Research Article

An MSCN-Based Virtual Computing Cell-Oriented BSM Dissemination Mechanism

Lei Ye,1,2 Yuping Wu,1 Qingwen Han,1 Xiaoyuan Zhang,1 Lingqiu Zeng,2,3,4 Yuqing Chen,1 and Xinhai Chen4,5

1School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China
2Key Laboratory of Advanced Manufacture Technology for Automobile Parts (Chongqing University of Technology), Ministry of Education, Chongqing 400054, China
3School of Computer Science, Chongqing University, Chongqing 400044, China
4Chongqing Engineering Research Center of Research and Testing for Automated Driving System and Intelligent Connected Vehicle, Chongqing 401122, China
5Chongqing Vehicle Test & Research Institute, Chongqing 401122, China

Correspondence should be addressed to Qingwen Han; hqw@cqu.edu.cn

Received 29 June 2020; Revised 3 August 2020; Accepted 24 March 2021; Published 17 April 2021

Academic Editor: Alessandro Bazzi

Copyright © 2021 Lei Ye et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the concept of multiaccess edge computing (MEC) being put forward, Roadside Unit (RSU) is considered as a valid application provider, which not only executes transmission resource allocation and data processing-related computing but also provides real-time applications to road vehicles. However, when fixed roadside nodes communicate with mobile vehicles, the high service migration rate could influence real-time feature of corresponding service. Moreover, vehicle density also affects service performance. Hence, in this paper, a two-processing layer architecture is constructed. A new concept, mobile secondary computing node (MSCN), which is used to compose mobile computing layer, is defined, and the number of MSCN changes dynamically with the vehicle density. Then, MSCN oriented virtual computing cell (VirCC), while corresponding to resource allocation approach and vehicle message dissemination mechanism, is designed. A network simulator (NS-3.28) is employed to investigate the performance of the proposed architecture. The simulation results show that the proposed architecture significantly improves both communication performance and computing efficiency.

1. Introduction

C-V2X technology will serve as the foundation for vehicles to communicate with each other and everything around them, providing 360° nonline-of-sight awareness and a higher level of predictability for enhanced road safety and autonomous driving. Both academic and industry researchers agree that vehicle's real-time data are an important and valuable source for automatic driving applications. Therefore, edge computing nodes, such as multiaccess edge computing nodes (MECNs) [1] and fog computing nodes (FCNs) are introduced to establish an open cloud environment in a close proximity to the radio access network (RAN) accessible by third parties in an effort to overcome the shortcomings of centralized cloud computing in terms of latency and throughput [2]. In this case, vehicle to everything (V2X) network not only serves as a bit pipe that is used to exchange messages but also executes data processing task. Moreover, the real-time and effective performance of data processing task deadly influences application's performance. MEC oriented data processing performance is decided by two key factors. The one is the performance of vehicle data dissemination, and the other is the rationality of computing resource allocation.

RSU, whose main function is to facilitate communication between vehicles and transportation infrastructure by transmitting data, is considered as one of the key computing resources in fog computing infrastructure and also serves as computing node in MEC infrastructure [3, 4]. Both researchers and industries believe that RSU oriented MEC could provide real-time service for autonomous vehicles, such as local driving trajectory plan and regional vehicle distribution information sharing. It is well known that RSUs
Mobile Assisted Computing Node MSCN

Mobile assisted computing nodes (MSCNs) are a set of specific vehicle nodes, such as buses and taxis, which serve as secondary computing node to execute local computing tasks. MSCN provides computing resource to regional road vehicle and upload computing results to RSU to support corresponding service.

Due to the instinct features of vehicle nodes, such as dynamic topology, dynamic computing demand, and dynamic cruising speed, two problems are posed for the selection of MSCN. Firstly, the number of MSCN relates to regional vehicle density, which changes dynamic in real-time. Secondly, computing efficiency of MSCN is decided by the relationship between MSCN and surrounding vehicles. Hence, validity of MSCN is associated with investigating the problem of cluster head selection. A composite weight parameter, which considers mobile similarity, position factor, and distance factor, is defined to select MSCN. Then, we consider both regional density feature and intervehicle relationship as the basis of VirCC generation mechanism design.

In general, the purpose of layering network structure is to make better use of network resources and transmission media. In the proposed architecture, both RSU and MSCN could provide computing service. The only problem is who acts the role of resource allocator. In general, RSU could be used to allocate frequency points and time slots to regional vehicle nodes. However, in high vehicle density condition, both computing resource and communication resource are inadequate. To relief the crisis of communication resource scarcity, a transmission range control approach should be used, and MSCN should serve as local resource allocator. Then, in this paper, a message dissemination mechanism, which includes both resource allocation method and corresponding BSM transmission cycle design, is proposed.

