This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3390991

ECMAC: Edge-Assisted Cluster-Based MAC Protocol in Software-Defined Vehicular Networks

Yiwen Shen, *Member, IEEE*, Jaehoon (Paul) Jeong, *Member, IEEE*, Junghyun Jun, *Member, IEEE*,
Tae (Tom) Oh, *Senior Member, IEEE*, Youngmi Baek, *Member, IEEE*

***Abstract*—Vehicular networks have emerged as a promising means to mitigate safety hazards in modern transportation systems. On highways, emergency situations associated with vehicles necessitate a reliable media access control (MAC) protocol that can provide timely warnings of possible vehicle collisions. In this paper, we present an edge-assisted cluster-based MAC protocol (ECMAC) for packet dissemination in software-defined vehicular networks. To reduce the control messaging overhead for clustering, ECMAC separates the cluster control plane (i.e., managing cluster formation) from the data plane (i.e., actual data transmission and forwarding) by using a software-defined network controller in a cellular network edge server. For transmitting packets, we design a time-division multiple access (TDMA) schedule algorithm to guarantee a high reliability and a low latency. The TDMA schedule in ECMAC is determined by a joint optimization process in the cellular edge, which is formulated as a binary integer linear programming problem and solved by a heuristic approach based on the divide-and-conquer paradigm. This joint optimization process minimizes the signal interference by jointly considering channel assignment and time slot allocation, thereby ensuring reliable communication. Through extensive simulations, our performance results show that ECMAC improves the successful delivery ratio of emergency packets by at least 25 %, compared with state-of-the-art approaches.**

***Index Terms*—MAC protocol, vehicular networks, software-defined networking, edge computing, safety.**

I. INTRODUCTION

Vehicular networks have emerged as a promising technology to enhance road safety in modern transportation systems, particularly in the context of future connected and automated vehicles (CAVs) [1], [2]. A vehicle in motion on a roadway experiencing an emergency situation (e.g., emergency braking and a collision in foggy weather) has the potential to initiate a chain of collisions, potentially leading to injuries or fatalities. Timely transmission of emergency messages to neighboring and multihop vehicles constitutes a mission-critical service capable of mitigating such tragic events. In order to enable such a mission-critical service to future CAVs, IEEE has published the next generation of dedicated short-range communications (DSRC)-based Vehicle-to-Everything (V2X) communication standard, designated as IEEE 802.11bd [3], building upon the previous 802.11p standard [1], [4], [5]. Concurrently, the 3rd generation partnership project (3GPP) has proposed the 4G LTE/5G V2X standards [6], [7], which facilitate direct communication among user equipments (UEs) and vehicular user equipments (VUEs) without the intervention of a base station (BS) such as eNodeB in 4G LTE and gNodeB in 5G. Notwithstanding these efforts and initiatives aimed at establishing a more intelligent CAV environment, a number of challenges impede the widespread adoption of vehicular networks for the automated vehicles in terms of reliability and scalability [8].

In the realm of vehicular networks, a significant challenge lies in the broadcast storm problem [9], which has the potential to impair the effective dissemination of information. Among various approaches, vehicle clustering has emerged as a promising solution to mitigate this issue [9]–[14]. A clustering approach typically entails grouping a number of vehicles and scheduling packet transmissions in a manner that minimizes packet collision probability. However, such approaches inevitably introduce substantial overheads associated with cluster management. Research efforts have been directed towards investigating mobility-aware cluster-based media access control (MAC) protocols [10] in vehicular ad hoc networks (VANETs). Additionally, the integration of cellular networks in VANETs has led to the proposal of LTE-assisted cluster-based protocols for packet relaying [11], [15]. More recently, several cluster-based schemes have been proposed, including a unified clustering framework for VANETs [12], traffic-differentiated clustering routing in hybrid vehicular networks [13], and reinforcement learning-based cooperative vehicle cluster scheduling [14]. Despite these advancements, the management of vehicle clusters remains a challenge, potentially affecting channel utilization and compromising the efficiency of communication systems.

The 5G and beyond 5G cellular networks have undergone a significant advancement in various dimensions, including support for new wideband carriers, varying sub-carrier spacings, a customizable slot-based framework, and cutting-edge channel coding methods [16]. This advancement has paved the way for the segregation of control and data planes in cluster-based vehicular networks, as the cluster control messages consume substantial channel resources, resulting in additional overhead. Additionally, software-defined networking (SDN) technology, when coupled with edge computing, has facilitated

---

Y. Shen and J. Jeong (the corresponding author) are with the Department of Computer Science & Engineering, Sungkyunkwan University, Suwon 16419, Republic of Korea (e-mail: {chrisshen, pauljeong}@skku.edu).

J. Jun was with the Department of Computer Science & Engineering, Indian Institute of Technology at Ropar, Rupnagar 140001, India. (e-mail: peterjun@iitrpr.ac.in).

T. Oh is with the Department of Information Sciences & Technologies, Rochester Institute of Technology, NY 14623-5603 USA. (e-mail: tom.oh@rit.edu).

Y. Baek is with the Department of Computer Science & Engineering, Changshin University, Changshin 51352, Republic of Korea (e-mail: ymbaek@cs.ac.kr).

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3390991

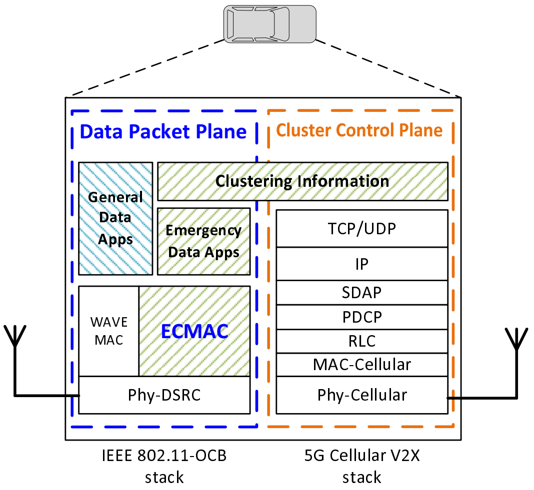


Fig. 1. Dual protocol stacks in a vehicle.

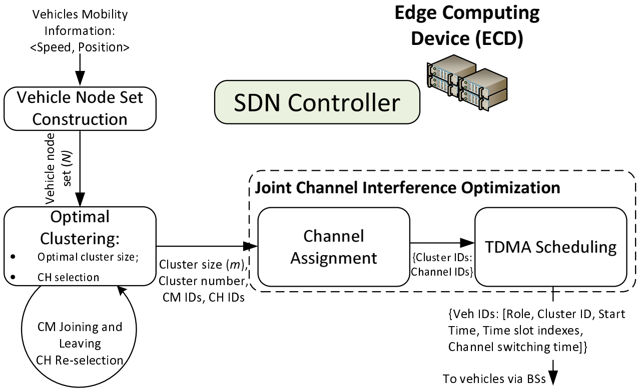


Fig. 2. The workflow of the joint interference optimization in SDN controller located in the ECD.

more adaptable and efficient network control and management [17]. An SDN controller operating within an edge computing device (ECD, also known as an edge server) [18] in vehicular networks can gather network parameters and expeditiously update the network configuration, particularly in light of the dynamic nature of vehicular network topologies. Hence, in this paper, we propose an edge-assisted cluster-based MAC (called ECMAC) protocol in software-defined vehicular networks (SDVN). A vehicle implementing ECMAC protocol has dual protocol stacks, as illustrated in Fig. 1. Vehicles routinely upload their mobility attributes (e.g., position, speed, and direction), which is called vehicle mobility information, to an SDN controller in an ECD through cellular networks (e.g., 4G LTE and 5G). Based on the collected vehicle mobility information, the SDN controller calculates *an optimal cluster size* for virtually clustering vehicles. Subsequently, it executes *a joint interference optimization* for the channel access schedule of vehicles, which results in minimized channel and signal interference, as depicted in Fig. 2. The joint interference optimization is formulated as a binary integer linear programming problem and is addressed using a heuristic method based on the divide-and-conquer approach, given its NP-hardness. The SDN controller then distributes the optimization results to vehicles over the cellular links. Upon receiving the optimized schedule, each vehicle can determine its role in the current network and transmit packets over a DSRC channel during its allocated time slot. Note that ECMAC can facilitate the dissemination of both emergency and non-emergency packets.

The main contributions of this work can be summarized as follows:

- *An edge-assisted network architecture based on SDN for multihop packet dissemination*. In the proposed architecture, a centralized SDN controller in an ECD virtually manages vehicle clusters, and a controlled VANET is responsible for a reliable packet dissemination. This architecture can enable new safety and non-safety applications in SDVN. (see Section III).
- *A delay-bounded clustering structure for reliable packet dissemination*. We analyze a delay-bounded optimal clustering structure in an SDVN, which reveals a design principle for cluster-based time-division multiple access (TDMA) protocols. The analysis suggests that a balanced tree structure shall be built for a TDMA protocol in SDVN and the optimal cluster size shall be the square root of total count of nodes. (see Section IV-A).
- *A joint optimization process to minimize channel and signal interference for reliable information dissemination*. The proposed optimization process is formulated as a binary integer linear programming problem and solved by a heuristic approach. With this process packet delivery among vehicles has a higher successful rate and lower delay. (see Section IV-D).

