Design of Joint Congestion Simulation Experiment Based on Cross-Layer Collaboration in VANETs

Hui Wang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: wanghuihy@xatu.edu.cn

Qi Yu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: yuqi@st.xatu.edu.cn

Xueqiao Cao
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: caoxueqiao@st.xatu.edu.cn

Dongsheng Li
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: lidongshengxatu@st.xatu.edu.cn

***Abstract*—Aiming at the link-level congestion caused by frequent network topology changes and limited communication resources and the node-level congestion caused by overload of cluster head nodes in VANETs, this paper proposes a Joint Congestion Strategy based on Cross-layer Collaboration in VANETs (JCSCCV). By designing a joint congestion overall architecture for cross-layer collaboration, a multi-criteria double cluster head dynamic clustering strategy based on motion similarity and a multi-path congestion-free routing algorithm based on genetic-ant colony fusion are proposed. The routing problem is transformed into a multi-objective optimization problem. The genetic-ant colony fusion meta-heuristic algorithm is used to solve the construction of multipath congestion-free routing, which avoids network congestion caused by link instability and uneven load distribution. Finally, through the NS3 simulation platform, it is verified that JCSCCV has better performance in V2X applications than CLCC, VC-TCP and NC, TOSCL, NewReno and other algorithms, and has higher application value.**

Keywords-Vanets; Cross-Layer Collaboration; Congestion Strategy; Dynamic Clustering; Multipath Congestion-Free Routing

I. INTRODUCTION

Vehicular Ad-hoc Networks (VANETs) serve as a key component of Intelligent Transportation Systems (ITS) [1], wherein low-latency data transmission holds significant importance for both safety-critical applications and high-throughput applications [2]. However, the rapidly changing network topology, coupled with a sharp increase in vehicle nodes and the massive propagation of various application service messages, can easily trigger channel collisions [3][4]. In severe cases, this leads to node-level congestion or link-level congestion, resulting in increased transmission delay and packet loss rates. Consequently, the inability to receive or transmit critical information in a timely manner rapidly degrades the service capability of VANET applications [5].

Addressing the congestion issues within the complex communication environment of VANETs, this paper adopts a cross-layer collaboration approach and proposes a Joint Congestion Strategy based on Cross-layer Collaboration in VANETs (JCSCCV). By transforming the traditional layered control architecture, JCSCCV utilizes a cross-layer collaboration method to jointly optimize resource allocation. This strategy enhances network performance and stability, ensures network connectivity during the communication process, and effectively mitigates node-level congestion and link-level congestion caused by frequent topology reconstruction and insufficient bottleneck link resources, thereby improving the overall network performance of VANETs.

Main Research Contents

The main research contents of this paper are as follows:

1) A cross-layer collaboration overall architecture is proposed, which allows protocols across different layers to share network status information and facilitates the real-time update of network resource information, such as channel status.

2) Targeting the issue where cluster head nodes are prone to node-level congestion in traditional single-cluster-head routing topologies, a multi-criteria double cluster head dynamic clustering strategy based on motion similarity is proposed. In addition, a dynamic maintenance algorithm is employed to ensure reliable communication and superior network performance during cluster variations. To address network congestion caused by unbalanced inter-cluster routing traffic, a multi-path congestion-free routing algorithm based on genetic-ant colony fusion is introduced. The genetic-ant colony fusion meta-heuristic algorithm is utilized to solve the construction of multi-path congestion-free routing, thereby avoiding network congestion arising from link instability and uneven load distribution.

3) Addressing the synchronization issue, a clock synchronization mechanism for both inter-cluster and intra-cluster nodes is proposed. This mechanism synchronizes the local clocks of all cluster member nodes to a common clock to enhance channel utilization.

4) An NS3 simulation platform was constructed, and simulation experiments were designed to compare performance with algorithms such as CLCC, VC-TCP/NC, TOSCL, and NewReno. The results verified that the JCSCCV strategy demonstrates superior performance in terms of average delay, average hop count, and system throughput.

II. RELATED WORK

The Wireless Access in Vehicular Environments (WAVE) protocol, a U.S. standard for VANETs, is currently the most mature Vehicle-to-Vehicle (V2V) communication protocol in use [6]. It primarily comprises the IEEE 802.11p protocol, the IEEE 1609.x series protocols, and the SAE J2735 safety service message set standard, as shown in Fig. 1.

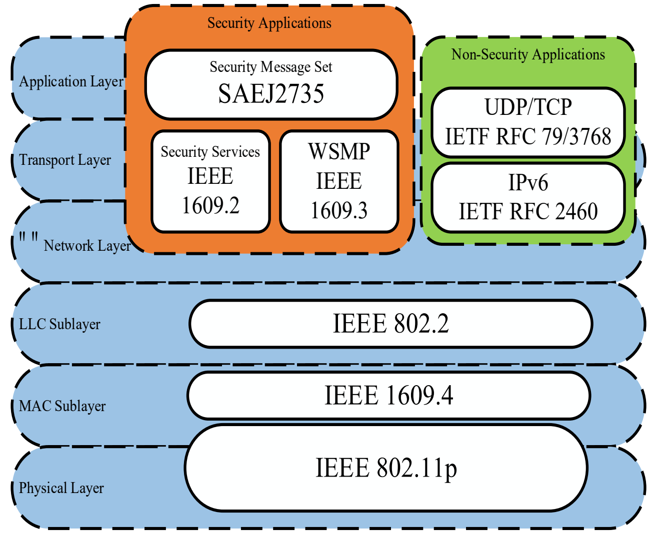


Figure 1. WAVE Protocol System

Among them, IEEE 802.11p defines the physical layer and the MAC sublayer of the protocol, primarily taking responsibility for medium access control [7]. The IEEE 1609.x series protocols involve the MAC extension sublayer as well as the network and transport layers related to safety applications, mainly governing functions such as channel allocation, channel access, service requests, and service advertisements during communication. The SAE J2735 standard was defined by the Society of Automotive Engineers (SAE) specifically for VANET safety application scenarios; while it classifies as an application layer standard based on the WAVE framework [8], it strictly speaking does not belong to the WAVE protocol framework.

There are two basic communication types in VANETs: Vehicle-to-Vehicle (V2V) communication and Vehicle-to-Infrastructure (V2I) communication [9]. IEEE 802.11p, based on Dedicated Short-Range Communication (DSRC) technology, was proposed as the communication standard for both V2V and V2I. In mobile ad-hoc networks where vehicles serve as topology nodes, each vehicle node is regarded as an independent mobile communication carrier responsible for packet relay and forwarding [10]. However, due to the rapid changes in network topology, as well as the limited transmission range of DSRC and the lack of supporting infrastructure, DSRC fails to meet certain requirements of V2V communication.

With the continuous expansion of 5G communication, its application to VANETs can optimize the VANET architecture to a certain extent and address the problems and challenges currently faced in the field.

The congestion control problem in VANETs has become a research hotspot. Traditional TCP algorithms include NewReno [11], Cubic, Vegas [13], and Westwood [14]. Among them, the NewReno and Cubic algorithms are suitable for traditional Internet network environments; they utilize packet loss as an indicator of congestion and subsequently reduce the congestion window. Vegas employs delay to detect congestion. Westwood predicts the link transmission capacity to assess congestion status; it determines the transmission rate based on returned ACK packets and further adjusts the window size and slow start threshold.

In recent years, with the rapid development of VANETs, scholars both domestically and internationally have conducted extensive research on congestion control in VANETs. Based on a designed information delivery rate model, Liu Biao et al. [15] proposed a distributed channel congestion control strategy for VANETs to ensure the propagation efficiency of Basic Safety Messages (BSM). Targeting congestion control at the MAC layer in VANETs, Sun Youwei et al. [16] proposed a congestion control strategy based on the Tabu Search algorithm, which determines whether to initiate congestion control by comparing values against a threshold. Tan Guozhen et al. [17] proposed UBRCC, a distributed congestion control strategy based on network utility maximization theory. This algorithm obtains the packet transmission rate by updating the vehicle congestion "price," ultimately realizing channel allocation that meets the safety requirements of individual vehicles.

