Design of a Reliable Transmission Mechanism for Vehicle Data in Mobile Internet of Vehicles Driven by Edge Computing

Wenjing Liu
Chongqing Technology and Business Institute, Chongqing 401520, China

***Abstract*—In order to meet the business requirements of different applications in heterogeneous, random, and time-varying mobile network environments, the design of a reliable transmission mechanism is the core problem of the mobile Internet of vehicles. The current research is mainly based on the computing power support of roadside units, and large delays and high costs are significant defects that are difficult to overcome. In order to overcome this deficiency, this paper integrates edge computing to design task unloading and routing protocol for the reliable transmission mechanism of mobile Internet of vehicles. Firstly, combined with edge computing technology, a mobile-aware edge task unloading mechanism in a vehicle environment is designed to improve resource utilization efficiency and strengthen network edge computing capacity so as to provide computing support for upper service applications; Secondly, with the support of computing power of edge task unloading mechanism, connectivity aware and delay oriented edge node routing protocol in-vehicle environment is constructed to realize reliable communication between vehicles. The main characteristics of this research are as follows: firstly, edge computing technology is introduced to provide distributed computing power, and reliable transmission routing is designed based on vehicle-to-vehicle network topology, which has prominent cost advantages and application value. Secondly, the reliability of transmission is improved through a variety of innovative technical designs, including taking the two hop range nodes as the service set search to reduce the amount of system calculation, fully considering the link connectivity state, and comprehensively using real-time and historical link data to establish the backbone link. This paper constructs measurement indicators based on delay and mobility as key elements of the computing offloading mechanism. The offloading decision is made through weighted calculation of delay estimation and computing cost, and a reasonable computing model is designed. The experimental simulation shows that the average task execution time under this model is 65.4% shorter than that of local computing, 18.4% shorter than that of cloud computing, and the routing coverage is about 6% higher than that of local computing when there are less than 60 nodes. These research and experimental results fully demonstrate that the mobile Internet of vehicles based on edge computing has good reliable transmission characteristics.**

Keywords*—*Mobile network; internet of vehicles; reliable transmission; edge computing

I. Introduction

Vehicular ad hoc networks (VANETs) are the basic networking mode of mobile Internet of vehicles. It mainly relies on a vehicle-to-vehicle (V2V) and vehicle-to-roadside unit (V2R) to provide a variety of data transmission and information interaction services [1]. The concept of Internet of Vehicles is extended from the Internet of Things. With the rapid development of Internet of Things technology and applications, especially the significant progress of sensing technology in the perception layer of the Internet of Things, it provides real-time perception and feedback of operating vehicle and road condition information, providing an essential big data foundation for the research and application of the Internet of Vehicles.

Unlike traditional networks, the mobile Internet of vehicles has unique internal characteristics regarding the network environment, node mobility, channel characteristics, computing power, cache space, and energy constraints. In particular, the dynamic change of topology leads to frequent network segmentation, which is difficult to ensure the end-to-end connected link; In addition, the massive data of diversified new applications have great pressure on the response delay and network load, which poses a severe challenge to the computing power of car coupling network [2]. Therefore, studying the characteristics of mobile vehicle networks and building a reliable transmission mechanism is the core problem to be solved urgently.

Currently, the research and application of reliable transmission mechanisms of mobile vehicle networks far lag behind the industrial technology demand, which does not match the current development of mobile vehicle networking. On the other hand, the research of introducing edge computing into a reliable transmission of mobile Internet of vehicles has been widely concerned and applied. The overall research status in this field is briefly described below.

As early as 2009, the United States released the intelligent transportation strategic research plan. In 2016, China released the development plan for the innovation of the mobile Internet of vehicles, focusing on the deployment and promotion of common key technologies, standards, infrastructure construction, platform experimental verification, application, and promotion, and successively launched a series of standards and specifications for the mobile Internet of vehicles industry, including Tencent, Alibaba, and Baidu has also established corresponding cooperation with various car enterprises. International academic circles, such as IEEE, have founded IEEE Transactions on vehicular technology and other top journals, bringing together important innovative research achievements in the world. Although the research on the reliable transmission of the mobile Internet of vehicles has made great progress, considering the scale of the mobile vehicle network, the rapid movement of vehicles, and the complex channel environment, ensuring the reliable transmission of information is still a very challenging problem [3][4].

Firstly, the location-based routing mechanism is a widely used key technology to solve data transmission. However, the existing related work still has corresponding limitations in node mobility characterization, routing index modeling, and relay node selection. It is mainly reflected in the following:

1) The vehicle network's scale, complexity, and dynamics are not fully considered. The existing routing mechanism is often applicable to local or single network form, resulting in the algorithm falling into optimal local solution and difficulty in adapting to the dynamic changes of mobile vehicle network effectively;

2) Rely on historical traffic information to quickly find routes in sparse scenarios and alleviate the occurrence of local optima while ignoring the positive role of real-time link information in assisting data transmission and avoiding collisions in congested flow scenarios.

