applied sciences

MDPI

Article

Design and Optimization of Cluster-Based DSRC and C-V2X Hybrid Routing

Yi Zhang and Jixian Zhang *

School of Information Science and Engineering, Yunnan University, Kunming 650504, China; yamizy@mail.ynu.edu.cn
* Correspondence: denonji@163.com

Abstract: With the continuous development of connected and automated vehicles (CAVs) and Internet of Vehicle (IoV) technologies, various application scenarios have put forward higher requirements for vehicular communications. On the one hand, applications related to vehicle driving safety require lower latency and higher throughput. On the other hand, users who use cellular vehicle-to-everything (C-V2X) to transfer data will have to face high communication fees due to the increasing amount of data. Therefore, from the perspective of balancing quality of service (QoS) and user communication costs, this paper integrates dedicated short-range communication (DSRC) and C-V2X, two vehicular communication technologies with their own advantages, into a framework called cluster-based traffic differentiated hybrid routing (CTDHR), to provide services for in-vehicle communication. A vehicle clustering method based on hierarchical clustering is proposed to solve problems (e.g., the communication linking being difficult to maintain and the frequent cell handover due to high-speed movement of vehicles). The CTDHR framework is modeled on the resulting clusters and an objective equation was established. Finally, since the obtained objective equation is a nonlinear integer programming problem, we propose a heuristic algorithm to solve this optimization problem. In the simulation experiments, CTDHR shows better communication performance than the existing DSRC and C-V2X hybrid models. The experimental results show that CTDHR can reduce the communication costs of users while satisfying QoS.

Keywords: vehicular communication; DSRC; C-V2X; VANET; vehicle clustering

check for updates

Citation: Zhang, Y.; Zhang, J. Design and Optimization of Cluster-Based DSRC and C-V2X Hybrid Routing. *Appl. Sci.* **2022**, *12*, 6782. https://doi.org/10.3390/app12136782

Academic Editor: Juan-Carlos Cano

Received: 4 June 2022
Accepted: 29 June 2022
Published: 4 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Automatic driving, with unparalleled advantages over traditional driving, has always been a research hotspot in academia and the industry [1]. In recent years, automatic driving technology has rapidly improved with the development of deep learning, and level 4 automatic driving has been verified in experimental scenarios [2]. Automatic driving technology can be divided into two categories: (1) a single-vehicle system; that is, relying on a single vehicle to complete automatic driving; (2) CAVs, which emphasize the sharing of information through vehicle-to-vehicle-communication (V2V), vehicle-to-infrastructure communication (V2I), and vehicle-to-person communication (V2P). The latter has lower requirements for vehicle performance, which is conducive to saving vehicle costs, and the latter has a stronger perception of the environment [3]. Due to the advantages of CAV, it has gradually become a more popular research direction in automatic driving [4]. To support CAVs, vehicular communication must still be improved in terms of transmission delay and throughput. For example, since deep learning is widely used in the field of unmanned driving, edge computing needs to be introduced to solve the problem of in-vehicle hardware capability being insufficient at supporting the operations of deep neural networks, putting forward higher requirements for in-vehicle communication [5]; in the event of an emergency, relevant information needs to be broadcast to nearby vehicles in a very short time [6]. In addition, vehicular entertainment [7], the allocation of communication resources, etc., must also be considered [8].

The DSRC based on 802.11p, proposed by the IEEE research group [9], and the C-V2X based on the cellular network proposed by 3GPP [10], have become the most likely vehicular communication technologies to be put into practical application. Studies and simulation experiments have shown that, compared with DSRC, C-V2X has better performance in terms of throughput, data transfer rate, and response time [11]. However, vehicular ad hoc networks (VANETs) based on DSRC have advantages in certain aspects. For example, the average delay of VANETs between single hops is only half of a C-V2X transmission; C-V2X needs to charge users due to the need to use the network service provider's infrastructure, but DSRC does not charge users for direct communication between vehicles; in addition, in areas with insufficient infrastructure, DSRC can be used as an effective blindness measure [12]. Therefore, DSRC has not been abandoned in both academia and industry and is still widely used in road warning, vehicle networking, and other fields. In recent years, many scholars have combined two vehicular communication technologies into one in-vehicle communication framework for research, achieving good results [13–17].

Combining DSRC and C-V2X into a framework will increase latency compared to using C-V2X alone, but this is beneficial in many aspects, such as reducing the occupancy rate of cellular network communication resources, reducing energy consumption, improving throughput, reducing communication costs for users, etc. [13–15]. Research on hybrid applications of DSRC and C-V2X can be divided into two categories. One is the hybrid when different vehicles are equipped with different communications equipment (that is, some vehicles are only equipped with C-V2X equipment, and some vehicles are only equipped with DSRC equipment) [17]. In this scenario, the data format needs to be changed continuously during communication, and the communication management is more complicated. The other category is the hybrid implementation in the scenario where all vehicles are equipped with both communication devices [13,15,16]. Compared with the former, the latter has stronger operability, so it has received more attention from researchers. It is worth noting that due to the high mobility of vehicles, it is difficult to maintain the vehicular communications link, and frequent cell switching is required when using C-V2X [12]. To solve these problems, the cluster vehicle method is mostly introduced in the current vehicle communication. This method treats a certain number of vehicles as a cluster, and performs vehicular communication and link management in units of clusters [18–22].

We noticed that Qi W J et al. in [13] designed a routing mechanism named traffic differentiated clustering routing (TDCR), which integrates DSRC and C-V2X into a framework from the perspective of balancing QoS and communication costs. This routing mechanism has received much attention. However, we discovered that, to ensure the successful establishment of the communication link, TDCR uses a vehicle clustering method based on spatial sampling, and the sampling radius is too small, which leads to an excessive number of forwarding nodes in the routing path of TDCR. According to the routing method of TDCR, there is a certain probability that the routing path cannot be successfully established. As the amount of data increases, the performance of TDCR sharply deteriorates. Inspired by TDCR, we designed a new vehicle clustering method that does not rely on inter-vehicle negotiation and spatial sampling. Based on this clustering method, the number of hops in the communication link between vehicles and the roadside unit (RSU) in VANETs can be effectively reduced. Additionally, from the perspective of balancing QoS and communication costs, a hybrid routing method based on cluster and traffic differentiation was redesigned. This method is referred to as the cluster-based traffic differentiated hybrid routing (CTDHR).

The contributions of this paper are presented as follows:

- A vehicle clustering method based on the idea of hierarchical clustering is proposed. The clustering method—to ensure the stability of clusters—changes the conditions in the hierarchical clustering algorithm to measure whether two clusters can be merged, and also makes some modifications in other aspects to adapt the clustering method to the vehicle clustering scenario. The proposed clustering method not only ensures

the stability of the cluster, but also reasonably controls the number of clusters, reduces the number of hops in the VANET link, and reduces the delay in transmitting the information.
- From the perspective of balancing QoS and user communication costs, the vehicular communication in the CTDHR framework is modeled, and a heuristic algorithm is proposed to solve the model. In addition, the hybrid routing of DSRC and C-V2X based on this model is completed in the heuristic algorithm.
- A simulation environment that is as close to reality as possible was designed and utilized to test the performance of CTDHR. The experimental results show that CTDHR has better performance than the existing DSRC and C-V2X hybrid frameworks.

This paper is organized as follows: We review related research work in Section 2 and describe the CTDHR framework in detail in Section 3. In Section 4, the vehicle cluster model and hybrid routing model are presented. In Section 5, a heuristic algorithm based on the routing model is designed. In Section 6, the CTDHR performance is tested in a simulation environment that is as close to reality as possible. In Section 7, we summarize our research.

2. Related Works

2.1. Research on DSRC and C-V2X

In recent years, many scholars have performed much research to improve DSRC and C-V2X. Particularly, in 2019, the IEEE research group proposed IEEE 802.11bd to update IEEE802.11p. IEEE802.bd will realize vehicular communication with a relative speed of 500 km/h, while the communication distance will be extended to 2000 m [23]. With the development of 5G, 3GPPP proposes NR-V2X on the basis of LTE-V2X so that it can use the advanced technology and spectrum resources of 5G [24].

In [25], Tang Y J et al. proposed a routing mechanism based on DSRC to improve the network response time and bandwidth utilization. In this method, a machine learning method is used to predict the future state of the vehicles to optimize the communication link to reduce the delay. Many other scholars have integrated DSRC and C-V2X, and designed a mechanism to use these two communication methods simultaneously. In [14], Ghafoor K Z et al. proposed a QoS-aware relay algorithm in a scenario where some vehicles used DSRC and some vehicles used C-V2X, which allowed vehicles using two different communication technologies to successfully network and complete information transmission. In [15], Mir Z H et al. proposed a DSRC and C-V2X hybrid vehicle network architecture and protocol stack and designed a distributed radio resource management entity that could effectively manage and coordinate radio resources in both RTAs. In [16], Heo S et al. proposed an adaptive semi-persistent C-V2X and DSRC hybrid vehicle networking model called H-V2X. In this model, when C-V2X meets the performance index, C-V2X is used for data transmission, otherwise, DSRC is used for data transmission. Qi W J et al. proposed a routing mechanism called TDCR in [13], which realizes the hybrid application of DSRC and C-V2X under consideration of the balance between the QoS requirements of different traffic and communication costs of users. In [17], Elbal BR et al. realized a hybrid framework with C-V2X as the main communication means and DSRC as a blind-compensation communication measure. In this work, V2V communication is also used to enhance cellular network base station signals. Since the two communication standards, DSRC and C-V2X, have conflicts in the 5.9GHz frequency band, how to make two different new technologies work together in the same frequency band has also become an important research topic. Ansari K. solved the problem of DSRC and C-V2X sharing in the 5.9 GHz band in [26], which enables DSRC and C-V2X to transmit data concurrently under the same framework.

In addition to making vehicular communication more efficient and reliable, how to allocate the resources of the communication system also has a crucial role in the performance of vehicular communication. In [27], Zhang J X et al. proposed an auction mechanism combined with blockchain technology, which maximizes social welfare and optimizes the allocation of communication resources on the Internet of Vehicles. In [28], Yan Y et al.

proposed a multi-type and multi-resource auction mechanism, which greatly improves resource use efficiency. The mechanism has significance in promoting the progress of communication resource allocation in vehicular communication systems.