The aforementioned TCP variants are primarily protocols based on packet loss feedback, delay feedback, or link capacity. These protocols belong to traditional layered congestion control algorithms, all of which utilize specific rule sets to make adjustments at the transport layer. However, the information required by these rule sets is difficult to share rapidly within a strictly layered congestion control architecture, and thus fails to adapt to the high mobility and low latency requirements of V2X. Consequently, researchers have proposed cross-layer congestion control protocols.

The core concept of cross-layer design involves coordinating the operations across different layers of the protocol stack by exchanging specific information. This approach blurs the boundaries between traditionally independent layers to enhance overall network performance. By sharing information and relevant parameters across network sub-layers, cross-layer design can satisfy the diverse requirements of different services and optimize the overall performance of the network [18].

Combining autonomic computing and cross-layer optimization technologies, Cheng Mingyang et al. [19] proposed a cross-layer transmission optimization strategy for wireless sensor networks. They utilized virtual pipes to transmit parameters across layers, realizing the planning and scheduling of limited resources within wireless sensor networks. Based on the primal-dual interior-point method, Feng Wei et al. [20] designed an algorithm with second-order convergence performance; by employing matrix splitting, they achieved distributed updates of service flow source rates and node link power to maximize network utility. Chen Jie et al. [20] proposed VC-TCP/NC, a cross-layer joint optimization method based on random linear network coding, and redesigned the transmission strategy at the sender of the network coding layer, thereby improving latency and network throughput. Yu Haiyang et al. [21] adopted a cloud model within the buffer queue to detect network congestion status and adjusted the data transmission rate, thus realizing cross-layer congestion control for Ad-hoc networks.

However, the high density and complexity of vehicular networks dictate the difficulty of achieving low-latency processing for different services. In view of this, this paper aims to implement joint congestion control in VANETs using a cross-layer collaboration approach. This strategy is designed to meet the diverse and differentiated demands of VANET services and effectively guarantee the real-time performance and reliability of information interaction within the network.

III. METHODOLOGY

Targeting node-level and link-level congestion in VANETs, this paper designs and optimizes the VANET architecture by integrating the interrelationships among the Application, Network, MAC, and Physical layers. While maintaining layer separation, the design allows protocols across different layers to rapidly share network status information. By adopting a cross-layer collaboration approach to effectively allocate network resources, the strategy achieves timely and effective control over node-level and link-level congestion, thereby improving the overall congestion status of VANETs.

A. Overall Architecture Design

Based on the analysis of VANET communication protocols and congestion control, the overall architecture of JCSCCV based on cross-layer collaboration is established. JCSCCV comprises the Application layer, Network layer, MAC layer, and Physical layer, as shown in Fig. 2.

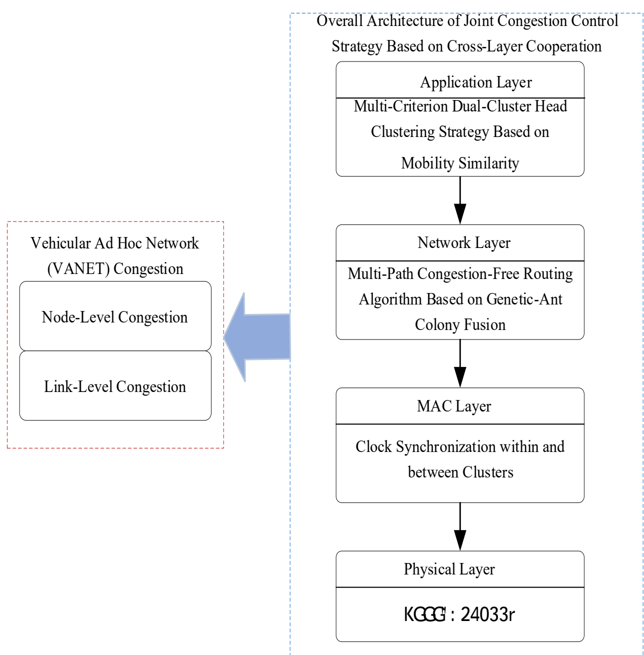


Figure 2. Overall Architecture for Cross-Layer Congestion Collaborative Scheduling

1) Application Layer — Resolving Node-level and Link-level Congestion in VANETs A clustering method based on motion similarity is designed to reduce link-level congestion caused by broadcast storms and channel collisions in VANETs, thereby satisfying the requirements for low-latency data transmission. A double cluster head mechanism is proposed; by introducing redundant cluster heads, this mechanism alleviates the pressure on a single cluster head, improves the data processing capability of cluster heads, and mitigates node-level congestion. An evaluation model is constructed using the CRITIC objective weighting method and the TOPSIS multi-criteria decision-making algorithm to elect the double cluster heads. Simultaneously, a packet scheduling algorithm targeting the double cluster head mechanism is proposed to achieve load balancing among cluster heads and reduce queuing delay.

2) Network Layer — Resolving Link-level Congestion Targeting link-level congestion caused by irrational routing, a genetic-ant colony fusion routing algorithm is designed to implement multi-path congestion-free routing and mitigate network congestion. First, a link stability metric is defined to evaluate the link status between vehicle nodes, and routing congestion degree and delay metrics are introduced to provide a decision-making basis for path selection. Second, the message and path discovery processes are designed, utilizing the genetic algorithm to construct the initial routing. Next, the probability transition formula and pheromone update strategy of the ant colony algorithm are improved; the routing is optimized through the efficient local search capability of the improved ant colony algorithm to obtain stable multi-path congestion-free routing. Finally, a link repair algorithm is proposed to reduce delay overhead caused by path instability and mitigate network congestion.

3) MAC Layer and Physical Layer — Implementing Clock Synchronization and Avoiding Potential Collisions Clock synchronization serves as the foundation for the collaborative operation of intra-cluster and inter-cluster nodes in VANETs. Addressing the synchronization issue, the MAC layer synchronizes the local clocks of all cluster member nodes to a common clock to enhance channel utilization and effectively resolve VANET congestion. The Physical layer adopts safe speeds based on the IEEE 802.11p standard to avoid potential collisions.

B. Cross-layer Joint Congestion Strategy

1) Application Layer

Targeting the issue where cluster head nodes in traditional single-cluster-head routing topologies are prone to node-level congestion, the Application layer considers both the instantaneous position and future movement trend of vehicles, proposing a multi-criteria double cluster head dynamic clustering strategy based on motion similarity. Taking position proximity and velocity consistency as evaluation indicators, vehicles with close spatial distance and similar movement states are rapidly grouped into a single cluster. A multi-criteria decision-making algorithm is used to construct an evaluation model to elect primary and secondary double cluster heads. The degree of congestion is measured based on the status of the cluster head buffer queue, and task traffic is scheduled to alleviate cluster head congestion. In addition, a dynamic maintenance algorithm is employed to ensure reliable communication and superior network performance even when the cluster undergoes changes.

a) Definition of Motion Similarity

First, motion similarity is defined to evaluate the motion similarity relationship between vehicles. Let siml be defined as position proximity and simv as velocity consistency. When the distance between two vehicles approaches 0, the position proximity reaches a maximum of 1; when the distance between two vehicles approaches +∞, the position proximity should be 0. The calculation formula is shown in (1).

$$\mathrm{siml}_{ij} = \begin{cases} e^{-kd_{ij}} & d_{ij} \le R \\ 0 & d_{ij} > R \end{cases} \tag{1}$$

Here, k denotes the position proximity sensitivity coefficient, and R represents the communication distance threshold between two vehicles. If the distance between two vehicles exceeds R, communication is considered impossible. The calculation of velocity consistency is presented in (2).

$$\mathrm{simv}_{ij} = \frac{v_i v_j}{|v_i||v_j|} \tag{2}$$

In (2), vi denotes the velocity vector information of the i-th vehicle, and vj denotes the velocity vector information of vehicle node j. Each vehicle can acquire its own velocity vector information and broadcast it to neighboring nodes. Consequently, the motion similarity between node i and node j is obtained by calculating the weighted sum of velocity consistency and position proximity, as shown in (3).