Secondly, the effective calculation of data plays a great role in strengthening the performance of vehicle communication, and the calculation of massive data brings great pressure on the network bandwidth. By sinking the cloud computing function to the user side, edge computing will greatly reduce the network load and network delay; the current research work has the following deficiencies:

1) Most of the existing computing unloading mechanisms rely on the assistance of roadside unit RSU and fail to fully explore a large number of idle intelligent vehicle resources to improve the edge performance of the network;

2) Although some edge computing designs consider the impact of intermittent connection caused by vehicle movement on computing unloading, they do not make full use of the service opportunities created by vehicle movement, ignoring that service vehicles far away (vehicles providing computing services) still have the opportunity to enter the communication range of task vehicles (vehicles requiring data computing) and participate in task computing.

To sum up, it is an important research and application direction of the current mobile vehicle networking to effectively combine data communication with calculation to provide computational support for the routing and distribution of mobile vehicle network data with the formulation of calculation unloading decisions [5] and to achieve the reliable transmission of mobile vehicle network data.

The rest of this paper will be organized as follows: Section II gives research contribution of this paper, Section III presents design of edge task unloading mechanism, Section IV elucidates edge routing design, Section V concludes the paper.

II. Research Contribution of this Paper

It is estimated that the scale of the Internet of vehicles industry will reach 200 billion yuan in 2025. Benefiting from the development of new generation communication technology, the mobile Internet of vehicles has strong ubiquitous interconnection ability, intelligent processing ability, and big data processing ability. It organically connects the traffic elements of the on-board network, including people, vehicles, roads, and clouds, breaks through the limitations of single-vehicle information perception and processing, and achieves the purpose of strengthening safety, improving efficiency, improving the environment and saving energy, It has become the core field of scientific and technological innovation and industrial development in the world, and has spawned a series of new technologies, new products, and new services.

Firstly, in terms of the outstanding feature of the dynamic topology of the mobile Internet of vehicles, different from the traditional wireless mobile network, the rapid movement of vehicle nodes is easy to cause the interruption of transmission links and affects the successful reception of information. Urban buildings, obstacles, and random channel environments exacerbate signal transmission instability, making the traditional network transmission mechanism difficult to work in the environment of the Internet of vehicles. Because the traditional network data transmission protocol is difficult to adapt to the frequent changes of topology and high-speed movement of nodes in the vehicle environment and is limited by the communication range of vehicles and the scale of vehicle network, in the design of multi-hop information transmission mechanism widely used in the industry, path selection and relay node selection are two key problems. This paper constructs a delay model based on the distribution and motion characteristics of road nodes, and weights the calculation of delay and computational cost for the next hop forwarding node's routing selection. The experiment shows that the coverage can be improved by 6% in sparse scenarios.

Secondly, in terms of the outstanding characteristics of the massive data of the mobile Internet of vehicles, the endless on-board applications pose a severe challenge to the computing power of the mobile Internet of vehicles, especially the new services promoted by communication technology and equipment manufacturing, such as augmented reality (AR) and automatic driving. A single vehicle with limited resources cannot meet the computing requirements of the above services. The on-board network based on cloud computing can improve service performance by integrating communication and computing resources, but it will lead to unpredictable delays. Therefore, this paper introduces vehicular edge computing (VEC). By taking delay and mobility as the key elements of the measurement indicators to build the measurement indicators of an effective calculation unloading mechanism, a weighted calculation model based on weight is designed, and experimental data validation is carried out to make up for the shortcomings of local computing and cloud computing.

In short, the mobile Internet of vehicles is the key technical means to realize the smart city and intelligent transportation. In view of the prominent characteristics and application challenges of its lack of dynamic topology and computing power, this paper designs a reliable transmission mechanism for the mobile Internet of vehicles driven by edge computing, which provides a certain reference value for scientific research and application.

III. Design of Edge Task Unloading Mechanism

Most current research relies on roadside service units with rich computing and storage resources or integrates multiple edge servers to calculate vehicle unloading tasks. However, deploying edge servers in all sections will bring huge economic costs. The ideal way is to unload the computing tasks of task vehicles to multiple service vehicles. The service vehicle executes each subtask and feeds back the results to the task vehicle [6]. After receiving all the calculation results, the task vehicle starts to run the application. Therefore, from the perspective of vehicle unloading mechanism research, this paper integrates the parked idle vehicle resources to provide edge computing power and overcomes the defect that the influence of vehicle mobility on the unloading decision is not fully considered in the existing unloading mechanism. In the design of the unloading mechanism, the core measurement indicators introduced include response delay (including local processing time, data upload time, processing time, and feedback time), incentive mechanism (encouraging vehicles participating in task computing to overcome the selfishness of nodes), mobility (establishing a mobile model to predict and evaluate the effective link time of vehicles).