Research in the field of vehicle communication has made great progress. However, there are few considerations concerning the user level in the existing results, and the high throughput and low latency requirements for CAVs are still not fully met.

2.2. Research on Vehicle Clustering

In recent years, many scholars have proposed vehicle clustering methods for vehicle communication application scenarios:

Hang S et al. [18] proposed a multi-channel MAC protocol based on vehicle clusters and VANETs that could effectively guarantee QoS. This protocol uses different channels to transmit data within the cluster and outside the cluster, which greatly saves communication resources. In [19], Abboud K et al. introduced vehicle clustering into VANETs and achieved good results. They also employed a stochastic cluster instability model to assess the rate of change in a cluster structure with cluster membership changes and cluster overlap states. In [20], Prajapati J et al. proposed a vehicle cluster method based on vehicle location, destination, and distance. Compared with the previous vehicle cluster algorithm, this method effectively reduces the resource occupation during vehicle communication. Qi W J et al. [13] proposed a clustering method based on spatial sampling. This method ensures the stability of clusters and communication links. In [21], from the perspective of improving user experiences in high-speed mobile scenarios, Kumar D V et al. proposed a method to realize vehicle clustering using common OFDM parameters and clustering mechanisms. In this method, hybrid DSRC/WiMAX technology was utilized. The realized vehicular communication has achieved good results in various communication scenarios in the simulation experiments. In [22], Sanker S et al. proposed a vehicle clustering protocol named MADCR, which first uses Euclidean distance to obtain vehicle clusters, and then uses the mayfly optimization algorithm to select the cluster head (CH). This method not only reduces the CH selection time but also effectively reduces the end-to-end delay and increases the transmission rate.

In existing studies, clustering methods mostly rely on vehicle-to-vehicle negotiation, which increases the time overhead of clustering. This method, especially in highly delay-sensitive scenarios, such as CAVs, has difficulty meeting the requirements of a high response ratio and low delay. There is still a lack of efficient and fast clustering methods.

Tabel 1 is the abbreviations used in this paper.

Table 1. Acronyms Used in Paper.

Acronyms	Definition
CAVs	connected and automated vehicles
IoV	Internet of Vehicles
DSRC	dedicated short-range communication
C-V2X	cellular vehicle-to-everything
VANETs	vehicular ad hoc networks
QoS	quality of service
CTDHR	cluster-based traffic differentiated hybrid routing
TDCR	traffic differentiated clustering routing
RSU	roadside unit
BS	base station
CH	cluster head
CM	cluster member

3. CTDHR Framework

In the CTDHR framework, the road is divided into segments with a fixed length, and each segment is called a service domain. There is an RSU deployed in each service domain, and the RSU participates in the transmission of data sent through VANETs. The

base station (BS) is used to support the transmission of C-V2X data, and usually one BS can cover multiple service domains. As shown in Figure 1, the vehicles in the service domain are divided into several clusters (the method of cluster division will be shown in Section 4), and a cluster contains one cluster head (CH) and several cluster members (CMs). The CH is the agent for the cluster to communicate with the outside world. When a CM needs to send a data packet, it will first send the data packet to the CH, and then the CH will select DSRC or C-V2X to forward the data to the RSU/BS. In order to reduce the delay caused by intra-cluster communication as much as possible, the distance between the CM and CH in the CTDHR cluster is strictly limited within the DSRC communication radius, which enables the CM and CH to communicate directly without any relay forwarding (i.e., CM and CH are connected in one hop). It is worth noting that each CH needs to establish a VANET communication link with the RSU as much as possible. When the CH of the cluster cannot directly establish communication with the RSU, it can rely on the VANET communication link with other CHs to realize multi-hop communication with the RSU. Therefore, in CTDHR, the communication range of the RSU does not need to cover the entire service domain. In this way, even if there is no VANET link between the two clusters, VANET communication between clusters can be realized through the relay and forwarding of the RSU (of course, each CH establishes a C-V2X link with the BS, so clusters can also rely on C-V2X to achieve inter-cluster communication) [29]. In addition, this cannot only reduce the deployment density of RSUs to reduce infrastructure construction costs, but also make good use of the redundancy of other CH communication capabilities.

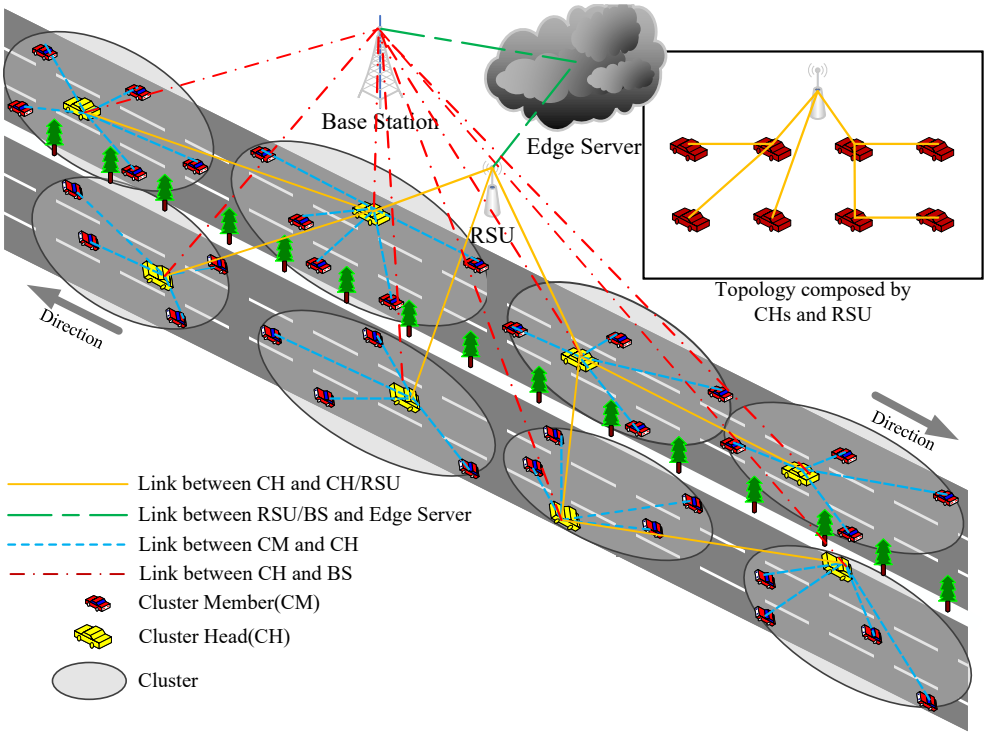


Figure 1. CTDHR framework.

To complete the above data transmission process, we designed three core algorithms. As shown in Figure 2, all vehicles first report their own statuses (including driving speed, direction, and communication capability) to the edge server. After the data collection is complete, the edge server runs Algorithm 1 (see Algorithm 1 in Section 4) according to the information reported by the vehicle to assign the vehicle to different clusters, and selects the CH for each cluster. Then, according to the clustering results and the communication capabilities of the CHs, Algorithm 2 is run (see Algorithm 2 in Section 4) to obtain the VANET routing path from each CH to the RSU. Finally, the edge server distributes the clustering results and routing path information to the vehicles. At this time, the vehicles

selected, such as CH, will run Algorithm 3 (see Algorithm 3 in Section 4). The main task of Algorithm 3 is to select DSRC/C-V2X according to the current perceived network state to forward the data packets to RSU/BS .

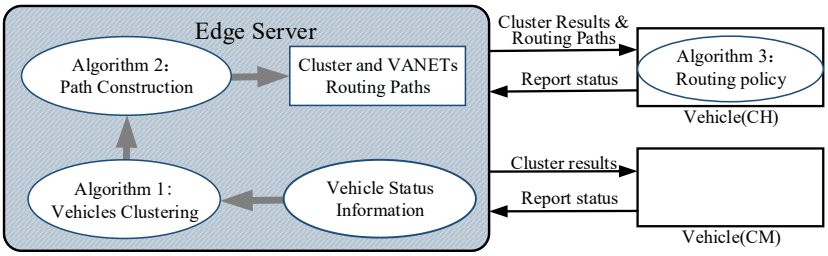


Figure 2. Basic operation logic of the three core algorithms in CTDHR.

Of course, in addition to the need to complete vehicular communication, we also need to consider balancing QoS and user communication costs. In order to satisfy QoS, it is necessary to reduce the delay generated during data packet transmission as much as possible and to reduce the communication costs of users, it is necessary to transmit more data through DSRC. This is contradictory, therefore, we will model the communication process in CTDHR and balance QoS and communication costs according to the model.

4. Model

Because the vehicular communication process is the same in each service domain, in the following research, we put the view in one service domain.

4.1. Vehicle Cluster Model and Algorithm

Definition 1. *Stability of vehicle clusters. Typically, the stability of a cluster is measured by the keeping time of the cluster. The keeping time of the cluster refers to the time elapsed from the moment when the cluster is established until a CM exists in the cluster and the distance between this CM and the CH is greater than the DSRC communication radius.*

When performing vehicle clustering, we have to consider the stability of the cluster. A stable vehicle cluster can reduce the risk of communication interruptions due to cluster dissolution. In addition, the more stable the vehicle cluster is, the longer the interval for reestablishing the cluster is required, thereby reducing the time and resource overhead (e.g., communication resources, computing device resources, etc.) required for rebuilding the cluster [18]. The number of CMs included in a cluster should also be reasonably controlled, because the communication data of the CM needs to be forwarded through the CH, and the communication capability of the CH is limited. In this paper, the number of CMs in a single cluster must not exceed N [15]. Reasonable control of the number of clusters is conducive to saving communication resources, reducing the occupation of the cellular network bandwidth, and can also affect the number of forwarding hops when CH and RSU perform multi-hop transmissions to a certain extent. Generally, as the number of clusters increases, the number of forwarding hops will also increase slightly, resulting in more delay [19]. Table 2 is the explanation of the mathematical symbols used in the paper, and these symbols will also be explained in the text when they appear.

