Dynamic Clustering Hierarchy for Vehicular Communication (DCHVC)

A Multi-Criteria Dual-Head Strategy Based on Motion Similarity

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

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

***Abstract*—The number of vehicles with network access keeps growing, and such growth brings more serious channel congestion and node overload problems to VANET systems, these problems will become much worse if the network only uses one cluster head node, to solve the above existing defects, this paper puts forward a dynamic clustering scheme named DCHVC, this scheme takes the motion similarity of vehicles as a reference standard, and uses a multi-index dual cluster head selection rule to raise the efficiency of vehicle data transmission, the whole scheme first divides stable vehicle clusters according to the similarity of vehicle speeds and the space distance between vehicles, the scheme also builds a network structure with two cluster heads, the main cluster node and the backup cluster node are selected through the CRITIC-TOPSIS multi-index evaluation method, this selection process will take both relative speed differences of vehicles and sustainable time of communication links into consideration, the system sets up a congestion control module based on cache queue occupancy, this module distributes data transmission tasks between the two cluster heads dynamically, meanwhile, adjustable cluster splitting and merging steps are used to keep the overall working state of clusters all the time, this paper builds a simulation environment by combining two simulation tools NS3 and SUMO, under test scenes with different vehicle moving speeds and different vehicle quantities, the operation performance of DCHVC is compared with three existing schemes including VWCA, KMRP and DCM, when the total number of vehicles in the test scene reaches 300, the DCHVC scheme can reach a network throughput of about 800 kbps, this value is 8.1% higher than the VWCA scheme, the average end-to-end transmission delay of this scheme can drop to around 0.69 seconds, which has a 37.3% delay reduction compared with VWCA, all data from simulation tests can prove that the DCHVC scheme can work steadily in V2X scenarios where vehicle positions change frequently, it can effectively improve the running stability of clusters and balance the data transmission load of every node.**

Keywords-Vanets; Double Cluster Head; Dynamic Clustering; Multi-Criteria Decision-Making; Congestion Control;

I. INTRODUCTION

Intelligent transport systems keep developing faster than ever before, and VANET technology takes a more important place to make road travel safer and more efficient, but the constant movement feature of such networks brings many tough troubles, such as frequent network structure adjustments, unstable signal connections and lasting channel blockages, all these troubles will stop low-delay vehicle business from working normally, traditional wireless communication rules are made for networks with fixed node positions, these rules cannot fit the fast-changing running state of VANET systems, and they easily lead to mass broadcast signals, channel resource competition and data packet collision, these faults will create link congestion, and lower the service

https://doi.org/10.58244/ijanmc.263992

quality of key safety functions such as automatic driving control and vehicle collision prevention, many previous research works regard cluster division as a feasible way to solve these problems, cluster technology separates all vehicles into small groups which are easy to manage, it cuts down the amount of control signals and limits the bad influence caused by network structure changes, every cluster will pick one cluster head node to manage internal signal transmission, collect data from other vehicles in this group and send messages between different clusters, this layered structure can cut useless data transmission and fully utilize limited network bandwidth, still, cluster division cannot fix all defects completely, the cluster head node needs to handle most data calculation and forwarding work, when there are too many vehicles or data tasks, the limited processing capacity of this node will be used up, this situation is called node congestion by most researchers, once node congestion happens, data packets will pile up in the storage space of cluster heads, signal transmission delay will rise sharply, and the storage space will even overflow, the unstable cluster structure will make this problem worse, vehicles keep moving all the time, so clusters will break apart and rebuild constantly, every rebuild process will waste computing resources and send lots of extra control signals, these extra data loads will put more pressure on the whole network and make congestion and delay even more serious [8][12][14][17], most existing papers point out a core design requirement, new cluster algorithms need to build long-lasting vehicle groups and avoid making cluster heads become network bottlenecks, based on this core requirement, this paper designs a new cluster division method which can keep stable cluster structure and balance data load at the same time, the designed scheme is named DCHVC, it judges vehicle movement similarity and uses two cluster head nodes to dynamically separate data transmission pressure, this paper has four main research innovations, first, a cluster dividing algorithm is put forward, it judges vehicle movement similarity by vehicle speed matching degree and vehicle space distance, it can build clusters that will not break easily and reduce extra resource loss from repeated cluster reconstruction, second, this paper uses CRITIC-TOPSIS tool to finish the selection of two cluster heads, this objective and standardized selection method picks the main cluster head and backup cluster head properly, it lowers the risk that the whole cluster stops working when one single cluster head fails, third, a signal scheduling module considering congestion state is designed, it checks the storage space usage of two cluster heads and distributes data tasks flexibly, it solves the uneven load problem that exists in most former dual-cluster-head schemes, fourth, plenty of simulation experiments are carried out, the test results prove that DCHVC has better performance than other classic algorithms in many evaluation standards.

II. RELATED WORK

Lots of research papers about VANET cluster division have been published in the last ten years, researchers in this field all agree that cluster technology is a basic method to make V2X communication expandable and stable [1][2], most early research works only studied cluster structures with just one cluster head, every single group relies on one node to finish all coordination and data forwarding work, the Weighted Clustering Algorithm, short for WCA, is one well-known single-cluster-head solution, many people choose to use it because it runs without central control and is simple to set up, the main logic of WCA is to calculate a total weight value for every vehicle node, the calculation takes node connection quantity, moving state and vehicle spacing as reference standards, then the node with the smallest weight value will be picked as the cluster head, this algorithm is easy to understand and does not take much computing power, but it has an obvious shortcoming, all weight values are fixed and set by past experience, so the algorithm cannot adjust itself when road traffic changes, there is an even bigger defect for single-cluster-head structures, the only cluster head has to take all communication tasks inside the cluster, if more vehicles appear or data transmission tasks increase, the cluster head will quickly run out of processing and forwarding ability, this node will become a network bottleneck and trigger node congestion, to solve the bottleneck problem brought by only one cluster head, recent scholars start to design structures with two or more cluster heads, the basic logic of this design is simple, two cluster heads share all coordination and data forwarding work, so one single node will not be overloaded easily, many two-cluster-head algorithms have been put forward, most of them set a main cluster head to handle most work and a spare cluster head to offer auxiliary support, however, existing research materials show a common defect in these algorithms, most two-cluster-head plans split data flow through fixed distance rules, they do not check the real task load of each cluster head, this careless design will cause uneven data pressure when vehicles gather unevenly inside a cluster, one cluster head still bears most transmission work, the extra spare cluster head cannot fully play its role, another widespread shortage of these algorithms is that they ignore vehicle movement consistency when dividing initial clusters, many algorithms do not make use of the rule that vehicles moving in the same direction will stay close for a long time, and they lack adjustable maintenance modules that can react to real-time congestion signals, these missing functions make two-cluster-head clustering perform poorly in fast-changing, heavily blocked vehicle communication environments.

III. MULTI-CRITERIA DUAL-CLUSTER HEAD DYNAMIC CLUSTERING STRATEGY BASED ON MOTION SIMILARITY

This paper puts forward a DCHVC clustering strategy system, and Fig.1 shows the overall structure of this system, this strategy mainly contains four core working steps, the first step is grouping vehicles by their movement similarity, the second step is selecting primary and secondary cluster heads with multiple evaluation standards, the third step is managing data congestion between the two cluster heads, the last step is carrying out real-time maintenance for all vehicle clusters.

