and V2I links for intra and inter-cluster communications. A QoS-guaranteed routing algorithm (QGR) is developed to find the optimal routing path within each cluster. Since topology variations may lead to network performance degradation, a distributed re-routing algorithm is developed. Within the global topology update period, the LPCA deployed at each relay vehicle along the original routing path re-selects the next hop based on the residual link lifetime estimated using local network observations. In this case, the routing paths are updated timely without the need for global topology updates and path calculations to maintain the required QoS.

In summary, the contributions of this paper are three-fold:

- We introduce a novel flexible protocol architecture to provide guaranteed QoS for various applications while adapting to network dynamics. In our approach, a protocol control agent is designed to perform network control based on the collection of multi-layer information such as service requirements, network topology, and link condition, to support the stringent QoS requirements and adapt to network dynamics.
- We propose a novel protocol named QCRP based on the novel architecture. QCRP can support ESS application in vehicular networks which requires stringent E2E QoS support for multiple receivers in a high mobility environment. To address the challenge, two types of PCA, namely, GPCA and LPCA are designed to perform global and local network control, respectively.
- We show that both global control and local control are essential to maintaining the E2E QoS performance over time via extensive simulation experiments, where the former enables control optimality while the latter contributes to network dynamics adaptation.

The rest of the paper is organized as follows. We introduce the related work in Section II. In Section III, we explain the proposed network architecture and system model. Next, we formulate our problems in Section IV and propose the corresponding solutions in Section V. In Section VI, we evaluate our work with extensive simulations followed by conclusions and further research issues in Section VII.

II. Related Work

A. Distributed solutions

The routing protocols that compute routing paths in a distributed fashion without the need for a global controller are referred to as distributed solutions. Distributed solutions have the benefits of low overhead and high adaptiveness to topology changes but suffer from frequent link outage issues in highly mobile environments [17]. To address this issue, many solutions have been proposed in the literature. They can be divided into several sub-categories by their methods. Location-based methods adopt geographic locations for route selection. Most location-based methods are developed based on the Greedy Perimeter Stateless Routing (GPSR) protocol [10], [18], for example, an enhanced GPSR protocol was proposed in [19] to reduce link outage and improve throughput by stabilizing the routing path. A connectivity-aware routing (CAR) protocol [20] was proposed to address the high delay issue caused by the carry-and-forward mechanism, where the route with less network disconnection probability was selected for packet transmission. Broadcast-based solutions improve network performance by suppressing the broadcast messages in data dissemination. In [21], a distributed vehicular broadcast (DV-CAST) protocol relying only on the local topology was proposed to address the broadcast storm and low connectivity issues with a small overhead. Later, in [22], a velocity and position-based broadcast suppression protocol (VP-CAST) was proposed for VANETs. The infrastructure-based methods utilize roadside infrastructures to improve network connectivity while satisfying QoS requirements [23].

B. Centralized solutions

Centralized solutions require the knowledge of global network topology before route calculation. Conventional routing protocols such as Optimized Link State Routing Protocol (OLSR) and its variants are initially designed for mobile ad hoc networks and later extended to vehicular networks [24]–[26]. However, they suffer from high control overhead for topology updates and long convergence delay [27]. Later, SDN attracted great attention which decouples the control plane and data plane. A new paradigm named SDVN leveraging SDN in vehicular networks has been explored in literature [28], [29] where the SDN controller is deployed at the edge cloud or road-side units (RSUs) for centralized route calculation and faster response to topology changes. More recently, with the development of edge computing and learning algorithms, researchers propose to adopt learning algorithms at mobile edge [30]–[32]. In [33], an artificial neural network-based vehicle mobility prediction model was developed to assist route calculation at the SDN controller. [34] proposed a greedy routing algorithm based on Graph Convolutional Network using the SDVN architecture to improve packet delivery ratio and delay.

While these works have shown promising performance in certain use cases, they are insufficient for ESS. The distributed solutions employ a best-effort strategy for path discovery and fail to guarantee any QoS requirements due to a lack of global network knowledge. The centralized solutions require global knowledge for path calculation consuming high control overhead and time to synchronize and update frequently to maintain the network performance. The cost can grow exponentially with the increase of ESS subscribers and network size which are uneconomical in practice. Therefore, a scalable, adaptive, and low cost routing protocol is desired to bridge the gap.

III. System Model

TABLE I: Notations and definitions

Symbol	Definitions
u_i	Member vehicle i
$l_{u_i},\ v_{u_i}$	Location and speed of u_i
$\mathcal{U}$	Sets of member vehicles
$\mathcal{M}$	Sets of mobility information
d_{u_i}	Inter-vehicle distance of u_i
$\mathcal{Q}$	Sets of QoS requirements
$B_w,\ B_v$	Bandwidth allocated for V2I and V2V links
$P_w,\ P_v$	Transmission power for V2I link and V2V links
$R_w,\ R_v$	Theoratical link capacity for V2I link and V2V links
$d_{\min},\ d_{\max}$	Derived transmission distance lower and upper bounds
$\mathcal{C}$	Sets of clusters
$\boldsymbol{CH}$	Sets of cluster heads
G_k	Trellis graph representation of cluster k
$\mathcal{P}_k$	Routing path for cluster k
GPCA	Global protocol control agent
LPCA	Local protocol control agent

In this section, we first introduce the network architecture followed by a detailed explanation of the assumptions and network settings adopted in our work. Next, we discuss the theoretical models established for our problem formulation. To avoid confusion, we summarize the frequently used symbols in Table I.

A. Network Architecture

We develop QCRP following the SET protocol architecture [16], where the control decisions are generated using cross-layer information. Each network entity deploys a PCA to configure protocols based on the observed network information and application requirements as shown in the left figure in Fig. 1. For each packet/flow with specific QoS requirements (e.g., delay, data rate, and reliability), PCA determines a control decision based on the current network conditions. In addition, PCA periodically evaluates and updates its control strategy to maintain QoS performance and reports errors to the application if the network cannot support the requirements.

We show the proposed network architecture in Fig. 1 where two types of PCAs are defined, namely, GPCA and LPCA, to collect and respond to global and local network information, respectively. While the global network knowledge is essential for computing the optimal routing paths, it causes high overhead and delay to update, especially in vehicular networks, due to high network dynamics. On the other hand, local network knowledge reflects the network conditions in real time and is less costly to update in the neighbourhood. GPCA is deployed at the central controller to determine the optimal routing path for guaranteeing the E2E service requirements using the global network knowledge. LPCA is deployed in each network entity along the routing path which is responsible for adjusting GPCA's control decisions using real-time local network observations.

B. Assumptions and Network Settings

In our scenario, we refer to all vehicles that subscribed to one ESS session as member vehicles and those selected to relay data packets as relay vehicles. The source vehicle is also considered a relay vehicle. We assume each vehicle has a unique identification number and periodically uploads its mobility information (e.g., locations, speed, link conditions, etc) to the base station (BS) via the physical uplink control channel (PUCCH) to construct a global network topology. GPCA is deployed at the BS side. We define five stages in a complete E2E data transmission process in the proposed network architecture, namely, service request, global control, control decision release, local control, and control adjustment as shown in the right figure of Fig. 1.

In the first stage, the ESS application initializes the service by sending its QoS requirements[2] to the BS. Next, given the global network topology and ESS service requirements, GPCA computes the optimal control decisions, i.e., global network topology update period and routing path, in the second stage and broadcasts it to the network using the physical downlink control channel (PDCCH) in the third stage. While the source vehicle starts data transmission after the third stage, the LPCA deployed at each relay vehicle performs local control to maintain the network performance by collecting the mobility and link information of its neighbours and adjusting the routing paths when link breakage occurs as indicated by stage four and five. Similar to [15], [33], we assume all vehicles are equipped with transceivers for V2V and V2I communications using the physical sidelink shared channel (PSSCH) and physical uplink shared channel (PUSCH) and physical downlink shared channel (PDSCH) for data transmission, respectively. Member vehicles periodically broadcast their mobility information to their one-hop relay vehicles via beacon messages using the physical sidelink control channel (PSCCH). This setting addresses frequent topology variations via V2V communications and ensures coverage via V2I communications.

C. Network Model

We consider a highway scenario as depicted in the right side of Fig. 1, where vehicles traverse a high-speed highway and fall within the coverage range of a solitary BS. Our analysis specifically focuses on a single source case, wherein only one vehicle shares data with other vehicles within the network.

[2] For example, E2E delay, data rate, reliability, coverage, etc.