The rest of this paper is organized as follows. We summarize related work in Section II. Section III describes the problem formulation and statement. Section IV explains the design of ECMAC protocol, including cluster formation and channel interference mitigation. Section V demonstrates the performance of our ECMAC with other baselines. Section VI examines the issues that arose in the development of ECMAC. We conclude this paper along with future work in Section VII.

II. RELATED WORK

In this section, we discuss several related approaches, such as contention-based approaches [5] and probability-based approaches. Then, we discuss several state-of-the-art cluster-based MAC protocols. At the end, we highlight the differences between our work and the related work.

Contention-based Approach: The IEEE 802.11-OCB (outside the context of a basic service set, i.e., previously called IEEE 802.11p) [5] defines a basic MAC process that adopts the EDCA [19] mechanism for quality of services (QoS) with carrier-sense multiple access with collision avoidance (CSMA/CA) mechanism in vehicular networks. A vehicle in the OCB mode can operate in control channel (CCH) and service channel (SCH) alternatively. However, since the IEEE 802.11-OCB is designed for broadcasting instead of unicasting or forwarding packets, i.e., it does not suggest using request-to-send (RTS)/clear-to-send (CTS) and acknowledgement (ACK) mechanisms, vehicles in OCB mode will try to access the channel when the channel is idle and do random backoff after collision detected. This approach may suffer from unreliable delivery due to no acknowledgment for successful packet delivery and also a well-known issue due to no provisioning of both RTS and CTC (e.g., a hidden terminal problem). A vehicle using an ALOHA-like protocol will try to transmit a packet whenever it has a packet to send. If more than one node tries to access the channel simultaneously, a collision happens and all nodes will try to re-transmit the packets after random backoff time [20]. A successful transmission shall trigger the receiver to send an ACK back to the sender. The channel access mechanism of both OCB mode and ALOHA-like protocols is similar in the sense that they all have random backoff time for retransmission and there is no RTS/CTS process. The IEEE 802.11-OCB mode tends to have an ALOHA-like performance when vehicle density is high, which causes serious packet loss [8]. For the pure and slotted ALOHA protocol, they do not check for an idle channel before transmitting, which is different from the OCB mode, indicating that they should have worse performance than the OCB mode when vehicle density is high. Thus, for emergency packet disseminations, the IEEE 802.11-OCB may not be reliable.

The authors in [9] proposed several probabilistic and timer-based techniques for mitigating the broadcast storm problem in VANET. The *weighted p-persistence* scheme in the work [9] enables a node to forward a packet with a higher probability p that depends on a sender-receiver distance. The *slotted 1-persistence* scheme assigns an intermediate node a time slot to forward packets with probability 1. The farther the node is, the earlier time slot is assigned. The *slotted p-persistence* combines the above two schemes, where a node forwards a packet with a probability p in its assigned time slot. Although the three techniques can increase a certain reliability for packet forwardings, they are not able to guarantee that emergency information can be disseminated to all vehicles. Also, it may take a long time to forward an emergency packet when some forwarded emergency packets are lost.

Cluster-based Approach: The work in [10] presented a distributed multi-channel and mobility-aware cluster-based MAC protocol (DMMAC) for VANET. DMMAC includes a channel scheduling and an adaptive learning mechanism via a fuzzy-logic inference to organize vehicles into more stable clusters. The study in [11] proposed a multihop cluster-based IEEE 802.11p and LTE hybrid architecture for VANET safety message dissemination (VMaSC-LTE). VMaSC-LTE combines an IEEE 802.11p-based multihop clustering scheme and a data packet relay process via a 4G LTE system. VMaSC-LTE uses BSs to forward packets, which can increase the packet delivery delay and reduce the packet delivery ratio when the number of cellular users increases due to the capacity issue in cellular systems. The work in [12] introduced a unified framework of clustering (called UFC) in VANET. UFC has a backoff-based *cluster head* (CH) selection process to reduce the cluster management overhead, where each vehicle competes for CH by a backoff timer. For further reducing cluster control message exchanges, a neighbor sampling scheme was proposed so that the CH selection process is only conducted by a stable set of neighbors. However, one limitation in UFC is that the multichannel of DSRC was not considered, which significantly reduced the system usability. To investigate the collaboration of road-side unts (RSUs) and vehicles for message exchange in VANET, Ko et al. [21] proposed an RSU cooperation-based adaptive scheduling scheme that includes centralized and ad hoc data scheduling with cluster management. When considering to mitigate network congestion caused by pedestrian safety messages, Sewalkar et al. [22] introduced a multi-channel clustering-based congestion control algorithm that groups pedestrians based on their location and direction as well as uses separate channels for cluster management.

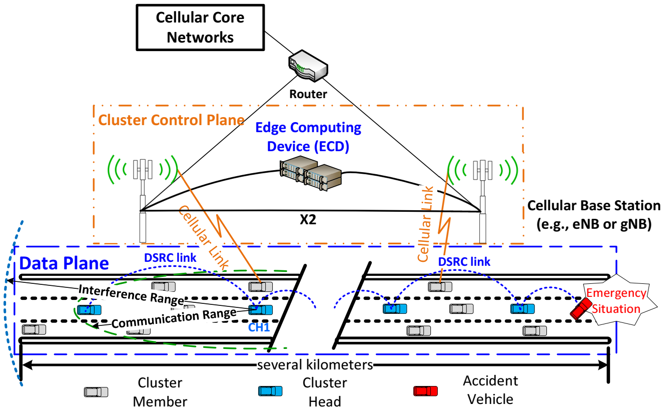


Fig. 3. Edge-assisted vehicle networks in a highway environment.

Unlike previous work, our ECMAC explores an edge-assisted cluster-based MAC protocol in vehicular networks, aiming at high reliability and low packet delivery delay. ECMAC separates the cluster management from the data plane in DSRC channels and moves it to the ECD side behind the cellular BSs. Furthermore, we propose a joint interference optimization to improve the delivery reliability of packets in a realistic wireless environment. The joint interference optimization is modeled by a binary integer linear programming problem, which is solved by a heuristic method using the concept of divide-and-conquer. In the next section, we will first formulate the problem and highlight the challenges.

III. PROBLEM FORMULATION

The objective of ECMAC is to enable vehicles to reliably and rapidly disseminate packets to multihop-away vehicles with an optimized cluster structure in a highway environment. In this section, we first give a system model for ECMAC protocol, then identify several design challenges and problems, and give several assumptions for the design of ECMAC.

A. System Model

In ECMAC, a vehicle can be a *cluster member* (CM) or a *cluster head* (CH). As shown in Fig. 3, we consider several cellular BSs connected to an ECD for cluster management in control plane. The ECD has an enough computation power, and can collect highway traffic through multiple BSs.

On the cellular network side in Fig. 3, the cluster control plane is in charge of the cluster management (i.e., cluster formation and maintenance). A vehicle periodically uploads

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3390991

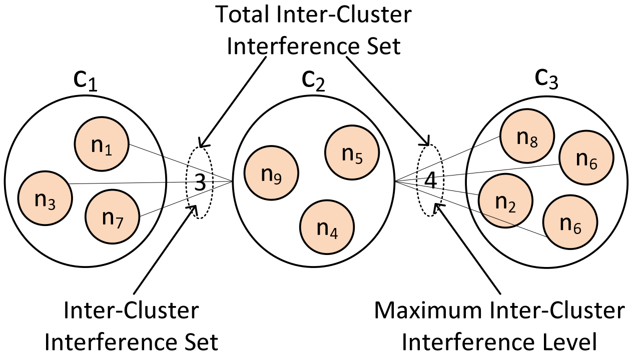


Fig. 4. Inter-cluster interference definition.

its mobility information (e.g., vehicle speed and position) to the ECD via cellular links. For scheduling transmissions of the managed vehicles, the ECD constructs a directed interference graph $G_I = (V, E)$, where V is a set of nodes (i.e., vehicles), and E is a set of edges in which a directed edge $e_{i,j}$ indicates an interference from node n_i to n_j in terms of interference range of n_i. That is, node n_j is within the interference range of node n_i, which may result in node n_j not receiving its packets correctly while node n_i is transmitting. Note that throughout this paper, we use node and vehicle interchangeably. A definition of the interference set for a node n_i is given below, suggesting that during the transmission time of some node n_j, node n_i may not be able to transmit or receive any packets due to the signal interference from node n_j, which is denoted as $e_{j,i}$.