$$\mathrm{sim}_{ij} = \omega_l \mathrm{siml}_{ij} + \omega_v \mathrm{simv}_{ij} \tag{3}$$

Here, ωl and ωv represent the weights for position proximity and velocity consistency, respectively, satisfying ωl + ωv = 1. In this section, ωl is set to 0.6 and ωv is set to 0.4 to prioritize vehicle distance during the clustering process.

b) Clustering via Agglomerative Hierarchical Clustering Algorithm

After obtaining the motion similarity of all nodes, the agglomerative hierarchical clustering algorithm is utilized to generate vehicle clusters. First, all vehicles are regarded as Non-Cluster Members (NCM). An NCM broadcasts its own velocity and position information to surrounding nodes while receiving similar information from other vehicles. It then calculates the motion similarity with neighboring nodes via (3). If the vehicle with the highest motion similarity is already a cluster member, the NCM sends a join request to it; otherwise, it sends a merge request. Once all vehicles belong to a cluster, clusters are merged based on the same principle. Finally, when the number of members in a cluster exceeds the maximum cluster member limit, the cluster ceases to participate in further merging and rejects join requests from other vehicles.

This process ensures that vehicles within a cluster possess high velocity consistency and position proximity, while the number of vehicles in each cluster does not exceed the maximum value. Consequently, the number of generated clusters is neither too large nor too small, thereby avoiding negative impacts on network performance. An excessive number of clusters results in reduced network coverage per cluster, which lowers routing reliability and coverage area. This prevents the full utilization of network channel resources, causing channel wastage. Furthermore, it generates excessive routing hops, increasing communication delay in VANETs. Conversely, too few clusters result in an excessive number of vehicle nodes per cluster. This increases the probability of channel conflicts and packet collisions, leading to network congestion and increasing the computational complexity of cluster head nodes. The specific clustering process is illustrated in Fig.3.

The process of the clustering algorithm based on motion similarity demonstrates a dynamic and self-organizing vehicle clustering management method. By dynamically merging and controlling the number of intra-cluster members, the system can adapt to network changes under different conditions, maintaining the effectiveness and stability of the generated clusters. Simultaneously, this clustering method helps control cluster size, ensuring an optimal balance between the number of clusters and the number of cluster members. This effectively improves the stability of the vehicle clusters and maximizes the cluster maintenance time.

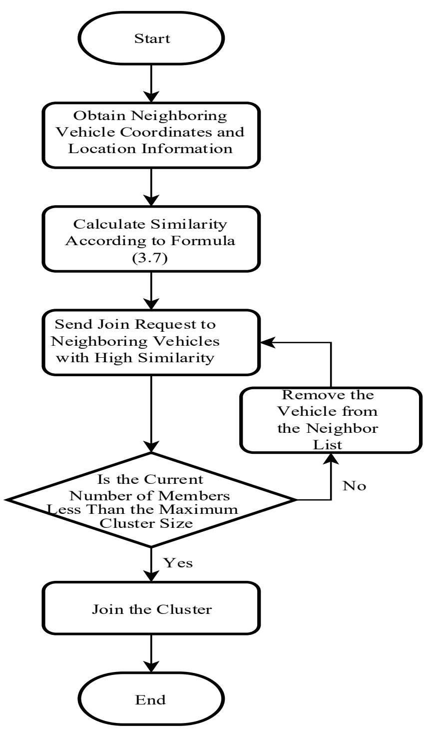


Figure 3. Flowchart of Clustering Based on Motion Similarity

The Application layer transmits messages, such as estimated collision probability and safe speed, to the Network layer before they are propagated to the target node.

Under the assumption that vehicles are randomly and independently distributed, given a vehicle density of ρ, the vehicle distribution on any road segment of length L follows a Poisson distribution with intensity ρL. The probability density function for the existence of k vehicles on this segment is shown in (4).

$$\partial(k,l) = \frac{(\rho l)^k}{k!} e^{-\rho l} \tag{4}$$

Based on the surrounding traffic flow density information, a communication demand estimation model is defined to characterize the data communication demand in VANETs, as shown in (5).

$$x(t) = f \times P \times R \times \rho \tag{5}$$

Here, f is the broadcast frequency of vehicle status messages, and the WAVE standard stipulates that the broadcast frequency $f \in [1, 10]$. To ensure driving safety between vehicles, the more frequent the message exchange, the larger the value of f required; in the experimental phase, f can be set to 10 Hz. R represents the communication range of the vehicle. P represents the traffic flow density surrounding the current vehicle, namely the number of programs sending WSM data packets in the current vehicle, which can be directly obtained through information exchange with Road Side Units (RSUs). k_R denotes the number of vehicles within the communication distance, and $P \times R \times k_R$ represents the number of programs sending WSM data packets within the vehicle communication range. Utilizing (5), the total communication volume X for safety-related applications can be calculated.

To realize the transmission of safety-related application data in the channel, a channel backoff access mechanism based on Markov chains is employed. The entire backoff process is modeled and described using the Markov chain shown in Fig.4, through which channel sharing is achieved to avoid channel congestion.

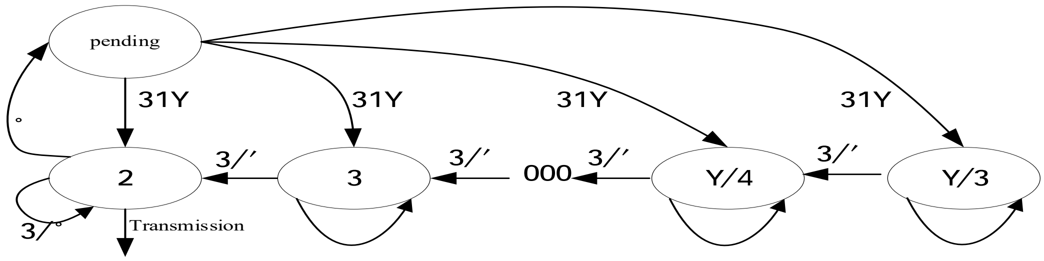


Figure 4. Markov Chain-Based Retreat Simplification Model

In the figure, W denotes the minimum contention window size, α represents the probability that a node has at least one data packet waiting to be sent at any instant, and β represents the probability that the channel is busy.

When a node has data to send, it enters the "wait to send" state and randomly selects an integer from $(0,1,\cdots,W-1)$ as the backoff value. Then, the node monitors whether the channel is in a busy state. If the channel is idle, the counter is decremented by 1; otherwise, it remains unchanged. When the counter value decreases to 0, the data is sent; if there is still data waiting to be sent, it enters the next "wait to send" state. Solving the Markov chain allows for obtaining the probability b_0 that the counter backs off to 0, as shown in (6).

$$b_0 = 2(1-\beta)/2-3\beta+\beta W \tag{6}$$

The data transmission rate follows a Poisson distribution. If the average time to transmit a data packet is T_v, then the probability α that there is at least one data packet waiting to be sent at any instant can be calculated by (7):

$$\alpha = 1-\frac{(VT_v)^k}{k!}e^{-VT_v}\Big|_{k=0} = 1-e^{-VT_v} \tag{7}$$

Based on Equations (6) and (7), the data transmission probability τ at any instant is calculated as shown in (8):

$$\tau = \frac{2(1-\beta)}{2+\beta W-3\beta}(1-e^{-VT_v}) \tag{8}$$

The probability β that the channel is busy is shown in (9):

$$\beta = 1-\sum_{k=0}^{\infty}(1-\tau)\frac{(2R\rho)^k}{k!}e^{-2R\rho} = 1-e^{-2R\rho\tau} \tag{9}$$

Here, R represents the transmission radius of the node. Equations (8) and (9) form a system of nonlinear equations, and it can be proven that there is a unique solution within the interval $(0,1)$. By solving this system of nonlinear equations, the values of β and τ can be obtained.