Table 2. Notation Table.

Notation	Implication
$G(\mathcal{V}, \mathcal{E})$	Topological diagram composed of CH and RSU
$\mathcal{V}$	The set of CHs and RSU
$\mathcal{E}$	The set of edges in $G(\mathcal{V}, \mathcal{E})$
a_i	The i-th CH in the $\mathcal{V}$
$e(i,j)$	Edge between a_i and a_j
$\mathcal{C}$	The set of clusters
c_i	The i-th cluster in the $\mathcal{C}$
R	Communication radius of DSRC
$D^{cluster}$	Merge parameter used to measure whether two clusters can be merged
$D^{vehicle}$	Used to measure whether two vehicles can establish communication
$D^{vehicle}_{threshold}$	If $D^{vehicle} \geq D^{vehicle}_{threshold}$, the two vehicles can establish stable communication
Y_{ij}	The remaining time that c_i and c_j are within the DSRC communication radius
$d^{cluster}, d^{vehicle}$	The distance between the two clusters and the distance between the two vehicles
d_{limit}	Maximum mergeable distance
N	Maximum number of CMs that can be included in a single cluster
v^{min}, v^{max}	Maximum and minimum speed of CMs in a cluster
$v^{vehicle}$	The speed of the vehicle
$Path_i$	VANETs communication link from a_i to RSU
$\mathcal{B}_i$	Data packet buffer of a_i
b_{ik}	k-th packets in $\mathcal{B}_i, i \in \{1, \ldots, M\}$
S_{ik}	Size of b_{ik}
f^{CV2X}, f^{DSRC}	The transmission rates of DSRC and C-V2X, respectively
ω_{ik}	Weight of b_{ik}
l_{ik}	Maximum delay allowed for b_{ik}
l^{min}, l^{max}	The upper and lower boundaries of l_i^k
T	The weighted delay generated in the system per unit time
p	Price per MB of cellular data
P	Communication costs incurred in the system per unit time
q_{ik}^{CV2X}	Sum of queuing delay and processing delay when b_{ik} is sent through C-V2X
q_{ik}^{m}	The sum of queuing delay and processing delay generated when b_{ik} is forwarded at m-th of $Path_i$
r	Vehicle's data generation rate
u_i^{CV2X}	The average delay generated by the packets sent using C-V2X in a_i
u_i^{DSRC}	The average delay generated by the packets sent using VANETs in a_i

Based on the above considerations, we were inspired by the idea of the hierarchical clustering algorithm and designed a vehicle clustering method that satisfies the above conditions. Hierarchical clustering is a bottom-up clustering method. It first initializes a class for each element. Then, we continue to merge the two closest clusters until only one class remains. Finally, the hierarchical clustering algorithm will obtain a clustering tree, and then select the appropriate number of classes from this clustering tree [30].

We first define some symbols to facilitate the following description. We use $G(\mathcal{V}, \mathcal{E})$ to denote the topological map composed of CH and RSU and use $\mathcal{V} = \{a_1, a_2, \ldots, a_M, RSU\}$ to denote the set composed of CH and RSU, where $a_i, i \in \{1, \ldots, M\}$ represents the CH of the i-th cluster and CH is a vehicle entity that includes attributes, such as speed, location, packet buffer, and data generation rate. $\mathcal{E} = \{\ldots, e(i,j), \ldots, e(k, RSU), \ldots\}$ represents the set of edges in the topology graph, and $e(i,j)$ represents the edge from a_i to a_j. $\mathcal{C} = \{c_1, c_2, \ldots, c_M\}$ is the set of clusters, where $c_i, i \in \{1, \ldots, M\}$represents the i-th cluster.

Different from using Euclidean distance in hierarchical clustering to measure whether two clusters can be merged, in the clustering algorithm of this paper, the two clusters with the largest value of $D_{ij}^{cluster}$ will be merged. $D_{ij}^{cluster}$ is the weighted sum of Y_{ij} and $(R - d_{ij}^{cluster})^{-1}$, Y_{ij} refers to the estimated remaining time of c_i and c_j within the DSRC communication range, R refers to the DSRC communication radius, and $d_{ij}^{cluster}$ refers to the

distance between c_i and c_j. $(R - d_{ij}^{cluster})^{-1}$ can describe whether c_i and c_j are on the edge of each other's DSRC communication range. The detailed definition of $D_{ij}^{cluster}$ is as follows:

$$D_{ij}^{cluster} = \begin{cases} \alpha Y_{ij} + \frac{\beta}{R - d_{ij}^{cluster}}, d_{ij}^{c} \leq d_{limit} \\ 0, d_{ij}^{c} > d_{limit} \end{cases} \quad (1)$$

When c_i and c_j are at the edge of each other's DSRC communication range (that is, when $d_{ij}^{cluster}$ is close to R), if the speed (the speed here refers to the average speed of all vehicles in the cluster) difference between c_i and c_j increases, it will cause c_i and c_j to quickly leave the DSRC communication range. Therefore, in Equation (1), the difference between R and $d_{ij}^{cluster}$ is used to limit the merging of clusters at the edge of the DSRC communication range. β is a negative value whose absolute value is greater than 1. When $(R - d_{ij}^{cluster})$ becomes smaller, $(R - d_{ij}^{cluster})^{-1}$ increases, $D_{ij}^{cluster}$ becomes smaller. d_{limit} is the maximum allowed distance that clusters can merge. α is the weight of Y_{ij}, and the calculation formula of Y_{ij} is:

$$Y_{ij} = \frac{R \pm d_{ij}^{cluster}}{\max\left\{\left|v_i^{max} - v_j^{min}\right|, \left|v_j^{max} - v_i^{min}\right|\right\}} \quad (2)$$

where v^{max} and v^{min} represent the maximum speed and minimum speed, respectively, of the CMs in the cluster. In particular, v_i^{min} represents the minimum velocity of the CM in cluster c_i. For two clusters, c_i and c_j within the DSRC communication range traveling in the same direction, if the average speeds of c_i and c_j do not change for a period of time in the future, and $d_{ij}^{cluster}$ tends to shrink (that is, c_i and c_j are close to each other)), then Y_{ij} should be the sum of $d_{ij}^{cluster}$ and R divided by the speed difference between c_i and c_j; that is, the numerator of Equation (2) should be added. Conversely, if $d_{ij}^{cluster}$ tends to increase (that is, c_i and c_j are far away from each other), then Y_{ij} should be the difference between $d_{ij}^{cluster}$ and R divided by the speed difference between c_i and c_j, and the numerator of Equation (2) should be subtracted. In addition, since the velocities between CMs within a cluster are different, we used the difference between the maximum speed of vehicles in one cluster minus the minimum speed of vehicles in the other cluster as the speed difference between the two clusters, which can make the resulting cluster more stable.

Since the number of CM members in the cluster is limited to N, even if the $D_{ij}^{cluster}$ of c_i and c_j is the largest, once the sum of the number of CMs of c_i and c_j exceeds N, ci and cj cannot be merged. The stopping condition of the clustering algorithm in this paper is until no two clusters with a distance less than d_{limit} can be found. In summary, the clustering algorithm process in this paper is as follows:

1. Vehicles report their current state information to the edge server;
2. Initialize clusters, each cluster contains one vehicle;
3. According to Equations (1) and (2), find c_i and c_j in all clusters, such that $D_{ij}^{cluster}$ is the largest. If the total number of CMs in c_i and c_j is less than N, then c_i and c_j are merged; otherwise, to ensure that the algorithm does not fall into an infinite loop, it is necessary to mark that $\max\{c_i, c_j\}$ does not enter the next cycle.
4. Repeat step 3 until no more than two clusters that meet the conditions can be identified.

The pseudocode of the clustering algorithm is shown in Algorithm 1. Lines 1 to 5 initialize variables. Lines 9 to 10 merge vehicle clusters when the conditions are met. The meaning of the 12 to 13 lines is that if the total number of CMs in the two clusters is greater than N, the larger one of the two clusters will not enter the next loop.

Algorithm 1: Vehicle Clustering. The algorithm is used to complete the vehicle clustering.

Input: $\mathcal{V}, d_{limit}, \alpha, \beta, N$
Output: $\mathcal{C}$

$\mathcal{C} \leftarrow \varnothing, d \leftarrow +\infty$;
for *all* $a_i \in \mathcal{V}$ **do**
  add a_i to c_i add c_i to $\mathcal{C}$;
end
$d_{ij}^{cluster} \leftarrow 0$;
while $d_{ij}^{cluster} \leq d_{limit}$ **do**
  Find c_i and c_j to minimize $D_{ij}^{cluster}$;
  $d_{ij}^{cluster} \leftarrow$The distance between c_i and c_j ;
  if $d_{ij}^{cluster} \leq d_{limit}$ *and* $\left\|c_i \bigcup c_j\right\| \leq N$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \left(c_i \cup c_j\right)$;
    $\mathcal{C} \leftarrow \mathcal{C} - \left\{c_i, c_j\right\}$;
  else if $\left\|c_i \bigcup c_j\right\| \leq N$ **then**
    Mark $\max\left\{c_i, c_j\right\}$ does not enter the next cycle;
  end
end
return $\mathcal{C}$;

This paper uses a simple example to illustrate the clustering process of Algorithm 1. Assume that there are five vehicles $\{a_1, a_2, a_3, a_4, a_5\}$ traveling in the same direction on the current road segment. The velocity of a_1 is 23 and the coordinates are (1, 450); the velocity of a_2 is 20 and the coordinates are (6, 350); the velocity of a_3 is 30 and the coordinates are (3, 240); the velocity of a_4 is 21 and the coordinates are (5, 100); the velocity of a_5 is 22, and the coordinates are (2, 0). Moreover, suppose the parameters are set to: $\alpha = 1$, $\beta = -200$, $d_{limit} = -200$, $R = 4$, $N = 3$. The clustering process is shown in Figure 3. Initialize each vehicle as a cluster to obtain $\mathcal{C} = \{c_1, c_2, c_3, c_4, c_5\}$, where $c_i = \{a_i\}$. In the first round of the loop, $\max\{D_{ij}^c\} = D_{4,5}^c = 499.45$, $d_{4,5}^{cluster} = 100.124$, $d_{4,5}^{cluster} < d_{limit}$, so merge c_4 and c_5 to c_6. In the second round of the loop, $\max\{D_{ij}^c\} = D_{1,2}^c = 99.29$, $d_{1,2}^{cluster} = 110.00$, $d_{1,2} < d_{limit}$, so merge c_1 and c_2 to c_7. In the third round of the loop, $\max\{D_{ij}^c\} = D_{3,7}^c = 65.05$, $d_{3,7}^{cluster} = 160.00$, $d_{4,5}^{cluster} < d_{limit}$, so merge c_3 and c_7 to c_8. In the fifth round of the loop, $d_{6,8}^{cluster} = 263.33$, $d_{6,8}^{cluster} > d_{limit}$, at this time, $\max\{D_{ij}^c\} = D_{6,8}^c = 0$, the clustering process is completed. The resulting clustering result is $\mathcal{C} = \{c_6, c_8\}$, and $c_6 = \{a_4, a_5\}$, $c_8 = \{a_1, a_2, a_3\}$.