Definition III.1. ***Interference Set*** *(I_{n_i}) All other nodes that can interfere transmissions of a node n_i, where the interfering edges to node n_i can be expressed as $I_{n_i} = \{e_{j,i} | e_{j,i} \in E\}$.*

The ECD calculates a cluster structure and a TDMA schedule for all covered vehicles, and disseminates the schedules to the vehicles via BSs. The ECD monitors the mobility of the covered vehicles to determine whether the cluster structure and schedule need to be updated. We give a series of definitions of clusters as follows and an example shown in Fig. 4.

Definition III.2. ***Cluster Set*** *(C) A group of clusters, where each cluster c_x contains a number of vehicles (i.e., a subset of V), can be expressed as $C = \{c_x | c_x \subseteq V\}$.*

Definition III.3. ***Inter-cluster Interference Set*** *($I_{c_x}^{c_y}$) The interference (i.e., an edge) from cluster c_y affects a cluster c_x, which can be expressed as $I_{c_x}^{c_y} = \{e_{j,i} | n_j \in c_y$ and $n_i \in c_x\}$.*

Definition III.4. ***Total Inter-cluster Interference Set*** *(I_{c_x}) The interference set of a cluster c_x with all other clusters, i.e., the interference from all other nodes that are not in the c_x, which can be expressed as $I_{c_x} = \{e_{j,i} | n_i \in c_i$ and $n_j \notin c_i\}$.*

Definition III.5. ***Maximum Inter-cluster Interference Level*** *($|I_{c_x}^{max}|$) The maximum (i.e. top) interference count of a cluster c_x from another cluster c_y among all other clusters, which can be expressed as $|I_{c_x}^{max}|$.*

As depicted in Fig. 4, the *inter-cluster interference set* of c_2 from c_1 includes three incident edges. The *total inter-cluster interference set* of c_2 has seven incident edges from both c_1 and c_3. The *maximum inter-cluster interference level* of c_2 is 4, which is contributed by c_3 having four nodes.

On the side of the DSRC link in Fig. 3, a vehicle can transmit packets to its CH through a channel assigned by a schedule received from the ECD. The CH also follows the schedule to relay the packets to other CHs through a common channel used by CHs. Eventually, any other CHs can receive the information from the original sender and disseminate it to their CMs. In the proposed system model, we employed a practical channel model, called Two-Ray Interference (TRI) model [23]. The TRI model considers both the freespace path loss model with shadowing and the ground reflection effects in vehicular network environments, which can capture a more realistic vehicle mobility scenario.

B. Problem Statement

When the SDN controller schedules packet transmissions for vehicles via a cluster structure, we may have the following challenges:

- The size of a cluster (i.e., the number of vehicles in a cluster) can affect the packet delay. Finding *an optimal cluster size* that provides the *optimal delivery delay* is challenging. With a given cluster structure, *a transmission schedule* shall also serve the timely delivery of emergency packets.
- Channel interference between clusters can cause packet collisions that may reduce the reliability of packet disseminations. Thus, another challenge is how to *assign available channels* to different clusters that can minimize the interference.
- Although channel interference can be reduced by a channel assignment plan, CMs can still experience *packet collisions* at a high vehicle density, especially if the interference range of vehicles is considered. Finding *an interference minimized TDMA schedule* for all vehicles can be another challenge.

To deal with the above challenges, we formulate the problem as a binary integer linear programming (BILP) problem as follows. Let C be the set of all clusters, the channel interference can be optimally minimized by solving the following BILP problem:

$$\begin{aligned} \min_{\substack{\forall c_x, c_y \in C \\ x \neq y}} \quad & \Bigg(\sum_{c_x, c_y \in C} \mathcal{H}(c_x, c_y) \sum_{\substack{n_i \in c_x \\ n_j \in c_y}} \mathcal{I}(n_i, n_j) \mathcal{T}(n_i, n_j) \Bigg), \\ \text{subject to} \quad & \sum c \leq |C|, \\ & \sum n \cdot c \leq |V|, \\ & |c_x| > 0, |c_y| > 0, \end{aligned} \tag{1}$$

where c_x and c_y are clusters in cluster set C and n_i is a node in cluster c_x. $\mathcal{H}(c_x, c_y)$, $\mathcal{I}(n_i, n_j)$, and $\mathcal{T}(n_i, n_j)$ are indicator functions of common channel use, interference range,

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3390991

and common time slot use, respectively, which are defined as follows:

$$\mathcal{H}(c_x, c_y) = \begin{cases} 1, & \text{if } c_x \text{ and } c_y \text{ with the same channel} \\ 0, & \text{if } c_x \text{ and } c_y \text{ with different channels} \end{cases}, \tag{2}$$

$$\mathcal{I}(n_i, n_j) = \begin{cases} 1, & \text{if } n_i \text{ and } n_j \text{ in interference range} \\ 0, & \text{if } n_i \text{ and } n_j \text{ not in interference range} \end{cases}, \tag{3}$$

and

$$\mathcal{T}(n_i, n_j) = \begin{cases} 1, & \text{if } n_i \text{ and } n_j \text{ in the same time slot} \\ 0, & \text{if } n_i \text{ and } n_j \text{ in different time slots} \end{cases}. \tag{4}$$

In (2), $\mathcal{H}(c_x, c_y)$ equals to one if clusters $c_x \in C$ and $c_y \in C$ operates with the same channel, and zero otherwise. Similarly, in (3) and (4), when $n_i \in c_x$, $n_j \in c_y$, and $c_x \neq c_y$, $\mathcal{I}(n_i, n_j)$ equals to one if vehicles i and j are within each other's interference range, and zero otherwise, and $\mathcal{T}(n_i, n_j)$ equals to one if vehicles i and j are assigned to a temporally overlapping time slot, and zero otherwise. $\sum \mathcal{I}(n_i, n_j), n_i \in c_x, n_j \in c_y$ gives the interference level for each node. When $\mathcal{T}(n_i, n_j)$ always equals one for all pairs of vehicles, that is, if two or more vehicles are assigned into the same time slot, they shall be assigned with different channels to transmit packets to reduce transmission collisions, which is a special case of the time slot allocation problem having only one time slot available. However, the number of channels is limited, which leads the channel interference optimization problem to be equivalent to a graph coloring problem (e.g., vertex coloring) [24], which is one of the NP problems. We are using this special case to prove that this problem is an NP since if one of the cases of the problem is NP it makes the whole problem NP. Thus, we solve this BILP problem by a heuristic approach described in Section IV-D.

C. Assumptions

Along the design of ECMAC, we have a number of assumptions as follows:

- A vehicle is equipped with both a DSRC interface and a vehicular UE (VUE) interface inside a vehicle On-Board Unit (OBU) [25].
- We assume that the time synchronization in each vehicle is done through the cellular networks. In practice, the global positioning system (GPS) and cellular networks can provide a reliable time synchronization function.
- For handover of nodes and packet forwarding between BSs, the 3GPP X2 interface is used.

Note that if a vehicle is equipped with only the DSRC interface, the proposed model can also work with the sacrifice of extra time slots for uploading vehicle mobility information toward the ECD for channel interference optimization. In the following sections, we first analyze an optimal cluster formation, and then develop a joint optimization process to minimize the interference impact.

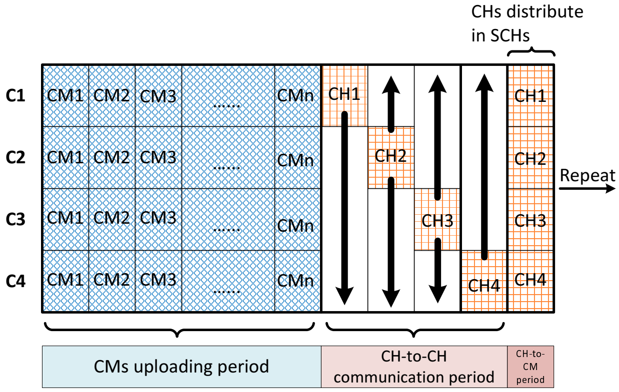


Fig. 5. An optimal scheduling for 4 clusters.

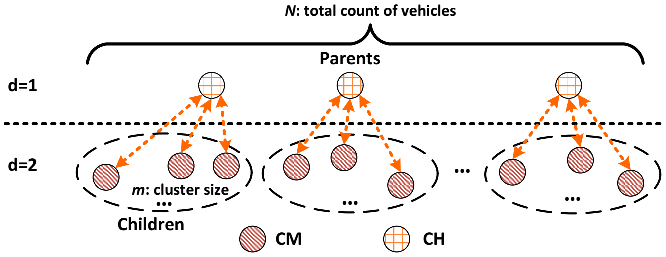


Fig. 6. A tree structure with 2 layers for forming clusters.