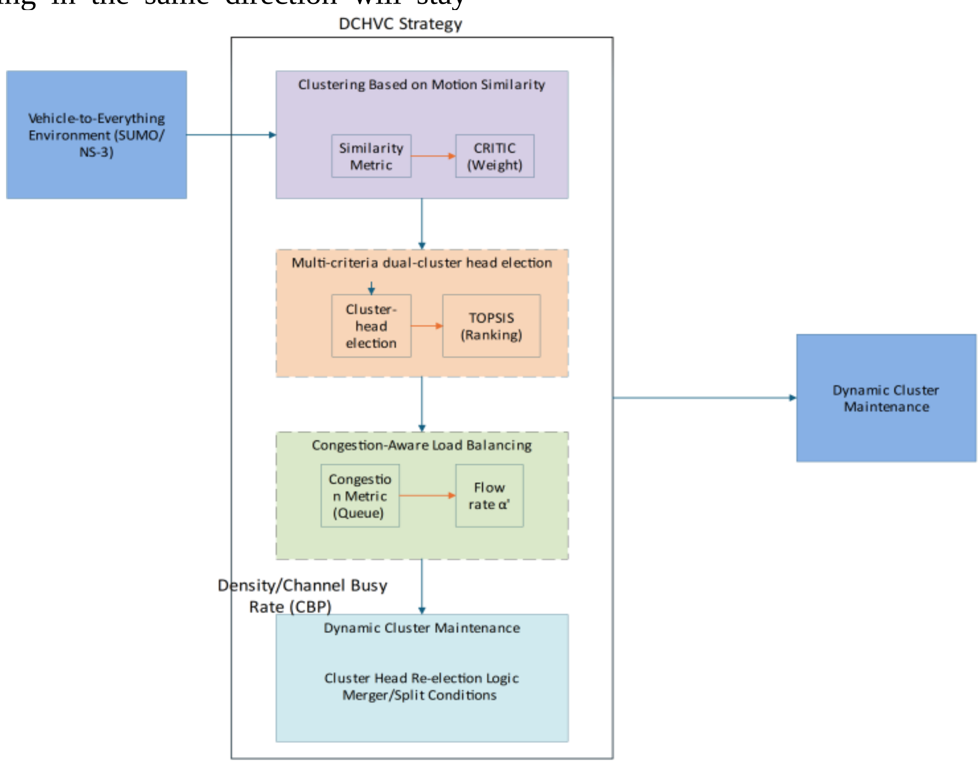

Figure 1. Dynamic Clustering Strategy Framework Diagram

A. Clustering Algorithm Based on Motion Similarity

To build vehicle clusters that can keep stable for a long time in VANET systems, researchers cannot only judge based on vehicle locations captured at a single moment, people also need to consider the predicted moving paths of every vehicle node, the service life of each cluster mainly depends on whether all vehicles inside it will keep consistent moving states later, this paper uses two matching judging standards to finish cluster grouping work, which are the space distance between vehicles and the consistency of vehicle driving speeds, vehicles that stay close to each other and share similar driving directions and driving speeds will be divided into the same cluster, two measurement indicators are set at first, one is the position proximity value siml, the other is the speed consistency value sim_v, the calculation formula used to work out the position proximity siml is displayed in the following formula (1).

$$\mathrm{siml}_{ij} = \begin{cases} \varepsilon^{-\mathrm{id}_j} & \mathrm{d}_j \le \mathrm{R} \\ 0 & \mathrm{d}_j > \mathrm{R} \end{cases} \tag{1}$$

Here, dij denotes the Euclidean distance between vehicles i and j, R represents the maximum communication range of the vehicle, and k is the sensitivity coefficient. The formula for calculating speed consistency sim_v is shown in (2).

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

In this formula, vi stands for the velocity vector of vehicle i, and vj represents the velocity vector of vehicle j separately, after finishing the above two index calculations, the motion similarity value Simij between vehicle node i and vehicle node j can be calculated through weighted summation, the specific calculation equation of Simij is presented in the subsequent formula (3).

$$\mathrm{sim}_{ij} = \omega_1 \mathrm{sim}_{ij} + \omega_v \mathrm{simv}_{ij} \tag{3}$$

The weights ω_l and ω_v satisfy $\omega_l + \omega_v = 1$. This

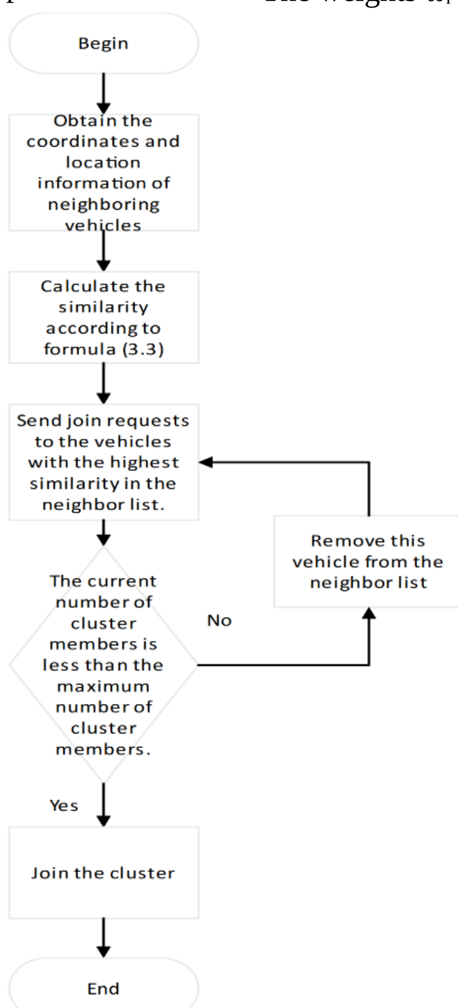

Figure 2. Flowchart of Clustering Based on Motion Similarity

This paper attaches more importance to the distance between vehicles, so we set the weight coefficient ω_l to 0.6 and the weight coefficient ω_v to 0.4, after we calculate the motion similarity value of every pair of vehicle nodes, we use the agglomerative hierarchical clustering algorithm to split all vehicles into different clusters, Fig.2 displays the complete execution flow of this algorithm.

This algorithm regards every single vehicle as a separate cluster at the very start, then it repeatedly combines two clusters that own the largest motion similarity value, if the total number of vehicles inside one cluster goes over the pre-set upper limit, this cluster will no longer join any follow-up merging steps, this grouping way can automatically create stable clusters with proper amounts and sizes, and it prevents the whole network performance from dropping due to too many clusters or too few clusters.

B. Multi-Criteria-Based Dual Cluster Head Election

This paper puts forward a two-cluster-head structure with main and backup cluster heads to solve the performance bottleneck brought by only one cluster head, the whole selection process of cluster heads includes two separate steps, the first step is to pick out all vehicle nodes that can become candidate cluster heads, the second step uses the combined CRITIC-TOPSIS evaluation algorithm to select the most suitable main cluster head and backup cluster head from these candidates.

We need to pick proper vehicle nodes to act as cluster heads so that we can build communication clusters with stable and reliable operation, this section uses several different measurement indicators to judge candidate cluster heads and turns the whole cluster head selection task into an optimization solving problem.