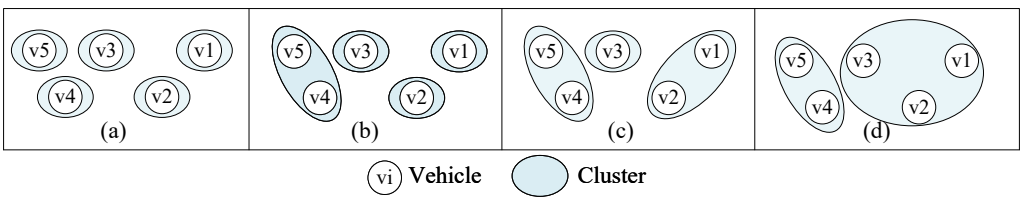


Figure 3. Schematic diagram of the cluster process. (**a**) The initialization before starting the cluster, (**b**–**d**) the process of the cluster

It is worth noting that in the above clustering process, no CH is generated. Next, we will discuss how to choose CH for each cluster. The principle is to select the vehicle in the cluster that can establish the most stable connections with other CMs as the CH. In a cluster, all CMs and CH in the cluster are connected by one hop. The parameter $D_{ij}^{vehicle}$ is introduced to evaluate the stability of the connection between the two vehicles. Different from $D_{ij}^{clucter}$, $D_{ij}^{vehicle}$ is used to directly estimate the remaining time that the

vehicle is within the mutual DSRC communication range based on the current situation. $D_{ij}^{vehicle}$ does not consider whether the vehicles are on the edge of the mutual DSRC communication range, because due to the limitation of d_{limit}, the distance between the vehicles in a cluster and the center of the cluster is less than R, and the final selected CH will also be closer to the center of the cluster. $D_{ij}^{vehicle}$ is defined as follows:

$$D_{ij}^{vehicle} = \begin{cases} \frac{R+d_{ij}^{v}}{\left|v_i^v - v_j^v\right|}, v_i \text{ and } v_j \text{ close to each other} \\ \frac{R-d_{ij}^{v}}{\left|v_i^v - v_j^v\right|}, v_i \text{ and } v_j \text{ away from each other} \end{cases} \tag{3}$$

The symbol $d_{ij}^{vehicle}$ represents the distance between two vehicles, and v^v represents the speed of the vehicle. When $D_{ij}{}^{vehicle} \geq D_{threshold}^{vehicle}$, then a stable connection can be established between CM_i and CM_j, and $D_{threshold}^{vehicle}$ is a settable threshold. The value of $D_{threshold}^{vehicle}$ in this paper is the same as that in the TDCR model [13]. In addition, this formula is used to determine whether communication can be established between two CHs when establishing a VANET routing path in Section 5.1. We did not focus on intra-cluster communication, so intra-cluster communication will not be discussed. Notably, the algorithm works even when there are extremely sparse vehicles on the road. In the most extreme case, each cluster contains only one vehicle.

4.2. Vehicular Communication Model

The data packets of the CM must be forwarded to the CH. Then, the CH selects the communication method to send the packets according to the QoS requirements of the packets. If C-V2X is selected, the CH directly sends the data packet to the nearby cellular network base station (BS); otherwise, the packet is sent to the next node according to the VANET routing path until the packet is sent to the RSU. Last, the packet is sent to the edge server by the BS or RSU.

Although through the CH, relay forwarding the packets adds extra delay, it is easier to optimize the network and reduce the overhead of communication resources [19]. If all data packets are sent through the cellular network, it will lead to a large increase in the communication cost for network users and increase the burden on the cellular network. Therefore, we combine DSRC and C-V2X communication methods in one framework and consider the balance between QoS and user communication costs in the following modeling.

First, we introduce the decision variable θ_{ik}, its meaning is:

$$\theta_{ik} = \begin{cases} 0, \text{use VANETs send } k\text{-th packet of } a_i \\ 1, \text{use C-V2X send } k\text{-th packet of } a_i \end{cases} \tag{4}$$

We use $\mathcal{B}_i = \{b_1, b_2, \ldots, b_k, \ldots\}$ to represent the data packet buffer of a_i, where $b_k, k \in \{1, 2, \ldots\}$ is the k-th packet in $\mathcal{B}_i$. In the following, we denote b_k in $\mathcal{B}_i$ by b_{ik}. Moreover, θ_{ik} is used to represent how b_{ik} is sent. When θ_{ik} is equal to 1, it means that VANET is used to send b_{ik}; when θ_{ik} is equal to 0, it means that C-V2X is used to send b_{ik}.

To balance the QoS and the user's communication cost, we next model the total delay and total communication cost generated in the service domain per unit time. In the service domain, the delay is partly caused by packets sent via VANETs and partly by packets sent via C-V2X. Therefore, the total delay should be the sum of these two parts. No matter which part of the delay, it is composed of the following four types: processing delay, queuing delay, transmission delay, and propagation delay. As the propagation delay is extremely short, it is ignored in this paper. The transmission delay can be calculated by dividing the packet size by the data transmission rate. We used S_{ik} to represent the size of b_{ik}, f^{CV2X} to represent the data transmission rate of C-V2X, and f^{DSRC} to represent the data transmission rate of VANETs. In addition, we measure the processing delay and forwarding delay with the time interval from when a data packet enters $\mathcal{B}_i$ to when the data packet is sent from

$\mathcal{B}_i$, and call it forwarding delay. q_{ik}^{CV2X} represents the C-V2X forwarding delay of b_{ik}, and q_{ik}^{m} represents the forwarding delay generated when b_{ik} is forwarded by the m-th node in $Path_i$. $Path_i$ represents the routing path of VANET from a_i to RSU. Different packets have different QoS requirements. Packets with high QoS usually require lower transmission delays, so they should also be forwarded preferentially to reduce their forwarding delays, so we multiply the forwarding delay of each packet by a weight. w_{ik} is the weight of b_{ik}, and its calculation formula is as follows:

$$\omega_{ik} = 2 \times \left[1 - \left(\frac{l^{max} - l_{ik}}{l^{max} - l^{min}}\right)\right] \tag{5}$$

where l_{ik} represents the maximum delay allowed for b_{ik}, which is the maximum time allowed from the start of the packet generation to the time the packet is sent to the RSU/BS. l_{ik} will subtract the elapsed time in real time after the data packets are generated, which will ensure that the data packets with lower QoS requirements can also be processed in time. $\left[l^{min}, l^{max}\right]$ is the value range of l_{ik}. The larger ω_{ik} is, the higher the priority of the data packet is.

Finally. We use T to represent the total delay generated in the service domain per unit time. The calculation formula of T is as follows:

$$T(\boldsymbol{\theta}) = \sum_{i=1}^{M} \sum_{b_k \in \mathcal{B}_i} \left[\theta_{ik}\left(\frac{S_{ik}}{f^{CV2X}} + \omega_{ik} \times q_{ik}^{CV2X}\right) + (1 - \theta_{ik})\left(\frac{S_{ik}}{f^{DSRC}} + \omega_{ik} \times q_{ik}^{i}\right)\right] \tag{6}$$

where $\theta_{ik}\left(\frac{S_{ik}}{f^{CV2X}} + \omega_{ik} \times q_{ik}^{CV2X}\right)$ is the delay generated when the data packet is sent through C-V2X, and $(1 - \theta_{ik})\left(\frac{S_{ik}}{f^{DSRC}} + \omega_{ik} \times q_{ik}^{i}\right)$ is the delay generated when the data packet is sent through VANET. It should be noted that if a packet is sent through the VANET link, the packet will enter the packet buffer of each node in the link. Therefore, when a data packet is sent through VANET, it is only necessary to accumulate the delay generated at each node that the data packet passes through. This is why the superscript of q_{ik}^{i} is i. The topology map composed of CH and RSU in the service domain is always determined, and once the topology map is determined, the optimal routing path can always be found according to algorithms such as Dijkstra. At this time, the total delay in the service domain is only related to which method is used to send the data packet. Therefore, T is a function of $\boldsymbol{\theta}$, which is the same in the following formulas; $\boldsymbol{\theta}$ is a vector consisting of all θ_{ik}.

Assume that the unit price of the cellular network traffic is p dollar/MB. The total communication cost generated in the system per unit time is represented by P:

$$P(\boldsymbol{\theta}) = p \times \frac{\sum_{i=1}^{M} \sum_{b_k \in \mathcal{B}} \theta_{ik} \times S_{ik}}{1024^2} \tag{7}$$

This formula is quoted from [13]. P can also reflect the level of the cellular network bandwidth consumption.