IV. DESIGN OF ECMAC PROTOCOL

For emergency packet dissemination, it is very important for a MAC protocol to guarantee a high successful packet delivery ratio within a bounded time. Our ECMAC is designed to achieve a high delivery ratio while minimizing the guaranteed worst-case delay. ECMAC consists of the following three components: clustering, channel assignment, and TDMA scheduling. Since these three components are interdependent, it is not feasible to solve all of them at once by a best-effort solution. Instead of solving the problems simultaneously, we address the problems one at a time. First, we propose a procedure for optimal clustering. Next, we design a channel assignment scheme for clusters. Finally, we find the best TDMA schedule by assigning a unique time slot within a frame to each vehicle and cluster head. Note that as described in Section III due to the NP-hardness of the formulated problem, identifying an optimal solution within polynomial time is challenging, thus we rely on multiple heuristic approaches to attain the best-effort solutions.

A. Analysis of Optimal Cluster Formation

Suppose that there are N nodes. A maximum slotted delay f can be formulated as follows:

$$f = m + \left\lceil \frac{N}{m} \right\rceil \text{ for } m > 0, \tag{5}$$

where m denotes a number of children per parent (i.e., cluster size), and the second term $\lceil \frac{N}{m} \rceil$ is the number of parents (i.e., cluster head). Therefore, the worst-case delay from any CM to its CH would be m in a TDMA manner. There are $\lceil \frac{N}{m} \rceil$ CHs and each cluster owns a time slot for communication among CHs, as they share one common channel. Thus, the worst-case communication delay among CHs equals $\lceil \frac{N}{m} \rceil$. This is illustrated in Fig. 5. Note that it is possible that a cluster does not have enough cluster members, but it shall build a cluster that has a cluster head. That is why we use the ceiling function to reflect this scenario. We can obtain an optimal m^* by differentiating f and solving it when it is equal to zero as follows.

$$f^{'} = 1 - Nm^{-2} = 0 \text{ for } m > 0, \tag{6}$$

and the optimal m^* is

$$m^* = \sqrt{N}. \tag{7}$$

Let d denote the number of layers of a balanced tree constructed by vehicles. For example, when $d = 2$ as shown in Fig. 6, the minimum slotted delay f^* is

$$f^* = 2N^{\frac{1}{2}}. \tag{8}$$

Considering a vehicular network with vehicle size $N = 100$ has the optimal cluster size of $m^* = 10$, and the optimal slotted delay is $f^* = 20$, indicating that the maximum delay is 20 time slots. Note that obtaining the optimal cluster size m^* requires the current number of all vehicles managed by the ECD. In our design, vehicles periodically upload their mobility information to the ECD to calculate the optimal cluster size. Other approaches can also be used to estimate the total number of vehicles in the network, such as an adaptive compressive sensing method [26]. Although we can get the optimal cluster size through this analysis, we may fail to provide reliable emergency packet disseminations in realistic wireless environments (having signal interference, radio fading, and multipath interference). To cope with this kind of failure, in the next section, we will describe the design of ECMAC protocol with an optimal clustering process and a joint optimization process for interference minimization.

B. Clustering Process

Our analysis in Section IV-A reveals that the minimum worst-case delay can be achieved by routing a packet along a full balanced tree. Obtaining an optimal tree that minimizes the worst-case delay from any general graph is *NP-hard.*

Therefore, we propose *GreedyFullBalanceTree*, which is a heuristic procedure for constructing an optimal tree from any general graph. This heuristic procedure is an iterative process and has two phases: grouping (i.e., clustering) and parent (i.e., cluster head) selection. The procedure first determines the number of children according to (7) in the analysis of Section IV-A. Suppose that the total size of the nodes is N and the optimal number of children is m. We sort all nodes based on their positions in the ascending order on the road. Then, we divide the nodes into $\lceil \frac{N}{m} \rceil$ groups, which can minimize the maximum packet forwarding delay by balancing the cluster size of each group, and also follow the analysis of optimal cluster size described in Section IV-A. For example, the first m nodes in the sorted list will form a group. Next, we select one node from each group as the parent node. This selection process considers connectivity to other members, geographic position, and relative speed of a vehicle. The connectivity of a vehicle i within its cluster c can be expressed as:

$$CON_i = \{e_{j,i} | e_{j,i} \in E \text{ and } n_j \in c\}, \tag{9}$$

where $e_{j,i}$ represents an interference source from n_j. We use an eccentric distance $\overline{d_i}$ for a vehicle i in its cluster c to select a potential geographic center node. The center point of the cluster is determined by the two farthest nodes in this cluster, and the distance between the two farthest nodes is defined as d_{max}. An average relative speed $\overline{v_i}$ of a vehicle i in its cluster c can be represented as:

$$\overline{v_i} = \frac{\sum_{n_j \in c} |v_{i,j}|}{m - 1}, \tag{10}$$

where $v_{i,j}$ is the relative speed between node i and j (i.e., speed difference).

Based on the metrics defined above, we define a CH weight metric, $W_v(n_i)$, for vehicle n_i as:

$$W_v(n_i) = \frac{|CON_i|}{|CON_{max}|} + (1 - 2\frac{\overline{d_i}}{d_{max}}) + (1 - \frac{\overline{v_i}}{v_{max}}), \tag{11}$$

where $|CON_{max}|$ and v_{max} are the maximum connection number for a vehicle to its neighbors within the cluster and the maximum speed limit, respectively. A node with the highest weight shall be selected as the CH for its cluster.

C. Cluster Control

For efficiently forwarding packets, clusters in vehicular networks should be well maintained. Here, we consider two major cluster control functions, member joining and leaving and head re-selection.

Member Joining and Leaving: When a CM moves fast, it can leave the current assigned cluster and join another cluster. Different from traditional approaches where a CM shall inform its CH for leaving and notify a new cluster CH for joining, a CM in our proposed architecture informs the ECD about its leaving from its current cluster. ECD then notifies the CH of the cluster where the CM is leaving and the CH of a cluster where the CM will be joining.

Head Re-selection: When a CH of a cluster is not capable of carrying on the role of cluster head, the ECD will determine a new CH for this cluster based on the mobility information reported by the vehicles in this cluster. The re-selection of CH is determined by the metric defined in (11). Once a new CH is selected, the ECD will notify all members of the cluster for the updated roles.

D. Minimizing Channel Interference

An optimal solution of channel interference minimization would require a joint optimization of channel assignment and a TDMA time scheduling. This joint optimization problem is formulated in Section III, which is an NP-hard problem. Therefore, we propose a heuristic method that basically divides this joint optimization into two phases. The first phase attempts to minimize *the total inter-cluster interference* by reusing the available channels. The second phase attempts to minimize *the maximum inter-cluster interference*. The channel assignment step (i.e., the first phase) receives a series of inputs, including cluster size, cluster number, CM IDs, and CH IDs, and it produces an array of channel assignments for clusters, which can be represented by a mapping data structure having pairs of key (i.e., cluster ID) and value (channel ID). The TDMA scheduling step (i.e., the second phase) receives the output from the previous step and mainly creates an allocation of channel access time for vehicles, which can be represented by a mapping data structure having pairs of vehicle id and a series of data that have role, cluster ID, start time, time slot indexes, and channel switching time. Fig. 2 illustrates the inputs and outputs of different steps in the proposed channel interference minimization.

1) Channel Assignment: We first construct an interference graph of clusters, $G_c = (V_c, E_c)$, where V_c is a set of clusters and E_c is a set of edges representing interference between clusters. We define $W_c(e_{i,j})$ as a weight of an edge $e_{i,j} \in E_c$ connecting clusters c_i and c_j such that

$$W_c(e_{i,j}) = \max_{c_i,c_j \in C}(|c_i|, |c_j|) - |\cup(n_x \in c_i \wedge n_y \in c_j)\,|. \tag{12}$$

The $W_c(e_{i,j})$ represents the interference level between cluster c_i and c_j, and the term on the right hand side represents the number of unique edges between two clusters. For example, if there are two edges $e_{1,2} \in E_c$ (connecting c_1 and c_2) and $e_{2,1} \in E_c$ (connecting c_2 and c_1), only one edge is counted since c_1 appears on both edges.

Then we apply the well-known K-way max-cut algorithm [27]. This algorithm basically finds a cut set which divides the nodes into K sets such that the total weight of edges in the cut set is maximized. A common channel is assigned to all nodes in the same set. Note that the DSRC regulates four SCHs for service communications, and two pairs of them are adjacent channels that may cause adjacent channel interference. Due to the constraint of adjacent channel interference, the algorithm may not find the solution for the problem. For this case, the four SCHs can be assigned to clusters by geographical locations as long as the neighbor clusters do not use the adjacent channels, since the one-way highway scenario can ensure that the distance between any two clusters with the same channel is the longest.