1) Relative velocity difference (v_{si})

The driving speed of vehicles is a key factor that decides whether clusters can stay stable, we have to take the relative speed between candidate cluster heads and other vehicles in the same cluster into account when choosing cluster heads, every vehicle can receive speed data of all nearby vehicles within its communication coverage through broadcast signals at every fixed time slot, we first calculate the average speed difference Rvi between vehicle node i and its surrounding neighbor vehicles, and the corresponding calculation formula is shown in the following equation (4).

$$Rv_i = \frac{1}{N_{nbr}-1}\sum_{j=1}^{v_i-1}|v_j - v_i| \tag{4}$$

Vehicle j is treated as a neighboring vehicle of vehicle i, vi means the driving speed of vehicle i and v_j stands for the driving speed of vehicle j separately, N_{nbr} refers to the total count of vehicles inside the cluster, we define the relative speed difference vs_i to measure how close the speed of vehicle i is to the speeds of other vehicles in the same cluster, the calculation formula of vs_i is given in the corresponding equation (5).

$$VS_i = \frac{Rv_i}{v_i} \tag{5}$$

The smaller the vs_i value, the closer the speed of vehicle node i is to the average speed of its neighbors, and the better the neighborhood stability of vehicle node i.

2) Link Availability Time

We make a basic assumption here, two vehicles can keep stable direct communication if their mutual distance is smaller than the preset threshold R, LR_{Ij} stands for the predicted duration that the communication link between vehicle i and vehicle j can stay usable, Δv_{ij} represents the relative moving speed of these two vehicles, we do not take real-time speed changes of vehicles into account while selecting cluster heads, the formula used to calculate LRij is listed in the corresponding equation (6).

$$LR_{ij} = \frac{R_{ij}}{\Delta V_{ij}} \tag{6}$$

Where R_{ij} is the relative transmission range between vehicles i and j. When both vehicles are traveling in the same direction, R_{ij} is

$$R_{ij} = \begin{cases} R + d_{ij}, & v_j > v_i \\ R - d_{ij}, & v_j < v_i \end{cases} \tag{7}$$

In this formula, d_{ij} means the distance between two vehicles, and this distance value is calculated by Euclidean distance formula, if vehicle j runs in front of vehicle i, d_{ij} will be set as a positive number, if vehicle j falls behind vehicle i, d_{ij} will become a negative number, we take the shortest usable time of communication links connecting to surrounding neighbor vehicles as one judgment standard to pick cluster heads in the selection process.

3) Average Neighbor Distance ($dnbr_i$)

In vehicle communication networks, the space gap between vehicles will directly change the quality of signal transmission, cluster head nodes are expected to stay near the middle area of the whole cluster, or keep the smallest average distance to all other vehicles inside the group, we use the indicator d_{nbr_i} which stands for average neighbor distance to judge the distance relation between one vehicle and its surrounding nodes, and the relevant computing formula is shown in the corresponding equation (8).

$$d_{\text{nbr}_i} = \frac{1}{N_{\text{nbr}}} \sum_{j=1}^{N_{\text{nbr}}} d_{ij} \tag{8}$$

The smaller the average neighbor distance d_{nbr_i}, the lower the overall communication overhead within the cluster, resulting in higher network performance.

This method takes three different evaluation indicators into full consideration, which are relative speed difference vs_i, shortest usable link duration LR and average distance d_{nbr_i} to nearby vehicles, after combining all these standards to pick cluster heads, the communication links between the chosen cluster head and other vehicles in the group can stay stable for a longer time, fewer repeated cluster head selection operations will generate extra control data, the wireless channel resources can be used more fully, and the overall network congestion situation will also get relieved.

4) Electing Cluster Heads Using CRITIC-TOPSIS

We use the classic TOPSIS multi-standard decision method to find out the most suitable vehicle node to act as cluster head, the relative speed difference vs_i between candidate node i and other group vehicles can show how stable their communication links are, and we mark this indicator as vi1, the minimum available time of communication link LR is recorded as v_{i2}, and the average distance d_{nbr_i} from nearby vehicles is set as vi3, if there are n candidate nodes for cluster head, we build a decision matrix M, and the specific expression of this matrix is displayed in the corresponding formula (9).

$$M = \begin{pmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ \vdots & \vdots & \vdots \\ v_{n1} & v_{n2} & v_{n3} \end{pmatrix} \tag{9}$$

We perform standardization processing on all evaluation indicators, so every single judging index can exert balanced influence in the selection process, two indicators named relative speed difference vs_i and average neighbor distance d_{nbr_i} have a feature that lower numerical values represent better performance as cluster head, Formula (10) is used to standardize these two indexes, while Formula (11) handles the standardization calculation of the minimum usable duration LR of communication links.

$$v_{ij}^* = \frac{\max(\mathbf{v}_{ij}) - \mathbf{v}_{ij}}{\left[\max(\mathbf{v}_{ij}) - \min(\mathbf{v}_{ij})\right]}, \quad j = 1,3 \tag{10}$$

$$v_{ij}^* = \frac{v_{ij} - \min(v_{ij})}{\left[\max(v_{ij}) - \min(v_{ij})\right]}, \quad j = 2 \tag{11}$$

$v^*{}_{ij}$ represents the normalized value of the corresponding indicator. After normalizing each vehicle node, the resulting normalized decision matrix M^* is expressed as shown in Equation (12).

$$M^* = \begin{pmatrix} V_{11}^* & V_{12}^* & V_{13}^* \\ V_{21}^* & V_{22}^* & V_{23}^* \\ \vdots & \vdots & \vdots \\ V_{n1}^* & V_{n2}^* & V_{n3}^* \end{pmatrix} \tag{12}$$

All indicators are normalized to eliminate the interference caused by inconsistent dimensions and value ranges. After constructing the normalized matrix, the CRITIC algorithm is adopted to calculate the weight of each evaluation index. CRITIC is an objective weighting algorithm oriented to multi-criteria decision-making, which allocates weights according to the degree of differentiation and correlation conflict between various indicators. Let I_j denote the independence of the j-th indicator with respect to all other indicators, and its computational formula is given in Equation (13).

$$I_j = \sum_{i=1}^{n} (1 - r_{ij}) \tag{13}$$

In the above formula, r_{ij} stands for the Pearson correlation coefficient between indicator i and indicator j. This coefficient quantifies the degree of variation brought by combining different indicators as well as the independence among criteria. The information content C_j of each evaluation indicator is calculated via Equation (14).

$$C_j = \sigma_j I_j, \quad j = 1, 2, 3, \ldots, n \tag{14}$$

Where σj denotes the standard deviation of the j-th criterion. A larger standard deviation means the criterion has more obvious differentiation among samples and contains more information, so it should be assigned a higher weight. If a criterion is more independent of other criteria, it carries more unique information and thus has higher importance. The weight of each criterion is determined by the ratio of its own information content to the sum of all criteria's information contents, as shown in the weight calculation Formula (15).

$$W_j = \frac{C_j}{\sum_{j=1}^{n} C_j}, \quad j = 1, 2, 3, \ldots, n \tag{15}$$

The weight matrix for each criterion obtained using the CRITIC algorithm is multiplied by the normalized decision matrix M^* to yield the weighted normalized decision matrix Z, as shown in Equation (16).

$$Z = M^* \cdot W = \begin{pmatrix} v_{11}^* & v_{12}^* & v_{13}^* \\ v_{21}^* & v_{22}^* & v_{23}^* \\ \vdots & \vdots & \vdots \\ v_{n1}^* & v_{n2}^* & v_{n3}^* \end{pmatrix} \cdot \begin{pmatrix} W_1 \\ W_2 \\ W_3 \end{pmatrix} = \begin{pmatrix} V_{11} & V_{12} & V_{13} \\ V_{21} & V_{22} & V_{23} \\ \vdots & \vdots & \vdots \\ V_{n1} & V_{n2} & V_{n3} \end{pmatrix} \tag{16}$$

After standardizing the data for each indicator, the positive ideal solution PIS^+ and negative ideal solution NIS^- are calculated using Equations (17) and (18).