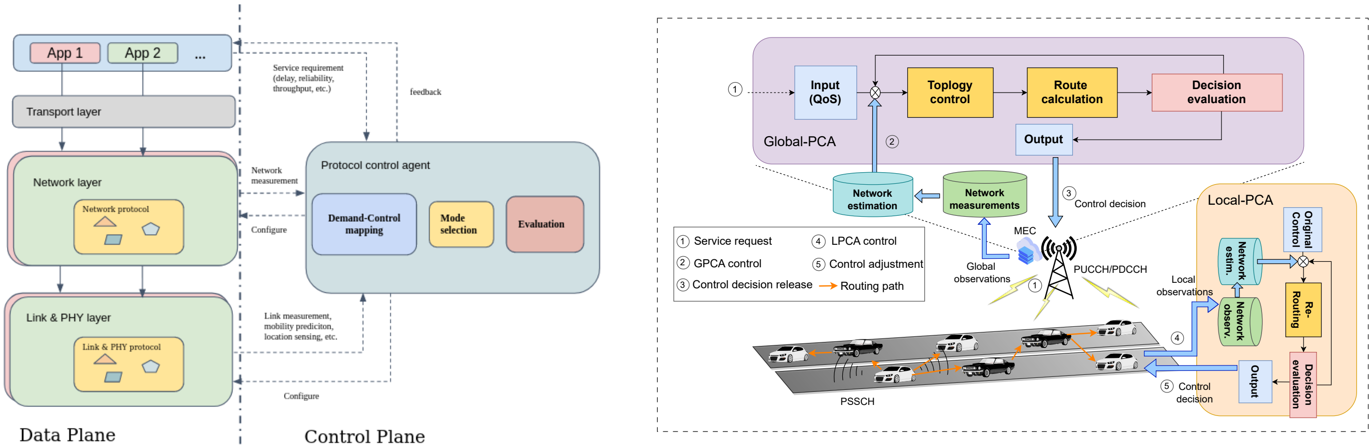


Fig. 1: Left: QCRP protocol architecture; Right: network architecture for ESS applications in vehicular networks.

Denote $\mathcal{U}$ as the complete set of member vehicles (we use vehicles for short in the rest of the paper). In total, there are U vehicles. Each vehicle uploads its mobility information denoted by $M_{u_i} = \{l_{u_i}, v_{u_i}\}$, $u_i \in \mathcal{U}$ to the BS, where $l_{u_i} = (x_{u_i}, y_{u_i})$ is the location of vehicle u_i and v_{u_i} is its speed. Thus, the global network knowledge can be represented as $\mathcal{M} = \{M_{u_1}, M_{u_2}, \cdots, M_{u_N}\}$. On receiving $\mathcal{M}$, GPCA computes the inter-vehicle distance:

$$d_{u_i} = \sqrt{(x_{u_i} - x_{u_j})^2 + (y_{u_i} - y_{u_j})^2},\ u_i, u_j \in \mathcal{U}. \quad (1)$$

In addition, we denote the QoS requirements by a tuple $\mathcal{Q} = \{\theta_{\text{rel}}, \theta_{\text{rate}}, \theta_{\text{del}}, \theta_{\text{cov}}\}$, where θ_{rel} is the E2E reliability requirement, θ_{rate} is the data rate requirement, θ_{del} is the E2E delay requirement, and θ_{cov} is the coverage requirement. $\mathcal{Q}$ is known by all vehicles and the same for each member vehicle subscribing for the same ESS session.

D. Channel Model

The uplink and downlink average data rate of the V2I link can be estimated by

$$R_w = \eta \cdot B_w \cdot \log_2 \left(1 + \frac{P_w}{N_0} \delta d_w^{-\alpha} |h|^2\right),\ w \in \{\text{up}, \text{dw}\}, \quad (2)$$

where $\eta \in (0, 1]$ is a constant coefficient determined by the transceiver hardware efficiency, B_w is the V2I link bandwidth, P_w is the transmission power, d_w is the uplink/downlink transmission distance, δ is the shadowing component, h is the Rayleigh-distributed fading coefficient with $\mathbb{E}(|h|^2) = 1$, N_0 is the power of additive white Gaussian noise (AWGN), and α is the path loss component. Similarly, the average data rate of a V2V link can be estimated:

$$R_v = \eta \cdot B_v \cdot \log_2 \left(1 + \frac{P_v}{N_0} \delta d_v^{-\alpha} |h|^2\right), \quad (3)$$

where B_v, P_v, and d_v are the V2V communication bandwidth, transmission power, and inter-vehicle distance, respectively. Given the data rate requirement θ_{rate}, the transmission distance constraint on data rate for V2V communication can be derived as follows[3]

$$d_{\text{rate}} \leq ((2^{\frac{\theta_{\text{rate}}}{\eta B_v}} - 1) \cdot \frac{N_0}{P_v \delta |h|^2})^{-\frac{1}{\alpha}}. \quad (4)$$

i.e., the transmission distance d_v can be d_{rate} at maximum to satisfy the data rate requirement.

E. Delay Model

There are mainly five delay components at each hop, namely, transmission delay T_t, medium access delay (or contention delay) T_c, queuing delay T_q, propagation delay, and processing delay, where the propagation and processing delay are negligible (at the level of a few microseconds). In this case, the E2E time cost of a path $\mathcal{P}$ can be calculated as

$$tc_{\mathcal{P}} = \sum_{i \in \mathcal{P}} tc_i = \sum_{i \in \mathcal{P}} T_t^i + T_c^i + T_q^i, \quad (5)$$

where tc_i refers to the time cost at each hop i along the path. Since we consider time-sensitive packet transmission, we assume the sum of medium access delay and queuing delay can be upper bounded by a constant value τ. The transmission delay can be computed by $T_t = L/R_v^i$, where L is the packet size and R_v^i is the data rate of the i-th hop[4]. Thus, the E2E time cost of a path $\mathcal{P}$ can be rewritten as

$$tc_{\mathcal{P}} = \sum_{i \in \mathcal{P}} (\frac{L}{R_v^i} + \tau_i). \quad (6)$$

Furthermore, given the E2E delay, coverage, and data rate requirements, the transmission distance constraint in terms of time cost is given by

$$d_{\text{del}} = Dist \times \frac{L + \tau_{\max} \cdot \theta_{\text{rate}}}{\theta_{\text{del}} \cdot \theta_{\text{rate}}}, \quad (7)$$

[3] Note that in practical systems, the above channel models may not be accurate due to randomness. In our design, the channel model is used by the GPCA to estimate link data rates and perform route planning.

[4] The delay model can be extended by considering variable per-hop delay as shown in [35].

where $Dist$ is the distance from the source to the furthest member vehicle and $\tau_{\max}$ is the maximum delay constant introduced by queuing delay and contention delay. In other words, the transmission distance d_v is lower bounded by d_{del} to satisfy the delay requirement.

F. Connectivity Model

Since wireless links are highly dynamic, analyzing the connectivity of each link is fundamental to achieving the QoS requirements. We use the packet delivery ratio (PDR) of a link to characterize the per-hop transmission reliability

$$P_e = (1 - p_b)^L, \tag{8}$$

where p_b is the bit error rate (BER)[5]. Similarly, the E2E reliability P_s can be derived as follows,

$$P_s = \prod_{i \in \mathcal{P}} P_e^i = \prod_{i \in \mathcal{P}} ((1 - p_b)^L)^i \geq \theta_{\text{rel}}. \tag{9}$$

Since p_b is determined by the modulation scheme and signal-to-noise ratio (SNR), it can be represented by a function of distance, i.e., $p_b = f(d)$[6]. Thus, the transmission distance constraint on reliability can be derived as follows

$$(1 - f(d))^L \geq \theta_{\text{rel}}. \tag{10}$$

d_{rel} denotes the solution to (10) when the equality sign holds.

In summary, we define d_{del} as the minimum reference distance $d_{\min}$ and $d_{\max} = \min(d_{\text{rate}}, d_{\text{rel}})$ as the maximum reference distance to facilitate topology control which characterizes the lower and upper bound of a transmission range. In this context, GPCA and LPCA determine whether the current routing plan can support the QoS requirements by comparing the distance between two relay candidates with $d_{\min}$ and $d_{\max}$. An error message will be sent to the ESS application for adjusting QoS requirement when $d_{\min} > d_{\max}$, i.e., more E2E delay time budget should be given to ensure a guaranteed performance. If there exists a non-empty relay candidate set that satisfies the transmission range, a routing path is calculated by selecting relay vehicles from the candidate set.

IV. Problem Formulation

In this section, we discuss how QCRP guarantees the E2E QoS performances in a nutshell. Three problems are formulated to obtain the optimal control strategy, namely, network clustering, routing path planning, and re-routing.

A. Network Clustering

Since not all vehicles in the network $\mathcal{U}$ can be reached using V2V links due to the transmission distance constraints imposed by the QoS requirements, GPCA divides the network into clusters where all vehicles within the same cluster are connected[7]. In this context, we propose a two-step clustering algorithm as summarized in Algorithm 1 to guarantee the connectivity of the vehicular network.