2) TDMA Scheduling: Now, we discuss how we can schedule the vehicle transmissions such that the maximum interference between vehicles is minimized. We assign unique time slots to all the vehicles in the cluster $c^{'}$ such that

$$c^{'} \leftarrow \arg\max(|c_i|), \forall c_i \in C. \tag{13}$$

To solve the problem, we propose a greedy time slot allocation (GTA) algorithm, which is shown in Algorithm 1. In this algorithm, we let C be a collection of all clusters, $c_i \in C$ be a cluster in this collection, $n_i \in c_i$ be a node that belongs to a cluster c_i. We denote the degree of a node n_i as $deg(n_i)$, which is the number of one-hop neighboring nodes of node n_i. We define $deg(n_i, c_k)$ as the number of edges incident to the node n_i in the cluster c_k. Also, let

$$deg(n_i, c^*) \leftarrow \max_{c_k \in C}\{deg(n_i, c_k)\}. \tag{14}$$

Algorithm 1 Greedy_Time_slot_Allocation_Algorithm (GTA)

```
function GREEDY_TIME_SLOT_ALLOCATION($S$)    ▷ $S$: vehicle node set
    $C \leftarrow$ClusterGroupSet($S$)    ▷ $C$ contains all clusters
    BuildNeighborList($S$)    ▷ build a neighbor list in each node
    $schedule \leftarrow \emptyset$    ▷ initialize time slot schedule set
    while $S \neq \emptyset$ do
        Construct $\{deg(n_i, c_k) | n_i \in S, c_k \in C\}$
        $deg(n_i, c^*) \leftarrow \max_{c_k \in C}\{deg(n_i, c_k)\}$
        Sort($S$)    ▷ sort nodes in a descending order by $deg(n_i, c^*)$
        for all $n_i \in S$ do
            if $i = 0$ then
                $TS \leftarrow TS \cup \{n_i\}$    ▷ assign $n_i$ with the current time slot
                $S \leftarrow S \setminus \{n_i\}$    ▷ remove $n_i$ from $S$
            else if $n_i \notin TS$ and $\{n_i \notin n_j.neig | \forall n_j \in TS\}$ then    ▷ $n_i$ is not in $TS$ and is not in the neighbor list of a node $n_j$ that is in $TS$.
                $TS \leftarrow TS \cup \{n_i\}$
                $S \leftarrow S \setminus \{n_i\}$
            end if
        end for
        $schedule \leftarrow schedule \cup \{TS\}$    ▷ put the current time slot into the schedule.
    end while
    return $schedule$
end function
```

The input S in Algorithm 1 is the vehicle node set. The algorithm process is as follows:

- Each node is assigned into the cluster group set C, and builds its neighbor list.
- Compute $deg(n_i, c^*)$ for all node $i = 1, \ldots, N$.
- Sort the nodes in the descending order of $deg(n_i, c^*)$.
- Assign an available time slot to the first node in the list.
- Go through the list and allocate a time slot to every node that is not connected to the allocated nodes and the time slot has never been assigned to another node in the same cluster.
- Cross out all the nodes allocated to the time slot by updating their neighbor lists.
- Repeat the process from the beginning on the unassigned nodes with a new or available time slot.

A flowchart of the GTA algorithm for greedy time slot allocation is shown in Fig. 7. The *GTA* algorithm is an extension of the greedy coloring algorithm (also known as Welsh-Powell Algorithm [28]). Therefore, it also shares the same computation complexity of $O(N^2)$ as it iterates through all the nodes with their neighbors in polynomial time. More importantly, this is an approximation algorithm that bounds the maximum number of time slots to less than $T^* = \max_i\{min(deg(n_i), i) | i = 1, \ldots, N\}$. It means that the longest delay between any pair of vehicles is bounded by T^*. The inter-cluster interference is completely eliminated after the time slot allocation phase is over. Also, there is no

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3390991

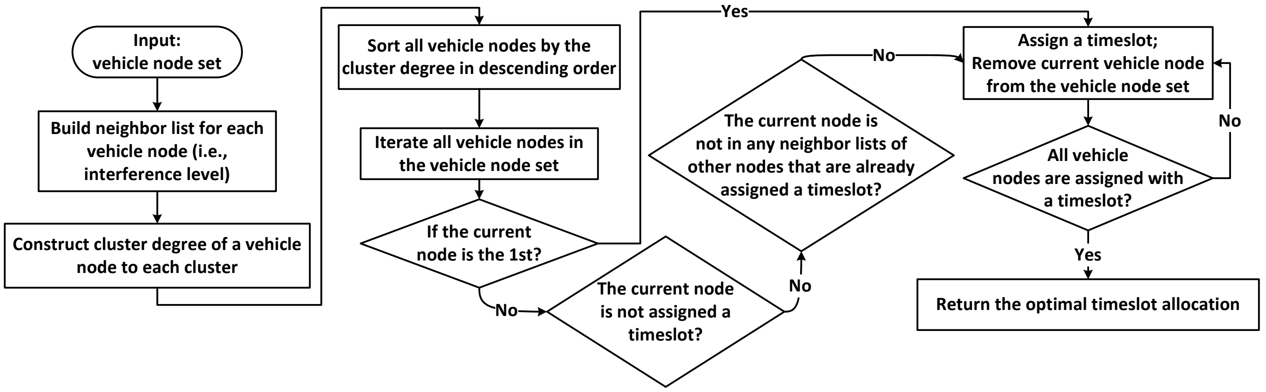


Fig. 7. The flowchart of GTA algorithm.

TABLE I
SIMULATION CONFIGURATION

Parameter	Description
Vehicle density (D)	The occupancy of all vehicles on road segments, indicating vehicle traffic volume. The default vehicle density is 0.3 veh/m.
Speed limit (v_{max})	The highest speed of a road segment. The default is 75 MPH (i.e., 120 km/h).
Packet interval (I)	The packet generation interval. The default is 0.1 s (i.e., 10 packets per second).
Data bit rate	The data transmission rate in physical layer. The default is 18 Mbps.
Radio delay	The time taken to switch from RX to TX mode for OFDM PHY. The default is 1 us.
Channel switch interval	The guard time for channel switch. The default is 2 ms.
Emergency packet size	The default packet size is 50 Bytes.
Interference range	The interference range in wireless communications. The default is 1100 m.
Transmission power	The default power is 20 mW.
Receiver sensitivity	The signal receiving threshold. The default sensitivity is -89 dBm.
Background noise	The default noise level is -110 dBm.
Path loss exponent	To calculate the signal attenuation. The default path loss exponent is 2.

TABLE II
PARAMETERS IN CELLULAR NETWORK

Parameter	Description
The number of BS	The default number is 2.
The BS TX power	The default value is 46 mW.
Upload & download resource block (RB)	Available spectrum resources for uploading and downloading flows, the default is 100.
The number of sub-carriers per RB	The default number is 12
The number of OFDM symbols per slot	The default number is 7
Maximum HARQ Re-transmission count	The default number is 3.
VUE TX power	The default value is 36 mW.
VUE mobility update interval	The default is 0.1 s.

intra-cluster interference since a unique time slot is allocated to each node within the same cluster. Note that the major difference between the original greedy coloring algorithm and the proposed GTA algorithm is that GTA allocates the same time slots (i.e., color) to nodes that do not have interference, where the original greedy coloring algorithm (e.g., Welsh-Powell Algorithm) colors vertex having the highest degree each time. In the next section, we demonstrate the performance of ECMAC protocol.

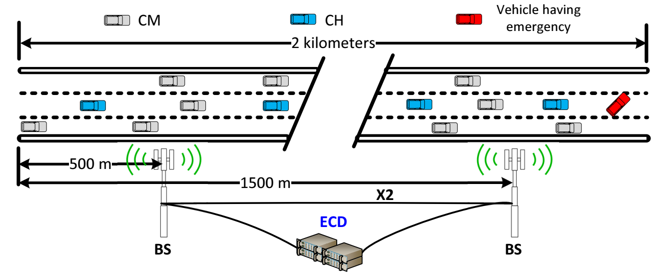


Fig. 8. Simulation setup for a highway environment.

V. PERFORMANCE EVALUATION

In this section, we show the performance evaluation of ECMAC protocol. We developed our protocol based on Veins [29] and SimuLTE [30] with a realistic vehicle mobility from the SUMO simulator [31]. The SUMO simulator uses an integrated and verified microscopic vehicle movement model to generate moving vehicles, having such as vehicle speed deviations and imperfect ratio for a driver's reaction time. We use a three-lane highway with a length of 2 km. The length of a vehicle is 5 m, and the minimum gap between any two vehicles is 2.5 m. In the simulations, vehicles are randomly generated according to different vehicle densities, and use Krauss car-following model [31] with the LC2013 lane-changing model [32]. When arriving at the end of the highway, vehicles are removed from the simulation and new

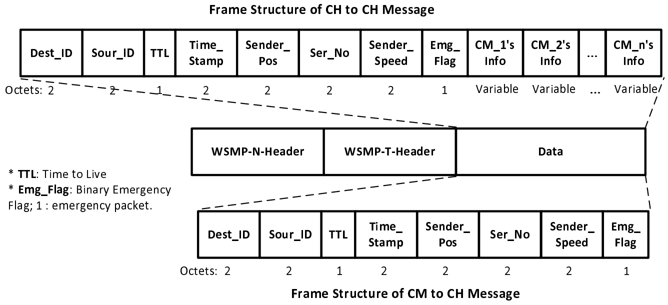


Fig. 9. The structure of frames in ECMAC.