Although the backoff mechanism can effectively avoid disorderly occupation of the channel by multiple nodes, it cannot completely eliminate data collisions. A data packet is successfully received by other nodes only if there is exactly one node transmitting within the sender's transmission range; otherwise, a collision will occur. Based on the vehicle distribution probability formula (4) and the data packet transmission probability formula (7), the probability of successful data packet transmission, P_a, can be derived as shown in (10):

$$P_a = k\sum_{k=1}^{\infty}\tau(1-\tau)^{k-1}\frac{(2R\rho)^k}{k!}e^{-2R\rho} \tag{10}$$

Simplifying yields (11):

$$P_a = 2R\rho\tau \cdot e^{-2R\rho} \tag{11}$$

2) Network Layer

Targeting network congestion caused by unbalanced inter-cluster routing traffic, and combining the strong global search capability of the Genetic Algorithm (GA) with the fast local search capability of Ant Colony Optimization (ACO), a multi-path congestion-free routing algorithm based on genetic-ant colony fusion (GAAC) is proposed. This algorithm alleviates link-level congestion in VANETs by optimizing path selection. A link stability metric is designed; by combining factors such as transmission delay and congestion degree to construct an objective function, the routing problem is transformed into a multi-objective optimization problem. The genetic-ant colony fusion meta-heuristic algorithm is utilized to solve for and construct multi-path congestion-free routing, thereby avoiding network congestion caused by link instability and uneven load distribution. Finally, a link repair algorithm is proposed. While ensuring communication continuity and meeting the low-latency requirements of VANETs, it reduces the need for route re-establishment and effectively alleviates network congestion.

First, a link stability metric is defined to evaluate the link status between vehicle nodes. Additionally, routing congestion degree and delay metrics are introduced to jointly provide a decision-making basis for path selection.

Considering the impact of congested nodes on vehicle communication performance, the residual buffer queue length of the node is used to represent the congestion degree N_c of node n. The calculation formula is shown in (12).

$$N_c(n) = \frac{q_{\text{use}}(n)}{Q_{\max}} \tag{12}$$

In the above equation, q_{use} denotes the used buffer queue length of node n, and Qmax represents the maximum buffer queue length of node n. The congestion degree of the route $R(S, D)$ connecting the source node S to the destination node D is defined as the maximum value of the node congestion degrees along this route, as shown in (13).

$$F_c(R(S,D)) = \max\{N_c(n) \mid n \in R(S,D)\} \tag{13}$$

Routes in VANETs are composed of multi-hop wireless links. If the Link Life Time (LLT) of each link is known, the RLT is shown in (14).

$$RLT_R = \min_{\text{link } R}(LLT_i) \tag{14}$$

Here, R represents the route connecting the source node and the destination node, composed of multiple wireless links l. Generally, if the distance between two vehicle nodes is less than the transmission distance T_R, it is considered that the two vehicles can perform direct communication.

Three new routing metrics are used to replace the minimum hop count for measuring wireless routing performance, transforming the routing problem into a multi-objective optimization problem. The objective function is shown in (15):

$$f = \omega_1 F_c - \omega_2 F_{LT} + \omega_3 F_d \tag{15}$$

In the objective function, F_c represents the route congestion degree, FLT denotes the route stability, and F_d indicates the route delay metric, while ω_1, ω_2, and ω_3 correspond to the weights of F_c, FLT, and F_d, respectively. Utilizing meta-heuristic algorithms such as Genetic Algorithms and Ant Colony Optimization for rapid route search requires the definition of a reasonable fitness function. The fitness function is employed to evaluate the quality of a solution, guide the algorithm's search process, and drive the algorithm to evolve towards superior solutions; it is typically derived from the objective function. The fitness function defined in this paper is shown in (16):

$$FF = \frac{1}{\omega_1 F_c - \omega_2 F_{LT} + \omega_3 F_d} \tag{16}$$

Secondly, the message and path discovery processes are designed, utilizing the Genetic

Algorithm (GA) to construct the initial routing. During the path discovery phase, multiple paths are obtained via broadcasting to serve as the initial population for the GA. Subsequently, the GA and Ant Colony Optimization (ACO) are designed. In the GA phase, the discovered paths are optimized using operations such as crossover and mutation to maximize the fitness value. Next,

The probability transition formula and pheromone update strategy of the ACO are improved. The routing is optimized through the efficient local search capability of the improved ACO to obtain stable multi-path congestion-free routing.

Finally, a link repair algorithm is proposed to reduce delay overhead caused by path instability and mitigate network congestion. The flowchart of the proposed genetic-ant colony fusion multi-path congestion-free routing optimization algorithm is shown in Fig. 5.

This congestion-free routing process is capable of identifying the optimal route, enhancing network throughput, and reducing packet loss rates and end-to-end delay. Consequently, it ensures the timely delivery of warning messages, with the Network layer transmitting these warning signals to the MAC layer.

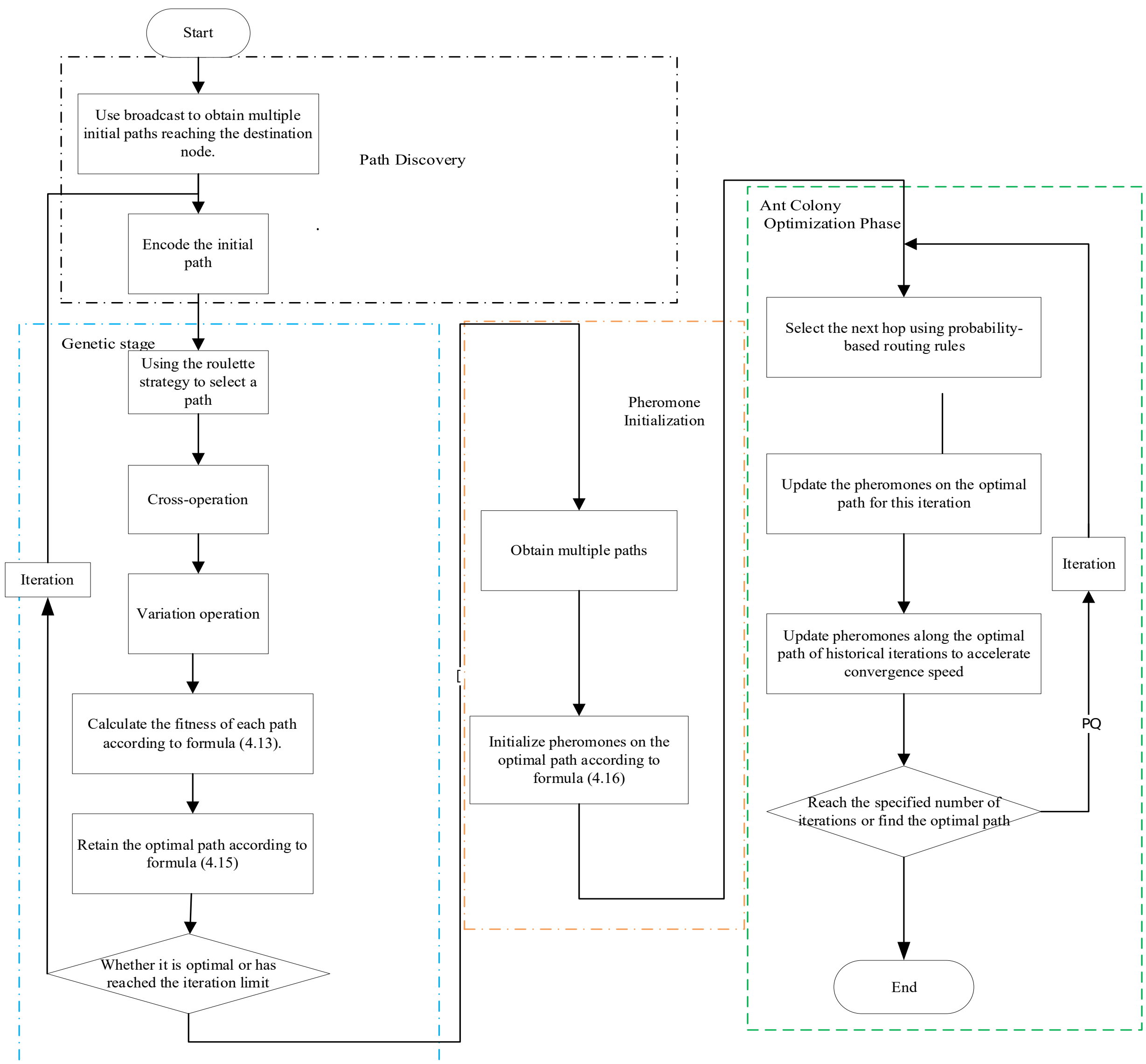


Figure 5. Flowchart of the Genetic-Ant Colony Fusion Congestion-Free Routing Algorithm

3) MAC Layer and Physical Layer

Addressing the synchronization issue, the MAC layer implements clock synchronization for both inter-cluster and intra-cluster nodes. JCSCCV synchronizes the local clocks of all cluster member nodes to a common clock to enhance channel utilization.