$$PIS^+ = (V_1^+, V_2^+, V_3^+) = \left[\max_i V_{ij} \mid j = 1, 2, 3 \right] \tag{17}$$

$$NIS = (V_1, V_2, V_3) = \left[\left(\min V_{ij} \mid j = 1, 2, 3 \right) \quad \text{and} \quad i \in [1, n] \right] \tag{18}$$

After obtaining the positive ideal solution PIS^+ and the negative ideal solution NIS^-, the Euclidean distance is used to calculate the separation distances S^+ and S^- between each alternative and PIS^+ and NIS^-, respectively.

$$S^+ = \sqrt{\sum_{j=1}^{n} (PIS^+ - V_{ij})^2} \tag{19}$$

$$S = \sum_{i=1}^{3} \sqrt{\frac{(n+1)}{j=1} \left(V_{ij} - NIS \right)^2} \tag{20}$$

In the last step, we compute the proximity coefficient C. This coefficient reflects the proximity degree between each candidate cluster head node and the ideal optimal solution, so we can sort all candidate nodes in descending order of this value. The corresponding calculation formula is given in Equation (21).

$$C = \frac{S^-}{S^+ + S^-} \tag{21}$$

Sort all candidate cluster heads by proximity coefficient C from highest to lowest and select the top two nodes as primary and secondary cluster heads.

C. Congestion Control Between Primary and Secondary Cluster Heads

After selecting the primary and secondary cluster heads, this paper puts forward a packet scheduling algorithm based on the cache queue status of the two cluster heads. The algorithm realizes load balancing between dual cluster heads and relieves congestion at the individual node level.

Node congestion level measurement the congestion level of a node is measured using the length of its cache queue. The calculation formula for ϕ_t is shown in (22).

$$\varphi_t = (1 - w_q) \times \varphi_{(t-1)} + w_q \times \frac{Q_{\text{occupy}}(t)}{Q_{\text{sum}}} \quad (22)$$

Where $Q_{occupy}(t)$ refers to the occupied buffer capacity at time t; Q_{sum} is the total buffer capacity; $\phi_{(t-1)}$ stands for the congestion degree of the previous time slot; w_q denotes the weight coefficient for balancing historical congestion state and real-time congestion state. The closer the value of ϕ_t is to 1, the more serious the congestion of this node.

This packet distribution method draws lessons from the AQM queue control method, researchers set two boundary values ϕ_{min} and ϕ_{max}, these two values can separate the jam degree of each network node into three different conditions, including no data jam state, slight data jam state and heavy data jam state, the specific distribution rules are listed below:

When the congestion states of the two cluster heads diverge, dynamically migrate part of CH1's communication tasks to CH2 until their congestion states converge.

If the two cluster management nodes fall into the identical jam grade, the data packet distribution process will use the rules of backpressure transmission, all internal cluster terminal nodes pick the forwarding cluster node for next data jump by relying on the judgment formula PCH(t), the computational expression of this function is displayed in formula (23);

$$P_{CH}(t) = w_1\varphi_{CH1}(t) - w_2\left(\frac{D_{CH2}(t)}{D_{CH1}(t)}\right)^{C_s}\varphi_{CH2}(t) \quad (23)$$

In this formula, $\phi_{CH1}(t)$ and $\phi_{CH2}(t)$ separately stand for the jam degrees of the two cluster management nodes, $D_{CH1}(t)$ and $D_{CH2}(t)$ mean the straight-line distances between cluster internal nodes and these two cluster heads, w_1 and w_2 act as weight coefficients, C_s serves as an impact parameter, if the calculated value of $P_{CH}(t)$ is smaller than zero, data will be sent through *CH1*, if the calculation result is above zero, nodes will pick *CH2* to complete transmission;

D. Dynamic Maintenance Algorithm for Clusters

The whole vehicle communication network has complicated structures and keeps changing all the time, vehicles move around frequently so the overall network layout will keep changing without stop, meanwhile all kinds of unexpected incidents can take place on the network, people need to keep adjusting cluster structures to match this changing network environment, dynamic cluster adjustment work mainly covers four different processing steps, these steps include updating cluster internal nodes, picking new cluster management nodes again, dividing one cluster into multiple small groups and combining separate clusters into one single group, the detailed operation steps of every single process are written down below.

For cluster member updates, when new vehicle nodes drive into the cluster coverage area or original cluster members move out of the range, the system will send broadcast request messages and confirmation messages, these messages can dynamically refresh the cluster member list and network routing table, this way can guarantee the network information stays up to date all the time.

The system will trigger a new cluster head selection operation in several common situations, the current cluster management node may move to the edge of the cluster coverage area, its comprehensive performance index may drop below the average level of all cluster nodes, or the node cannot be connected for a long time, this triggering mechanism can effectively keep the cluster communication process efficient and stable.

Multiple clusters may produce overlapping coverage areas during network operation, when such overlapping ranges keep expanding, the network will generate too many gateway nodes and cause more communication conflicts, the system will judge according to the preset distance threshold and overlap time threshold, once the relevant conditions are satisfied, these adjacent clusters will be combined into one large cluster, and the network will select a new cluster management node to improve the usage efficiency of network resources.

When the number of vehicles inside a single cluster gets too large, or the channel busy rate goes beyond the preset limit value, the cluster will face possible network jams, the algorithm will split the original complete cluster into two independent small clusters under this situation, each new subcluster will select its own cluster management node again, this processing method can make the whole communication state keep stable.

CBP is one important index used to judge the load situation of communication channels, it can show how much channel resource has been taken up, this parameter also works as a core standard to judge link transmission quality in vehicle mutual safety communication research, Equation (24) records the specific computational formula of CBP, researchers can rely on this formula to accurately calculate the usage rate of communication channels, this calculation result can offer reference standards to carry out cluster management work.

$$\mathrm{CBP} = \frac{t_{\mathrm{Busy}}}{t_{\mathrm{CBP}}} \times 100\% \tag{24}$$

Here, t_{CBP} denotes the CBP measurement window, while t_{Busy} represents the period during which the device measurement channel is occupied. Additionally, the density of the vehicle cluster is evaluated to determine whether congestion occurs within the cluster, with the cluster density formula shown in (25).

$$\mathrm{Ns}(k) = \lambda \cdot N(k) + (1-\lambda)(\mathrm{Ns}(k-1)) \tag{25}$$

In this computational formula, $N_{(k)}$ stands for the total vehicle amount recorded at the k-th sampling moment, $Ns_{(k-1)}$ refers to the vehicle density value worked out during the (k-1) sampling period, λ is a parameter that balances the weight of past vehicle density data and the vehicle quantity detected in real time, when λ is assigned a value of 0.5, the system will treat real-time detection data and historical density data with identical weight proportions, this setup can avoid unnecessary cluster splitting triggered by vehicles that just briefly pass through the cluster coverage area, the preset limit value of cluster density is fixed at 12, and the threshold value of CBP is set as 98%, if the actual cluster density and real-time CBP value both go over their respective preset limits, all vehicle nodes inside the original cluster will be separated into two small subclusters, the upper limit of node quantity for each newly divided subcluster is controlled to 0.6 times the total node count of the initial cluster, after finishing the cluster separation operation, the system will launch another round of cluster head selection for each subcluster, the whole system keeps track of real-time CBP data and vehicle density status all the time, it will adjust cluster structures and cluster head selection schemes dynamically according to the monitored data, this processing mode can lower the occurrence of channel collision problems and network jam issues brought by too many vehicles within a single cluster.

Through dynamic cluster maintenance management, the issue of cluster state changes caused by vehicle movement is effectively mitigated, ensuring stable communication and robust network performance during dynamic cluster transitions.