Specifically, the contributions of this paper are as follows:

1. Mobile secondary computing node (MSCN) is defined to collect and process road vehicle information;
2. A two-layer processing architecture based on MSCN is proposed, in which RSU serves as the main computing node and the selected mobile node serves as the secondary computing node;
3. An adaptive VirCC generation mechanism is proposed.
4. A VirCC-based vehicle message dissemination mechanism is proposed.

The rest of this article is organized as follows: Section 2 presents the related work, and the proposed two-layer processing architecture is in Section 3. In Section 4, the generation mechanism of VirCC is introduced. The information distribution mechanism inside the virtual area is introduced in Section 5. Section 6 is the analysis of simulation results. The conclusion is given in Section 7.

2. Related Work

In the past few years, researchers devote to improving reliability and real-time feature of V2X network and realize that MEC could provide computing capability and real-time service for road vehicles. Hence, MEC-related technologies have attracted many researchers’ attention. In this section, we analysis related works from three aspects, which are network architecture improvement, vehicle cluster management, and rationality of resource allocation.

2.1. Network Architecture. In traditional cloud computing architecture, all computing tasks are executed at cloud data center [7]. Communication delay of data uploading and downloading process deadly influences application performance. In order to meet the requirements of autonomous driving, researchers intend to take full advantage of roadside equipment and vehicle on-board equipment, which are considered as edge devices and could provide an approach to overcome the shortcomings of centralized cloud computing in terms of latency and throughput.

In terms of architecture, MECs use centralize mode, while fog computing employs distribute mode [8, 9]. MEC node, which is defined as an implementer of edge computing, could bring computational and storage capacities to reduce latency and improve context awareness. The MEC nodes are usually co-located with the radio network controller or a macrobase-station [10]. In a vehicle-oriented network, MEC platform can be flexibly deployed on various nodes, such as Roadside Units (RSU) and evolving Node B (eNodeB). In literature [11], an MEC server is deployed at RSU to improve real-time feature. However, both RSU and eNodeB are fixed nodes, which could not provide effectual computing support to mobile vehicle nodes for two reasons. Firstly, V2I communication performance is deadly
influenced by Doppler shift. Secondly, due to vehicle node’s high mobility feature, the handover connection between vehicle nodes and MEC platform will be established frequently; this way not only increases the cost of maintaining the handover connection but also reduces the efficiency of computing.

On the other hand, fog computing nodes (FCNs) utilize devices such as M2M gateways and wireless routers to provide computing capability. FCNs can be any node between the vehicle terminal and the core cloud architecture used to calculate and store data from vehicle terminal locally. In literature [12], the authors discuss the feasibility fog node and think that quality of service and application could be greatly improved by making full use of computing capability of vehicular nodes. However, industry doubts the feasibility of FCNs, which have no operation managers to ensure their credibility.

As one of the basic service tools, city bus could be considered as a special kind of vehicle nodes, which could provide credible service to social vehicles. Hence, in our previous work [2], we analysis the coverage feature of bus net and proposed a fog computing architecture. The experimental results show that the structure can effectively improve the communication efficiency.

Therefore, in this paper, we propose a two-layer processing architecture, in which a new concept, MSCN, is defined.

2.2. Vehicle Clustering. Clustering method is a traditional hot thesis in the field of vehicular network. In the past few years, researchers proposed a large amount of clustering algorithm, which could be divided into three categories: (1) ID-oriented, (2) location-based, and (3) weight-based.

ID-oriented clustering algorithm uses node ID as the basis of cluster head selection. In 2002, Lin [13] employed lowest-ID algorithm in mobile node clustering. Fan [14] improved the lowest-ID algorithm and introduced the concept of cluster head working time. However, the stability of clusters should decrease with the increase of cruising speed. Hence, ID-oriented clustering algorithm will always be used in conjunction with other methods, such as location-oriented method and intervehicle relationship-oriented method. Literature [15] proposed a synthesis algorithm, which considers node ID, geographical proximity, and speed difference as the basis for cluster header selection.

On the other hand, location-based clustering algorithms concern about intervehicle spatial relationship. In [16], a region-based clustering mechanism is proposed to cluster vehicles according to region location to reduce the contention period introduced by vehicle access channels in the MAC protocol. However, road vehicle clustering is influenced by both spatial relationship and vehicle density feature. Literature [17] employs a density-related factor, number of neighbors, as the cluster head selection criteria. Deng et al. [18] proposed an effective spatial clustering algorithm based on grid and density, which supports multi-density clustering.