JCSCCV sets the severity level based on the collision probability calculated by the Application layer. The larger the ACI, the higher the priority of the information, and the higher the degree of importance and urgency. The data flow weight w_a can be quantified as the urgency of the data currently waiting to be sent, namely the priority of the data flow, as shown in (17).

$$w_a = \frac{ACI + 1}{N_{ACI} + 1} \tag{17}$$

In this context, $N_{ACI} = 4$ represents the number of ACI types. In VANETs, different types of data possess different priorities. According to the importance and urgency of the messages, four types of Access Categories (AC) data are defined, denoted by N_{ACI} . Each AC corresponds to an Access Category Index (ACI), where $ACI \in [0, 1, 2, 3]$. The larger the ACI, the higher the priority of the information, and the greater the degree of its importance and urgency. 1 is added to both the numerator and denominator to avoid zero values. Obviously, since $ACI < N_{ACI}$, the value range for data priority is $w_a \in (0, 1)$.

Given the time sensitivity of warning messages, JCSCCV proposes a three-level priority allocation algorithm to realize the transmission of critical warning messages with higher priority. The first level considers the direction factor and, adopting a method similar to that of the Network layer, utilizes Equation (3) to select relay nodes with higher consistency in motion similarity. The second level differentiates between warning messages and non-warning messages. The third level differentiates warning messages based on varying severity levels, which facilitates the propagation of highly critical warning messages with the highest priority. JCSCCV establishes the severity level based on the collision probability calculated by the Application layer. The larger the ACI, the higher the priority of the information, and the greater the degree of importance and urgency, as shown in Table 1.

TABLE I. SEVERITY LEVEL OF WARNING MESSAGES

L	numerical value
NW	Pc = 0.00
L0	（Pc >= 0.00） and （Pc <= 0.33）
L1	（Pc >= 0.33） and （Pc <= 0.66）
L2	（Pc >= 0.66） and （Pc <= 1.00）

The MAC layer prioritizes warning messages as follows:

Switch (L)

Case: L_0

Wait for an idle time slot

Case: L_1

Request to release a time slot

Case: L_2

Release a reserved time slot via non-warning or low-priority warning messages

End Switch

The aforementioned priority propagation process improves channel utilization and throughput, while reducing the warning message loss rate and end-to-end delay.

JCSCCV adopts the IEEE 802.11p standard at the Physical layer. Once the destination node receives a warning message, it decrypts and authenticates the message, and adopts a safe speed to avoid potential collisions.

IV. SIMULATION AND ANALYSI

A. Simulation Experiment Environment

The simulation environment was established based on the Linux system, incorporating the SUMO and NS-3 simulation software [23]. SUMO was employed to conduct traffic simulation, while NS-3 was responsible for network simulation. The Linux system used was Ubuntu 16.04, with NS-3 version 3.27 and SUMO version 0.32.0. The overall scheme of the simulation platform is shown in Fig. 6.

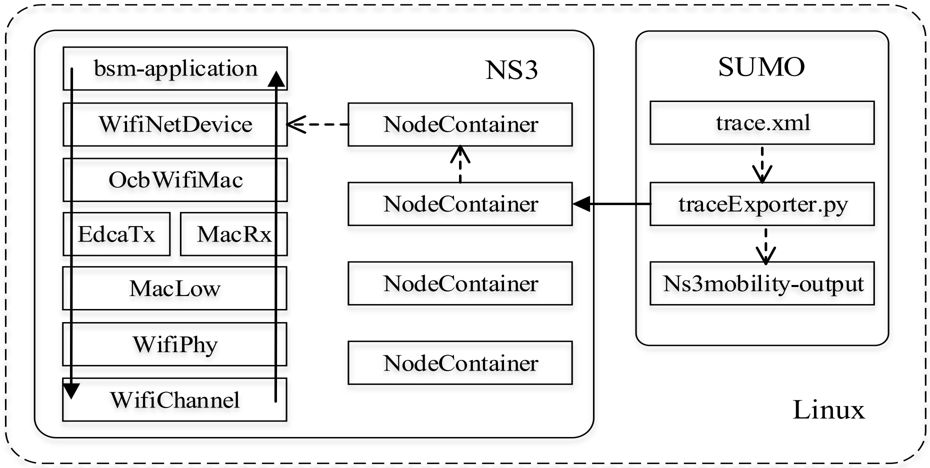


Figure 6. Overall Solution for Simulation Platform

First, a traffic simulation scenario was constructed using SUMO, with the traffic scenario parameters shown in Table 2.

TABLE II. TRAFFIC SIMULATION SCENARIO PARAMETERS

Parameter	Value
Scenario Area	5km*5km
Number of Vehicle Nodes	0-500
Maximum Speed	60m/s
Minimum Speed	0 m/s
Average Speed	0、5、10、20、30、40、50、55 m/s
Road Density	0.00001
Highway Speed Limit	60m/s
Number of Lanes	2

Considering actual urban traffic scenarios, this scenario includes 0-500 vehicle nodes, and the generated scenario possesses randomness; the higher the average speed of the nodes, the more frequently the topology changes. Secondly, the traffic flow information from SUMO was imported as parameters into the NS-3 simulation software to verify the performance of the JCSCCV strategy. The network simulation scenario parameters are shown in Table 3.

TABLE III. NETWORK SIMULATION PARAMETERS

Parameter	Value
Physical Layer	WireliessPhyExt
MAC Layer	802.11
Frequency Band	5.9GHz
Bandwidth	10Mbps
Communication Range	300m
Sensing/Listening Range	600m
Packet Size	1000Bytes
Transport Layer	TCP New Reno

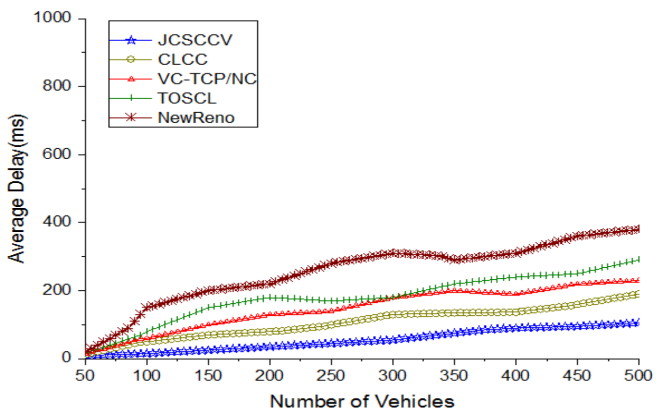
B. Analysis of Experimental Results

This section performs simulation verification on the throughput performance of safety-related applications in VANETs. In high-reliability applications, broadcast transmission with small payload sizes is typically employed. Tests were conducted on frequently occurring data streams with different priorities. Under different vehicle density conditions, the proposed JCSCCV strategy is compared with cross-layer algorithms such as NewReno [11], TOSCL [19], VC-TCP/NC [21], and CLCC [22]. The performance pros and cons of each algorithm are analyzed in terms of average delay, average throughput, packet loss rate, collision probability, packet delivery ratio, and routing overhead.