To achieve the goal of balancing the QoS of packets and user communication costs, we use the following objective function:

$$J(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \left(\ln T(\boldsymbol{\theta}) + P(\boldsymbol{\theta})\right) \tag{8}$$

$$(1 - \theta_{ik}) \times \sum_{e(m,n) \in Path_k} \left(\frac{S_{ik}}{f^{DSRC}} + q_{ik}^{m}\right) \leq l_{ik}, \theta_{ik} = 0 \tag{8a}$$

$$C_{e(m,n)}^{DSRC} - \sum_{b_k \in \mathcal{B}} (1 - \theta_{ik}) \times S_{ik} \geq 0, e(m,n) \in Path_k \tag{8b}$$

$$C^{CV2X} - \sum_{b_k \in \mathcal{B}_i} \theta_{ik} \times Size_{ik} \geq 0 \tag{8c}$$

$$\theta_{ik} \in \{0, 1\} \tag{8d}$$

where C^{CV2X} represents the capacity of the link of the cellular network. $C^{DSRC}_{e(m,n)}$ represents the capacity of the VANETs link a_m to a_n in $Path_k$. Equation (8a) specifies that if VANETs are used to transmit b_{ik}, the total delay generated must be less than l_{ik} for b_{ik}. Equations (8b) and (8c) stipulate that the capacity of the cellular network links and VANET links cannot be exceeded at any time. Equation (8d) illustrates the range of values for θ_{ik}.

5. Heuristic Algorithm

The objective Equation (8) is a nonlinear 01 integer programming problem, which has been proven to be NP-hard. Therefore, according to the previously established model, we designed a heuristic algorithm to approximately solve the problem. It takes more time and communication resources to make the CH aware of the global information of the network. Therefore, in practice, the network is usually not optimized as a whole, but each node is optimized according to the local information easily perceived by CH [19]. This approach is similar to that employed in the greedy algorithm; if each node can achieve the optimal result, the entire network will obtain the optimal result.

In this subsection, first, let the edge server identify the optimal VANET routing path according to the cluster results and collected vehicle information. Second, each CH selects DSRC or C-V2X to send data packets according to its own situation and the local information it can perceive.

5.1. VANET Routing Path

CH and RSU are used as nodes to establish the topology graph $G(\mathcal{V}, \mathcal{E})$. First, initialize $\mathcal{E} = \varnothing$. Second, establishing a communication link between CHs is similar to establishing a communication link between a CM and a CH in Section 4.1, so we also use Equation (3) to determine whether a VANET communication link can be established between the two CHs. If $D^{vehicle}_{ij} \geq D^{vehicle}_{Threshold}$, then $\mathcal{E} \bigcup \{e(i, j)\}$. We use the average of the delays generated by the past x packets transmitted through $e(i, j)$ to represent the weight w_{ij} of $e(i, j)$.

After completing the establishment of $G(\mathcal{V}, \mathcal{E})$, we discuss how to establish the routing paths of VANET. First, the CHs are sorted by distance to the RSU from near to far. Second, the Dijkstra algorithm is selected to identify the shortest path from each node to the RSU. Due to the capacity limitation of the communication link, every time a path is identified, it is necessary to check whether the remaining capacity of the link in the path can accommodate the data volume of the newly added node, and if not, a new path needs to be determined.

In practice, there are cases where there is no path between the CH and the RSU in the topology graph. If there is no path between a_i and RSU, then find $c_j, j \in \left\{ j \middle| d^{vehicle}_{j,RSU} \leq d^{vehicle}_{i,RSU} \right\}$ closest to c_i. Split $\max\{c_i, c_j\}$ and remove $\max\{c_i, c_j\}$ from $\mathcal{C}$ and add to $\mathcal{C}$ the resulting clusters from the split. The topology map is updated, and the routing path can be identified. If the path by which the remaining capacity can meet the needs of the current cluster is not identified, then the current cluster will be set to an unreachable state, and all data for the cluster will be sent over the cellular network.

The pseudocode for the establishment of routing paths for VANETs is shown in Algorithm 2, where r represents the data transmission rate of the vehicle. The functions involving Lines 4 to 17 are to try to find the shortest paths. If the identified path is not available, mark the weight of the edge with insufficient capacity as infinite according to Line 12. If there is no path between the CH and the RSU, the processing is performed according to Lines 18–20.

Algorithm 2: Path Construction. The algorithm is used to establish VANET communication links between clusters.

Input: $G(\mathcal{V}, \mathcal{E}), \mathcal{C}, r$
Output: *Paths*

$Path \leftarrow \varnothing$;
$Nodes \leftarrow \mathcal{V} \backslash \{RSU\}$;
Sort *Nodes* in ascending order based on the distance from CH to RSU;
while $Nodes \neq \varnothing$ **do**
  for *all* a_i *in Nodes* **do**
    if *There is a path between* a_i *and RSU* **then**
      $path_i \leftarrow \text{Dijkstra}(node_i, RSU, G)$;
      if $C^{DSRC}_{e(m,n) \in Path_i} - r \geq 0$ **then**
        $Path \bigcup \{Path_i\}$;
        $Nodes \leftarrow Nodes \backslash \{node_i\}$;
      else
        Set the weight of $e(m, n)$ as $+\infty$;
      end
    else
      break;
    end
  end
  Find $c_j, j \in \left\{ j \middle| d^{vehicle}_{j,RSU} \leq d^{vehicle}_{i,RSU} \right\}$ closest to c_i;
  Divide $\max\{c_i, c_j\}$ into two clusters;
  Insert two clusters obtained by divide $\max\{c_i, c_j\}$ into $\mathcal{C}$ and delete $\max\{c_i, c_j\}$ into $\{c_u, c_v\}$ from $\mathcal{C}$;
  Update $G(\mathcal{V}, \mathcal{E})$, *Nodes*, *Paths*;
end
return *Paths*;

Figure 4 shows the general workflow of Algorithm 2. Figure 4a is the topology diagram between CH and RSU, which includes RSU and four clusters. First, each cluster will be sorted by the distance to the RSU from near to far. Suppose the order is $\mathcal{C} = \{c_1, c_4, c_2, c_3\}$. Then, in the edge server, the Dijkstra algorithm is used to find the shortest path from c_1 to the RSU to obtain $Path_1 = \{e(1, RSU)\}$, as shown in Figure 4b. As shown in Figure 4c,d, we successively find paths $Path_4 = \{e(4, RSU)\}$ and $Path_2 = \{e(2, 1), e(1, RSU)\}$ for c_4 and c_2. However, when establishing the path of c_3, it is found that there is no path between c_3 and the RSU in the topology diagram. At this time, c_2 is the closest cluster to c_3 on the RSU side, and c_2 has more members than c_3, so c_2 is split into c_5 and c_6, as shown in Figure 4e. Insert c_5 and c_6 into the appropriate position of $\mathcal{C}$; delete c_2 from $\mathcal{C}$; update the topology map. Now, the vehicle cluster set is $\mathcal{C} = \{c_1, c_4, c_3, c_5, c_6\}$. As shown in Figure 4f, find the shortest path from c_5 to RSU to obtain $Path_5 = \{e(5, 4), e(4, RSU)\}$. Find the shortest path $Path_6 = \{e(6, 5), e(5, 4), e(4, RSU)\}$ from c_6 to RSU, as shown in Figure 4g. When seeking the shortest path of c_3, $Path_3 = \{e(3, 6), e(6, 5), e(5, 4), e(4, RSU)\}$ is obtained for the first time, but the capacity of $e(4, RSU)$ cannot meet the needs of c_3. Therefore, we change the weight of $e(4, RSU)$ to $+\infty$, determine the shortest path of c_3, Obtain $Path_3 = \{e(3, 6), e(6, 5), e(5, 1), e(1, RSU)\}$, and detect that all links are available, as shown in Figure 4h. Thus far, the establishment of VANET routing paths for all clusters in Figure 4a has been completed.

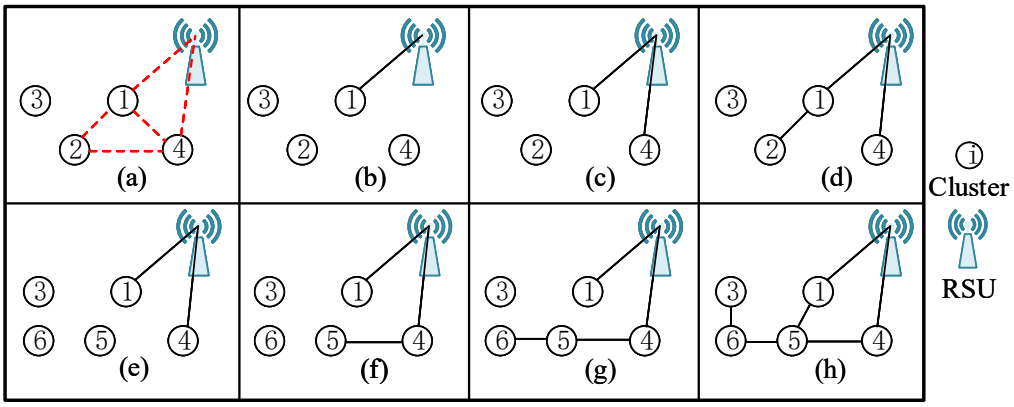


Figure 4. Example diagram of the running process of Algorithm 2. (**a**) The topology diagram between CH and RSU; (**b**–**h**) the running processes of Algorithm 2.

5.2. Routing Policy

To facilitate the CH in evaluating the current network conditions to decide how to forward the data packets to the RSU/BS, u_i^{CV2X} is used to represent the average delay of x packets transmitted from a_i via C-V2X, and u_i^{DSRC} is used to represent the average delay of x packets transmitted from a_i via DSRC. Their calculation formulas are expressed as follows:

$$u_i^{CV2X} = \frac{1}{x} \times \sum_{k=0}^{x} \left(\frac{S_{ik}}{f^{CV2X}} + q_{ik}^{CV2X} \right), \theta_{ik} = 1 \tag{9}$$

$$u_i^{DSRC} = \frac{1}{x} \times \sum_{k=0}^{x} \sum_{e(m,n) \in Path_k} \left(\frac{S_{ik}}{f^{DSRC}} + q_{ik}^{m} \right), \theta_{ik} = 0 \tag{10}$$

This paper assumes that $u_i^{CV2X} \leq l^{min}$ is always true [13]. When a_i needs to send data, it will first select the packet with the largest weight from $\mathcal{B}_i$ according to Equation (5). If $u_i^{DSRC} \leq l_{ik}$, a_i will use VANETs to send the packet; otherwise, C-V2X will be selected to send the packet after sending a packet. Update $l_{ik}, k \in \{1, 2, \dots\}$ and $\omega_{ik}, k \in \{1, 2, \dots\}$ of all packets in the buffer, and update u_i^{CV2X} or u_i^{DSRC}, every x packet sent via C-V2X or VANET. As long as the data packets generated by a_i are sent to the BS through the C-V2X method, the cost will be paid by a_i.