Obviously, ID-oriented clustering algorithms and location-based clustering algorithms could be considered as the basis of weight clustering algorithms. Then, the algorithm optimization problem is transformed into a weight factor optimization problem.

The weighted clustering algorithm (WCA) assigns different weight coefficients to various parameters affecting the performance of the cluster head and finally selects the node as the clustering head according to the weight [19]. Compared with the method of selecting the cluster head based on a single attribute, this algorithm takes into account the influence of different factors on the cluster head, and the generated cluster is more stable. In general, the weight factor is constructed according to a series factor, such as vehicle cruising characteristics (including direction, speed, and acceleration) [20, 21], moving similarity [22], and local vehicle density.

In this paper, a comprehensive weight factor, which considers moving similarity, position factor, and distance factor, is defined to select MSCN. A VirCC join index, which denotes the position relationship between MSCN and road vehicle, is used to generate VirCC.

2.3. Communication Resource Allocation. Until now, V2X system performance is restricted by communication resource.

In general, DSRC nodes use contention-based method to obtain communication resource. However, VANET also supports RSU centric allocation mode [23]. On the other hand, LTE-V employs resource pool mechanism, which supports both centric and distribute allocation mode, to alleviate channel congestion. However, due to vehicle’s mobility feature, RSU could not competent to the task of resource allocation [2]. In this paper, we use specific mobile nodes as the resource allocator. The only question is how to optimize resource allocation method and set transmission priority. The basic approach is to determine transmission priority according to request priority [23]. In the proposed architecture, MSCN, which provides computing resource to regional vehicle and supports RSU’s service providing, should be granted with high priority. Hence, we design a two-step communication resource allocation method. Firstly, RSU allocates transmission resource to MSCN. Then, the remaining resource is allocated to other road vehicles. In [24], mobile similarity is selected as the basis of priority setting. Kim et al. [25] proposed a resource allocation scheme that considers various attributes of the vehicle when allocating resources, such as speed, density, and direction. Aiming to provide computing ability to vehicle nodes, in the proposed architecture, road vehicles are divided into several VirCCs, who’s centric is MSCN. To improve the processing effectiveness of MSCN, we employ a VirCC oriented mechanism, which allocates adjacent time slots to vehicle nodes belonging to the same VirCC.

3. System Model

In this section, the proposed MSCN-oriented infrastructure is described in detail. As mentioned earlier, the proposed infrastructure includes two-processing layers, which are
3.1. MSCN-Oriented Infrastructure. The proposed MSCN-oriented infrastructure is shown in Figure 1.

As shown in Figure 1, the proposed infrastructure includes three layers, which are general vehicle node layer and the two-processing layer we defined, namely, MEC layer and MSCN layer.

1. Layer 1: MEC Layer
   MEC layer, which includes RSUs, corresponding edge computing server, and local database, is used to process regional vehicle information and provide processing results to support vehicle application implementation. RSUs could serve as communication and computing resource allocators.

2. Layer 2: MSCN Layer
   MSCN layer includes public operational vehicles, such as city bus, taxi, and other service vehicles. MSCNs are specific mobile nodes, which could provide computing resource to social vehicles and processing results to MEC layer. MSCNs could serve as communication resource allocators on demand.

3. Layer 3: General Vehicle Node Layer
   Layer 3 contains all general social vehicles, which communicate with each other via direct connection mode, such as PC5 and DSRC. In this paper, we assume that all road vehicles are equipped with a PC5 interface. Specifically, road vehicles receive application-oriented information through the PC5 interface and update BSMs to distributed computing nodes.

3.2. Service Flow. Corresponding service flow is described as follows:

   Step 1: road vehicles periodically upload BSM messages to local MSCN;
   Step 2: distributed MSCNs collect and process BSM messages and then upload processing results to upper-level RSU;
   Step 3: MEC layer collects MSCN message and generates regional traffic information, which could be used to support autonomous vehicle decision making progress;
   Step 4: according to road vehicle service application, MEC layer provides service information to regional vehicle via PC5 interface.

   Note that in this architecture, RSUs service as service provider, while MSCNs can be either the source node or the relay node.

3.3. Virtual Computing Cell (VirCC). Here, we assume that vehicle nodes in the geographic area of one RSU coverage belong to one cluster. According to road test results, provided by Chongqing Vehicle Test and Research Institute Co., Ltd., the reliable communication radius of LTE-V is about 150 meters. To ensure all road vehicles find an access RSU, here we assume the radius of basic vehicle cluster as 300 m.

Definition 2. Virtual Computing Cell
Virtual computing cell (VirCC) is a group of neighboring vehicles, which includes one MSCN and multiple vehicle nodes. A vehicle cluster could be divided into multiple VirCC, and the number of VirCC is decided by regional vehicle density.