A. Search and Optimization of Service Vehicles

By adding its computing resource information to the beacon, the service vehicle indicates its availability to the surrounding vehicle nodes and brings the one-hop and two-hop vehicles relative to the task vehicle into the search range (Fig. 1). It is necessary to study and design the construction method of the service vehicle group according to distance, moving direction, speed, and angle, and optimize and model the service vehicle group by using the result measurement index of throughput according to time delay [7].

Vehicles exchange speed and position information with each other through periodic broadcast beacons. Service vehicles can indicate their availability to surrounding vehicles by adding their computing resource information to the transmitted beacon information[8]. When a task vehicle needs to process a task, it can find the service vehicle that can be used to participate in the calculation and unloading by listening to the beacon information from the surrounding nodes.

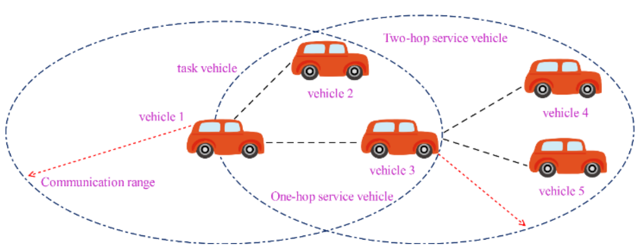

Fig. 1. Example of a two-hop search range.

1) One-hop service vehicles: For a task vehicle, the vehicles within its one-hop range are called one-hop vehicles, which are candidate service vehicles. This is because the task vehicle can directly communicate with vehicles within one hop, which provides favorable conditions for task unloading in the workshop. When both vehicle 2 and vehicle 3 are within the communication range of vehicle 1, vehicle 1 can unload tasks to vehicle 2 and vehicle 3 directly through workshop communication to seek assistance.

2) Two-hop service vehicles: Vehicles within the two-hop range of mission vehicles are potential service vehicles, such as vehicle 4 and vehicle 5, which are called two-hop service vehicles. Due to the limitation of communication range, although the two-hop vehicle cannot directly communicate with the task vehicle, with the help of vehicle mobility, the two-hop vehicle has the opportunity to travel within the communication range of the task vehicle so that it can serve the task vehicle. In the initial stage, although the task vehicle. Such as, vehicle 1 cannot directly communicate with its two-hop vehicle, but it can indirectly obtain the speed, location, computing resources, and other information of the two-hop vehicle through the relay of one-hop vehicles, such as vehicle 2 and vehicle 3. Based on the obtained information, the task vehicle can judge whether the two-hop vehicle meets the conditions of task unloading. If a two-hop vehicle is selected as a service vehicle to assist the task vehicle in processing the task, it first needs to obtain the task of unloading the task vehicle through the relay of the one-hop vehicle [9]. After the selected two-hop vehicle completes the assigned task, once it enters the communication range of the task vehicle, it will feed back the result to the task vehicle.

3) Service collection optimization: We introduce the concept of expected throughput (ET), the expected value of throughput between vehicle user nodes and sink nodes during road movement. ET comprehensively considers the average level of throughput within the physical coverage of sink nodes, which can be used to measure the effectiveness of service integration scheme optimization [10].

The expected throughput parameter can help design performance test scenarios. Based on the estimated throughput data, it can correspond to the frequency and number of transactions in the test scenario. After the test is completed, it can measure whether the algorithm achieves the expected goal according to the actual throughput.

If P_i is set as the actual link throughput when T (x) is equal to T_i, the corresponding probability can be seen from the concept of ET:

$$ET = \sum_{i=1}^{N} P_i T_i \quad (1)$$

To get the value of ET, we need to solve each P_i first. Let L be the distance between the sink node and the vehicle user node, and T (x) represents the throughput of the actual link. According to the knowledge of probability theory, the corresponding probability when T (x) is equal to Ti is:

$$P_i = P(l_i < L < l_{i+1}), \ i=1,2,\ldots,N-1; \quad (2)$$

$$P_i = P(0 < L < l_i), \text{ i=N} \quad (3)$$

According to the research results of reference [11], the specific form of the Formula (2) and (3) can be expressed as (4) and (5):

$$P_i(i \neq N) = \frac{(1+2\ln a)(a_i^2 - a_{i+1}^2)}{a^2} - \frac{2a_i^2 \ln a_i - 2a_{i+1}^2 \ln a_{i+1}}{a^2} + \frac{(1+2\ln a)(a_{2N-i}^2 - a_{2N-i+1}^2)}{a^2} - \frac{2a_{2N-i}^2 \ln a_{2N-i} - 2a_{2N-i+1}^2 \ln a_{2N-i+1}}{a^2} \quad (4)$$

$$P_i(i = N) = \frac{(1+2\ln a)(a_N^2 - a_{N+1}^2)}{a^2} - \frac{2a_N^2 \ln a_N - 2a_{N+1}^2 \ln a_{N+1}}{a^2} \quad (5)$$

Where a_i, a_{i+1}, a_{2N-i}, a_{2N-i+1} are the change points when t_i is the link throughput.