The pseudocode of the routing policy is shown in Algorithm 3. Lines 2 to 8 are used to select the current packet that needs to be sent. Lines 9 to 13 are used to select the sending method. Line 14 is used to update the packet weight and maximum allowable delay in real-time. Line 15 is used to update the average delay according to Equations (9) and (10). After obtaining the routing paths of VANETs, each CH independently runs Algorithm 3. Combined with Equations (9) and (10), the real-time updated u_i^{CV2X} and u_i^{DSRC} enable the CH to grasp the current network state to a certain extent. If congestion occurs in the network, u_i^{CV2X} and u_i^{DSRC} obtained by feedback will inevitably increase, which will cause the CH to adjust the method employed for sending packets.

In Algorithms 2 and 3, all constraints of Equation (8) are satisfied. By optimizing the VANET link, the delay of transmitting data packets through VANETs is reduced as much as possible, enabling as few packets as possible to be transmitted through C-V2X, reducing the communication costs of users. This approach achieves the goal of balancing QoS and communication costs.

Algorithm 3: Routing Policy. The algorithm used to choose which way (DSRC or C-V2X) to send the packet.

Input: $Path_i, a_i, l_i^{DSRC}, \mathcal{B}_i \in a_i, Path_i$
Output: *None*
while $\mathcal{B} \neq \varnothing$ **do**
  $max \leftarrow 0$;
  $b_K \leftarrow None$;
  for *all* b_k *in* $\mathcal{B}$ **do**
    if $\omega_{ik} \geq max$ **then**
      $b_K \leftarrow b_k$;
    end
  end
  if $l_{iK} \leq u_i^{DSRC}$ **then**
    Send b_K to the next node of $Path_i$ using VANETs;
  else
    Send b_K to BS using C-V2X;
  end
  Update l_{ik} and ω_{ik} for all packages in $\mathcal{B}$;
  Update u_i^{CV2X} and u_i^{DSRC};
end

6. Results

This paper compares the CTDHR framework with the TDCR framework. First, the three parameters α, β, and d_{limit} of Equation (1) are selected. Second, the clustering results and communication performance of CTDHR and TDCR are compared. In the experiment, we still do not care about the communication within the cluster, and only consider the communication between the clusters.

6.1. Experimental Setting

1. The establishment of the simulation experiment environment. Before the experiment started, we used Python3.8 to build a simulation environment for vehicle communication scenarios. First of all, we observed the changes in the traffic flows at ten highway intersections in reality at 7:00 a.m.–9:00 a.m. and 3:00 p.m.–5:00 p.m. through video tapes, and obtained the speed characteristics and the interval distributions of vehicles entering the intersection. According to the observation results, the vehicle speeds obeyed normal distributions with a mean of 23 m/s and a variance of 6; the interval between two vehicles on the same lane entering the intersection obeyed a normal distribution with a mean of 3.7 s and a variance of 1.2. We then simulated a two-way six-lane highway with a length of 2000 m and generated vehicles with different speeds and directions at the beginning of the lanes based on the survey results. In addition, each vehicle had a different data generation rate. In this way, after a period of simulation, the road was filled with a certain number of vehicles. At this time, the possible collision of the vehicle must be considered. In the simulation environment of this paper, when two vehicles might have collided, the rear vehicle may have taken measures to change lanes and brake to avoid the collision. The speed of the vehicle also had a probability to change. Another important point is that the RSU was located in the middle of the road in the experiment. In addition, in the simulation program, the position of the vehicle was updated every 0.1 s. In previous vehicular communication simulations, there were insufficient numbers of vehicles [16,17,19,20], or insufficient mobility of vehicles (vehicles either did not move [16] or moved at a constant speed [21]). In this paper, the driving conditions of the vehicle in the real environment were fully imitated in the simulation environment to reflect the real performance of the vehicle communication method.

2. Experimental parameter settings. To verify the applicability of CTDHR under the high-speed movement of the vehicle, the speed range of the vehicle was 18 to 28 m/s. Moreover, the number of vehicles in the experimental scene obeyed the normal distribution of $\mathcal{N}(130, 5)$. Although the communication radius was extended to 2000 m in IEEE802.11bd, the road was usually curved, and there were obstacles between two vehicles to block the transmission of signals, so the communication radius of DSRC was set to 400 m in this paper. l_{ik} obeys the normal distribution of $\mathcal{N}(17, 8)$ in milliseconds, but its value is still limited to $\left[l^{min}, l^{max}\right]$. Since it is difficult to obtain the accurate data transmission rate of DSRC in the current research, according to the spectrum resources and sub-carrier spacing of 802.11bd [31], the data transmission rate of DSRC was estimated to be 200 Mbps using the Nyquist criterion and Shannon formula. Refer to Table 3 for other parameter settings.

Table 3. Experimental parameter settings.

Parameter	Value
Parameter α of Equation (1)	0.1
Parameter β of Equation (1)	-150
Maximum allowable distance that two clusters can be merged d_{limit}(meter)	170
Threshold for establishing a robust communication link $D^{vehicle}_{threshold}$	6
Length of road (meter)	2000
Number of vehicles	$\mathcal{N}(130, 5)$
Communication radius of DSRC R (meter)	400
DSRC sub-carrier spacing (KHz)	156.25
Speed of vehicle v (meters per second)	$[18, 28]$
Data transfer rate of DSRC f^{DSRC} (Mbps)	200
Data packet *Size* (Byte)	$[200, 1800]$
Maximum delay allowed l_{ik} for b_{ik} (ms)	$\mathcal{N}(17, 8)$
Vehicle's data generation rate r (Mbps)	$[0.4, 8]$
The unit price of C-V2X traffic charges p (per MB)	0.1
Upper limit of vehicles in a cluster N	15

3. To facilitate the comparison of the CTDHR and TDCR clustering algorithms, this paper uses cluster stability and the number of clusters to measure the quality of cluster results. The stability of a cluster is measured by the cluster hold time, which refers to the time from when the cluster is established to when the distance between any CM and CH in the cluster exceeds the DSRC communication radius. In addition, the number of clusters in the system should be kept in an appropriate range because a smaller number of clusters can reduce the number of hops on the VANET link, thereby reducing the delay of packet transmission. However, if the number of clusters is too small, the interval between two clusters will be too large to establish a communication link, or congestion may easily occur during communication.
4. Both CTDHR and TDCR were tested in the same simulation environment.
5. We implemented CTDHR and TDCR using Python 3.8. The hardware configuration of the experimental platform is detailed as follows: the processor is an Intel(R) Core(TM) i5-10400F CPU with 8 GB of memory and a 256 GB SSD.

6.2. Experimental Results

6.2.1. Parameter Selection

Choosing a good value for α, β, d_{limit} can enhance the stability of the cluster and control the number of clusters. In the experiment, we used the dichotomy method to roughly determine the value of α in $[0, 2.0]$, the value of β in $[-50, -200]$, and the value of d_{limit} in $[100, 300]$. After determining the value range, each different parameter combination was tested in 20 different road simulation environments, and the clustered results were obtained. Figure 5 is a box plot obtained in the test of the effect of different values of α, β, d_{limit} on the number of clusters and cluster stability.

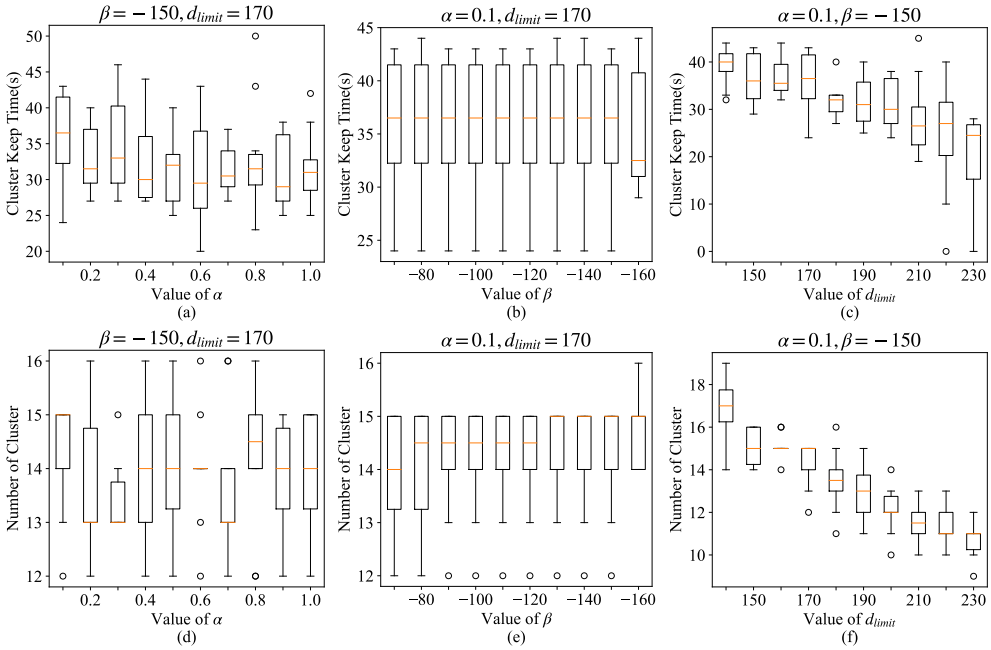


Figure 5. (**a**–**c**) are the effects of changing the α, β, d_{limit}, respectively, on the stability of clusters; (**d**–**f**) are the effects of changing the α, β, d_{limit}, respectively, on the number of clusters

According to Equation (1), α is the weight of Y_{ij}, but Y_{ij} is only affected by the current state of the vehicle, and the vehicle state may change at any time. Thus, reducing the influence of Y_{ij} on $D_{ij}^{clusterr}$ can increase the stability of the cluster. With a continuous decrease in α, the stability of the cluster shows an increasing trend and the outliers gradually decrease. So, the value of α is 0.1. If two clusters of c_i and c_j at the edge of each other's DSRC communication range are merged, the resulting cluster may be very unstable. By increasing β, the $D_{ij}^{clusterr}$ value of the two clusters can be reduced, thereby limiting the merging of these two clusters. However, as shown in Figure 5b,e, when β is greater than -160, the beta has less effect on cluster stability and the number of clusters. This is because the D-limit limits the maximum distance that clusters are allowed to merge, which makes it less likely that clusters at the edge of mutual DSRC communication can merge. At $\beta = -160$, the mean of the number of clusters increases, and in some cases the number of clusters starts to increase. So we take the value of BETA as -150. As shown in Figure 5e,f, d_{limit} has an important impact on the number and stability of clusters. The larger the d_{limit} is, the more distant the clusters can be merged. When d_{limit} is 170, a reasonable number of clusters can be obtained while maintaining good stability. At this time, the number of clusters will be reduced, and the clusters containing more CMs will be less stable. Therefore, as increases , both the stability of the cluster and the number of clusters tend to decrease. The final values of the three parameters are 0.1, -150, and 170.