Here, two cases, which are low-density case and high-density case, are discussed to explain the generation mechanism of VirCC.

3.3.1. Low-Density Case. At low density, we stipulate that VirCC is evenly arranged in the cluster and select the MSCN corresponding to a certain VirCC as the cluster head. In order to ensure that the communication range of the cluster head is the entire cluster, we set the number of VirCC to 3. Here, the total length of vehicle cluster is assumed as \( L \), while the diameter of three virtual cells is about \( L/3 \). Therefore, the MSCN of the middle VirCC is closest to the center of the cluster, and selecting this MSCN as the cluster head can ensure that the cluster head has the best communication coverage. VirCC of low-density case is shown in Figure 2. The red MSCN is considered as the cluster header, who serves as transmission controller for time-slot allocation. The other two MSCNs (blue) in Figure 2 implement computing task only.

3.3.2. High-Density Case. VirCC of high-density case is shown in Figure 3. The whole cluster is divided into 4 VirCCs. Diameter of VirCC is decided by road vehicle distribution factor.

   In high-density case, both computing resource and communication resource are insufficient.

   Obviously, regional computing ability increases with the increase of MSCN number, which should be decided by computing requirements. Hence, number of VirCC relates to regional vehicle density.

   On the other hand, transmission range control could be used to relief communication resource scarcity by improving resource reuse rate. Here, the transmission radius is calculated as follows:

   \[
   R = \frac{1}{\rho} \ln\left( 1 - \rho^{\frac{n}{n}} \right),
   \]

   where \( R \) is the transmission radius, \( \rho \) is the local vehicle density, \( P \) is the road connectivity probability, and \( n \) is the number of vehicle nodes. The relationship between transmission radius and node density under different connected probabilities is shown in Figure 4.

   When the connected probabilities are the same, the higher the node density, the shorter the transmission radius will be.
**Definition 3.** Density Threshold $\rho_0$. Here, we define a density threshold $\rho_0$ to distinguish between high- and low-density scenes of vehicles. According to equation (1), when the density is 0.067 vehicle/m, the communication distance is about 150 m. Therefore, we set the density threshold to 0.067 vehicle/m, when regional vehicle density higher than $\rho_0$, number of VirCC should increase according to vehicle number.

4. **VirCC Generation**

Obviously, performance of the proposed architecture is mainly determined by the feature of VirCC. In this section, we focus on VirCC generation process. Firstly, MSCN selection criteria are given. Then, the VirCC joining rule is proposed to generate VirCC dynamically.

4.1. **Number of VirCC.** Number of VirCCs is determined by the regional vehicle density and could be calculated as

$$N_{VirCC} = \frac{L}{R}$$

(2)

where $L$ is the total length of vehicle cluster; $R$ is the transmission radius. Note here, for low-density case, one vehicle cluster includes three VirCCs.
4.2 MSCN Selection Criteria. As mentioned earlier, MSCNs are specific mobile nodes, which could provide computing resource to social vehicles and processing results to MEC layer. The stability of VirCC largely depends on the reasonable selection of MSCN. To make VirCC more stable and sustainable, we define a weight factor, which considers moving similarity feature, relative position, and intervehicle distance, as MSCN selection basis, as follows:

\[ W = \alpha_1 \times (1 - S_{\text{similarity}}) + \alpha_2 \times f_{\text{position}} + \alpha_3 \times \text{Dis}_i, \]  

(3)

where \( S_{\text{similarity}} \) is the mobile similarity factor \([2]\), which is defined as

\[ S_{\text{similarity}} = \frac{\sum_{m=1}^{N} S_{\text{speed}} \times S_{\text{acceleration}} \times S_{\text{direction}}}{N}, \]  

(4)

where \( S_{\text{speed}}, S_{\text{acceleration}}, \) and \( S_{\text{direction}} \) are the similarity of speed, acceleration, and heading between candidate MSCN and neighbor node; \( N \) is the number of neighbors of candidate MSCN.

\( f_{\text{position}} \) is the position factor:

\[ f_{\text{position}} = \frac{\sqrt{(x_{\text{in}} - x_{\text{cn}})^2 + (y_{\text{in}} - y_{\text{cn}})^2}}{R}, \]  

(5)

where \((x_{\text{in}}, y_{\text{in}})\) is the coordinate of the candidate MSCN, and \((x_{\text{cn}}, y_{\text{cn}})\) are coordinates of the cluster head (CH), and \( R \) is the cluster radius.