IV. EXPERIMENT

This paper tests the performance of the DCHVC scheme with multiple groups of simulation experiments, the test platform combines NS3 software for network scene simulation and SUMO tool to build real vehicle traffic models, this combined testing environment can compare the comprehensive performance of DCHVC and several common cluster control algorithms, these comparison algorithms include VWCA, KMRP and DCM, all contrast tests run under different vehicle movement speeds and different vehicle density environments.

A. Experimental Setup

SUMO full name is Simulation of Urban Mobility, this open source tool is used to build vehicle traffic simulation scenes, the software can build detailed urban road traffic models, it can simulate single vehicle driving behaviors and the traffic running state of each independent road lane, SUMO has obvious advantages on tiny vehicle behavior simulation, users can set fixed driving rules for all vehicles, simulate sudden traffic accident scenes, set interaction relations among different transport tools, and build road network models divided by single lanes, this simulation software can read road network data exported from map service platforms, the OpenGL visual window inside SUMO lets researchers directly watch the change rules of vehicle flow, and check whether the cluster management method can work well in actual traffic scenes.

For network simulation parts of the experiment, the NS-3 simulation tool is adopted in this paper, this discrete event simulation software can support all kinds of communication protocols covering each layer of the communication system, it can complete simulation modeling for bottom physical signal waveforms and upper application layer data generating modules, the whole software adopts a split module design structure, and it can cooperate with different script programs flexibly, these features make NS-3 fit well for research on vehicle wireless communication networks, such research work needs to observe mutual cooperation effects of multiple communication protocols.

To connect these two simulation tools together, researchers turn the output data generated by SUMO into movement track files that NS-3 software can read normally, this conversion step relies on the ns2-mobility-trace auxiliary tool, the tool can read track files written under TCL format standards, and convert these file data to match the movement setting rules of nodes inside NS-3 simulation scenes, this whole data conversion process lets the NS-3 simulation platform borrow complete vehicle flow data exported from SUMO, the platform uses such detailed data to set the moving paths of all network nodes, and the simulation result can restore real vehicle driving states under actual road traffic environments accurately, this combined simulation method not only makes the whole experimental scene closer to real traffic running situations, but also offers an available testing tool for people to study vehicle wireless communication systems and test their operating performance under complicated road traffic conditions.

The traffic simulation vehicle flow used in this paper is generated by SUMO simulation parameters. The map road is a three-lane ring road, as shown in Fig.3.

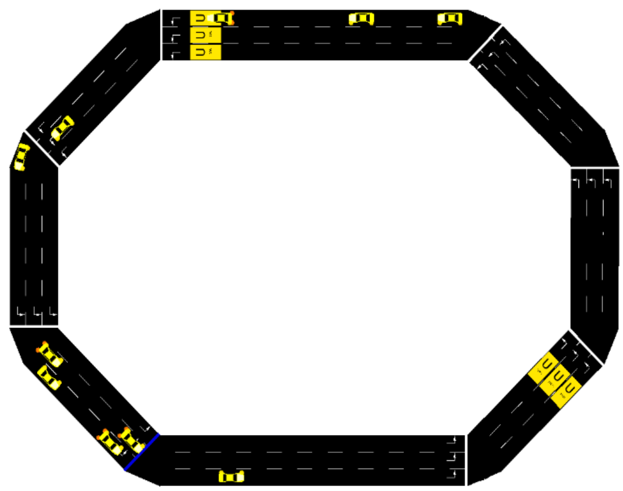

Figure 3. Transportation Map

All vehicles in this simulation scene carry GPS receiving equipment, every single vehicle can get its own real-time location data and driving speed information through this device, the vehicle access model uses Poisson distribution rules to simulate vehicles continuously driving into the test road section, the driving speed of each vehicle follows the distribution rule of normal distribution, all important parameter settings adopted in the traffic simulation experiment are recorded in Table 1;

TABLE I. TRAFFIC SIMULATION PARAMETERS TABLE

Parameters	**Value**
Road length	5KM
Number of lanes	3
Number of vehicles	100，150，200，250，300
Traffic Status	Vehicle density from low to high
Maximum vehicle speed	[60，80，100，120] Km/h

All parameter values used for network simulation tests are listed clearly inside Table 2, vehicles in the simulation exchange data signals by using the 802.11p communication protocol, each CBR data packet carries 512 bytes of information, the transmission speed of CBR data stream is fixed at 16 kb/s, the experiment picks the dual-ray ground model to simulate signal transmission paths between vehicles, all detailed numerical settings linked with network simulation can be found inside Table 2.

TABLE II. NETWORK SIMULATION PARAMETERS

Parameters	Value
MAC Protocol	IEEE 802.11p
Buffer queue length	50 packets
Radio Wave Propagation Model	Two-ray ground
CBR	16 kb/s
CBR group size	512 bytes
Total number of packets	2300000

B. Experimental Results and Analysis

1) Clustering Performance Comparison

This experimental part starts with testing two key properties of the proposed DCHVC algorithm, these two properties are the stable state of divided clusters and the convergence speed of the whole algorithm, three other clustering algorithms are chosen to finish contrast tests together with DCHVC, the VWCA algorithm picks cluster management nodes after calculating multiple indicators including vehicle driving speed, moving direction, relative space distance and node connection quantity, other ordinary vehicle nodes will join the most matching cluster following fixed judgment rules, this algorithm belongs to a traditional weighted clustering scheme widely used in vehicle communication networks, KMRP is a relatively new vehicle mutual communication clustering algorithm built on the K-means calculation method, it can maintain good comprehensive performance under most test scenes, DCM is also a double cluster head algorithm design, this algorithm uses the PO calculation method to split all network nodes and select main cluster heads and backup cluster heads separately, nodes inside each cluster send data forwarding tasks to the closest cluster head, this design can ease the data jam pressure on single cluster head nodes, the paper first analyzes cluster stability indexes to prove the advantages of the DCHVC algorithm, the measured stability standards mainly contain the continuous working time of cluster heads and the convergence performance of the whole algorithm, the comparison data collected from these four different algorithms are displayed below.

Comparison of Algorithm Stability

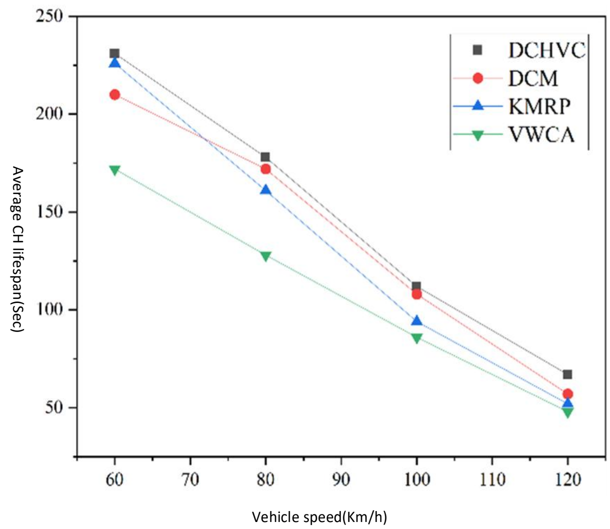

Figure 4. Comparison of Average CH Lifespan Among Four Algorithms at Different Vehicle Speeds

This paper first takes the duration of cluster head service time as a judgment standard to measure how stable each divided cluster is, the cluster head service time refers to the time length from one node being chosen as cluster management node until it turns back into a common cluster member, many existing research papers will adopt this index to complete performance tests, cluster heads need to finish network resource allocation and data message coordination work, longer cluster head service time means the system does not need to run cluster reconstruction operations frequently, and the extra system consumption brought by repeated reconstruction will drop down, the drawn curve figure shows the average service time of cluster heads will keep shrinking when vehicles move faster, there are two main situations that will trigger cluster head replacement in this paper’s designed algorithm, the first situation is that the current cluster head vehicle drives outside the cluster signal coverage range, all nodes inside the cluster have to launch another round of cluster head selection, the second situation comes from the position shift of vehicles, multiple clusters get closer to each other and trigger cluster combination processing, the original cluster head may not meet the selection standards after cluster merging and will be downgraded to ordinary vehicle node, the DCHVC algorithm in this paper also faces shorter cluster head working time under higher vehicle speed environments, but its average cluster head service time is still longer than the DCM algorithm and the KMRP clustering scheme, this comparison result can prove that the designed algorithm owns better cluster stability performance.