B. Modeling and Analysis of Link Connectivity

The connectivity time between vehicle nodes is used to describe the connectivity of links, which represents the duration of effective links when the workshop distance is less than the vehicle communication range. It is necessary to study and design the network connectivity model under the relative position of one hop or two hops, opposite or reverse driving, to provide a basis for selecting service vehicles for task unloading [12].

For vehicles A and B, if the workshop distance between them is less than the communication range of the vehicle, there is an effective link between the two vehicles. Therefore, we can use the connectivity time of the workshop to describe the connectivity of the link.

Suppose (x_A, y_A) and (x_B, y_B) are their respective coordinates, and V_A and V_B are their respective velocities. At the same time, D_{AB} is defined as the initial distance between two vehicles:

$$D_{AB} = (x_B - x_A) \quad (6)$$

According to the respective states of the two vehicles, we calculate the connection time between the two vehicles through the following two cases.

1) Driving in the same direction: If $V_A > V_B$, the direct connection time between the two vehicles can be expressed as:

$$T_{life} = \frac{R+D_{AB}}{|V_B-V_A|} \quad (7)$$

Where R is the communication radius of the two vehicles.

Otherwise, the connection time of the two vehicles can be expressed as follows:

$$T_{life} = \frac{R-D_{AB}}{|V_B-V_A|} \quad (8)$$

2) Driving in different directions: When two vehicles are driving in opposite directions, the connection time of the two vehicles can be expressed as:

$$T_{life} = \frac{D_{AB}+\sqrt{R^2-(y_A-y_B)^2}}{V_B+V_A} \quad (9)$$

Otherwise, the connection time of the two vehicles can be expressed as follows:

$$T_{life} = \frac{D_{AB}-\sqrt{R^2-(y_A-y_B)^2}}{V_B+V_A} \quad (10)$$

C. Edge Unloading Algorithm

It is defined here that edge service vehicles have different computing power, and the computing power of edge service vehicles is expressed by C_e. The vehicle generates a calculation task T_A at A:

$$T_A = \{D_{in}, D_{out}, C_{comp}, T_{max}\} \quad (11)$$

where D_{in} represents the data amount of the task, D_{out} represents the output amount of the result, C_{comp} represents the calculation amount of the task, and T_{max} represents the maximum completion time that the task can tolerate.

1) Upload phase: We use t to represent the period when the number of hops does not change, which is determined by the vehicle's movement. In a period t, the equivalent bandwidth BW does not change, so in a period t_x, the amount of uploaded data $D_{upx} = t_x \times BW_x$. We assume that at a time t_i, the number of hops changes n times. When the first (n-1) change is completed, it is t_j.

$$\sum_{x=0}^{n-1} D_{upx} + (i-j)\, BW_n = D_{in} \quad (12)$$

When the above formula is satisfied, the task upload is completed, and the whole-time span T_U is recorded as the upload time of the task.

2) Calculation stage: Take the calculation unloading of the task once, that is, switching from task vehicle A to service vehicle B as an example. Here, E_A represents the computing power of vehicle A and E_B represents the computing power of vehicle B.

Then the task calculation time T_C is:

$$T_C = {}^{C}/_{E_A} \quad (13)$$

The download time of the calculation result is:

$$T_D = {}^{D_{out}}/_{BW_A} \quad (14)$$

Where BW_A represents the equivalent bandwidth of the result downloaded from task vehicle A to service vehicle B.

Then the task completion time is:

$$T_{comp} = T_C + T_D \quad (15)$$

3) Download phase: The principle is the same as that of the upload stage, so in a certain period t_x, the amount of downloaded data $D_{D_x} = t_x \times BW_x$. We assume a time t_1, when the number of hops changes n times, and the time when the (n – 1)st change is completed is t_2.

$$\sum_{x=0}^{n-1} D_{D_x} + (t_1 - t_2)\, BW_n = D_{out} \quad (16)$$

When the above formula is satisfied, the task upload is completed, and the whole-time span T_D is recorded as the download time of the task.