\( \text{Dis}_i \) is the distance factor:

\[ \text{Dis}_i = \frac{\sum D_{ij}}{N}, \]  

(6)

where \( N \) is the number of neighbors of candidate MSCN, and \( D_{ij} \) is the normalized distance between node \( i \) and node \( j \):

\[ D_{ij} = \frac{\sqrt{(y_i - y_j)^2 + (x_i - x_j)^2}}{TR}, \]  

(7)

where \((x_i, y_i)\) is the coordinate of \( i \), and \((x_j, y_j)\) are coordinates of \( j \), and \( TR \) is the radius of coverage area.

\( \alpha_1, \alpha_2, \) and \( \alpha_3 \) are the weighting coefficients of the three attributes. Note here, \( W \in [0,1] \). The less the \( W \) value of MSCN candidate, the more correlative with surrounding vehicles, and correspondingly, the more stable its VirCC is.

Then, BSM message is defined as

\[ \text{BSM}_{\text{vehicle}} = [\text{ID}, \text{Vel}, \text{Acce}, \text{Dir}, \text{Pos}, \text{Time}]. \]  

(8)

The MSCN selection process is given in Algorithm 1, and corresponding variables and functions are explained in Table 1.

4.3 VirCC Joining Rule. VirCC joining rule is established based on VirCC index, which is calculated as

\[ \text{VirCCindex} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (\text{vec}_i - \text{vec}_j)^2 + (\text{acce}_i - \text{acce}_j)^2 + (\text{dir}_i - \text{dir}_j)^2}. \]  

(9)

![Figure 4: The relationship between node density and propagation radius.](image)
Here, \((x, y)\), vec, acce, and dir represent the coordinates, speed, acceleration, and cruising direction of general vehicle nodes, and \((x_j, y_j)\), vec \(_j\), acce \(_j\), and dir \(_j\) represent the coordinates, speed, acceleration, and cruising direction of MSCN, respectively.

After selecting the MSCN, ordinary nodes join the most suitable VirCC. The VirCC joining progress is shown in Algorithm 2, while related variables and functions are presented in Table 2.

### 4.4. VirCC Update

Due to the mobility of MSCN or VirCC members, VirCC will also change. The VirCC update procedure is needed to maintain the stability of the VirCC structure according to the changes in the topology. The node perceives the dynamic change of the VirCC structure through periodic messages.

### 5. Message Dissemination Mechanism

In the proposed architecture, message dissemination progress varies according to local vehicle density. As mentioned earlier, in this paper, we consider two cases: low-density case and high-density case.

#### 5.1. Low-Density Case

For low-density case, cluster header serves as controller, which allocates transmission time slot to all cluster members. The communication resource allocation progress includes three steps, described as follows:

1. Step 1: RSU allocates time slot to MSCNs, which are given with high transmission priority to communicate with RSU. Note here, RSU should decide MSCN’s occupied time-slot number according to the upload throughput.
(ii) Step 2: RSU transfers allocation right and available resource table to cluster header (CH) via announcement.

(iii) Step 3: CH allocates timeslot to other vehicle nodes (excluding MSCN) of cluster. Here, all cluster members share a same SCH channel via TDMA mechanism for BSM dissemination. Corresponding time-slot allocation procedure includes two steps.

(i) Step 1: cluster header divides VeMAC time into three time periods, each of which includes several time slots.

(ii) Step 2: cluster header allocates time slots to VirCC. Vehicle nodes belonging to the same VirCC should be allocated with adjacent time slots. Note here, on the one hand, all MSCNs shall be allocated with two time-slots, the one is used to send local vehicle information, while the other is used to upload computing output to RSU. On the other hand, compared with ordinary road vehicles, MSCN should have higher channel occupancy priority. The dissemination cycle is shown in Figure 5.

5.2. High-Density Case. When the number of cluster members larger than the allowable time-slot number, a control right hand out procedure should occur. The procedure includes three steps as follows:

(i) Step 1: control right hand out application: cluster header sends control right hand out application message to RSU via unicast mode.

(ii) Step 2: transmission parameter setting: RSU sends message with corresponding transmission parameters, such as transmission power and SCH channel, to all vehicle nodes via the CCH channel.

(iii) Step 3: control right takeover confirmation: MSCNs feedback control right takeover confirmation message to RSU and allocate time slots to vehicle nodes belonging to the corresponding virtual cell. The dissemination cycle is shown in Figure 6.

Detail of the proposed distribution right scheduling, which includes both low-density case and high-density case is given by Algorithm 3, and corresponding variables and functions are explained in Table 3.

6. Simulation and Experiment Results

6.1. Simulation Setup. In this paper, we conduct experiments on the NS-3.28 simulation platform to study the effectiveness of the message distribution mechanism. The road model parameters are shown in Table 4:

As shown in Table 4, we select a 600-meter straight road as simulation background. The range of vehicles number is set as [10, 70]. Corresponding vehicle density is 0.017 vehicles/m to 0.117 vehicles/m. Vehicle’s velocity is set as a random number between 20 m/s and 50 m/s. Corresponding communication parameters are set as shown in Table 5.