6.2.2. Comparison of Clustering Methods

After completing the selection of parameters, Algorithm 1 was tested in 100 different simulation environments. Table 4 shows the comparison between Algorithm 1 and the TDCR clustering algorithm in 100 tests.

Table 4. Statistics on the stability of clusters and the number of clusters for CTDHR and TDCR.

	Framework	Average Value	Max Value	Min Value	Median Value
Cluster Keep Time	CTDHR	35.07	50.00	24.00	34.50
	TDCR	38.48	51.00	28.00	38.00
Number of Clusters	CTDHR	13.90	16.00	11.00	14.00
	TDCR	20.00	20.00	20.00	20.00

In TDCR, the method of spatial sampling is used to cluster vehicles, and the sampling radius is half of the DSRC communication radius (when R is 400 m, the sampling radius is 200 m), so the number of clusters in TDCR is constant, 20. In the clustering method of CTDHR, the parameter d_{limit} is used to control the size of the cluster, so the coverage of the cluster is not fixed. When d_{limit} is determined to be 170, the theoretical maximum coverage of the cluster is 340 m. Therefore, the clustering method of CTDHR can obtain fewer clusters. Moreover, in the clustering method of this paper, the one-hop connection between the CM and the CH is also realized. The small number of clusters means that the number of hops in the VANET link can be reduced as much as possible. When there is no congestion in the network, fewer hops make each packet forward as few times as possible, thereby reducing the total delay in the network. Moreover, fewer clusters mean fewer communication resources of the cellular network are required. Fewer clusters also facilitate cluster management and reduce the consumption of communication resources.

Due to the speed difference between vehicles, the vehicle may change its current state at any time. Therefore, the more vehicles in the cluster, the worse the stability of the cluster. When the number of vehicles on the road is the same, the smaller the number of clusters, the higher the average number of vehicles in the cluster. Therefore, the cluster stability of CTDHR is slightly worse than that of TDCR. In Equation (1), Y_{ij} considers merging clusters that can stay connected longer, while $(R - d_{ij})^{-1}$ restricts the merging of clusters at the edge of the mutual DSRC communication radius. Therefore, the stability of CTDHR is also fully guaranteed, and the loss of various statistical features is within an acceptable range compared with TDCR. In addition, the number of vehicles in a single CTDHR cluster will not exceed N, which avoids the situation that the number of CMs in a single cluster is too much to exceed the communication capability of the CH. However, this situation may appear in TDCR.

6.2.3. Comparison of VANET Communication Link

After completing the establishment of the cluster and the election of the CH, the communication link is established on the cluster obtained in Section 5.1 using Algorithm 2. For comparison, the VANET link of TDCR was also established. Table 5 shows a comparison of 100 groups of data obtained by the test.

Table 5. Statistics on the stability of clusters and the number of clusters for CTDHR and TDCR.

	Framework	Average Value	Max Value	Min Value	Median Value
Total number of hops in a single test	CTDHR	30.75	40.00	23.00	30.00
	TDCR	47.88	61.00	40.00	47.00
Maximum number of hops in a single test	CTDHR	3.80	5.00	3.00	4.00
	TDCR	4.36	6.00	3.00	4.00
Average number of hops in a single test	CTDHR	2.21	2.92	1.87	2.17
	TDCR	1.54	2.00	1.15	1.50

Notes: In each test, each CH generates a VANET communication link to the RSU. The total number of hops in a single test is the sum of the hops of all links in a single test. The maximum number of hops in a single test refers to the number of hops of the longest link in a single test. The average number of hops in a single test is the total number of hops in a single test divided by the number of CHs.

Since the number of clusters in CTDHR is smaller than TDCR, its total hop count is also smaller than TDCR in the test. In Algorithm 2, when the VANET link capacity cannot meet the node's demand, the node will look for other VANET paths. The newly identified path is no longer the shortest, so CTDHR is larger than TDCR in terms of the average hop count. However, due to the small number of clusters in CTDHR, the maximum number of hops of a single link in CTDHR is still smaller than that in TDCR. Thus, the maximum number of 'forwarding' in CTDHR is less than TDHR for packets sent via VANETs when

another condition is the same. Therefore, CTDHR generates less forwarding delay and can better satisfy the QoS of data packets.

6.2.4. Vehicle Communication Performance Test

After completing the link construction, the performances of CTDHR and TDCR were compared in the test, where the data generation rate of the vehicles gradually increased from 0.4 to 8 Mbps in steps of 0.4. At each data generation rate, tests were performed in 100 different simulation environments and the test results were averaged. The duration of a single test was 1 s. Figure 6 shows the results of the objective equations, total delay, and cellular network bandwidth consumption obtained from the CTDHR and TDCR comparison experiments.

The overall latency of CTDHR is less due to the smaller number of clusters and shorter VANET link for CTDHR at the same vehicle density. Moreover, the routing policy of CTDHR uses VANET to transmit data packets as much as possible, so the cellular network bandwidth consumption of CTDHR is also smaller than that of TDCR. As shown in Figure 6b, when the data generation rate of the vehicle is greater than 4.4 Mbps, the delay of TDCR suddenly increases, indicating that congestion occurs in the network at this time. However, CTDHR does not experience congestion until the data generation rate is greater than 5.2 Mbps. CTDHR reduces the time delay and cellular network bandwidth consumption generated in the system as much as possible, so the result of its objective equation is also smaller than that of TDCR.

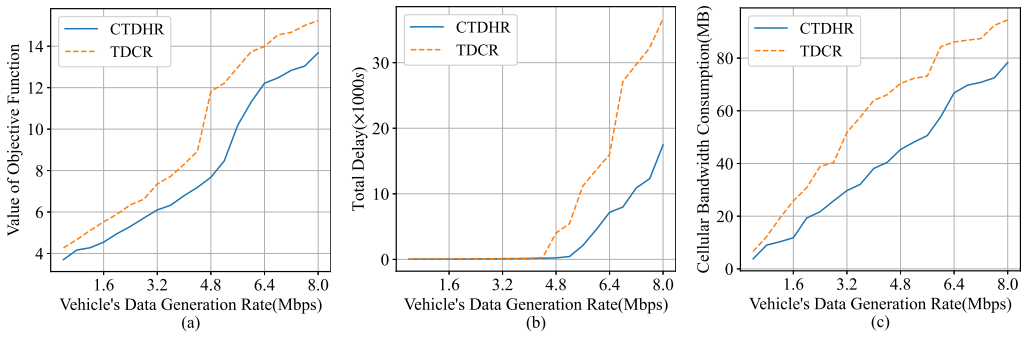


Figure 6. (**a**–**c**) The comparisons of CTDHR and TDCR for objective function results, total delay, and cellular network bandwidth consumption, respectively.

In addition to cellular network bandwidth consumption and delay, we also compared the QoS satisfaction rate and packet loss rate of the two methods. In the experiment, if the real delay from the generation of b_{ik} to the transmission to the RSU/BS was less than l_{ik}, the QoS requirements were considered as met.

As shown in Figure 7, CTDHR has better performance in terms of both packet loss rate and QoS satisfaction rate. CTDHR can effectively address the situation in which the remaining capacity of the link is less than the data generation rate of the newly added node when generating the routing path. Therefore, CTDHR avoids the situation where traffic is concentrated on one node, which effectively relieves the pressure on each forwarding node in the topology and is less likely to cause congestion than TDCR. The final performance is that with an increase in the vehicle data generation rate, the packet loss rate of CTDHR is much smaller than that of TDCR. In addition, the CTDHR routing policy is more flexible. According to the QoS of each packet, the appropriate transmission methods are selected, which also makes CTDHR satisfy the QoS of the packet at a higher rate.

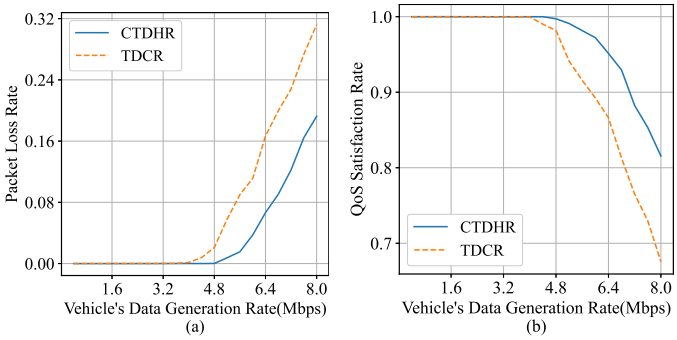


Figure 7. Comparison of CTDHR and TDCR in the communication performance. (**a**) The comparison of the packet loss rate, (**b**) the comparison of the QoS satisfaction rate

7. Conclusions

This paper redesigns the vehicle cluster method, communication link establishment method, and routing policy based on the TDCR idea. In the vehicle clustering method, we no longer adopt the spatial sampling method of TDCR; we redesigned a clustering method that is suitable for vehicle clustering based on the idea of hierarchical clustering. In CTDHR, a new cluster model was designed, and the parameters in Equation (1), which reduce the number of clusters while ensuring the stability of clusters, were carefully selected. The link capacity was fully considered when establishing the communication link so that the network obtained a greater throughput. The flexible routing strategy enables data packets to choose the appropriate transmission method according to the environmental conditions.