To optimize the completion time of tasks in unloading, the problem optimization model is as follows:

$$\text{Min}\ [T_U + T_C + T_D] \tag{17}$$

D. Edge Unloading Simulation

The statistical data of this experiment is obtained from five groups of simulation experiments based on six vehicle nodes. Under the constraints of the same amount of task calculation, compare the simulation results of the average task execution time and average task completion time of the 2-hop experimental scenario, and keep two decimal places for all parameters.

It can be seen from the Fig. 2 that in terms of average task execution time, the execution time of tasks under the edge unloading calculation mode is the shortest, which is 62.38 seconds, and compared with the average task execution time calculated by itself of 179.4 seconds, it is shortened by 65.4%; The execution time in the cloud is 76.46 seconds, which is 57% shorter than the vehicle's own calculation. This is because once the task is generated, the calculation amount is a fixed value, sorted according to the calculation power: edge > cloud > itself.

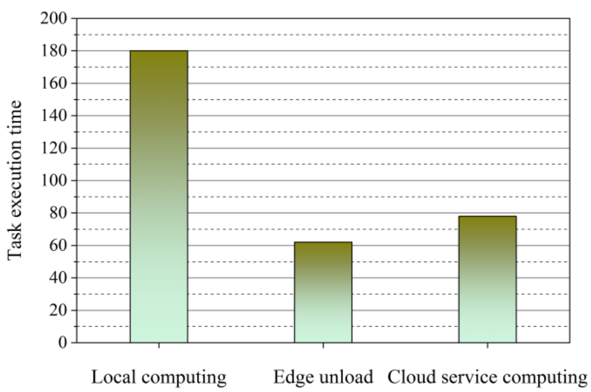

Fig. 2. Task unloading algorithm performance results.

The experimental results show that when only the task execution time is considered, the task will not be calculated locally and should be unloaded to the edge node with strong computing power. It can be seen that the execution time and communication time of the generated tasks need to be considered. When the tasks in the cloud are executed and coordinated, although the cloud has rich computing resources, if a large number of tasks are uploaded to the cloud for execution, it is bound to increase the burden on the cloud in the core network.

IV. Edge Routing Design

It is designed to build a delay model based on road section nodes' distribution and movement characteristics. Combined with the connectivity model established in the previous discussion, it is necessary to introduce the road section (as shown in Fig. 3) evaluation mechanism for the evaluation of road weight and then establish a backbone link composed of a series of intersection edge nodes and in road edge nodes for all roads. The intersection edge node is responsible for calculating and distributing the road weight, while the inner edge node is used for transmitting information [13]. The optimal path selection algorithm is designed according to the road weight. According to the complexity, the node can calculate the route itself or through task unloading to avoid local optimization and data congestion.

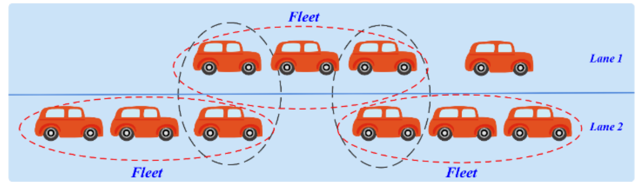

Fig. 3. Example of road section composition.

A. Algorithm Framework

V2V communication mostly adopts two specific data packet formats, including cooperative sensing messages. It is mainly used to periodically broadcast vehicle information and distributed environment notification message [14][15]. It is based on event triggers and is mostly used for vehicle safety alarms and emergency notifications. The cooperative sensing message can periodically broadcast application requests and network transport layer requirements (network heartbeat) according to the specified broadcast frequency. Its broadcast frequency is determined by the communication management entity based on the application scenario and network status. By analyzing the received data packet, the vehicle node can obtain other vehicle information within its driving range so as to obtain the network topology information and vehicle node information in this area, including vehicle position, direction, speed, stability, i.e., acceleration and destination, etc.

The broadcast of security alarm messages and other burst service information is mostly sent through data packets. It has the highest access priority and simple Dayton format design and can flexibly transmit various business applications [16]. The data format of the emergency message mainly consists of starting characters, identification code, data unit, and check code, with a total length of 25 bytes.

This routing mechanism design applies the advantages of real-time traffic estimation in vehicle and road condition information prediction to the broadcast routing protocol, fully considers the relevant factors affecting vehicle communication, is committed to maintaining the improvement of the overall performance of the network, and designs a comprehensive weighted multi-hop broadcast routing algorithm based on real-time traffic estimation.