In the first step (line:3-9), GPCA computes the inter-vehicle distance $d_{u_i}(t)$ for each $u_i \in \mathcal{U}$ at time t and estimates the future inter-vehicle distance after t_0 time slots, i.e., $\hat{d}_{u_i}[t] = d_{u_i}[t+t_0]\ \forall u_i \in \mathcal{U}$. Next, GPCA divides the network into sub-networks if $\max\{d_{u_i}[t], \hat{d}_{u_i}[t]\} \geq d_{\max}$. This step guarantees the connectivity of each sub-network, where all vehicles within the same sub-network can be reached while satisfying the per-hop QoS requirement. By the end of this step, $\mathcal{U}$ is divided into K_0 sub-networks. In the second step (line:10-18), GPCA further evaluates the coverage distance of each sub-network $\text{cov}(\hat{\mathcal{U}}_k)$ and a sub-network will be clustered again if the E2E time cost exceeds the delay budget:

$$Tc_{\max} = \frac{\text{cov}(\hat{\mathcal{U}}_k)}{d_{\min}} \cdot \left(\frac{L}{\theta_{\text{rate}}} + \tau_{\max}\right) > \theta_{\text{del}},\ \mathcal{U}_k \in \hat{\boldsymbol{\mathcal{U}}}, \tag{11}$$

where $\frac{\text{cov}(\hat{\mathcal{U}}_k)}{d_{\min}}$ computes the maximum number of hops in a sub-network and $\frac{L}{\theta_{\text{rate}}} + \tau_{\max}$ is the maximum per-hop time cost. Thus, a sub-network $\hat{\mathcal{U}}_k$ can be divided into $K = \lceil \frac{\theta_{\text{del}}}{tc_{\max}} \rceil$ clusters. Next, GPCA uniformly divides $\hat{\mathcal{U}}_k$ into K clusters $\mathcal{C}_k = \{C_1, \ldots, C_K\}$ and the vehicle closest to the cluster center is selected as the cluster head (CH), i.e.,

$$\text{CH}_k = \underset{u_j \in C_k}{\arg\min}\ l_{u_j} - \sum_{u_j \in \mathcal{C}_K} \frac{l_{u_j}}{|C_k|}, \tag{12}$$

where $|C_k|$ is the total number of vehicles within C_k and $\sum_{u_j \in \mathcal{C}_K} \frac{l_{u_j}}{|C_k|}$ yields the centroid of cluster C_k. Finally, we obtain the cluster set $\boldsymbol{\mathcal{C}} = \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ and the CH set $\mathbf{CH} = \{\text{CH}_1, \text{CH}_2, \cdots, \text{CH}_K\}$. In our design, each CH is responsible for receiving data packets from the source vehicle via V2I links and distributing them within its cluster.

However, estimating the inter-vehicle distance in t_0 can be challenging due to the time-varying vehicular mobility patterns, which further affect the connectivity of the network. Therefore, we aim to develop an effective solution to predict the connectivity duration time based on the proposed clustering algorithm and proactively re-cluster the network before the cluster loses its connectivity.

B. Route Planning

We assume guaranteed QoS performance for V2I links as they use dedicated frequency bands for data transmission in a single hop. As for intra-cluster V2V communications, we focus on a multi-hop scenario considering the transmission distance constraints and coverage requirements.

We consider a single-trip data transmission in our scenario, where the sender only transmits each packet once without packet retransmission. To support the E2E QoS requirements, the available routing paths need to satisfy the following constraints: (i) Guaranteed throughput, i.e., each V2V link

[5] BER can be improved using physical-layer error correction code (ECC) such as forward error correction (FEC) [36], etc.

[6] We use the BPSK modulation scheme as an example to derive this function as shown in Appendix VIII.

[7] In this paper, we define a network is connected if there exists a path connecting any pair of vehicles while satisfying the E2E QoS requirements.

Algorithm 1 Clustering algorithm

```
Input: Network mobility information $\mathcal{M}$ and reference distances $(d_{\min}, d_{\max})$.
Output: Cluster set $\boldsymbol{\mathcal{C}}$ and cluster head set CH.
// First-step clustering
Obtain network mobility at time $t$:
for each $u_i$ in $\mathcal{U}$ do
    Compute $d_{u_i}[t]$ and $\hat{d}_{u_i}[t]$
    Split network if $\max\{d_{u_i}[t], \hat{d}_{u_i}[t]\} \geq d_{\max}$
end for
Obtain $\hat{\boldsymbol{\mathcal{U}}} = \{\hat{\mathcal{U}}_1, \hat{\mathcal{U}}_2, \cdots, \hat{\mathcal{U}}_{K_0}\}$
// Second-step clustering
for each $\hat{\mathcal{U}}_k \in \hat{\boldsymbol{\mathcal{U}}}$ do
    Compute cov($\hat{\mathcal{U}}_k$) and $K$
    Uniformly divide $\hat{\mathcal{U}}_k$ into clusters $\mathcal{C}_k = \{C_1, \ldots, C_K\}$.
    for each $C_k \in \mathcal{C}_k$ do
        Compute: $\text{CH}_k = \underset{v_j \in C_k}{\arg\min}\ l_{v_j} - \sum_{v_j \in C_k} \frac{l_{v_j}}{|C_k|}$.
    end for
end for
```

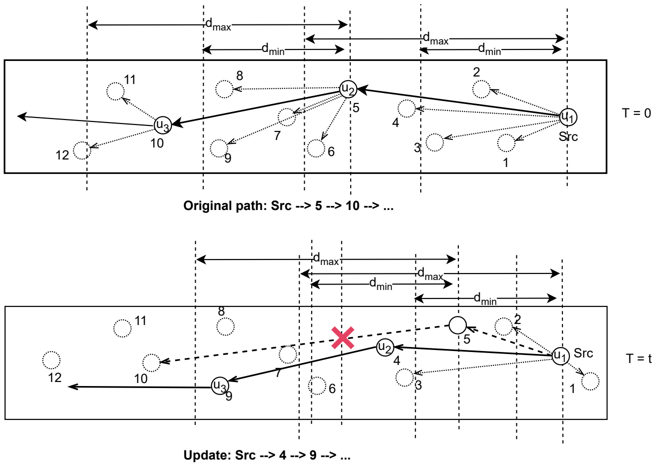


Fig. 2: In-network control. The original path fails the transmission distance constraints after t time slots due to mobility. LPCA performs re-routing to locally update the routing path.

should guarantee a minimum data rate of θ_{rate}; (ii) Guaranteed E2E reliability, i.e., the E2E PDR should be greater than θ_{rel}. While ESS applications are time-sensitive, we formulate the routing problem to minimize the E2E delay:

$$
\begin{aligned}
\text{P1:}\quad & \underset{\mathcal{P}}{\arg\min} \quad \sum_{u_i \in \mathcal{P}} \frac{L}{R_v^{(u_i)}} + \tau_{u_i} && \text{(13a)}\\
\text{s.t.}\quad & R_v = B_v \cdot \log_2\left(1 + \frac{P_v}{N_0}\delta d_v^{-\alpha}|h|^2\right) \geq \theta_{\text{rate}} && \text{(13b)}\\
& (P_v)_{\text{dB}} - PL(d_v)_{\text{dB}} - N_{\text{dB}} \geq \theta_{\text{SNR}}, && \text{(13c)}\\
& (1-p_b)^L \geq \theta_{\text{rel}}, && \text{(13d)}\\
& \prod_{i \in \mathcal{P}} ((1-p_b)^L)^i \geq \theta_{\text{rel}},. && \text{(13e)}
\end{aligned}
$$

where u_i is the relay vehicle selected at hop i, (13b) is the the per-hop data rate constraint, (13c) is the link condition constraint, and (13d) and (13e) are the per-hop and E2E reliability constraints, respectively. The solution to P1 yields the optimal routing path.

C. In-network Control

We use a simple example as shown in Fig. 2 to explain the in-network control performed by each LPCA. After the GPCA broadcasts the control decisions to the network, all relay vehicles start data transmission following the control decision. In the meantime, member vehicles periodically send beacon messages containing mobility information to their corresponding relay vehicles after packet reception. In this context, each relay vehicle can construct a one-hop local network topology.

Although GPCA provides optimal control decisions, the network topology changes over time due to mobility, leading to QoS performance degradation. To address this issue, the LPCA at each relay vehicle adjusts the original control decision by selecting a different next-hop relay based on the local network observations. For simplicity, we refer to all vehicles within the transmission range as relay candidates. Such location-based routing strategy is heavily discussed in literature [10]. However, most protocols are developed in a best-effort fashion and are insufficient to guarantee E2E performance due to the lack of global knowledge and QoS requirements. Thus, how to efficiently select each relay candidate to adapt to network dynamics while maintaining the E2E QoS guarantee remains a challenging issue.