Through the above improvements, CTDHR reduces the packets sent through the cellular network while ensuring QoS, so that more packets are sent through VANETs, which reduces the communication costs of the user. In the comparative experiment with TDCR, CTDHR achieves superiority in the following aspects: the number of clusters, the maximum number of hops of the VANET communication link, and the communication performance. Although CTDHR is slightly inferior to TDCR in terms of cluster stability and maximum hop count of VANET links, this is within the acceptable range. It can be seen that even with the above optimizations, the throughput of CTDHR is still low. In future research, we will attempt to (1) introduce deep learning methods to solve the problem of link generation in VANETs, and (2) use reinforcement learning to obtain better routing strategies.

Author Contributions: Conceptualization, Y.Z. and J.Z.; methodology, Y.Z. and J.Z.; software, Y.Z.; formal analysis, Y.Z.; investigation, Y.Z.; resources, J.Z.; data curation, Y.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, Y.Z. and J.Z.; visualization, Y.Z.; supervision, J.Z.; project administration, J.Z.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by the National Natural Science Foundation of China (nos. 62062065, 12071417, 61962061), a project of the Natural Science Foundation of Yunnan Province of China (2019FB142 and 2018ZF017), the Education Foundation of Yunnan Province of China (2022J002), and the Program for Excellent Young Talents, Yunnan, China.

Data Availability Statement: The code used in the experiment can be obtained through the e-mail of the first author or corresponding author.

Conflicts of Interest: The authors declare no conflict of interest. All authors agree with all content of this paper without any objection. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Luo, Q.Y.; Hu, S.H.; Li, C.L; Li, G.H; Shi, W.S. Resource Scheduling in Edge Computing: A Survey. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 2131–2165. [CrossRef]
2. Fu, Y.C.; Li, C.L.; Yu, F.R; Tom, H.L; Zhang, Y. A Survey of Driving Safety With Sensing, Vehicular Communications, and Artificial Intelligence-Based Collision Avoidance. *IEEE Trans. Intell. Transp. Syst.* **2021**, *19*, 1–30.
3. Xia, Z.C.; Wu, J.; Wu, L.B; Chen, Y.J; Yu, P.S. A Comprehensive Survey of the Key Technologies and Challenges Surrounding Vehicular Ad Hoc Networks. *ACM Trans. Intell. Syst. Technol.* **2021**, *12*, 1–22. [CrossRef]
4. Ahmed, E.; Gharavi, H. Cooperative vehicular networking: A survey. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 996–1014. [CrossRef] [PubMed]
5. Lu, S.D.; Yuan, X.; Shi, W.S. Edge Compression: An Integrated Framework for Compressive Imaging Processing on CAVs. In Proceedings of the 2020 IEEE/ACM Symposium on Edge Computing (SEC), San Jose, CA, USA, 12–14 November 2020; pp. 125–138.
6. Azzedine, B.; Zhang, Q. Countermeasures against Worm Spreading: A New Challenge for Vehicular Networks. *ACM Comput. Surv.* **2020**, *52*, 1–25.
7. Blesson, V.; Wang, N.; David, B.; Eyal, D.L.; Shi, W.S.; Christopher, S. A Survey on Edge Benchmarking. *ACM Comput. Surv.* **2022**, *54*, 1–33.
8. Zhang, J.X.; Yang, X.T.; Xie, N. An online auction mechanism for time-varying multidimensional resource allocation in clouds. *Future Gener. Comput. Syst.* **2020**, *111*, 27–38. [CrossRef]
9. Yang, H.J.; Shen, J.; Zhou, T.Q.; Ji, S.; Pandi, V. A Flexible and Privacy-Preserving Collaborative Filtering Scheme in Cloud Computing for VANETs. *IEEE Trans. Internet. Technol.* **2022**, *22*, 1–19. [CrossRef]
10. Badreddine, Y.Y.; Toufik, A.; Mohamed, M. Implementation and Assessment of IEEE 802.11BD for Improved Road Safety. In Proceedings of the IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2021; pp. 1–6.
11. Mohammad, K.; Mohammadreza, M.; Edward, A.; Rachel, D.; Sarthake, C.; Lou, Y.Y.; Aviral, S. A Survey on Intersection Management of Connected Autonomous Vehicles. *ACM Trans. Cyber-Phys. Syst.* **2020**, *4*, 1–27.
12. Won, M. L-Platooning: A Protocol for Managing a Long Platoon With DSRC. *IEEE Trans. Intell. Transp. Syst.* **2021**, 1–14. [CrossRef]
13. Qi, W.J.; Landfeldt, B.; Song, Q.Y.; Guo, L.; Jamalipour, A. Traffic Differentiated Clustering Routing in DSRC and C-V2X Hybrid Vehicular Networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 7723–7734. [CrossRef]
14. Ghafoor, K.Z.; Guizani, M.; Kong, L.; Maghdid, H.S.; Jasim, K.F. Enabling Efficient Coexistence of DSRC and C-V2X in Vehicular Networks. *IEEE Wireless Commun.* **2020**, *27*, 134–140. [CrossRef]
15. Mir, Z.H.; Toutouh, J.F.; Ko, Y.B. Enabling DSRC and C-V2X Integrated Hybrid Vehicular Networks: Architecture and Protocol. *IEEE Access* **2020**, *8*, 180909–180927. [CrossRef]
16. Heo, S.; Yoo, W.; Jang, H.; Chung, J.M. H-V2X Mode 4 Adaptive Semipersistent Scheduling Control for Cooperative Internet of Vehicles. *IEEE Internet Things* **2021**, *8*, 10678–10692. [CrossRef]
17. Elbal, B.R.; Rupp, M. Coexistence of DSRC and C-V2X communication: modeling a competing scenario. In Proceedings of the IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Helsinki, Finland, 13–16 September 2021; pp. 1151–1156.
18. Su, H.; Zhang, X. Clustering-Based Multichannel MAC Protocols for QoS Provisionings Over Vehicular Ad Hoc Networks. *IEEE Trans. Veh. Technol.* **2007**, *56*, 3309–3323.
19. Abboud, K.; Zhuang, W.H. Impact of Microscopic Vehicle Mobility on Cluster-Based Routing Overhead in VANETs. *IEEE Trans. Veh. Technol.* **2015**, *64*, 5493–5502. [CrossRef]
20. Prajapati, J.; Chauhan, P. User-Interest Based Clustering for Efficient Vehicle Communication in VANET. In Proceedings of the 2018 3rd International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 15–16 November 2018; pp. 247–253.
21. Kumar, V.D.; Kanagachidambaresan, G.R.; Chyne, P.; Kandar, D. Extended Communication Range for Autonomous Vehicles using Hybrid DSRC/WiMAX Technology. *Wirel. Pers. Commun.* **2022**, *123*, 2301–2316. [CrossRef]
22. Sennan, S.; Ramasubbareddy, S.; Balasubramaniyam, S.; Nayyar, A.; Kerrache, C.A.; Bilal, M. MADCR: Mobility aware dynamic clustering-based routing protocol in Internet of Vehicles. *China Commun.* **2021**, *18*, 69–85. [CrossRef]
23. Vasilis, M.; Dries, N.; Seilendria, H.; Daniel, V.D.A.; Johann, M.B. Experimental V2X Evaluation for C-V2X and ITS-G5 Technologies in a Real-Life Highway Environment. *IEEE Trans. Netw. Serv. Manag.* **2021**, *19*, 1–19.
24. Pablo, B.B.; Luis, F.U.; Mnica, A.I.; Diego, J.R.; Martha, C.P.P. An Evaluation of OMNeT++-based V2X Communication Frameworks: On the Path Towards 5G-V2X Simulations. In Proceedings of the 24th International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '21), New York, NY, USA, 22 November 2021; pp. 75–78.
25. Tang, Y.J.; Cheng, N.; Wu, W.; Wang, M.; Dai, Y.P.; Shen, X.M. Delay-Minimization Routing for Heterogeneous VANETs With Machine Learning Based Mobility Prediction. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3967–3979. [CrossRef]
26. Ansari, K. Joint use of DSRC and C-V2X for V2X communications in the 5.9 GHz ITS band. *IET Intell. Transp. Syst.* **2021**, *15*, 213–224. [CrossRef]
27. Zhang, J.X.; Lou, W.L.; Sun, H. Truthful auction mechanisms for resource allocation in the Internet of Vehicles with public blockchain networks. *Future Gener. Comput. Syst.* **2022**, *132*, 11–24. [CrossRef]

28. Yan, Y.; Zhang, J.X.; Lou, W.L. Double Action Mechanism for Vehicle Edge Computing Resource Allocation and Pricing. In Proceedings of the 2021 4th International Conference on Information Communication and Signal Processing (ICICSP), Shanghai, China, 24–26 September 2021; pp. 547–552.
29. Quy, V.Khanh.; Nam, V.H.; Linh, D.M.; Ban, N.T.; Han, N.D. Communication solutions for vehicle ad-hoc network in smart cities environment: a comprehensive survey. *Wirel. Pers. Commun.* **2021**, *18*, 1–25.
30. Verbraeken, J.; Wolting, M.; Katzy, J.; Kloppenburg, J.; Verbelen, T.; Rellermeyer, J.S. A survey on distributed machine learning. *ACM Comput. Surv.* **2021**, *53*, 1–33. [CrossRef]
31. Khan, M.A.; Sayed, H.E.; Malik, S.; Zia, T.; Khan, J.; Alkaabi, N. Level-5 autonomous driving are we there yet? a review of research literature*ACM Comput. Surv.* **2022**, *27*, 1–38.