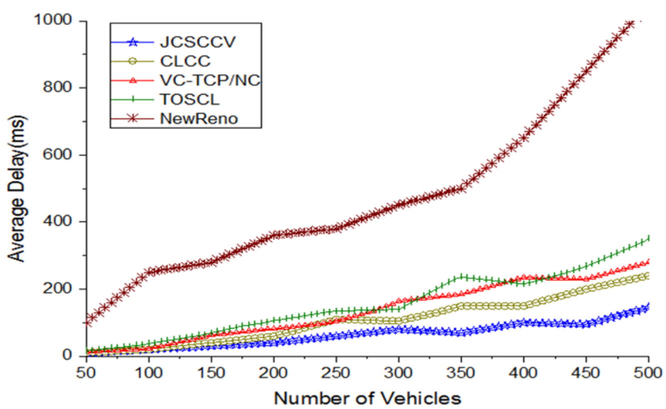
The results verify that the JCSCCV strategy can effectively alleviate network congestion, improve system throughput and stability, reduce delay, and enhance VANET transmission performance in urban scenarios with high traffic density. Its advantages are particularly significant in environments characterized by high-speed node movement, frequent topology changes, and poor channel quality.

1) Comparison of Average Delay

Under different Bit Error Rates (BER), the transmission of information service data, such as dynamic maps, was performed using JCSCCV, NewReno, TOSCL, VC-TCP/NC, and CLCC, respectively. The average delay was compared, as shown in Fig. 7.



(a) The bit error rate is 5%.



(b) The bit error rate is 20%.

Figure 7. Comparison of Average Latency at Different Bit Error Rates

As the number of vehicles increases from 50 to 500, under low Bit Error Rate (BER), the average delay of the JCSCCV strategy increases from 9.5 ms to 105 ms. Under high BER, the average delay increases to a maximum of 150 ms. Under both low and high BER conditions, the average data transmission delay of JCSCCV generally remains within 150 ms with little difference; slight fluctuation in average delay is observed only under high BER.

Whether under low or high BER, the average delay of NewReno is higher than that of the JCSCCV, TOSCL, VC-TCP/NC, and CLCC algorithms. Particularly under high BER, the average delay increases correspondingly from 19. ms to 1055 ms. Overall, congestion strategies using the cross-layer approach result in lower delay than those using the layered approach. The main reason is that NewReno is not adapted to the wireless network environment. Once NewReno detects packet loss, it frequently reduces the congestion window. When the multiplicative decrease mechanism is triggered, NewReno directly reduces the window value to the minimum, resulting in a blind reduction of the congestion window value whenever packet loss occurs, which leads to increased delay.

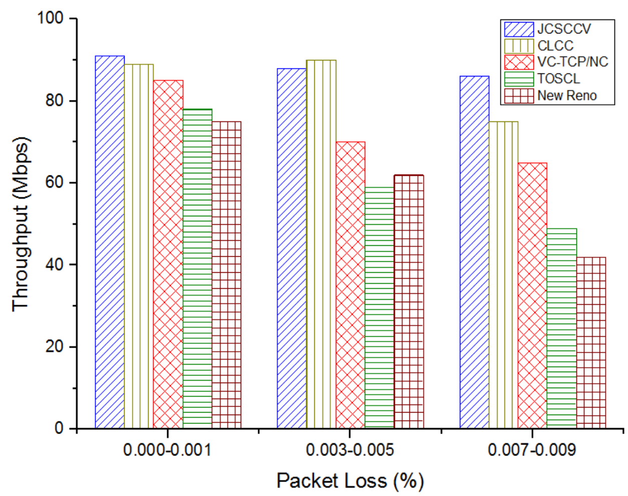
The average delay of the JCSCCV method is consistently lower than that of TOSCL, VC-TCP/NC, and CLCC. This is primarily because the cross-layer congestion algorithm combines the expected queue value, utilizes the utility function U_a^c and the adjustment factor θ to alter the transmission rate, and incorporates the data flow priority weight w_a to adjust the link rate. This reduces the computational delay at the MAC layer, thereby reducing the average network delay.

2) Throughput Comparison

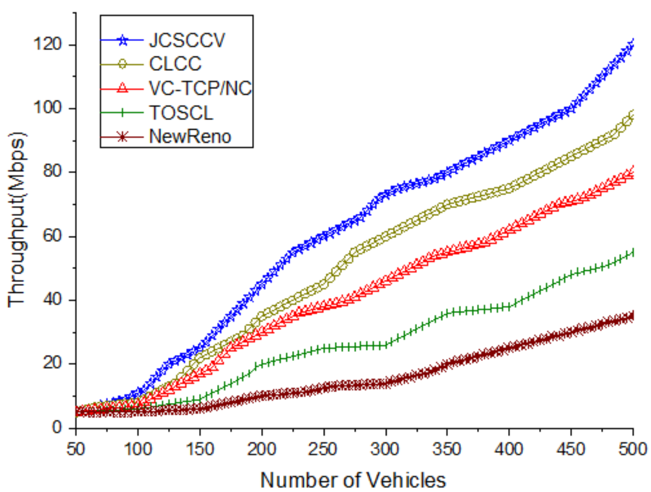
To analyze algorithm performance more effectively, assuming vehicles are in a high-density state, an FTP application was utilized to transmit files ranging from 50 MB to 500 MB within the same simulation environment. The average throughput of JCSCCV was compared with that of NewReno, TOSCL, VC-TCP/NC, and CLCC under different packet loss rates. The experimental results are shown in Fig. 8.

When the packet loss rate is low, the throughput of all five algorithms is relatively high, and the throughput of JCSCCV is superior to that of CLCC, VC-TCP/NC, TOSCL, and NewReno. As the packet loss rate increases, the throughput of all five algorithms decreases to some extent. However, JCSCCV maintains a consistently high level of average throughput during file transmission and remains relatively stable. In contrast, NewReno exhibits a significant drop in throughput as the packet loss rate increases, due to its inability to rapidly distinguish the causes of packet loss.

Subsequently, the average throughput of different strategies under different vehicle densities was compared at higher packet loss rates (0.007-0.009), as shown in Figure 8(b).



(a) When the packet loss rate is low



(b)When packet loss rates are high

Figure 8. Throughput Comparison Chart

As the number of vehicles increases from 50 to 500, the average throughput using JCSCCV increases by 28\%, consistently remaining higher than that of the CLCC, VC-TCP/NC, TOSCL, and NewReno methods; whereas the throughput using NewReno increases by only 7.7\%.

By introducing data flow weight, the JCSCCV algorithm compares the priorities of different data types via this parameter and rationally adjusts the link rate according to the importance and urgency of the messages. This effectively prevents scenarios where the congestion window drops rapidly and frequently, which would otherwise lead to reduced throughput. Compared with NewReno, TOSCL, VC-TCP/NC, and CLCC, JCSCCV exhibits more stable throughput, reflecting a significant enhancement in throughput performance throughout the entire data transmission process.

3) Comparison of Packet Loss Rate

Fig.9 illustrates the packet loss rate performance under different strategies. The results indicate that with 500 vehicles, the packet loss rate of the JCSCCV strategy is 17.5%, whereas that of the NewReno strategy is 57.4%. The packet loss rate of the JCSCCV method is consistently lower than that of the CLCC, VC-TCP/NC, TOSCL, and NewReno strategies. This is primarily because the introduction of a channel backoff mechanism based on Markov chains achieves channel sharing, which effectively reduces packet loss and accommodates the dynamic nature of the network topology.