The physical layer function is realized based on YANSPHY model. During CCH period, vehicle nodes broadcast emergency and control information via CSMA mode, while
SCH period is used to broadcast vehicle node information via VeMAC mode. Moreover, the processing output of MEC at RSU end side should be feedback to road vehicles via the CCH channel. The allowable time-slot number is set to 30, while BSM packet size is set as 200 bytes.

6.2. Simulation Result

6.2.1. VirCC Generation and Maintenance. Here, we analyze VirCC generation and maintenance process to verify the effectiveness of the proposed VirCC working progress.

(1) VirCC Generation Time. Here, we use classic minimum ID algorithm and the highest node degree clustering algorithm as contrast. Experiment result is given in Figure 7.

As shown in Figure 7, VirCC generation time of the proposed method is shorter than the other two algorithms and performs a better real-time feature.

(2) MSCN Number. As mentioned earlier, MSCN number is decided by regional vehicle density. Moreover, during MSCN selection process, MSCN number is set as 30, while BSM packet size is set as 200 bytes.

(Figure 5: Message dissemination cycle of low-density case.)

(Figure 6: Message dissemination cycle of high-density case.)

6.2.2. Message Distribution Mechanism Simulation Results.

Three performance parameters, which are average backhaul delay, packet delivery rate, and effective feedback ratio, are considered.

(1) Average Backhaul Delay. As mentioned earlier, the proposed two-layer MSCN-oriented architecture is used to provide high real-time service to road vehicles. Hence, corresponding real-time feature should consider both transmission delay and computing processing delay. Then, average backhaul delay of vehicular service is defined as the time interval between the time point of vehicle’s BSM sent out and the time point that the vehicle obtained corresponding service information, which is generated according to the previous BSM. Simulation results are shown in Figure 10. The average backhaul delay of the two-layer MSCN-oriented static processing architecture proposed in our previous work [22], pure RSU processing architecture, pure CSMA processing architecture, lowest-ID algorithm, and highest-degree algorithm are used as a comparison.

As shown in Figure 10, comparing with pure RSU and pure CSMA processing architecture, and lowest-ID and highest-degree algorithm, the average backhaul delay of MSCN-oriented architectures, including both static and dynamic architectures, is relatively stable with the increase of road vehicle density. Moreover, average backhaul delay of the proposed dynamic architecture is lower than that of the static architecture.

(2) Packet Delivery Rate (PDR). Packet delivery rate (PDR) is the most important parameter to evaluate the performance of message dissemination mechanism. Here, two kinds of no-clustering methods, which are pure CSMA mode and RSU control mode, and three kinds of clustering methods, which are lowest-ID algorithm, highest-degree algorithm, and two-layer MSCN-oriented static processing method, are used as a contrast to verify the effectiveness of the proposed method. Simulation results are shown in Figure 11.

As shown in Figure 11, PDR performance of no-clustering methods is deadly influenced by local vehicle density, while that of clustering methods is less affected by local vehicle density and
**Input:** vehicle set $V$, density threshold $\rho_0$, MSCN set $mscn_set$.

1. Density $\leftarrow$ calculate_density ( )
2. **If** density $< \rho_0$, **then**
3. Allocation_timeslots ($mscn_set$)
4. CH_id $\leftarrow$ getCH ($mscn_set$)
5. **For** each $v \in V$, do
   6. **If** CH_id $==$ getid ($v$), **then**
      7. scheduler_flag (CH_id, true)
      8. controlright_confirm ($v$)
   9. **Else**
      10. scheduler_flag (getid ($v$), false)
      11. scheduler (CH_id)
   12. **End if**
   13. **End for**
14. **Else**
15. Allocation_timeslots ($mscn_set$)
16. controlright_handout ( )
17. parameter broadcast ( )
18. **For** each $v \in V$, do
19.   poweradjust ($v$)
20.   mscnid $\leftarrow$ getmscnid ($mscn_set$)
21.   **If** mscnid $==$ getid ($v$), **then**
22.      scheduler_flag (mscnid, true)
23.      controlright_confirm ($v$)
24.   **Else**
25.      scheduler_flag (getid ($v$), false)
26.      scheduler (mscnid)
27.   **End if**
28. **End for**
29. **End if**