The design idea of the algorithm can be summarized as follows: firstly, the real-time traffic estimation algorithm with excellent performance is used to accurately predict the target road condition and vehicle behavior, including the vehicle density, vehicle speed, vehicle geographical location and destination of the road section; Secondly, the target vehicle uses the relevant information obtained from real-time traffic estimation to design a comprehensive weighting algorithm, which maps the network topology and node information into weight factors, and uses the comprehensive weighting algorithm to obtain the weighted value of each node in the target road section relative to the information source node; Thirdly, the ranking of the reliability of forwarding nodes is realized according to the size of the comprehensive weighted value, and the two nodes with the highest reliability (optimal and suboptimal nodes) are encapsulated into the broadcast packet as the destination forwarding nodes, so as to suppress the number of forwarding nodes; Finally, the forwarding node parses the broadcast packet. Suppose it can parse its own relevant information. In that case, it uses the weight mapping algorithm to map the corresponding weighted value into the forwarding waiting time. The optimal forwarding node has the shortest waiting time to effectively reduce the forwarding delay and ensure the real-time effectiveness of the broadcast information [16]. In addition, to maximize the network's reliability, the algorithm also introduces a timeout retransmission mechanism.

B. Flow Design Algorithm

In this algorithm, in order to effectively overcome the network dynamics caused by the high-speed mobility of vehicles, effectively estimate the network topology and estimate the real-time traffic; In order to reduce the number of forwarding nodes and avoid the broadcast storm, a comprehensive weighting algorithm is introduced to select trusted relay nodes; And to ensure the real-time performance of secure broadcast messages, an adaptive waiting slot mechanism based on comprehensive weighted value mapping is needed[17][18]; For ensuring the reliability of security alarm message propagation, the packet timeout retransmission mechanism is introduced[19][20]. The main implementation process of the algorithm is shown in the figure below. The algorithm mainly includes the related processing between nodes and candidate forwarding nodes. The detailed implementation process can be summarized as follows:

Step1- Broadcast: The vehicle node broadcasts relevant messages regularly to make the node become an information source node;

Step2- Determine impact factors: The information source node performs real-time traffic estimation on its on-board nodes' relevant information, outputs the nodes' pre-judgment information, and determines the relevant impact factors.

Step3-Comprehensive weighting: The information source node maps the relevant influence factors of each node into weight factors and uses the comprehensive weighting algorithm to calculate the comprehensive weighting value of each node, which is used as the basis for the selection of the next hop forwarding node.

Step4-Select the optimal forwarding node: The information source node selects the optimal and secondary forwarding nodes according to the weight value, encapsulates the relevant nodes' identification (and corresponding weight) into the secure broadcast packet, and the information source node sets a timeout retransmission timer.

Step5-Broadcast data analysis: After receiving the broadcast data packet, other nodes within the communication range parse it. Suppose they parse the identifier that matches themselves (they are determined as candidate nodes).

Step6-Weight mapping: The candidate node maps the corresponding weight to the forwarding waiting time. The larger the weight, the smaller the forwarding waiting time;

Step7-Waiting for timeout processing: If the same packet is received within the waiting time, the waiting time will be stopped, and the packet will be discarded directly. If the same packet is still not received after the waiting timeout, the candidate node will forward the packet.

Step8-Timeout Retransmission: If the information source node receives the same broadcast packet, the timeout retransmission timer will stop. Otherwise, if the retransmission timer times out, it will broadcast the packet again.

Step9-Update forwarding node set: The candidate node forwards the packet and resets the forwarding hops of the packet. At this time, the forwarding node becomes a new information source node and repeats the process. After the final packet exceeds the forwarding times, stop forwarding and discard it.

C. Relay Selection Strategy

The one-hop communication distance R of the information source node is evenly divided into N segments according to the distance from near to far. If the relative distance between the target node and the information source node is D_R, the given time slice of the vehicle in each section can be expressed as:

$$T_d = S_{ij} * \partial \tag{18}$$

Where, ∂ is the estimated one-hop relay delay, including the access delay and propagation delay of the channel, and S_{ij} is the number of time slices corresponding to the location of a given target node, i.e.:

$$S_{ij} = N[1.8 - \frac{MIN(D_R,R)}{R}] \tag{19}$$

$$N = {}^{R}/_{2r} \tag{20}$$

here **r** is the relay selection radius.

The definition of the number of time slices S_{ij} is mainly based on the fact that the selection mechanism of the default waiting time T_d is a delay redundancy process. When the waiting time T_d is too large, the packet transmission delay will increase, but it is easier to avoid redundant data forwarding at the same time; When the waiting time T_d is too small, it will lead to frequent packet forwarding and increase redundant data, but it can also reduce the packet transmission delay and ensure the timeliness of the information. Therefore, considering the actual scenario and optimization mechanism, the confirmation of N can be realized through the above formula 20. Select the default waiting time t to be at least greater than N/2 time slices to weigh the real-time performance and efficiency.