Comparison of Algorithm Convergence

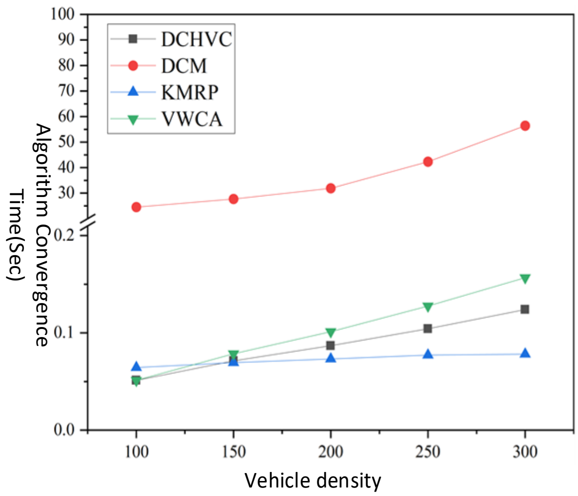

Figure 5. Algorithm Convergence Time Diagram at Different Vehicle Densities

Fig.5 draws the corresponding relation curve between algorithm convergence time and vehicle node density, every tested clustering scheme will spend more time to finish convergence when node quantity rises, yet the DCM algorithm's performance drop is far more obvious, the calculation load of DCM will grow exponentially along with rising node numbers, this feature makes it hard to put this algorithm into actual network usage, such poor convergence performance means DCM cannot adapt to real application scenes that need fast response speed, the K-means-based KMRP clustering algorithm gets little influence from node quantity changes on its convergence speed, the VWCA algorithm needs to collect data from lots of surrounding nodes when building clusters, so its convergence speed falls sharply once node density becomes higher, the DCHVC clustering algorithm designed in this paper also takes longer convergence time when there are more nodes, but the extra waiting time is still kept within a reasonable limit, this result shows the designed algorithm has actual usable value in real traffic scenes.

2) Network Performance Comparison

This section carries out detailed comparative analysis between the DCHVC algorithm and DCM, KMRP together with VWCA algorithms, all analysis data are obtained from NS3 simulation experimental results, the analysis mainly targets several core network performance indicators, these indicators contain system throughput, end-to-end transmission delay and data packet delivery success rate, all comparison tests adjust vehicle moving speeds and traffic densities to observe changes in network operating status, the whole contrast experiment intends to fully judge the practical effect and unique strengths of the DCHVC algorithm under vehicle all-scene communication environments, the three main network performance measurement indicators are listed as follows.

Throughput. The total amount of valid data successfully transmitted within a given time period in a network, reflecting the network's transmission efficiency.

End-to-end delay is one core measurement indicator, this index calculates the total time consumption of one data packet, from the moment the source node sends out the packet until the target node fully receives this packet without error, people can rely on this numerical value to judge how fast the whole network responds to data transmission requests.

Packet Delivery Ratio, abbreviated as PDR, is another important evaluation index, this value is calculated by dividing the amount of data packets received successfully by the total number of packets transmitted from the source node, this index can directly reflect how efficiently and stably the whole network processes data packets, researchers can also rely on this indicator to judge whether the network is facing serious data congestion.

Next, we will analyze the three indicators under different vehicle speeds and traffic densities.

1) Throughput Analysis

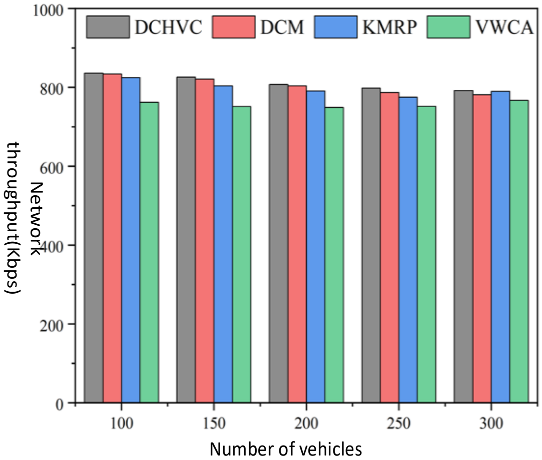

Figure 6. Network Throughput at Different Number of vehicles

Fig.6 records the changing trend of average throughput when the total number of vehicle nodes keeps increasing, every clustering algorithm will see throughput drop with more vehicles, this phenomenon is easy to explain, more nodes will compete for communication channel resources and cause frequent signal collisions, the system will start delay waiting and packet retransmission mechanisms accordingly, the DCHVC algorithm still maintains higher throughput values than DCM, KMRP and VWCA in all test environments, two core reasons bring this obvious performance advantage, the first reason is that DCHVC can reasonably divide all vehicles into independent clusters, data transmission inside each cluster relies on cluster heads for forwarding, and neighboring clusters adopt different communication channels to work at the same time, this design can greatly cut down signal interference between different clusters, besides, DCHVC adds a dedicated data packet scheduling mechanism, cluster heads can transfer part of overloaded data traffic to other idle cluster heads to finish forwarding tasks, this operation eases transmission bottlenecks formed when single cluster head carries too much data load, the DCHVC scheme put forward in this paper can well solve the problem of decreased network throughput, such performance decline usually happens when vehicle density grows high and leads to serious channel congestion.

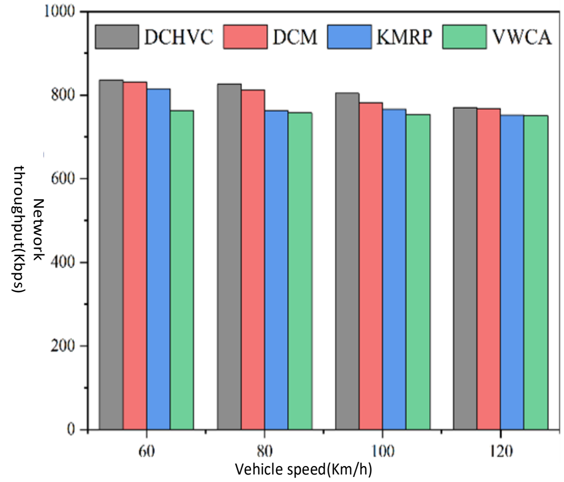

Figure 7. Network Throughput at Different Vehicle Speeds

Fig.7 shows how vehicle driving speed affects the throughput of DCHVC, DCM, KMRP and VWCA algorithms, all four clustering algorithms will have lower throughput when vehicles run faster, the main cause is that faster vehicle movement makes the network connection structure change more frequently and increases the chance of communication link breaking, the DCHVC algorithm designed in this paper can still keep a higher throughput level than the other three comparison algorithms under every speed test condition, the core reason is that the clustering judgment standard of DCHVC takes both similar driving speed and close geographic distance of vehicles into account, so the divided clusters have stronger overall stability, moreover, the cluster heads selected by the multi-index comprehensive judgment method inside DCHVC are less likely to be replaced frequently than cluster heads from other algorithms, as a result, high-speed vehicle movement will bring smaller negative influence on the throughput of DCHVC algorithm when compared with other clustering schemes.