V. QoS-guaranteed Routing Protocol

In this section, we explain QCRP in detail. First, we propose a data-driven vehicle mobility prediction model based on LSTM to facilitate network clustering in case of frequent network topology changes. Next, we discuss a QoS-guaranteed routing (QGR) algorithm employed by GPCA for route planning within each cluster. QGR essentially initializes a routing path satisfying the QoS requirements with prior knowledge of global network topology. Finally, a residual link lifetime-based relay selection algorithm is discussed. Each LPCA performs re-routing to maintain the network performance via local network observations.

A. Predictive Clustering Algorithm with Feedback Control

We assume MEC is adopted in an ESS-enabled system to execute heavy computation tasks to support GPCA. While vehicle mobility patterns are simple in a highway environment as opposed to urban settings, accurately modelling these patterns in practice remains challenging due to the inherent speed and behavioural diversity of vehicles. Furthermore, the accurate estimation of SNR and consequently, the cluster connectivity is contingent upon precise inter-vehicle distance prediction, taking into account path loss and fading/shadowing effects in the wireless channel. On the other hand, a rich set of vehicle

Algorithm 2 Predictive Clustering with Feedback Control

```
Obatain $\{\mathcal{C}, \mathbf{CH}\}$ and initialize $t_0, t_0^{\min}, t_0^{\max}$
// Mobility prediction phase
for each vehicle $u_i \in \mathcal{U}$ do
    Predict the trajectory $\mathbf{D}_{u_i}$ in $t_0$ slots:
    $\mathbf{D}_{u_i} = \{l_{u_i}(t), l_{u_i}(t+1), \ldots, l_{u_i}(t+t_0)\}$
end for
// Connectivity evaluation phase
for each cluster $C_k \in \mathcal{C}$ do
    if $\exists d_{u_i}(t+t_0') > d_{\max},\ u_i \in \mathcal{U}$ then
        Set $t_0'$ as the global upload period
    end if
end for
Broadcast $\{\mathcal{C}, \mathbf{CH}, t_0'\}$ to the network.
// Feedback control phase
if receive error message then
    $t_0 = \min\{t_0/2, t_0^{\min}\}$
else
    $t_0 = \max\{t_0 + t_0'', t_0^{\max}\}$
end if
```

mobility data is uploaded to the BS provisioning the fundamentals for data-driven prediction models. In this context, we propose a proactive clustering algorithm to dynamically determine the cluster update period using LSTM and historical vehicle mobility data[8].

		GPR	LR	SVR	LSTM
Speed	Slow	0.0008	0.0206	0.0225	**0.0001**
	Normal	0.0011	0.0205	0.0225	**0.0001**
	Fast	0.0015	0.0205	0.0227	**0.0001**

TABLE II: Mean square error (MSE) of four prediction models: Gaussian Process Regression (GPR), Linear Regression (LR), Support Vector Regression (SVR), and LSTM predicting vehicles' locations in 10 time slots with different speed.

An accurate prediction model not only helps to maintain the connectivity of each cluster but also reduces the control overhead between BS and vehicles. However, the prediction accuracy decays as the prediction period increases. While a fixed prediction period is inadaptive to the prediction performance, we propose a proactive prediction period control algorithm following the well-known Additive-Increase-Multiplicative-Decrease (AIMD) to dynamically adjust the prediction period based on the prediction results.

As shown in Algorithm 2, GPCA performs clustering at time t and predicts the mobility pattern of the network in t_0 time slots using LSTM. LSTM essentially predicts the trajectory of each vehicle in the network in time sequence, denoted by $\mathbf{D}_{u_i} = \{l_{u_i}(t), l_{u_i}(t+1), \ldots, l_{u_i}(t+t_0),\},\ \forall u_i \in \mathcal{U}$. Next, GPCA evaluates the connectivity of each cluster using the prediction results by calculating the inter-vehicle distance. A cluster is considered as disconnected if $\forall d_{u_i}(t+t_0') \geq d_{\max}$ at t_0', and $t_0' \leq t_0$. We then adopt t_0' as the upload period when all the member vehicles should upload their mobility data to the BS. Finally, the clustering results $\{\mathcal{C}, \mathbf{CH}, t_0'\}$ are broadcast to the network. After each upload period, GPCA extends its prediction period by t_0'' slots, i.e., $t_0 := t_0 + t_0''$, if no error message is reported to the BS due to disconnection. Otherwise, t_0 is reduced by half to restore its prediction accuracy. To avoid overwhelming the prediction process, we manually set the lower bound $t_0^{\min}$ and upper bound $t_0^{\max}$ for the prediction period t_0. In summary, the update process can be expressed as follows:

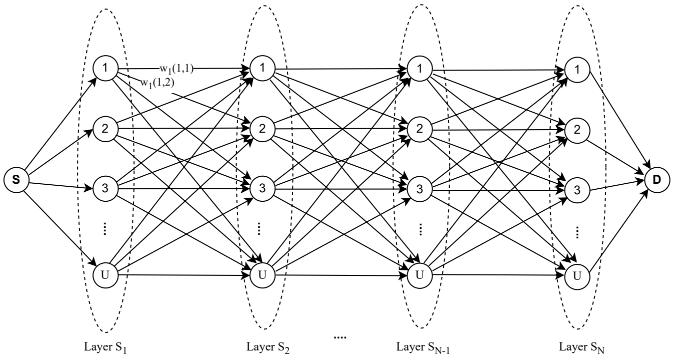


Fig. 3: Trellis graph representation

$$t_0 := \begin{cases} \max\{t_0 + t_0'', t_0^{\max}\} & \text{if } \textit{error} \text{ is false,} \\ \min\{t_0/2, t_0^{\min}\} & \text{if } \textit{error} \text{ is true.} \end{cases} \tag{14}$$

B. QoS-guaranteed routing algorithm

To better characterize the routing process, we use the Trellis graph $G =< V, N, E >$ to represent each cluster $\mathcal{C}_k$ as shown in Fig. 3. A graph consists of V nodes, E edges, and N layers (representing N hops between the source and destination nodes). Each layer S contains the complete V nodes and all nodes between two layers are fully connected with edges E. We regard the CH of each cluster as the source node and the vehicle at the two ends of each cluster as the destination nodes. Thus, two graphs can be constructed to represent a cluster, i.e., forward G_k^f and backward G_k^b, respectively[9].

Next, we explain the proposed QoS-guaranteed routing algorithm to perform route planning for each cluster as summarized in Algorithm 3. Specifically, to incorporate the QoS constraints in our network representation, we define the weight of each edge as follows

$$w_n(i,j) = \log(\bar{p}_b^{(n)} \cdot \bar{tc}^{(n)}),\ \forall i \in S_n, \forall j \in S_{n+1}, \tag{15}$$

[8] We adopt LSTM as a proof-of-concept to show the capability of the data-driven prediction model in QCRP. We compared its prediction performance with other regression models as shown in Table II. More advanced prediction model such as graph attention networks (GATs) [37], [38] can be adopted to improve prediction accuracy. In this paper, we adopt a single-layer shallow LSTM model and train offline using historical traffic data.

[9] However, the trellis graph can grow exponentially with the increased number of vehicles of each cluster, which increases the time complexity of finding the shortest path. To address this issue, we simplify the Trellis graph by merging multiple nearby vehicles as one node and randomly selecting one of them as the representation of that node.

Algorithm 3 QoS-guaranteed Routing Algorithm

```
1: Input: $\boldsymbol{\mathcal{C}}$, $\boldsymbol{CH}$, $\mathcal{Q}$, $d_{\max}$, and $(src, dst)$.
2: Output: Optimal path set $\mathcal{P} = \{\mathcal{P}_1, \cdots, \mathcal{P}_K\}$.
3: for $\mathcal{C}_k \in \boldsymbol{\mathcal{C}}$ do
4:     Construct graph $G_k = \{G_k^f, G_k^b\}$, where the edge weight is denoted by
       $w_n(i,j) = \begin{cases} \log(\bar{p}_b \cdot \bar{tc}_{i,j}) & d_{u_i} <= d_{\max}, \\ \infty & du_i > d_{\max}. \end{cases}$
5:     Compute optimal routing path for $G_k$:
       $\mathcal{P}_k = Dijkstra(G_k, src, dst)$.
6:     Evaluate $tc$ and $P_s$ for $\mathcal{P}_k$.
7:     Report error if $tc^{(k)} > \theta_{\text{del}}$ or $P_s^{(k)} < \theta_{\text{rel}}$.
8: end for
```

where $\bar{p}_b = \frac{\theta_{\text{rel}}}{p_{b_{i,j}}}$ and $\bar{tc} = \frac{tc_{i,j}}{\theta_{\text{del}}}$ are the normalized PDR and time cost between node i and node j of two adjacent layers. For instance, $w_1(1,2)$ represents the edge weight from node 1 in S_1 to node 2 in layer S_2. The benefit of such a weight metric design is that both E2E delay and reliability are considered in our graph representation thanks to the sum property of the logarithm. In addition, we set the weight to infinity if the inter-vehicle distance exceeds $d_{\max}$ to remove unsatisfactory links, i.e., $\{w_n(i,j) = \infty | d_{u_i} > d_{\max}\}$. Then, GPCA constructs the corresponding graph G_k for each cluster $C_k \in \mathcal{C}$ (Line:4). In total, the number of paths from the source node to the destination node can be derived $N_{\text{path}} = \prod_{n=1}^{N} |\mathbf{w}_n|$, where $|\mathbf{w}_n|$ is the number of non-infinity-weight edges between two layers in the graph, and there are at most V^N number of paths in a graph. Finally, GPCA computes the shortest path that minimizes the sum of the total weight from the source to the destination for each graph using Dijkstra's algorithm. The routing path $\mathcal{P}$ is then evaluated by the E2E QoS performance metrics before broadcast to the network:

$$tc(\mathcal{P}) = \sum_{n=1}^{N} tc^{(n)}, \ P_s(\mathcal{P}) = \prod_{n=1}^{N} P_e^{(n)}.$$