For the optimal forwarding node, because it corresponds to the maximum weighted value and the range of weighted value σ_i is between (0,1), we can design an adaptive forwarding waiting time determination mechanism:

$$T_r = T_d * (1 - \sigma_i) \tag{21}$$

This waiting time confirmation mechanism ensures that the optimal forwarding node forwards packets at the fastest speed. When the weight of the optimal forwarding node tends to the maximum value of 1, the T_r value tends to 0, indicating that this node fully meets the forwarding conditions. Therefore, the node hardly needs to wait and can directly forward packets. The waiting time of the suboptimal forwarding node will also be shorter than that of the slot persistence algorithm, that only depends on the confirmation of geographical location information, so it can effectively ensure the real-time performance of the forwarded packet. The dual guarantee of the optimal and suboptimal nodes can effectively improve the reliability of packet forwarding. Moreover, the optimal and suboptimal gradient waiting time settings can effectively avoid the collision of forwarded packets in competing channels.

D. *Experimental Simulation of Routing Effect*

In the simulation, the target road section with 80 vehicle nodes is set, and the contracting rate is 2 packet / s. Count the number of nodes receiving data packets at a specific time within the two-hop range, and calculate the corresponding coverage. It can be seen from the simulation results that the algorithm can achieve high area coverage in a very short time. With the increase in forwarding nodes, the network load index will increase and finally form a broadcast storm, resulting in serious network congestion. Therefore, it takes a long time to cover the road far enough effectively. The algorithm effectively suppresses the number of forwarding nodes, avoids the occurrence of a broadcast storm, and effectively improves the real-time performance of packet broadcasting.

The following Fig. 4 shows the comparison of experimental data:

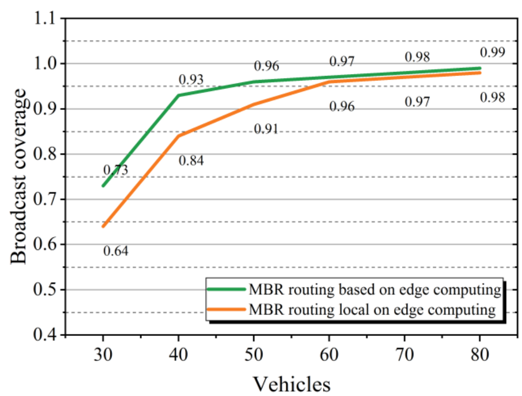

Fig. 4. Relationship between vehicle density and broadcast coverage.

As can be seen from the experimental data in Fig. 4, the real-time packet forwarding performance of the MBR routing algorithm based on edge computing is higher than that of the routing algorithm supported by local computing.

V. Conclusion

This paper constructs the measurement index of effective computing unloading mechanism, takes the delay and mobility as the key elements of the measurement index, discusses the reasonable allocation of weights, takes into account one-hop and two-hop vehicle nodes, and analyzes the impact of mobility in relative operation direction on link connectivity. In the aspect of task unloading decision-making, this paper analyzes how the task vehicle selects the task unloading object from the service vehicle set and how to allocate the task. This paper expounds on the weighted calculation design of time delay and calculation cost, how to localize or unload the adaptive decision task to one-hop / two-hop vehicles, and carries out the corresponding experimental verification. Finally, it explains how the mobile vehicle network constructs the backbone link composed of intersection edge nodes and inner edge nodes, as well as the calculation method and simulation experiment of the weight of each section.

The experimental simulation shows that the average task execution time under this model is 65.4% shorter than that of local computing, 18.4% shorter than that of cloud computing, and the routing coverage is about 6% higher than that of local computing when there are less than 60 nodes. These research and experimental results fully demonstrate that the mobile Internet of vehicles based on edge computing has good reliable transmission characteristics. However, there are still shortcomings in the study of dense vehicle nodes in complex scenarios. Further in-depth analysis of unloading and routing mechanisms in various types of road scenarios is needed, and the idle computing power of stationary vehicle nodes has not been included in the task calculation application.

In addition, with the vigorous development of various applications, some studies consider using the advantages of the global perspective of software-defined network (SDN) to provide rich traffic conditions and network information for vehicles to reduce the control cost of the workshop; There are also attempts to introduce UAVs into the mobile vehicle network, rely on the mobile and high-altitude characteristics of UAVs to improve link connectivity and provide relay services.

With the increase in the number of intelligent networked vehicles and the popularization of application services, it will also be necessary to consider the scenario of multi-level tasks, meet the development trend of differentiated services in terms of delay, energy, and economic cost, and provide differentiated resource allocation for different service levels.