2) Average End-to-End Latency Analysis

Fig.8 draws the variation curve of end-to-end transmission delay under different node densities, the delay values of all four algorithms keep rising steadily as vehicle nodes increase, this change rule is easy to understand, more nodes will make data cache queues longer and trigger fiercer competition for channel resources, the performance advantage of DCHVC becomes more obvious under high node density scenes, its dual cluster head load balancing mechanism can work best in such environments.

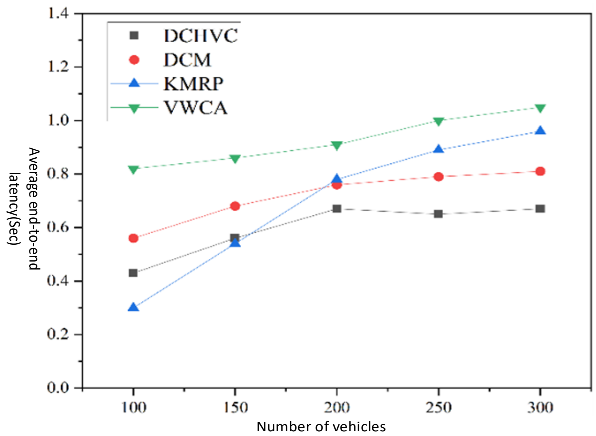

Figure 8. Average End-to-End Latency at Different Number of vehicles

For other contrast algorithms, their cluster head nodes will accumulate massive pending data packets and generate obvious queuing delay. The KMRP algorithm built on K-means clustering will form unbalanced network topological structures, clusters containing a large number of vehicle nodes bring heavy data forwarding pressure to their cluster heads, queuing delay and data processing delay will surge rapidly when the number of vehicles rises. The DCM algorithm also adopts the dual cluster head framework, but all nodes inside a cluster only send data to the closest cluster head, this operation leads to uneven task allocation between the two cluster heads, cluster heads located in dense vehicle areas need to handle far more transmission work, the design cannot fully exert the performance advantages brought by the dual-cluster-head structure.

Fig.9 compares the average end-to-end delay of the proposed DCHVC algorithm alongside DCM, KMRP and VWCA under different vehicle speeds. The curve shows that the average end-to-end delay of every algorithm rises as vehicle driving speed goes up.

The main factor behind higher delay at faster speeds is frequent shifts in network topology. The system has to spend extra time rebuilding clusters, re-electing cluster heads or letting vehicles switch to other clusters, and these repeated operations add extra transmission delay.

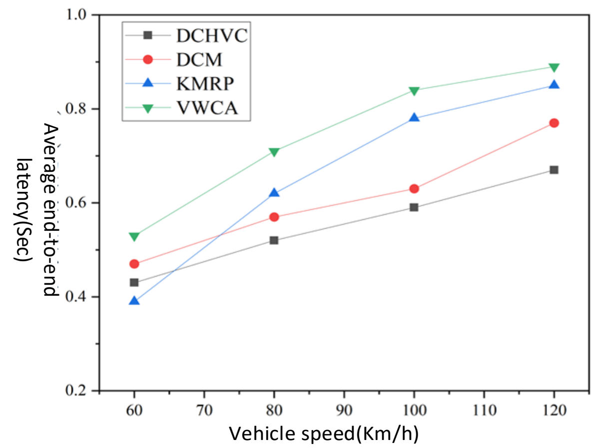

Figure 9. Average End-to-End Latency at Different Vehicle Speeds

Compared with the three contrast algorithms, the DCHVC algorithm groups vehicles according to their similar movement states, which greatly improves cluster stability and cuts down the number of re-clustering operations. Besides, its clustering and cluster head selection logic consume less computing time, and the communication links connecting cluster heads and ordinary nodes stay more stable. For these reasons, the DCHVC algorithm maintains lower end-to-end delay than the other three clustering schemes in all test speed scenarios.

3) Packet Delivery Rate

Fig.10 presents the change rule of packet delivery ratio under different vehicle node densities. For DCM, KMRP and VWCA algorithms, their packet delivery ratio will rise along with growing node density, this trend is reasonable because vehicles are distributed more closely, communication signal strength gets better and data transmission links become more stable.

The DCHVC algorithm keeps a persistently high packet delivery ratio in all density test scenes. Even though more vehicles raise the chance of data packet collision, the overall delivery success rate of DCHVC still stays at a higher level. The clustering strategy designed in this paper sets an upper limit on the number of vehicles in each single cluster, this restriction weakens the negative influence brought by the total number of vehicles on packet delivery performance. When vehicle density rises, each cluster only contains a fixed range of nodes, these vehicles gather around the cluster head more closely, which stabilizes communication links and greatly lifts the packet delivery ratio. In contrast with other clustering methods, DCHVC has more reliable data transmission control ability when the road is full of dense vehicles.

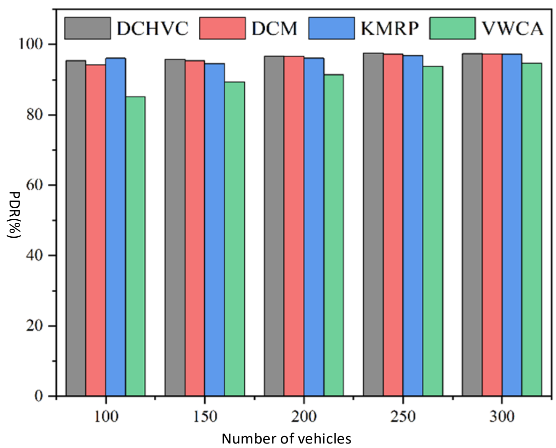

Figure 10. PDR under Different Number of vehicles

The root cause is that faster vehicle speeds lead to frequent changes of the vehicle network topology. Once vehicles drive to the edge of a cluster, the distance between these vehicles and the cluster head becomes larger, which causes signal attenuation. Meanwhile, frequent topological changes will directly trigger data packet loss in the transmission process.

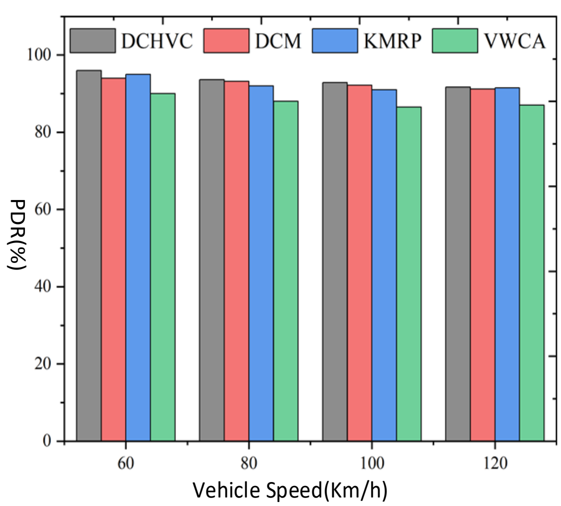

Figure 11. PDR at Different Vehicle Speeds

Different from the three comparison algorithms, the DCHVC algorithm adopts a dual-cluster-head structure to weaken the negative influence of high speed on packet delivery ratio. Two cluster heads jointly expand the effective communication coverage of the cluster and reduce performance losses brought by frequent topological updates.