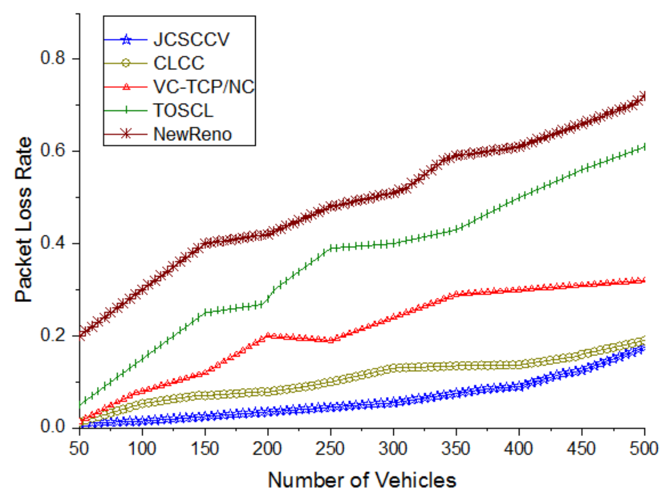


Figure 9. Packet Loss Rate Comparison

4) Comparison of Collision Probability

Fig.10 illustrates the collision probability performance under different strategies. The results indicate that with 500 vehicles, the collision probability of the JCSCCV strategy is 0.19, whereas that of the NewReno strategy is 0.95.

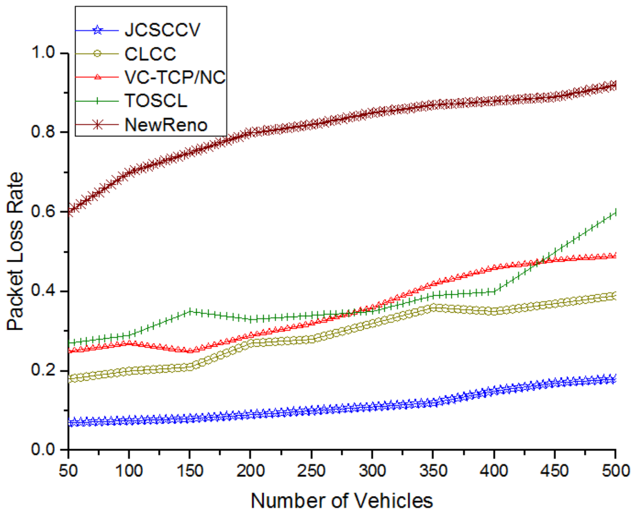
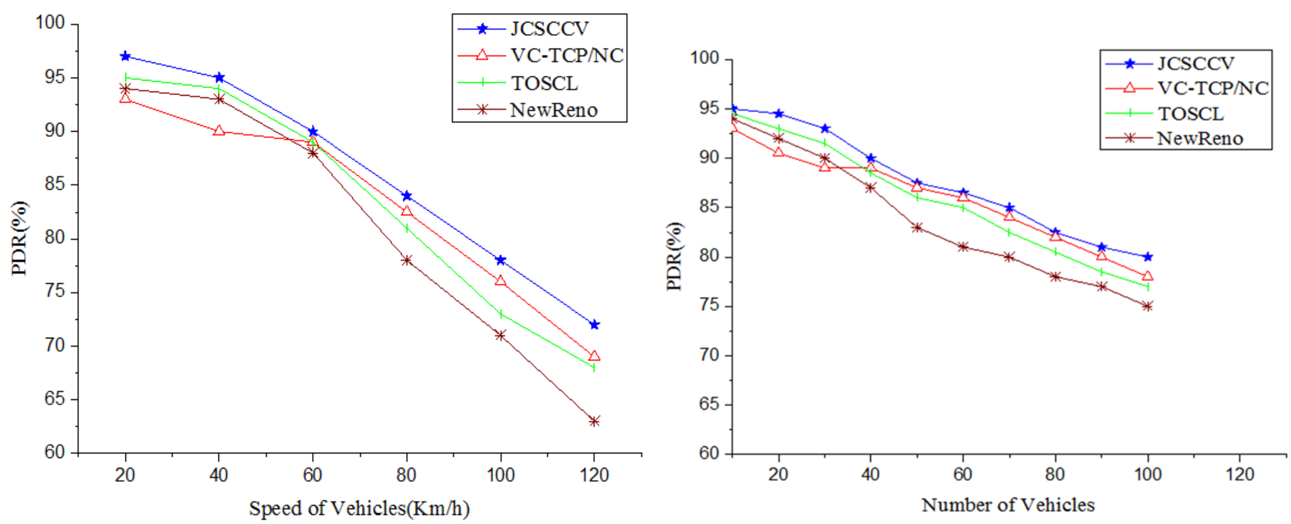


Figure 10. Conflict Probability Comparison

An increase in the number of transmitting nodes within the same channel leads to an increase in the data required to be transmitted per unit time and increased channel occupancy. As the number of vehicles within the communication range increases, the probability of nodes successfully transmitting data decreases rapidly, while the collision probability increases with the growth in vehicle numbers. The collision probability of the JCSCCV method is consistently lower than that of the CLCC, VC-TCP/NC, TOSCL, and NewReno strategies. The mode of processing vehicle node information via multi-threading facilitates synchronized information interaction between VANET vehicles, thereby reducing collisions caused by channel contention.

5) Packet Delivery Ratio (PDR)

With the number of vehicle nodes set to 20, as the vehicle speed increases, a continuous downward trend is observed in the PDR of the routing protocols. Similarly, under the condition of a vehicle speed of 40 km/h, as the number of vehicles increases, the PDR of all algorithms exhibits a downward trend, as shown in Fig. 11.



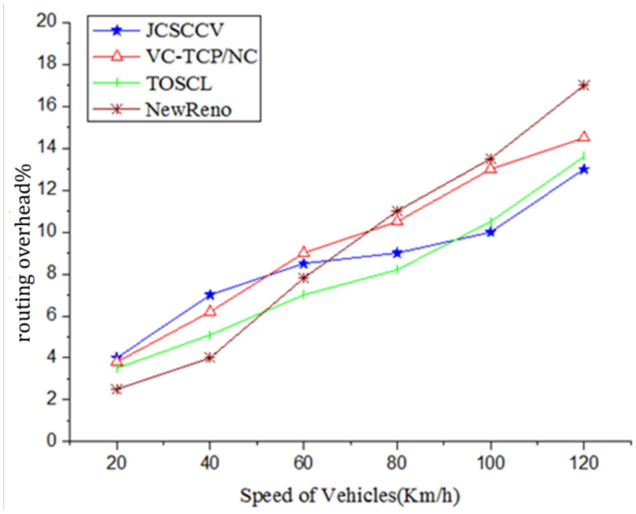
(a)Different speeds

(b) Different quantities

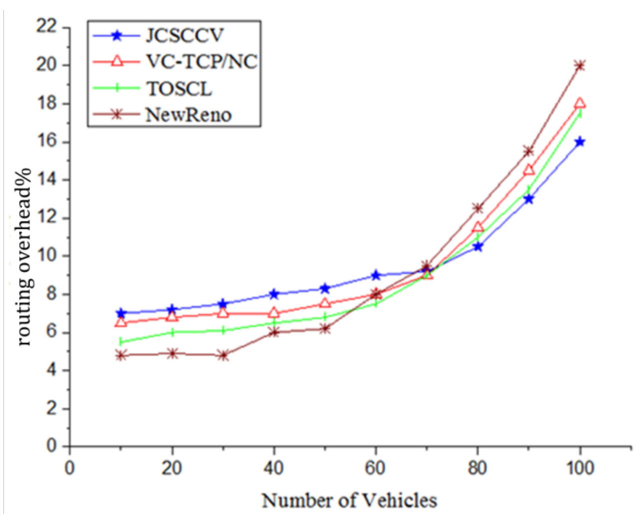
Figure 11. PDR Comparison

Although JCSCCV is similarly affected by the increase in speed, its PDR performance is 5.1% higher than that of the VC-TCP/NC protocol when the vehicle speed reaches 120 km/h, exhibiting superior performance advantages. Particularly, when the number of vehicles reaches 100, the PDR of the JCSCCV algorithm is 4.3%, 3.1%, and 1.8% higher than that of NewReno, TOSCL, and VC-TCP/NC, respectively. This result highlights the ability of the JCSCCV algorithm to maintain a high data transmission success rate even when dealing with high-density vehicle scenarios. This indicates that the JCSCCV algorithm demonstrates strong robustness and adaptability to variations in network density.