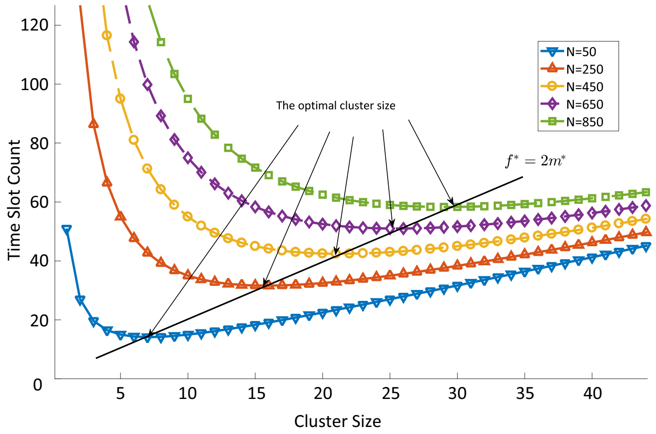


Fig. 10. The numerical results for the relation between the cluster size and the time slot count by different vehicle volume (N).

vehicles are injected from the entry of the highway following Poisson distribution. For other vehicle mobility-related parameters, we use the default values in the SUMO simulator. Two cellular BSs are placed at 500 m and 1500 m positions along the roadway, respectively. The two BSs can equally cover as many vehicles as possible. The simulation setup is shown in Fig. 8.

We have simulated an emergency packet dissemination scenario where a vehicle at the head of a highway encounters an emergency. The vehicle having the emergency floods emergency packets (EP) periodically to all other vehicles. The frame structures of CM-to-CH and CH-to-CH are shown in Fig. 9, which follows the IEEE WAVE Short Message Protocol (WSMP) standard [33] that uses the WSMP-N-Header and WSMP-T-Header to specify the network and transport information, respectively. The time-to-live (TTL) in the frame is set to 255 in our simulation. Other evaluation settings are as follows:

- **Performance Metrics**:
  - *Packet Delivery Ratio* (*PDR*): The total successful delivery ratio of packets among all transmitted packets to reach a node 2 km away.
  - *Mean End-to-End Packet Delay* (*ME2E delay*): The mean delay of a packet from a source node to a destination node. It includes queuing delays, transmission delays, forwarding delays at intermediate nodes, and any other delays.
  - *Mean Maximum E2E Packet Delay* (*MME2E delay*): The average E2E delay for the worst cases in packet forwarding.
- **Baselines**: (i) IEEE 802.11-OCB [5], (ii) DMMAC [10], (iii) slotted p-persistence [9], and (iv) Cellular-Relay [11].
- **Parameters**: (i) vehicle density (veh/m), (ii) with or without GTA algorithm, and (iii) background traffic interval (s).

For the IEEE 802.11-OCB baseline, we use a simple receive-and-forward process to disseminate packets with a single channel. For the slotted p-persistence baseline, we set forwarding probability p to 0.5 and the slot number to 5. The WAIT_TIME of the scheme is 5 ms. For each measured data point, we simulate 10 times, so all the performance results have error bars to show 90 % confidence interval. The rest of simulation configuration is shown in Table I and II.

A. Numerical Analysis for the Optimal Cluster Size

Fig. 10 shows the numerical results of the relation between the cluster size and the time slot count in different vehicle volumes, which was discussed in Section IV-A. From Fig. 10, we learn that one can select a cluster size near the optimal points for different traffic volumes, which gives a similar value for time slot count. Moreover, we can easily find that the optimal time slot count f^* has a linear increasing relation with the optimal cluster size m^*, which is $f^* = 2m^*$.

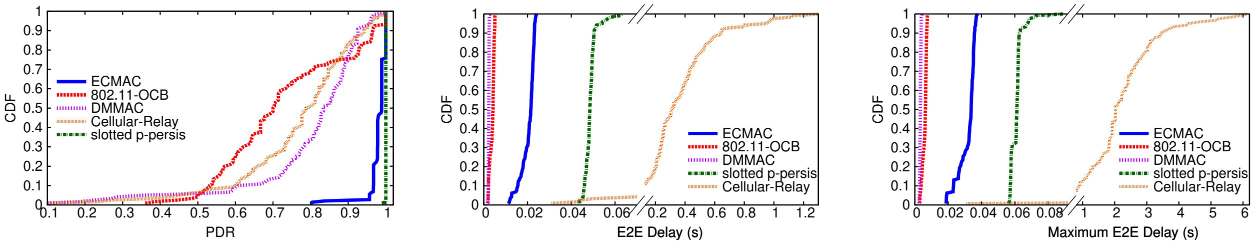
B. Impact of Vehicle Density

In order to compare the performance of ECMAC with the baseline schemes [5], [9], [10] in different vehicle density, we varied the vehicle density to run the simulations. The vehicle density is to measure the heaviness of road traffic. In our simulation, it ranges from 0.05 to 0.6 vehicle per meter in a 3-lane highway environment. The higher vehicle density indicates heavier road traffic. The results are shown in Fig. 11. From Fig. 11(a) and 11(d), we can see that the emergency PDR of ECMAC is close to 100 % at most densities except density 0.05. This good performance is because that ECMAC provides an optimum cluster structure through an interference optimization process that can minimize the channel interference. The performance downgrade of ECMAC at density 0.05 is due to the sparse vehicular network where some vehicles in the optimum cluster structure cannot connect with each other. The emergency PDRs of 802.11-OCB continuously decrease when the vehicle densities increase due to the broadcast storm. Both DMMAC and Cellular-Relay have a low PDRs as the vehicle density is low, and when the vehicle densities increase, the PDRs fluctuate between 80 % and 90 %. The slotted p-persistence scheme has almost 100 % PDRs at most of the time. This is because the scheme uses a probability based forward process that mitigate the broadcast storm problem.

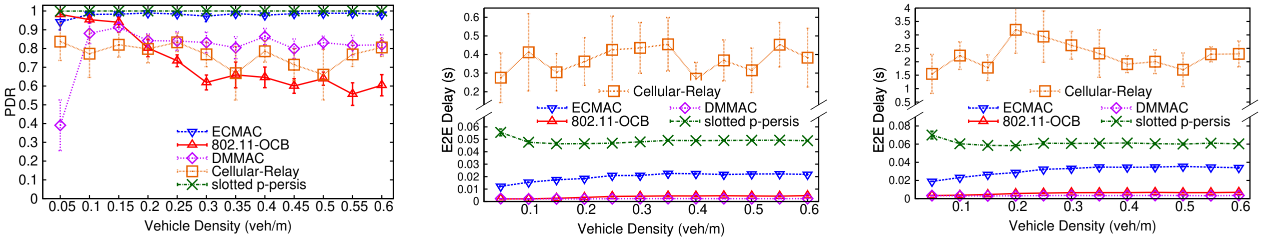
We also measure the ME2E and MME2E delays for an EP traveling to 2 km away in a highway. Fig. 11(b) and 11(e) show that the ME2E delay in ECMAC is worse than that of 802.11-OCB, DMMAC, but better than that of slotted p-persistence and Cellular-Relay. The CSMA-like process (802.11-OCB and DMMAC) enables a vehicle to transmit a packet whenever the wireless channel is not occupied. On the other hand, since ECMAC uses TDMA schedule for packet dissemination, a vehicle needs to wait for its time slot to transmit a packet, which may increase the packet delivery time. However, considering the high PDR of ECMAC, ECMAC can provide a more reliable emergency packet dissemination service than other baselines. Fig. 11(c) and 11(f) show the performance of the MME2E delay (i.e., worst cases). The MME2E delay of ECMAC is lower than that of slotted p-persistence and Cellular-Relay but higher than that of 802.11-OCB and DMMAC.

By the performance of ECMAC and other schemes in various vehicle densities, we can see that ECMAC obtains a good balance between reliability and efficiency for packet dissemination.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3390991



(a) CDF of Emergency PDR (b) CDF of E2E Delay (c) CDF of Maximum E2E Delay



(d) Emergency PDR (e) E2E Delay (f) Maximum E2E Delay

Fig. 11. Impact of vehicle density.

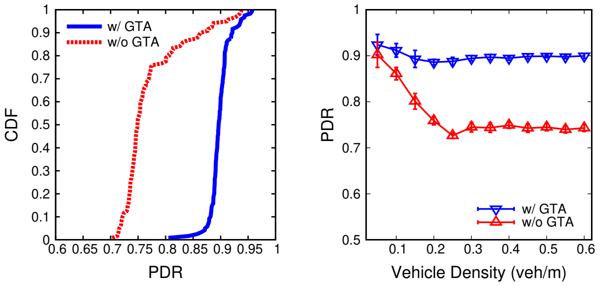
C. Effectiveness of GTA Algorithm

To verify the effectiveness of GTA in Algorithm 1, we use a sequential time slot allocation scheme as a baseline that does not consider the interference among vehicles (marked by “w/o GTA” in Fig. 12). In the simulation, vehicles periodically generate packets with random destinations, which is different from the other evaluation setting where a vehicle tries to disseminate EPs to multihop-away vehicles.