**Algorithm 3:** The process of distributing scheduling rights.

| Variable/function | Explanation |
|-------------------|-------------|
| Density | The vehicle density |
| calculate_density ( ) | Calculate road vehicle density |
| Allocation_timeslots ($X$) | RSU preferentially allocates time slots to nodes in MSCN set $X$ |
| getCH ($X$) | Find the cluster head in $X$ |
| CH_id | The node ID of CH |
| Controlright_handout ( ) | CH unicasts control right hand out message to RSU |
| poweradjust ($X$) | Adjust power size of vehicle $X$ |
| getmscnid ($X$) | Obtain the MSCN ID of the area where the $X$ vehicle is located |
| Mscnid | The node ID of MSCN |
| getid ($X$) | Get node ID of node $X$ |
| scheduler_flag ($X$, $Y$) | Set whether node $X$ is used as a resource scheduler by the Boolean value $Y$ |
| controlright_confirm ($v$) | $X$ sends control right takeover confirmation message to RSU |
| scheduler ($X$) | Set $X$ as the resource scheduler’s node ID |

**Table 3:** Explanation for Algorithm 3 variables and functions.

| Parameter | Value |
|-----------|-------|
| Road length | 600 m |
| Lane number | 3 |
| RSU number | 2 |
| Distance between RSUs | 300 m |
| Cluster number | 2 |
| Cluster diameter | 300 m |
| Vehicle’s velocity | 20 m/s–50 m/s |
| Vehicle’s number | 10–70 |

**Table 4:** Road simulation parameter.

| Parameter | Value |
|-----------|-------|
| Frequency band | 5.9 GHz |
| Channel bandwidth | 10 MHz |
| Data rate | 6 Mbit/s |
| Transmit power | 16 mW |
| Packet size | 200 bytes |
| Packet delivery frequency | 10–100 Hz |
| Allowable time-slot number | 30 |

**Table 5:** Communication simulation parameter.
changes smoothly. Moreover, the proposed message dissemination mechanism performs a stable packet delivery rate, which is obviously better than the other methods. In addition, the packet delivery rate of the dynamic processing framework is higher than that of the static architecture.

(3) Effective Feedback Ratio. Similar to average backhaul delay, effective feedback ratio also is a parameter that denotes service performance. There might be a situation that the responsible vehicle could not receive valid service
information. For example, a vehicle node sent BSM to RSU, and the service information feeding back from RSU could not receive by the vehicle, which already moves out of coverage area of the previous RSU. Here, we define the effective feedback ratio as the ratio of the received effective service message number to upload BSM number. Simulation results are shown in Figure 12. The effective feedback ratio of pure RSU and two-layer MSCN-oriented static processing architecture is used as a contrast.

As shown in Figure 12, the higher the vehicle velocity is, the lower the effective feedback ratio should be. Moreover, overall, the effective feedback ratio of MSCN-oriented architecture is higher than that of the static processing architecture and pure RSU processing architecture.

7. Conclusion and Future Work

This work investigates the unique feature of RSU-based service and proposed a service performance enhancement approach, in which a two-layer oriented MSCN information collecting and processing dynamic infrastructure is defined. Moreover, the concept of VirCC is proposed. A MSCN selection criteria and VirCC joining rule are presented, and a MSCN-oriented computing and transmission resource allocation method is given, and corresponding message dissemination mechanism is designed. Then, we analyse VirCC generation and maintenance process; the simulation results prove that the working process of VirCC is stable and effective. Finally, we have built the simulation model and given a comprehensive performance evaluation, which demonstrate that the proposed architecture is able to provide reliable RSU-oriented services. Three performance parameters, which are average backhaul delay, packet delivery rate, and effective feedback ratio, are considered to verify the corresponding service performance.

In our future work, the MSCN transfer scenario should be considered. Meanwhile, other road scenarios, such as bends, intersections, and slopes, should be considered in the experimental setting.

Data Availability

The simulation data supporting the system performance analysis can be obtained from the website: https://github.com/qxcs/MSCN-oriented-BSM-information-dissemination-mechanism.git

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the Key Laboratory of Advanced Manufacture Technology for Automobile Parts (Chongqing University of Technology), Ministry of Education, under grant 2017KLMTO4, and the Key Research and Development Projects of Chongqing Special Industries for Technological Innovation and Application Demonstration, under Grant cstc2018jjsx-cyjdX0064.