6) Routing Overhead



(a) Different speeds



(b) Different quantities

Figure 12. Routing Overhead Comparison

As shown in Fig. 12(a), the routing overhead of the four routing algorithms increases as the node speed increases. When the node speed is low, the GAAC routing algorithm proposed in this paper incurs relatively higher protocol overhead. However, as the speed increases, network link instability intensifies. Since the AODV and AOMDV algorithms utilize the minimum hop count without considering link stability, their routing overhead increases. Consequently, the gap in routing overhead among the four routing algorithms continuously narrows. Because the GAAC algorithm considers link stability when establishing routes and simultaneously employs a multi-path mechanism, it reduces the delay associated with waiting for route reconstruction, thereby lowering routing overhead.

As shown in Fig.12(b), the routing overhead of the four routing algorithms increases as the node speed increases. When the node speed is low, the routing overhead of AODV is relatively the lowest, while the GAAC algorithm proposed in this paper incurs relatively higher overhead. However, as the number of vehicles increases, the routing overhead of the GAAC algorithm grows at a slower rate. When the number of vehicles exceeds 70, its routing overhead gradually becomes lower than that of the AODV, EAA, and AOMDV algorithms. The GAAC protocol improves the load capacity and stability of the network at the cost of routing overhead.

V. CONCLUSIONS

This paper addresses the network congestion caused by the complexity of the communication environment in VANETs and the difficulty of rapid information sharing under traditional layered approaches by proposing a Joint Congestion Strategy based on Cross-layer Collaboration (JCSCCV). The JCSCCV algorithm utilizes the interrelationships among the Physical layer, MAC layer, and Transport layer to establish an overall architecture for the joint congestion strategy based on cross-layer collaboration.

Targeting the issue where cluster head nodes are prone to node-level congestion in traditional single-cluster-head routing topologies, a multi-criteria double cluster head dynamic clustering strategy based on motion similarity is proposed. This strategy measures the degree of congestion based on the buffer queue status of cluster heads and schedules task traffic to alleviate cluster head congestion. Targeting link-level network congestion caused by unbalanced inter-cluster routing traffic, a multi-path congestion-free routing algorithm based on genetic-ant colony fusion is proposed. The genetic-ant colony fusion meta-heuristic algorithm is utilized to construct multi-path congestion-free routing, thereby avoiding network congestion caused by link instability and uneven load distribution.

Finally, the JCSCCV algorithm was verified through simulation. Compared with algorithms such as CLCC, VC-TCP/NC, TOSCL, and NewReno, JCSCCV demonstrates superior performance in terms of average delay, average hop count, and system throughput, possessing high application value. The results verify that the JCSCCV strategy can effectively alleviate network congestion in urban scenarios with high traffic density.

References

[1] Li X, Noh H, Cottam A, et al. Leveraging connected-vehicle data to identify and analyze park service areas [J]. Cities, 2026,170106665-106665. DOI: 10.1016/J. CITIES. 2025. 106665.

[2] Zhang C, Chang X, Jia X, et al. Sybil-resistant and privacy-preserving authentication based on short-term pseudonym for internet of vehicles. [J]. Scientific reports, 2025, DOI:10.1038/S41598-025-28854-1.

[3] YUE Sheng. Research on Internet of Vehicles Technology [J]. Digital Communication World, 2020 (11):121-122.

[4] Saad M M, Khan M T R, Shah S H A, et al. Advancements in vehicular communication technologies: C-V2X and NR-V2X comparison[J]. IEEE Communications Magazine, 2021, 59(8): 107-113.

[5] Li H, Liu X, Han Z. Minimum Platoon Set to Implement Vehicle Platoons in the Internet of Vehicles Environment [J]. Sensors, 2025, 25(22): 7066-7066. DOI:10.3390/S25227066.

[6] Wang Junhua, Zhu Kun, Hossain E. Green Internet of Vehicles (IoV) in the 6G era: Toward sustainable vehicular communications and networking[J]. IEEE Transactions on Green Communications and Networking, 2021, 6(1): 391-423.

[7] Rakhi S, Shobha K R. An enhanced priority-based multi-hop clustering algorithm for vehicular ad hoc networks[J]. International Journal of Communication Networks and Distributed Systems, 2024, 30(1): 42-69.

[8] Liang C, Yichen J, Tianjiao X, et al. Cross-layer cooperative offloading in vehicular edge computing networks[J]. Vehicular Communications,2023,42 DOI:10.1016/J. VEHCOM.2023.100624.

[9] Li J, Li X, Zhang R. Cross-layer collaboration handoff mechanism based on multi-attribute decision in mobile computation offloading[J]. Soft Computing, 2019,23(1):323-341. DOI: 10.1007/s00500-017-2881-3.

[10] ZHOU H, XU W, CHEN J, et al. Evolutionary V2Xtechnologies toward the Internet of vehicles: Challenges and opportunities[J]. Proceedings of the IEEE, 2020, 108(2): 308-323.

[11] FLOYD S, HENDERSON T. The NewReno modification to TCP's fast recovery algorithm: RFC2582[A/OL]. [2020-07-01]. https://dl.acm.org/doi/pdf/10.17487/RFC2582.

[12] HA S, RHEE I, XU Lisong. Cubic: a new TCP-friendly high-speed TCP variant [J]. ACM SIGOPS Operating Systems Review, 2008, 42(5): 64-74.

[13] BRAKMO L S, O'MALLEY S W, PETERSON L L. TCP Vegas: new techniques for congestion detection and avoidance[C] //Proceedings of the Conference on Communication Architectures, Protocols and Application. New York, USA: ACM, 1994:24-35.

[14] MASCOLO S, CASETTI C, GERLA M, et al. TCP westwood: bandwidth estimation for enhanced transport over wireless links [C] //Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MOBICOM). New York, USA: ACM, 2001:287-297.

[15] Hemmati A, Zarei M. UFC3: UAV-Aided Fog Computing Based Congestion Control Strategy for Emergency Message Dissemination in 5G Internet of Vehicles[J]. Automotive Innovation, 2024, (prepublish):1-17. DOI:10.1007/S42154-024-00284-1.

[16] Hui W, Haoyu L, Yuan Z. An Intelligent Congestion Control Strategy in Heterogeneous V2X Based on Deep Reinforcement Learning[J]. Symmetry,2022,14(5):947-947. DOI:10.3390/SYM14050947.

[17] Mitra S, Mondal A. Joint Congestion Control Strategy During V2V Communication Among Authentic Vehicles in VANET. [J]. Wireless Personal Communications,2014,79(1):43-67. DOI:10.1007/s11277-014-1840-x.

[18] Gola K K, Khan G, Kumar S, et al. Intelligent node identification and dynamic clustering for underwater acoustic sensor networks [J]. Discover Computing, 2025,28(1):251-251. DOI:10.1007/S10791-025-09739-3.

[19] Ning F, Li Z, Lu J, et al.3D CAD model dynamic clustering based on inertial feature encoder[J]. Applied Soft Computing, 2025, 182113627-113627. DOI:10.1016/J. ASOC.2025.113627.

[20] Sabri M, Verde R, Balzanella A, et al. A Novel Classification Algorithm Based on the Synergy Between Dynamic Clustering with Adaptive Distances and K-Nearest Neighbors[J]. Journal of Classification,2024, 41(2):264-288. DOI:10.1007/S00357-024-09471-5.

[21] ChoksiA, ShahM. Machine learning based centralized dynamic clustering algorithm for energy efficient routing in vehicular ad hoc networks[J]. Transactions on Emerging Telecommunications Technologies, 2023,35(1): DOI:10.1002/ETT.4914.

[22] Gosai M A D, Raval M L. Grid Based Multipath with Congestion Avoidance Routing Protocol[J]. Journal of Trend in Scientific Research and Development, 2017,1(3): DOI:10.31142/ijtsrd72.

[23] Devangavi D A, Gupta R. Adaptive Congestion Controlled Multipath Routing in VANET: A Multiagent Based Approach[J]. International Journal of Agent Technologies and Systems (IJATS),2017,9(1):43-68. DOI:10.4018/IJATS.2017010103.