An error message will be reported if the current path fails to support the QoS requirements (Line:5-8), i.e.,

$$error = \begin{cases} 1 & tc(\mathcal{P}) > \theta_{\text{del}} || P_s(\mathcal{P}) < \theta_{\text{rel}}, \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

In addition, we assume the member vehicles covered between two consecutive relay vehicles can receive the same packet simultaneously during transmission. In this context, QGR essentially yields a tree-like routing path where the CH is the root node, the rely vehicles are child nodes, and the rest member vehicles are leaf nodes.

C. Residual link lifetime-based relay selection

Despite the routing paths computed by GPCA guaranteeing the required QoS, they fail to continuously maintain the performances over time due to network mobility. To address the issue without more frequent global network topology updates, we propose a distributed relay selection algorithm for the LPCAs deployed at the relay vehicles to perform re-routing when a link is anticipated to break due to increasing transmission distance.

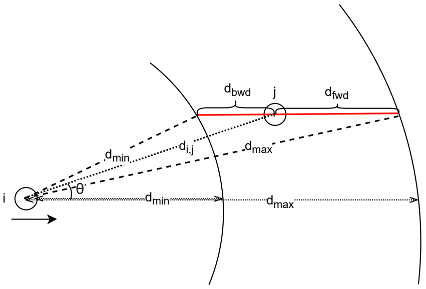


Fig. 4: Residual link lifetime evaluation

Denote $\mathcal{U}_{\text{ref}}^i$ as the set of vehicles resides within the reference transmission distance of u_i and we refer a vehicle $u_j \in \mathcal{U}_{\text{ref}}^i$, $i \neq j$ as a *relay candidate*. We evaluate the availability of each relay candidate using the residual link lifetime t_{res}^j, which is defined as the time a relay candidate u_j remains within the transmission range of u_i. Denote d_s, $s \in \{\text{fwd}, \text{bwd}\}$ as the distance of a relay candidate to the front and back edge of the transmission range as shown in Fig. 4. Thus, due to the speed differences between u_i and u_j and vehicle positions, two types of residual distance can be specified using the law of cosines:

When $v_j > v_i$, the residual distance d_{fwd} satisfies

$$d_{\max}^2 = d_{u_i}^2 + d_{\text{fwd}}^2 - 2 d_{u_i} d_{\text{fwd}} \cos(\pi - \theta), \tag{17}$$

where θ is the angle between vehicle i and j and $d_{i,j}$ is the inter-vehicle distance. By solving (17), we obtain

$$d_{\text{fwd}} = d_{u_i} \cos(\pi - \theta) \pm \sqrt{(d_{\max}^2 - d_{u_i}^2) + (d_{i,j} \cos(\pi - \theta))^2}. \tag{18}$$

Here, ± is determined depending on the sign of d_{fwd}. Similarly, when $v_j < v_i$, the backward residual distance d_{bwd} can be expressed as

$$d_{\text{bwd}} = d_{u_i} \cos\theta \pm \sqrt{(d_{\min}^2 - d_{u_i}^2) + (d_{i,j} \cos\theta)^2}, \tag{19}$$

where $d_{\min}$ is the minimum reference distance. Therefore, the residual link lifetime can be estimated by

$$t_{\text{res}} = \begin{cases} \frac{d_{\text{fwd}}}{|v_j - v_i| + 1} & v_j \geq v_i, \\ \frac{d_{\text{bwd}}}{|v_j - v_i| + 1} & v_j < v_i. \end{cases} \tag{20}$$

To avoid the oscillation problem, the re-selection procedure is called only if the residual link lifetime of the original relay is less than a threshold[10] and the vehicle with the maximum residual link lifetime is selected as the new relay, i.e.,

[10] In our implementation, we use the average residual link lifetime of the relay candidate set $\bar{t}_{\text{res}} = \sum_{j \in \mathcal{U}_{\text{ref}}^i} t_{\text{res}_j}$ as the threshold value.

Algorithm 4 Residual link lifetime-based relay selection

```
1: Input: Local network information $\mathcal{M}_L$, transmission range $\{d_{\min}, d_{\max}\}$, original relay vehicle $u_{\text{old}}$.
2: Output: Updated relay vehicle $u_{\text{new}}$
3: Update $T_{\text{budget}} = \theta_{\text{del}} - (t_{\text{start}} - t_{\text{now}})$
4: Update $Dist = Dist - d(l_{u_{new}}, l_{u_{now}})$
5: Update $d_{\min} = Dist \times \frac{L+\tau_{\max}\cdot\theta_{\text{rate}}}{T_{\text{budget}}\cdot\theta_{\text{rate}}}$
6: Compute $t_{\text{res}}^{\text{old}}$, $\bar{t}_{\text{res}}$ using (20)
7: if $t_{\text{res}}^{\text{old}} < \bar{t}_{\text{res}}$ then
8:     $u_{new} = \arg\max_{u_{j'} \in \mathcal{U}_{\text{ref}}^{i}} t_{\text{res}}$
9: else
10:     $u_{new} = u_{old}$
11: end if
```

$$u_{new} = \begin{cases} \arg\max\limits_{u_{j'} \in \mathcal{U}_{\text{ref}}^{i}} t_{\text{res}} & t_{\text{res}}^{(u_j)} < \bar{t}_{\text{res}}, \\ u_j & \text{otherwise.} \end{cases} \tag{21}$$

It is worth noticing that LPCA updates searching space at each hop by estimating the remaining delay budget T_{budget} and remaining coverage distance $Dist_{\text{rm}}$ as follows

$$T_{\text{budget}} = \theta_{\text{del}} - (t_{\text{start}} - t_{\text{now}}), \tag{22}$$

where t_{start} and t_{now} are the timestamps when the data packets are sent out from the source vehicle and the current reception time, respectively.

$$Dist_{\text{rm}} = \theta_{\text{cov}} - d(l_{u_{\text{now}}}, l_{u_{\text{nxt}}}), \tag{23}$$

where $d(l_{u_{\text{now}}}, l_{u_{\text{nxt}}})$ is the distance from the current vehicle to the next relay vehicle. In this context, $d_{\min}$ can be updated by

$$d_{\min} = Dist_{\text{rm}} \times \frac{L + \tau_{\max} \cdot \theta_{\text{rate}}}{T_{\text{budget}} \cdot \theta_{\text{rate}}}. \tag{24}$$

The updated $d_{\min}$ is carried in the packet header which will be used for relay selection in the next hop. The complete relay selection algorithm is summarized in Algorithm 4.

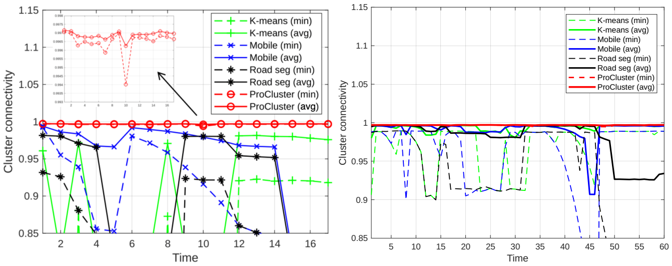
In a nutshell, when a data packet arrives at relay vehicle u_i, the corresponding LPCA first updates the remaining delay budget and coverage following (23) and (24) by inspecting the control information carried in the packet header and local network information (Line:3-5). Next, LPCA evaluates the original routing path by computing its residual link lifetime. A new relay vehicle will be selected if $t_{\text{res}}^{\text{old}}$ is smaller than a predefined threshold value (Line:6-11).