For vehicles at cluster edges with weak signals and packet loss risks caused by topology changes, nodes will choose the cluster head that is closer and has stronger signal to forward data under non-congestion conditions. This mechanism effectively improves the reliability of data transmission under high-speed moving scenarios.

V. CONCLUSIONS

This paper puts forward the DCHVC clustering algorithm with motion similarity awareness, which can jointly solve two main problems existing in V2X communication networks: network congestion triggered by frequent topology changes and transmission bottlenecks brought by single cluster head structure. The core design idea of this scheme is that the cluster division judgment should take not only the real-time distance between vehicles into consideration, but also the consistency of vehicle driving rules. This core idea runs through the design of the whole clustering flow and the double cluster head selection mechanism.

The paper uses the CRITIC-TOPSIS comprehensive evaluation method to complete the selection of main and backup cluster heads, this method offers objective evaluation standards and avoids the subjective weight setting defects of traditional weighted clustering algorithms. Besides, the algorithm adds a cache state aware data scheduling module, which can real-time monitor the data queue load of each cluster head and transfer redundant data flow reasonably. This function solves the uneven load distribution problem that restricts the performance of other existing dual-cluster-head algorithms.

A large number of NS3-SUMO joint simulation experiments prove that the DCHVC algorithm is superior to VWCA, KMRP and DCM algorithms in multiple performance indicators, such as cluster stability, algorithm convergence speed, network throughput, end-to-end transmission delay and packet delivery ratio. The above performance improvements verify the core viewpoint of this paper: combining dual-cluster-head architecture with motion similarity cluster division and dynamic load transfer mechanism can effectively relieve communication congestion at both data link layer and node layer.

The follow-up research plan of this paper is to expand the current algorithm framework to support inter-cluster routing function, so as to avoid mass data accumulation in large-scale vehicle network scenes and further improve the overall communication service quality of V2X system.

REFERENCES

[1] Oliveira A, Khatibi A,Suzuki D, et al. DL-based beam management for mmWave vehicular networks exploring temporal correlation [J]. Computer Communications, 2026, 257108606-108606. DOI: 10.1016/J.COMCOM. 2026. 108606.

[2] Khaleghirad H, Khonsari A, Dolati M. MoTiVE: A mobility and time-integrated graph autoencoder for trust management in VANETs [J]. Ad Hoc Networks, 2026, 192104328-104328. DOI: 10.1016/J. ADHOC. 2026.104328.

[3] Ren M, Khoukhi L, Labiod H, et al. A mobility-based scheme for dynamic clustering in vehicular ad-hoc networks (VANETs) [J]. Vehicular Communications, 2017, 9: 233-241.

[4] Hadeel A, S. J A, Mohammed M, et al. Enhanced Artificial Gorilla Troops Optimizer Based Clustering Protocol for UAV-Assisted Intelligent Vehicular Network [J]. Drones, 2022, 6(11): 358-358. DOI:10.3390/DRONES6110358.

[5] M. S K, Adib H, Hassen H, et al. A secure multifactor-based clustering scheme for internet of vehicles[J]. ournal of King Saud University - Computer and Information Sciences, 2023, 35(10): 101867-. DOI:10.1016/J. JKSUCI.2023.101867.

[6] R. K R, T. M. A novel clustering and optimal resource scheduling in vehicular cloud networks using MKMA and the CM-CSO algorithm[J]. International Journal of Communication Systems, 023, 36(5): DOI: 10.1002/DAC.5424.

[7] Abbas F, Fan P. Clustering-based reliable low-latency routing scheme using ACO method for vehicular networks [J]. Vehicular Communications, 2018,1266-74. DOI: 10.1016/j.vehcom.2018.02.004.

[8] Lin D, Wang Q. An energy-efficient clustering algorithm combined game theory and dual-cluster-head mechanism for WSNs[J]. IEEE Access, 2019, 7: 49894-49905.

[9] Xu Y, Liu Y, Zhou P. Three-way cluster-guided contrastive graph clustering [J]. Pattern Recognition, 2026,180(PC):114313-114313. DOI:10.1016/J. PATCOG. 2026. 114313.

[10] Foussoul A, Goyal V, Kumar A. Fully-Dynamic Load Balancing[J]. Mathematical Programming, 2025,216(1-2): 1-31. DOI:10.1007/S10107-025-02310-4.

[11] Asad A, Amir A, Farhan M, et al. Enhanced Fuzzy Logic Zone Stable Election Protocol for Cluster Head Election (E-FLZSEPFCH) and Multipath Routing in wireless sensor networks [J]. Ain Shams Engineering Journal, 2024,15(2):102356-. DOI: 10.1016/J. ASEJ. 2023.102356.

[12] Liu Y, Huang H, Zhou J. A dual cluster head hierarchical routing protocol for wireless sensor networks based on hybrid swarm intelligence optimization[J]. IEEE Internet of Things Journal, 2024, 11(9): 16710-16721.

[13] Li H, Ren Y, Luo W, et al. A Three-tier Load Balancing Model with Dynamic Data Partitioning Strategy for Distributed Stream Processing [J]. Journal of Grid Computing, 2026, 24(1): 3-3. DOI:10.1007/S10723-026-09823-Y.

[14] Istwal Y, Verma K S. Dual Cluster Head Routing Protocol with Super Node in WSN [J]. Wireless Personal Communications, 2019, 104(2):561-575. DOI:10.1007/s11277-018-6034-5.

[15] Jin L, Zhang G, Wang J, et al. Quantitative Survivability Analysis Using Probability Model Checking:A Study of Cluster-Based Vehicle Networks with Dual Cluster Heads[J]. 2020,17(2):206-219.

[16] Alghamdi T A. Energy efficient protocol in wireless sensor network: optimized cluster head selection model[J]. Telecommunication Systems, 2020, 74(3): 331-345.

[17] Reddy K M, Veluswamy S A, Selvanayaki S, et al.Modeling of metaheuristic-based dual cluster head selection with routing protocol for energy-efficient wireless sensor networks [J]. Soft Computing, 2025,29(6):1-22. DOI:10.1007/S00500-025-10563-6.

[18] Łukasz Piotr Łuczak,Przemysław Ignaciuk,Michał Morawski. Evaluating MPTCP Congestion Control Algorithms: Implications for Streaming in Open Internet [J]. Future Internet, 2023, 15(10):328-. DOI:10.3390/FI15100328.

[19] Khaled H, Imene T, Ping W Z, et al.Dynamic Base Station Clustering in User-Centric mmWave Networks: Performance Analysis and Optimization[J]. IEEE TRANSACTIONS ON COMMUNICATIONS, 2021, 69(7): 4847-4861. DOI:10.1109/TCOMM. 2021. 3070357.

[20] Liao C, Zhang X, Wang S, et al. Explicit-feedback TCP for congestion control in LEO satellite communications networks[J]. Ad Hoc Networks, 2026, 187104219-104219. DOI:10.1016/J. ADHOC.2026.104219.

[21] Alfoudi A S, Aziz M R, Alyasseri Z A A, et al. Hyper clustering model for dynamic network intrusion detection[J]. Iet Communications, 2025, 19(1): e12523.

[22] Yan C, Zhu W, Li R, et al. Mapping Dynamic Protein Clustering with AIEgen - Active Chemigenetic Probe[J]. Angewandte Chemie International Edition, 2025, 64(14): e202422996.

[23] Priya S S, Vijayabhasker R, Rajaram A. Advanced security and efficiency framework for mobile ad-hoc networks using adaptive clustering and optimization techniques[J]. Journal of Electrical Engineering & Technology, 2025, 20(3): 1815-1826.