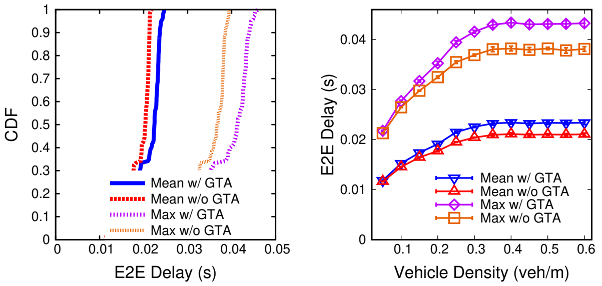
The results are shown in Fig. 12 where the “w/ GTA” indicates using GTA. It can be seen that the PDR increases from about 74 % to about 90 % compared with that of without GTA as the vehicle density increases. Since GTA algorithm removes the channel interference by extending the TDMA schedule of ECMAC, the ME2E delay and MME2E delay are also increased in comparison with that without it. It is also interesting to see the gains in PDR and losses in E2E delay at the same time. We show the trade-off between PDR and E2E delay in Fig. 13. Algorithm 1 improves the PDR by about 20 % at a high vehicle density (>0.25), whereas the ME2E and MME2E delay downgrade by about 10 % and 12 %, respectively. Because our primary concern is on the reliability of packet dissemination, this trade-off is acceptable as long as the E2E delay is within a safe-zone for emergency situations, which is less than 0.1 s [34].

D. Impact of Background Traffic Interval

To evaluate the impact of background traffic on the protocols, we conducted simulations with background traffic intervals (BTI) ranging from 0.01 s to 0.1 s. Note that BTI is the packet generation interval of each vehicle for broadcasting, where 0.1 s (i.e., 100 ms) is generally considered to be the largest interval of sending beacons for driving safety [35]. Fig. 14 and 15 show the performance of ECMAC and other baselines in terms of PDR and MME2E delay. Fig. 14(b) demonstrates that as the BTI increases, the PDRs in most of the schemes increase correspondingly except for 802.11-OCB.



(a) The CDF of PDR (b) The PDR



(c) The CDF of E2E delay (d) The E2E delay

Fig. 12. Effectiveness of the GTA algorithm.

As shown in Fig. 15, when the BTI is greater than 0.03 s, the MME2E delays of ECMAC fluctuate between 0.02 s and 0.04 s. The MME2E delays of ECMAC have a surge when the BTI is lower than 0.03 s. This is because ECMAC allocates time slots to vehicles for transmitting packets and the total time of the schedule depends on the count of vehicles. When the BTI is smaller than the period of a schedule, the generated packets have to be queued in the MAC layer to wait for the next transmission opportunity, which can cause a longer delay.

In sum, based on our simulation results, ECMAC performs well in emergency PDR compared to other state-of-the-art baselines. To distribute emergency packets to multihop-away vehicles, ECMAC can guarantee a high delivery ratio while

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3390991

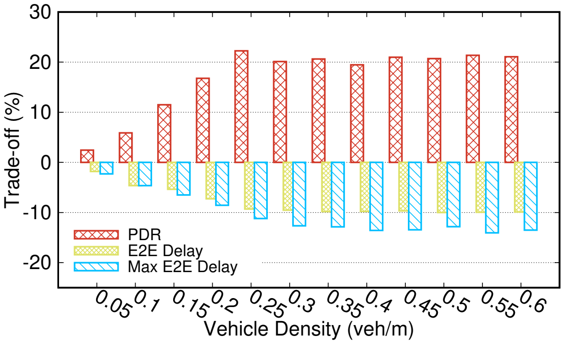
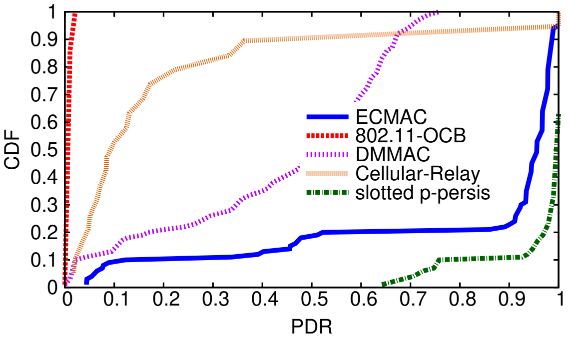
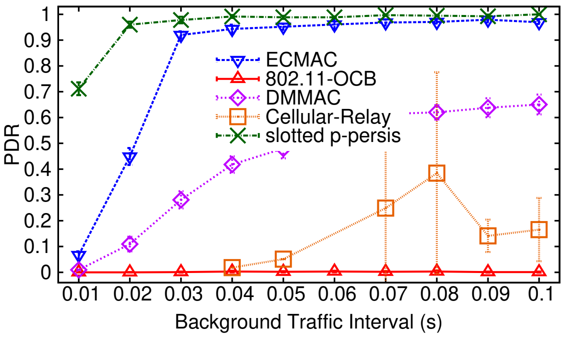


Fig. 13. Trade-off between PDR and E2E delay.



(a) CDF of Emergency PDR



(b) Emergency PDR

Fig. 14. Impact of background traffic on emergency PDR.

maintaining a proper delivery delay.

VI. DISCUSSION

In this work, we used the SUMO simulator to generate vehicle traffic in a highway environment for evaluating our ECMAC with other baselines. Our simulator can also accommodate real vehicle trajectory data for the evaluation, such as highD dataset [36], by importing a dataset of vehicle trajectory into the SUMO simulator. Generally, the order of magnitude in the packet E2E latency is in millisecond level, whereas relative speeds among vehicles are on the order of magnitude with seconds. That is, the relative distance differences between vehicles have less effect on the latency of the packet E2E. Since our focus is on the efficiency and reliability of the proposed protocol, using real vehicle trajectory data would have less impact on the performance evaluation.

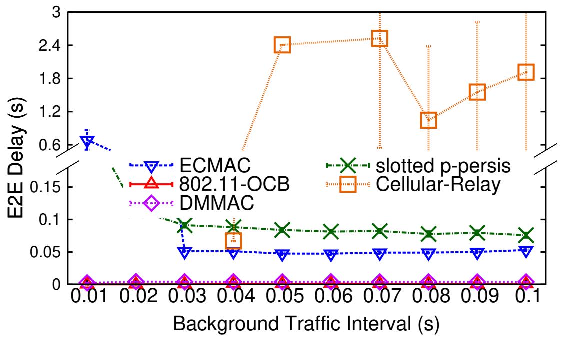


Fig. 15. Impact of background traffic on MM E2E delay.

Another issue may arise in collecting vehicle mobility information via BSs. Generally, a BS has its capacity to serve a certain number of UEs. If the number of UEs (including general UEs and VUEs) simultaneously accessing the cellular network channels exceeds the serving capacity of a BS in extreme cases (e.g., natural disasters), a congestion can happen and the update for vehicle mobility information in ECD can be delayed. In this situation, the messages transmitted by vehicles can be lost due to a stale transmission schedule. For solving this issue, dedicated cellular channels shall be guaranteed for VUEs to upload vehicle mobility information.

VII. CONCLUSION

In this paper, we have presented an edge-assisted cluster-based MAC protocol (called ECMAC) in software-defined vehicular networks. In our design, an edge computing device calculates an optimum cluster formation and conducts an channel interference optimization process, which guarantees the reliable packet forwarding. The delay-bounded optimal cluster structure reveals a design principle for a cluster-based protocol. As future work, we may extend ECMAC to a two-way traffic environment in which a dynamic channel switching will be applied for clusters in both directed roadways to minimize the frame collisions due to duplicate channel assignment.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government, Ministry of Science and ICT (MSIT) (No. 2023R1A2C2002990). This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the MSIT (No. 2022-0-01199, Regional strategic industry convergence security core talent training business). This work was supported in part by the DGIST R&D Program of the MSIT under Grant 18-EE-01. This research was also supported in part by Basic Science Research Program through the NRF of Korea funded by the Ministry of Education (No. 2022R1I1A1A01053915).

REFERENCES

[1] J. Jeong, Y. Shen, T. Oh, S. Céspedes, N. Benamar, M. Wetterwald, and J. Härri, "A Comprehensive Survey on Vehicular Networks for Smart Roads: A Focus on IP-based Approaches," *Vehicular Communications*, vol. 29, p. 100334, 2021.

[2] A. Ahmed, M. M. Iqbal, S. Jabbar, M. Ibrar, A. Erbad, and H. Song, "Position-based emergency message dissemination schemes in the internet of vehicles: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 13 548–13 572, 2023.

[3] IEEE, "IEEE Standard-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Next Generation V2X," *IEEE Std 802.11bd-2022*, pp. 1–144, 2023.