Acknowledgment

This work is supported by Science &Technology Research Program of Chongqing Municipal Education Commission(Grant No. KJQN202104009) and Science and technology projects in Hechuan District (Grant No.hckj-2022-87), partly funded by Chongqing Technology and Business Institute project (Grant No. ZZ2020-06).

References

[1] F. Yang, J. Li, T. Lei, & S. Wang. Architecture and key technologies for Internet of Vehicles: a survey[J]. Journal of Communications & Information Networks, 2017, 2(2), pp.1-17.

[2] H. Peng, Q. Ye & X. S. Shen. SDN-based resource management for autonomous vehicular networks: A multi-access edge computing approach[J].vol(18), 2019, IEEE Wirel. Commun.

[3] J. Zhang, M. Ren, H. Labiod, and L. Khoukhi, Link duration prediction in VANETs via AdaBoost[C], IEEE Global Communications Conference, 2017, 86(2), pp. 1-6.

[4] M. Al-Rabayahn & R. Malaney. A new scalable hybrid routing protocol for VANETs[J]. IEEE Trans. Veh. Technol., 2012, 61(6), pp.2625 – 2635.

[5] Q. Ding, B. Sun & X. Zhang. A traffific-light-aware routing protocol based on street connectivity for urban vehicular ad hoc networks[J]. IEEE Commun. Lett., 2016, 20(8), pp.1635 – 1638.

[6] R. Kaur, T. P. Singh and V. Khajuria, Security Issues in Vehicular Ad-Hoc Network (VANET)[C], 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), 2018, pp.884-889.

[7] Y. R. B. Al-Mayouf, N. F. Abdullah, O. A. Mahdi, S. Khan, M. Ismail, M. Guizani, & S. H. Ahmed. Real-time intersection-based segment aware routing algorithm for urban vehicular networks[J]. IEEE Trans. Intell. Transp. Syst, 2018, 19(7), pp.2125 – 2141.

[8] M. H. Eiza, T. Owens & Q. Ni. Secure and Robust Multi-Constrained QoS Aware Routing Algorithm for VANETs[J]. IEEE Trans. Depend. Sec. Comput , 2016, 13(1) , pp.32 – 45.

[9] L. LIU, C. CHEN, J. FENG, T. T. XIAO & Q. Q. PEI. A Survey of Computation Offloading in Vehicular Edge Computing Networks, ACTA ELECTRONICA SINICA[J], Vol. 49 No. 5, May 2021.

[10] J. Zhang & K. B. Letaief. Mobile edge intelligence and com- puting for the Internet of vehicles [J]. Proceedings of the IEEE， 2020，108（2）, pp.246-261.

[11] H. ZHAO, H. X. PENG, J. ZHU & D. Z. LI. Optimal Trunk Node Selection Based on Group-Based Acknowledgement Mechanism in Vehicular Networks[J], Journal of Northeastern University, Vo l. 34, No. 1 Jan. 2013.

[12] Y. Ge, S. Wen, Y. H. Ang & Y. C. Liang . Optimal relay selection in IEEE 802. 16j multihop relay vehicular networks[J] . IEEE Transactions on Vehicular Technology，2010，59(5) , pp. 2198 -2206.

[13] F. Samie, V. Tsoutsouras, L. Bauer, S. Xydis, D. Soudris & J. Henkel. Computation offloading and resource allocation for low-power IoT edge devices [C]// 2016 IEEE 3rd World Forum on Internet of Things（WF - IoT）. Reston，VA，USA, IEEE，2016, pp.7-12.

[14] Y. Liu, H. Yu, S. Xie & Y. Zhang. Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks [J]. IEEE transactions on vehicular technology， 2019，68（11）, pp. 11158-11168.

[15] X. Xu, Y. Xue, X. Li, L. Qi, & S. Wan. A computation offloading method for edge computing with vehicle- to- everything [J]. IEEE access, 2019，7, pp.131068-131077.

[16] X. Yang, X. Yu, H. Huang and H. Zhu, Energy Efficiency Based Joint Computation Offloading and Resource Allocation in Multi-Access MEC Systems[J], IEEE Access, 2019, 7, pp.117054-117062.

[17] S. Wang, J. Xu, N. Zhang and Y. Liu, A Survey on Service Migration in Mobile Edge Computing[J], IEEE Access, 2018, 6, pp.23511-23528.

[18] M. Singh and S. Kim, Security analysis of intelligent vehicles: Challenges and scope[C], 2017 International SoC Design Conference (ISOCC), 2017, pp. 13-14.

[19] F. Zhang, T. Luo, & J. Li. A multi-hop broadcast routing based on real-time traffic estimates for vanet[D], 03, 2015.

[20] J. Fan, L. Bin, and F. Chao, Mobile relay deployment in multi-hop relay networks[J], Computer Communications, 2017, 112(1), pp.14-21.