VI. Performance Evaluation

In this section, we verify our proposed QCRP protocol via extensive simulation experiments. We first introduce the experiment settings in our simulations, including traffic trace generation, channel parameters, and QoS requirements. Next, we theoretically compare the performances of each component in QCRP, i.e., clustering algorithm and QGR algorithm, with existing works using the models derived in Section III. Finally, we evaluate the complete QCRP protocol under different traffic densities, respectively using the ns-3 simulator [39].

TABLE III: Parameters Setting

Parameters	Values
Noise power spectrum density	-100 dBm/Hz
V2V and V2I channel bandwidth B_v, B_w	20 MHz
Transmission power	23 dBm
Carrier frequency	5 GHz
E2E delay requirement θ_{del}	30 ms
Data rate requirement θ_{rate}	80 Mbps
Reliability requirement θ_{rel}	99%
SNR threshold for data transmission θ_{SNR}	23 dB
Cluster coverage θ_{cov}	500 m



(a) Short-term performance (b) Long term performance

Fig. 5: Clustering performance comparison under high traffic density scenario.

A. Simulation setting

We used the Simulation of Urban MObility (SUMO) simulator to generate four mobility traces with different traffic densities (high, medium, low, and sparse) on a four-lane 5-km highway, where $463, 336, 219$, and 136 vehicles are injected into the system. The BS is deployed in the middle of the road covering the target region. Three types of vehicles with different speed ranges: $[25, 30]$ m/s, $[30, 35]$ m/s, and $[35, 40]$ m/s are considered. All vehicles adopt the Krauss car-following model and lane-change model [40] by default to simulate real car movements. We set all vehicles with the same transmission power of 23 dBm and assign a total bandwidth of 20 MHz in the 5 GHz frequency band for V2V data transmission. As for the QoS requirements, we set the E2E delay, data rate, reliability, and coverage requirements to be 30 ms, 80 Mbps, 99%, and 500 meters, respectively. We select one vehicle in the trace as the source vehicle and set the rest as member vehicles. The source vehicle starts transmission when it enters the target region and we evaluate the network performance over time until it leaves the target region. The complete simulation settings are summarized in Table III.

B. Clustering algorithm performance verification

First, we evaluate the connectivity performance of our proposed clustering algorithm, denoted by ProCluster, and compare it with two other clustering algorithms, namely, K-

TABLE IV: Connectivity performance comparison

Algorithms	High	Medium	Low	Sparse
ProCluster	**0.9974**	**0.9976**	**0.9969**	**0.9973**
	0.9967	**0.9968**	**0.9965**	**0.9959**
Mobile	0.9877	0.9872	0.9901	0.9911
	0.9271	0.8551	0.9379	0.9638
K-means	0.9871	0.9829	0.9924	0.9911
	0.9150	0.8096	0.9625	0.9638
Road seg	0.9212	0.9627	0.9796	0.9854
	0.6428	0.6776	0.8040	0.8861

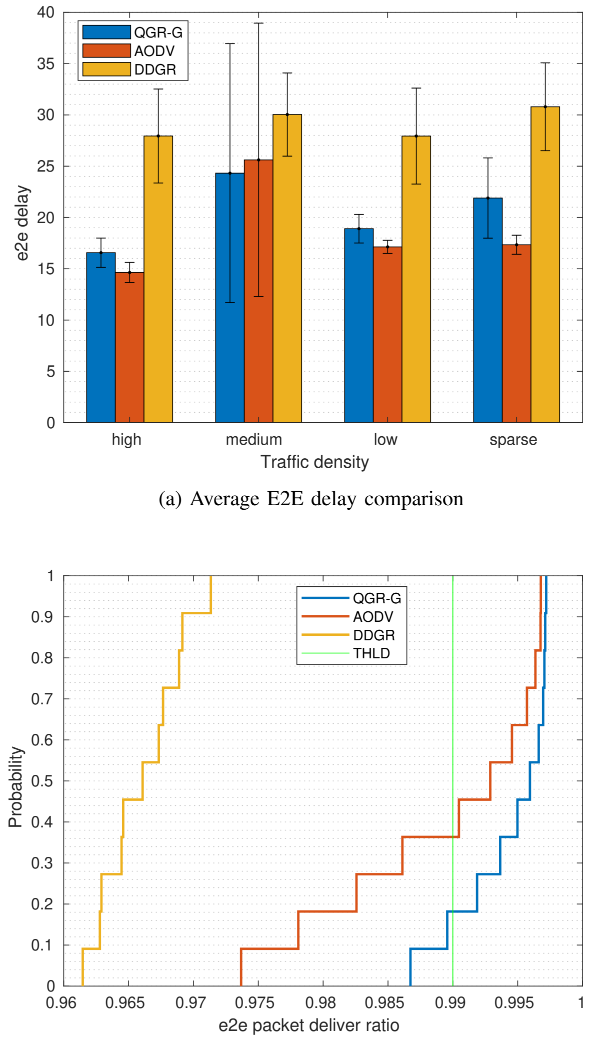
means and Road segmentation[11]. The connectivity of each cluster is evaluated using the connectivity model we derived in (9). In addition, we also evaluate the performance Algorithm 1 (denoted by Mobile) to highlight the effectiveness of mobility prediction in ProCluster.

In our simulation, we first evaluate all algorithms in a short period without updating the network topology. We use both the average network connectivity, i.e., average connectivity measurements of all clusters denoted in solid lines, and the minimal network connectivity, i.e., the cluster with the minimum connectivity denoted in dashed lines, to characterize the overall performance of each clustering algorithm. We use the high-traffic density scenario as an example. As shown in Fig. 5(a) ProCluster performed the best maintaining high connectivity throughout the experiment with tolerable fluctuations. The connectivity performance using Mobile performed the second best with its average connectivity satisfying the requirement most of the time thanks to its two-step cluster design, however, its minimal connectivity failed the requirement due to network mobility. On the other hand, K-means and Road seg performed worse with high fluctuation in their connectivity performances as they only depend on current topology and are unaware of network mobility.

Next, we fix the topology update period at 10 seconds to verify their long-term performance as shown in Fig. 5(b). Similarly, ProCluster maintained the network cluster connectivity throughout the simulation period while the rest restored their performance only after each update. The network is updated adaptively based on the prediction of vehicle mobility. Finally, we summarize the average and minimal connectivity performances under different traffic densities in Table IV. Overall, ProCluster achieves an improvement up to 55% compared to other clustering algorithms particularly when traffic density is high. This is because fewer clusters are formed in high-density networks causing higher link outage probability at the edge of each cluster due to vehicle mobility.

C. Global route planning performance verification

In this experiment, we evaluate the effectiveness of our proposed global routing algorithm employed by GPCA, i.e., the QGR algorithm without LPCA's local control (denoted by QGR-G), in guaranteeing the E2E delay and reliability. Specifically, we compare the E2E delay and reliability performance of QGR-G with the DGGR algorithm proposed in [12] and use AODV as the benchmark[12]. In our evaluation, we first compute the routing path for each cluster at time $t = 0$ using the mobility traces generated by SUMO and compute the theoretical average E2E delay and reliability performances of the entire network over one update period (i.e., the same routing paths are used for data transmission during this period). As shown in Fig. 6(a), both AODV and QGR-G compute routing paths satisfying the E2E delay under different traffic densities. AODV provided a slightly better delay performance in some cases, compared to QGR-G as it



(a) Average E2E delay comparison

(b) Average E2E PDR comparison

Fig. 6: Global routing algorithm comparison without local control.

[11] For K-means algorithm, we determine the number of cluster heads by dividing the network size, i.e., the distance between the front-edge vehicle and the back-edge vehicle, by the coverage requirement. For road segmentation (Road seg), we employ the approach adopted in [15] by uniformly clustering the target road into equal-length segments and selecting the centroid vehicle as the cluster head.

[12] We replaced the carry-and-forward mechanism in DGGR with V2I links for data transmission to reduce E2E delay. Here, AODV essentially finds the minimum-hop path for each cluster.

essentially computes the minimum-hop paths. On the other hand, DGGR fails to satisfy the delay performance as it requires more V2I links for data transmission when the inter-vehicle distance increases and lack of global network topology. Note that the E2E delay can easily reach up to a few seconds for the original DGGR algorithm based on the carry-and-forward mechanism.

As for the E2E reliability performance, we use the Dense traffic scenario as an example. As shown in Fig. 6(b), QGR-G performed the best achieving a 0.8 probability that satisfies the reliability requirement (99%) while AODV only provides a 0.6 probability guarantee. This is because QGR jointly considered the reliability and delay requirements when computing the routing paths. The routing path yield by DGGR fails to satisfy the E2E reliability requirement as it always selects the farthest vehicle within its reach for data transmission leading to low link reliability. However, although QGR-G provides better E2E reliability performance over time (improvement up to 28%), it failed to guarantee the E2E reliability due to topology changes between global topology updates at the BS. Thus, the E2E performance cannot be guaranteed.