[4] ASTM, "Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems," *ASTM E2213-03(2010)*, pp. 1–25, 2010.

[5] IEEE, "IEEE Standard-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pp. 1–4379, 2021.

[6] 3GPP, "Architecture Enhancements for 5G System (5GS) to Support Vehicle-to-Everything (V2X) Services," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.287, Dec. 2023, version 18.2.0.

[7] ——, "Vehicle-to-Everything (V2X) Services in 5G System (5GS); Stage 3," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 24.587, Jan. 2024, version 18.4.0.

[8] J. Thota, N. F. Abdullah, A. Doufexi, and S. Armour, "V2V for Vehicular Safety Applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2571–2585, 2020.

[9] N. Wisitpongphan, O. K. Tonguz, J. S. Parikh, P. Mudalige, F. Bai, and V. Sadekar, "Broadcast Storm Mitigation Techniques in Vehicular Ad Hoc Networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 84–94, December 2007.

[10] K. Hafeez, L. Zhao, J. Mark, X. Shen, and Z. Niu, "Distributed Multichannel and Mobility-Aware Cluster-Based MAC Protocol for Vehicular Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 8, pp. 3886–3902, Oct 2013.

[11] S. Ucar, S. C. Ergen, and O. Ozkasap, "Multihop-Cluster-Based IEEE 802.11p and LTE Hybrid Architecture for VANET Safety Message Dissemination," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2621–2636, April 2016.

[12] M. Ren, J. Zhang, L. Khoukhi, H. Labiod, and V. Vèque, "A Unified Framework of Clustering Approach in Vehicular Ad Hoc Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1401–1414, May 2018.

[13] W. Qi, B. Landfeldt, Q. Song, L. Guo, and A. Jamalipour, "Traffic differentiated clustering routing in dsrc and c-v2x hybrid vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7723–7734, 2020.

[14] Y. Xia, L. Wu, Z. Wang, X. Zheng, and J. Jin, "Cluster-enabled cooperative scheduling based on reinforcement learning for high-mobility vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 664–12 678, 2020.

[15] J. Garbiso, A. Diaconescu, M. Coupechoux, and B. Leroy, "Auto-adaptive multi-hop clustering for hybrid cellular-vehicular networks," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Oct 2017, pp. 1–6.

[16] G. Naik, B. Choudhury, and J. Park, "IEEE 802.11bd & 5G NR V2X: Evolution of Radio Access Technologies for V2X Communications," *IEEE Access*, vol. 7, pp. 70 169–70 184, 2019.

[17] A. Kashinath, M. Hasan, R. Kumar, S. Mohan, R. B. Bobba, and S. Padhy, "Safety Critical Networks using Commodity SDNs," in *IEEE INFOCOM 2021*, 2021, pp. 1–10.

[18] J. Wang, K. Zhu, B. Chen, and Z. Han, "Distributed clustering-based cooperative vehicular edge computing for real-time offloading requests," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 653–669, 2022.

[19] Y. Gao, X. Sun, and L. Dai, "IEEE 802.11e EDCA Networks: Modeling, Differentiation and Optimization," *IEEE Transactions on Communications*, vol. 13, no. 7, pp. 3863–3879, July 2014.

[20] J.-B. Seo, B. C. Jung, and H. Jin, "Modeling and online adaptation of aloha for low-power wide-area networks (lpwans)," *IEEE Internet of Things Journal*, vol. 8, no. 20, pp. 15 608–15 619, 2021.

[21] B. Ko, K. Liu, S. H. Son, and K.-J. Park, "Rsu-assisted adaptive scheduling for vehicle-to-vehicle data sharing in bidirectional road scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 977–989, 2021.

[22] P. Sewalkar and J. Seitz, "Mc-coco4v2p: Multi-channel clustering-based congestion control for vehicle-to-pedestrian communication," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 3, pp. 523–532, 2021.

[23] C. Sommer, S. Joerer, and F. Dressler, "On the applicability of two-ray path loss models for vehicular network simulation," in *2012 IEEE Vehicular Networking Conference (VNC)*, 2012, pp. 64–69.

[24] D. B. West, *Introduction to Graph Theory*, 2nd ed. Prentice Hall, September 2000.

[25] M. Segata, R. L. Cigno, T. Hardes, J. Heinovski, M. Schettler, B. Bloessl, C. Sommer, and F. Dressler, "Multi-technology cooperative driving: An analysis based on plexe," *IEEE Transactions on Mobile Computing*, vol. 22, no. 8, pp. 4792–4806, 2023.

[26] Y. Cai, H. Zhu, S. Chang, X. Wang, J. Shen, and M. Guo, "Peerprobe: Estimating vehicular neighbor distribution with adaptive compressive sensing," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1703–1716, 2022.

[27] Goemans, Michel X. and Williamson, David P., "Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming," *Journal of the ACM*, vol. 42, no. 6, pp. 1115–1145, Nov. 1995.

[28] Welsh, D. J. A. and Powell, M. B., "An upper bound for the chromatic number of a graph and its application to timetabling problems," *The Computer Journal*, vol. 10, no. 1, pp. 85–86, 1967.

[29] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, Jan 2011.

[30] A. Virdis, G. Stea, and G. Nardini, "Simulating lte/lte-advanced networks with simulte," in *Simulation and Modeling Methodologies, Technologies and Applications*, M. S. Obaidat et al., Ed. Cham: Springer International Publishing, 2015, pp. 83–105.

[31] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. WieBner, "Microscopic Traffic Simulation using SUMO," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 2575–2582.

[32] J. Erdmann, "Sumo's lane-changing model," in *Modeling Mobility with Open Data*, M. Behrisch and M. Weber, Eds. Cham: Springer International Publishing, 2015, pp. 105–123.

[33] IEEE, "IEEE Standard for Wireless Access in Vehicular Environments (WAVE)–Networking Services," *IEEE Std 1609.3-2020 (Revision of IEEE Std 1609.3-2016)*, pp. 1–210, 2021.

[34] T. K. Mak, K. P. Laberteaux, and R. Sengupta, "A Multi-channel VANET Providing Concurrent Safety and Commercial Services," in *Proceedings of the 2Nd ACM International Workshop on Vehicular Ad Hoc Networks*, ser. VANET '05. New York, NY, USA: ACM, 2005, pp. 1–9.

[35] S. Zhang, J. Chen, F. Lyu, N. Cheng, W. Shi, and X. Shen, "Vehicular communication networks in the automated driving era," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 26–32, 2018.

[36] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2118–2125.

Yiwen Shen received the Ph.D. degree from the Department of Electrical & Computer Engineering, Sungkyunkwan University, South Korea, in 2021. He is currently a Research Professor with the College of Computing and Informatics, Sungkyunkwan University, South Korea. His research interests include connected and automated vehicles (CAV), intelligent transportation systems (ITS), vehicular ad hoc networks (VANET), and IoT systems. He is a member of the IEEE.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2024.3390991

Jaehoon (Paul) Jeong received the B.S. degree from the Department of Information Engineering, Sungkyunkwan University, in 1999, the M.S. degree from the School of Computer Science and Engineering, Seoul National University, South Korea, in 2001, and the Ph.D. degree from the Department of Computer Science and Engineering, University of Minnesota, in 2009. He is currently an Associate Professor with the Department of Computer Science & Engineering, Sungkyunkwan University. His research interests include Internet of Things, vehicular networks, wireless sensor networks, and mobile ad hoc networks. He is a member of the ACM and the IEEE Computer Society.

Junghyun Jun received the MS degree in electrical engineering system from the USC Viterbi School of Engineering, in 2005, and the PhD degree in computer science and engineering from the University of Cincinnati, in 2011. He is a research engineer for Trimble New Zealand. Previously, he was an assistant professor at the Indian Institute of Technology, Ropar. Prior to this, he served as a postdoc research associate with the Singapore University of Technology and Design. He is a recipient of the best paper/demo awards from WCNC'13 and IPSN'12 (runner-up). He is a member of the IEEE.

Tae (Tom) Oh received the B.S. degree in electrical engineering from Texas Tech University in 1991, and the M.S. and Ph.D. degrees in electrical engineering from Southern Methodist University in 1995 and 2001, respectively, while working for telecommunication and defense companies. He is currently an Associate Professor with the Department of Information Sciences and Technologies and the Department of Computing Security, Rochester Institute of Technology. His research includes mobile ad hoc networks, vehicle area networks, sensor networks, and mobile device security. He is a member of the ACM.

Youngmi Baek received her Ph.D. degrees in Computer Engineering from the Kyungpook National University, Daegu, Korea, in 2015. In 2002 and 2015, she joined the PIRL (POSTECH Information Research Laboratory) and the CPS Global Center, DGIST, as a Researcher, respectively. Since February 2017, she has been with the Department of Information and Communication Engineering, DGIST, where she is a Research Professor. Her current research interests include communication protocols and network security in vehicular ad-hoc networks, in-vehicle networks, cyber-physical systems, and wireless sensor networks.