References

[1] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, “Predictive offloading in cloud-driven vehicles: using mobile-edge computing for a promising network paradigm,” IEEE Vehicular Technology Magazine, vol. 99, p. 1, 2017.
[2] L. Zeng, J. Zhang, Q. Han et al., “A bus-oriented mobile FCNs infrastructure and intra-cluster BSM transmission mechanism,” IEEE Access, vol. 7, pp. 24308–24320, 2019.
[3] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, “Vehicular fog computing: a viewpoint of vehicles as the infrastructures,” IEEE Transactions on Vehicular Technology, vol. 65, no. 6, pp. 3860–3873, 2016.
[4] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: the communication perspective,” IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2322–2358, 2017.
[5] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, “Power-delay tradeoff in multi-user mobile-edge computing systems,” in Proceedings of the 2016 IEEE Global Communications Conference, pp. 1–6, Washington, DC, USA, December 2016.
[6] Q. Han, Q. He, L. Zeng, L. Ye, and F. Li, “A bus oriented coordination method for intra-cluster BSM transmission,” in Proceedings of the 2018 IEEE International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, November 2018.
[7] D. H. Jacobson, “A view on cloud computing,” International Journal of Computers & Technology, vol. 4, no. 2, pp. 50–58, 2010.
[8] H. Li, G. Shou, Y. Hu, and Z. Guo, “Mobile edge computing: progress and challenges,” in Proceedings of the 2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), Oxford, UK, April 2016.
[9] A. Aljumah and T. A. Ahanger, “Fog computing and security issues: a review,” in Proceedings of the Annual Conference on Computers, pp. 237–239, Chandigarh, India, March 2018.
[10] K. Dolui and S. K. Datta, “Comparison of edge computing implementations: fog computing, cloudlet and mobile edge
computing,” in *Proceedings of the 2017 Global Internet of Things Summit (GIoTS)*, pp. 1–6, Geneva, Switzerland, August 2017.

[11] H. Bao, L. Qin, C. Huang, and X. Jia, “Minimal road-side unit placement for delay-bounded applications in bus Ad-hoc networks,” in *Proceedings of the 2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, San Diego, CA, USA, 2017.

[12] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, “Vehicular fog computing: a viewpoint of vehicles as the infrastructures,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, p. 1, 2016.

[13] C. Lin and M. Gerla, “Adaptive clustering for mobile wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 2002.

[14] P. Fan, “Improving broadcasting performance by clustering with stability for inter-vehicle communication,” in *Proceedings of the 65th IEEE Vehicular Technology Conference, VTC Spring 2007*, Dublin, Ireland, April 2007.

[15] Z. Y. Rawashdeh and S. M. Mahmud, “A novel algorithm to form stable clusters in vehicular ad hoc networks on highways,” *Eurasip Journal on Wireless Communications & Networking*, no. 1, pp. 15–20, 2012.

[16] Y.-C. Lai, P. Lin, W. Liao, and C.-M. Chen, “A region-based clustering mechanism for channel access in vehicular ad hoc networks,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 83–93, 2011.

[17] M. Gerla and J. T.-C. Tsai, “Multicluster, mobile, multimedia radio network,” *Wireless Networks*, vol. 1, no. 3, pp. 255–265.

[18] C. Deng, J. Song, R. Sun, S. Cai, and Y. Shi, “GRIDEN: an effective grid-based and density-based spatial clustering algorithm to support parallel computing,” *Pattern Recognition Letters*, vol. 109, pp. 81–88, 2018.

[19] M. Chatterjee, S. K. Das, and D. Turgut, “WCA: a weighted clustering algorithm for mobile ad hoc networks,” *Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2002.

[20] O. Sarah, A. Rachida, J. J. P. C. Rodrigues, and T. Saad, “Secure and stable vehicular ad hoc network clustering algorithm based on hybrid mobility similarities and trust management scheme,” *Vehicular Communications*, vol. 13, pp. 128–138, 2018.

[21] L. Dan, K. Jian, A. Squciarini, Y. Wu, and O. Tonguz, “MoZo: a moving zone based routing protocol using pure V2V communication in VANETs,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1357–1370, 2016.

[22] Q. Han, X. Zhang, J. Zhang, L. Zeng, and X. Peng, “Research on resource scheduling and allocation mechanism of computation and transmission under MEC framework,” in *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference-ITSC*, Auckland, New Zealand, October 2019.

[23] R. S. Tomar and S. Verma, “RSU centric channel allocation in vehicular ad-hoc networks,” in *Proceedings of the 2010 Sixth International Conference on Wireless Communication and Sensor Networks*, pp. 1–6, Allahabad, India, 2010.

[24] C. Wu, B.-J. Hu, Z.-H. Wei, and L. Lin, “An adaptive TDMA scheduling strategy based on beacon messages for vehicular ad-hoc network,” in *Proceedings of the 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, pp. 255–261, Guangzhou, China, 2017.

[25] J. Kim, J. Lee, S. Moon, and I. Hwang, “A position-based resource allocation scheme for V2V communication,” *Wireless Personal Communications: An International Journal*, vol. 98, pp. 1569–1586, 2018.