D. QCRP performance verification

Finally, we evaluate the overall performances of the complete QCRP protocol in a more practical setting using the ns-3 simulator. Using the same channel setting and QoS requirement as summarized in Table III, we implement a constant-bit-rate *on-off* application at the source vehicle sending 10000 500 byte-packets at the rate of 90 Mbps to fully utilize the V2V channels. At the beginning of each time period, GPCA computes the global routing path and generates the corresponding forwarding table based on the QGR algorithm. The BS then broadcasts the forwarding table to the network and the packets are relayed accordingly by each relay vehicle. As for in-network control, the forwarding table is updated locally by the LPCAs at the relay vehicles according to the proposed relay selection algorithm. We verify QCRP under different traffic densities in comparison with AODV and DGGR algorithms and adopt QGR-G alone as a benchmark to show the effectiveness of re-routing. In our simulation, we measure the average E2E throughput, latency, and reliability (characterized by PDR) performances, respectively by computing the average E2E QoS performance (i.e., from source to the edge vehicles) of all clusters over one second.

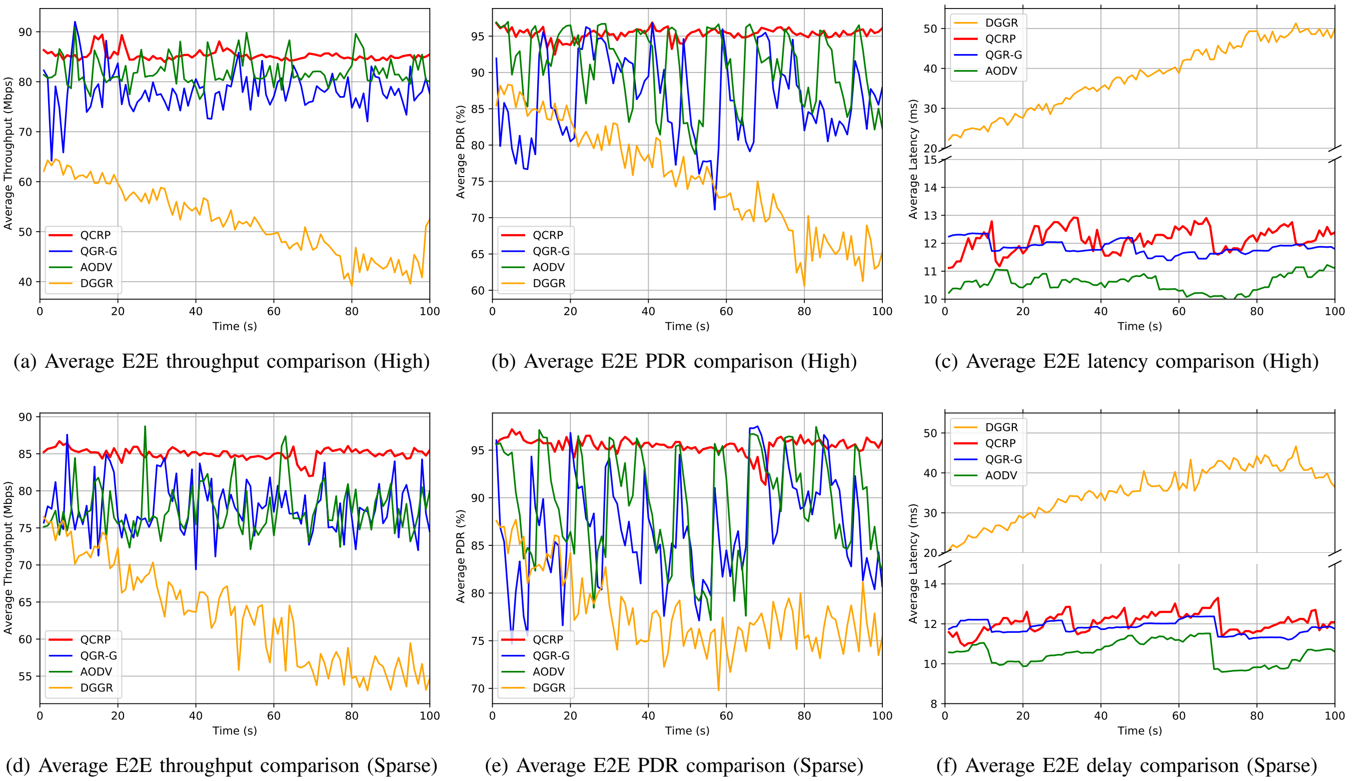
Due to page limit, we only present the simulation results under high density and sparse density for analysis as shown Fig. 7. The throughput performance is measured by the number of bits received at the edge vehicle of each cluster over one second and the reliability performance is measured by the number of packets being received by the edge vehicle over the number of packets being sent from the source vehicle. As for the E2E latency performance, we measure the average latency of the packets received by the edge vehicle within one second and compute the average E2E latency of all clusters.

QCRP marked in red curve performed the best over time under various traffic densities thanks to the joint global and local control. Particularly, QCRP showed high robustness to network mobility and traffic densities compared to the comparison methods in terms of both throughput and PDR performances. As shown in Fig. 7(a)(d) and Fig. 7(b)(e), both AODV and QGR-G marked in green and blue curves, with only global clustering and route planning, suffered from large performance variation with oblivious degradation and recovery pattern between each topology update. The variations increased when the traffic density was lower. This is because the V2V links of each cluster are more likely to break due to increased inter-vehicle distances. QCRP showed minimal E2E throughput and reliability performance variation throughout the experiment. Note that the cause of lower PDR in our simulations than in the theoretical analysis is the packet loss due to the limited buffer size we set at each relay vehicle. On the other hand, the E2E performances achieved by DGGR showed short-term variation and long-term decreasing patterns in general due to the lack of global route planning and its greedy-forwarding design. Interestingly, DGGR performed slightly better in throughput when the traffic density was lower. Similar patterns can be observed in its reliability performance. This is because we modified DGGR by replacing its carry-and-forward design with V2I links which are more stable than V2V links.

As it can be observed in Fig. 7(d)(f), QCRP showed slightly higher E2E latency with more frequent variations compared to QGR-G and AODV (but always satisfies the E2E latency requirement). This is because a new relay vehicle is re-selected whenever the original relay vehicle is out of the transmission range. Thus, the number of hops to reach the edge vehicles changes more frequently causing longer latency and higher variations. AODV provisioned the least E2E latency as expected since it uses the least-number-of-hops path for packet transmission. On the other hand, DGGR saw an increasing E2E latency under both traffic densities over time as more vehicles were injected into the network resulting in more hops. Overall, we conclude that QCRP provides improved and robust E2E throughput, reliability, and latency performance under high mobility with the help of joint global and local protocol control.

VII. Conclusion

In this paper, we investigate the network routing problem in vehicular networks. Future V2X applications require stringent QoS performance. However, due to the vulnerability of wireless communications and high mobility features, it is challenging to guarantee E2E QoS performances with conventional routing solutions. To bridge the gap, we propose a novel network architecture that leverages cross-layer information to generate corresponding control strategies. Following the architecture, a novel routing protocol named QCRP is proposed to guarantee the required QoS. Specifically, a predictive clustering algorithm based on LSTM is proposed to ensure network connectivity by dynamically adjusting the topology update period and network mobility prediction. A QoS-guaranteed routing algorithm is developed to perform route planning in



(a) Average E2E throughput comparison (High)
(b) Average E2E PDR comparison (High)
(c) Average E2E latency comparison (High)
(d) Average E2E throughput comparison (Sparse)
(e) Average E2E PDR comparison (Sparse)
(f) Average E2E delay comparison (Sparse)

Fig. 7: QCRP E2E performance comparison: first row under high traffic density and second row under sparse traffic density

a centralized manner based on QoS requirements and global network topology. In addition, a residual link lifetime-based re-routing algorithm is proposed to adapt to network dynamics and maintain QoS performance. We evaluate our solution with the state-of-art and benchmark using traffic traces with different densities. The results show that our solution is capable of guaranteeing the required QoS performances. However, the potential of the proposed architecture and QCRP is yet to be explored. For example, how QCRP can be extended to multiple source scenarios and enable serving multiple applications with various QoS requirements remains an open issue. In addition, the spectrum resource allocation and packet retransmission to reduce interference and improve packet delivery ratio are also worth investigating in our future work.

VIII. Appendix A

In a BPSK system, denote E_b as the signal energy associated with each bit, and N_0 as the noise power level per hertz. Let $\gamma_b = E_b/N_0$, then the BER is given by

$$p_b = Q(\sqrt{2\gamma_b}), \tag{25}$$

where the Q function is defined as:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{x^2}{2}} dx. \tag{26}$$

Since E_b/N_0 is a normalized form of SNR, it can be represented by

$$\frac{E_b}{N_0} = \text{SNR} \cdot \frac{B}{f_b}, \tag{27}$$

where B is the channel bandwidth and f_b is the channel data rate. The channel model in 3 can be adopted to estimate SNR

$$\text{SNR} = \frac{P_t}{N_0} \delta d^{-\alpha} |h|^2. \tag{28}$$

Bringing (28) and (27) into (25), we can obtain the mapping between BER and distance

$$BER = f(d) = Q\left(\sqrt{2\frac{B}{f_b} \cdot \frac{P_t}{N_0} \delta d^{-\alpha} |h|^2}\right). \tag{29}$$

References

[1] J. He, K. Yang, and H.-H. Chen, "6G cellular networks and connected autonomous vehicles," *IEEE Netw.*, vol. 35, no. 4, pp. 255–261, 2020.

[2] C.-X. Wang, X. You, X. Gao, X. Zhu, Z. Li, C. Zhang, H. Wang, Y. Huang, Y. Chen, H. Haas *et al.*, "On the road to 6g: Visions, requirements, key technologies and testbeds," *IEEE Communications Surveys & Tutorials*, 2023.

[3] M. Noor-A-Rahim, Z. Liu, H. Lee, M. O. Khyam, J. He, D. Pesch, K. Moessner, W. Saad, and H. V. Poor, "6g for vehicle-to-everything (v2x) communications: Enabling technologies, challenges, and opportunities," *Proceedings of the IEEE*, vol. 110, no. 6, pp. 712–734, 2022.

[4] M. M. Saad, M. A. Tariq, J. Seo, and D. Kim, "An overview of 3gpp release 17 & 18 advancements in the context of v2x technology," in *2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2023, pp. 057–062.

[5] R. Hussain and S. Zeadally, "Autonomous cars: Research results, issues, and future challenges," *IEEE Commun. Surv. Tutor*, vol. 21, no. 2, pp. 1275–1313, 2018.

[6] L. Barbieri, S. Savazzi, M. Brambilla, and M. Nicoli, "Decentralized federated learning for extended sensing in 6g connected vehicles," *Vehicular Communications*, vol. 33, p. 100396, 2022.

[7] H. Zhang, X. Zhang, and D. K. Sung, "A fast, reliable, opportunistic broadcast scheme with mitigation of internal interference in vanets," *IEEE Trans. Mob. Comput.*, 2021.

[8] T. Chatterjee, R. Karmakar, G. Kaddoum, S. Chattopadhyay, and S. Chakraborty, "A survey of vanet/v2x routing from the perspective of non-learning-and learning-based approaches," *IEEE Access*, vol. 10, pp. 23 022–23 050, 2022.

[9] A. Al-Habob, H. Tabassum, and O. Waqar, "Dynamic unicast-multicast scheduling for age-optimal information dissemination in vehicular networks," in *2022 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2022, pp. 1218–1223.

[10] A. Srivastava, A. Prakash, and R. Tripathi, "Location based routing protocols in vanet: Issues and existing solutions," *Vehicular Communications*, vol. 23, p. 100231, 2020.

[11] M. Ayyub, A. Oracevic, R. Hussain, A. A. Khan, and Z. Zhang, "A comprehensive survey on clustering in vehicular networks: Current solutions and future challenges," *Ad Hoc Networks*, vol. 124, p. 102729, 2022.

[12] C. Chen, L. Liu, T. Qiu, D. O. Wu, and Z. Ren, "Delay-aware grid-based geographic routing in urban vanets: A backbone approach," *IEEE/ACM Trans. Netw.*, vol. 27, no. 6, pp. 2324–2337, 2019.

[13] M. 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*, vol. 114, p. 101961, 2021.

[14] M. Yang, Y. Li, D. Jin, L. Zeng, X. Wu, and A. V. Vasilakos, "Software-defined and virtualized future mobile and wireless networks: A survey," *Mobile Networks and Applications*, vol. 20, pp. 4–18, 2015.

[15] 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. Vehicular Tech.*, vol. 69, no. 7, pp. 7723–7734, 2020.

[16] L. Cai, J. Pan, W. Yang, X. Ren, and X. Shen, "Self-evolving and transformative (SET) protocol architecture for 6G," *IEEE Wirel. Commun.*, 2022.

[17] H. Gong, L. Fu, X. Fu, L. Zhao, K. Wang, and X. Wang, "Distributed multicast tree construction in wireless sensor networks," *IEEE Transactions on Information Theory*, vol. 63, no. 1, pp. 280–296, 2016.

[18] B. Karp and H.-T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 243–254.

[19] A. BENGAG, A. Bengag, and M. E. Boukhari, "Enhancing gpsr routing protocol based on velocity and density for real-time urban scenario," in *2020 International Conference on Intelligent Systems and Computer Vision (ISCV)*. IEEE, 2020, pp. 1–5.

[20] V. Naumov and T. R. Gross, "Connectivity-aware routing (car) in vehicular ad-hoc networks," in *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*. IEEE, 2007, pp. 1919–1927.

[21] O. K. Tonguz, N. Wisitpongphan, and F. Bai, "Dv-cast: A distributed vehicular broadcast protocol for vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 17, no. 2, pp. 47–57, 2010.

[22] A. Khan, A. A. Siddiqui, F. Ullah, M. Bilal, M. J. Piran, and H. Song, "Vp-cast: Velocity and position-based broadcast suppression for vanets," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[23] B. L. Nguyen, D. T. Ngo, N. H. Tran, M. N. Dao, and H. L. Vu, "Dynamic v2i/v2v cooperative scheme for connectivity and throughput enhancement," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1236–1246, 2020.

[24] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," Tech. Rep., 2003.

[25] J. Toutouh, J. García-Nieto, and E. Alba, "Intelligent olsr routing protocol optimization for vanets," *IEEE transactions on vehicular technology*, vol. 61, no. 4, pp. 1884–1894, 2012.

[26] S. Gangopadhyay and V. K. Jain, "A position-based modified olsr routing protocol for flying ad hoc networks," *IEEE Transactions on Vehicular Technology*, 2023.

[27] P. K. Shrivastava and L. Vishwamitra, "Comparative analysis of proactive and reactive routing protocols in vanet environment," *Measurement: Sensors*, vol. 16, p. 100051, 2021.

[28] I. Yaqoob, I. Ahmad, E. Ahmed, A. Gani, M. Imran, and N. Guizani, "Overcoming the key challenges to establishing vehicular communication: Is sdn the answer?" *IEEE Communications Magazine*, vol. 55, no. 7, pp. 128–134, 2017.

[29] J. Bhatia, R. Dave, H. Bhayani, S. Tanwar, and A. Nayyar, "Sdn-based real-time urban traffic analysis in vanet environment," *Computer Communications*, vol. 149, pp. 162–175, 2020.

[30] L. Zhao, G. Han, Z. Li, and L. Shu, "Intelligent digital twin-based software-defined vehicular networks," *IEEE Network*, vol. 34, no. 5, pp. 178–184, 2020.

[31] L. Zhao, Z. Bi, A. Hawbani, K. Yu, Y. Zhang, and M. Guizani, "Elite: An intelligent digital twin-based hierarchical routing scheme for softwarized vehicular networks," *IEEE Transactions on Mobile Computing*, 2022.

[32] A. Nahar and D. Das, "Metalearn: Optimizing routing heuristics with a hybrid meta-learning approach in vehicular ad-hoc networks," *Ad Hoc Networks*, vol. 138, p. 102996, 2023.

[33] Y. Tang, N. Cheng, W. Wu, M. Wang, Y. Dai, and X. Shen, "Delay-minimization routing for heterogeneous vanets with machine learning based mobility prediction," *IEEE Trans. Vehicular Tech.*, vol. 68, no. 4, pp. 3967–3979, 2019.

[34] Z. Li, L. Zhao, G. Min, A. Y. Al-Dubai, A. Hawbani, A. Y. Zomaya, and C. Luo, "Reliable and scalable routing under hybrid sdvn architecture: A graph learning based method," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[35] X. M. Zhang, Y. Zhang, F. Yan, and A. V. Vasilakos, "Interference-based topology control algorithm for delay-constrained mobile ad hoc networks," *IEEE Trans. Mob. Comput.*, vol. 14, no. 4, pp. 742–754, 2014.

[36] S. Zaidi, S. Bitam, and A. Mellouk, "Enhanced adaptive sub-packet forward error correction mechanism for video streaming in vanet," in *IEEE Glob.*, Dec. 2016, pp. 1–6.

[37] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[38] X. Mo, Z. Huang, Y. Xing, and C. Lv, "Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9554–9567, 2022.

[39] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and tools for network simulation*. Springer, 2010, pp. 15–34.

[40] J. Song, Y. Wu, Z. Xu, and X. Lin, "Research on car-following model based on sumo," in *The 7th IEEE/International Conference on Advanced Infocomm Technology*. IEEE, 2014, pp. 47–55.