# RSU-Assisted Adaptive Scheduling for Vehicle-to-Vehicle Data Sharing in Bidirectional Road Scenarios

Byungjin Ko, *Student Member, IEEE*, Kai Liu, *Senior Member, IEEE*, Sang Hyuk Son, *Fellow, IEEE*, and Kyung-Joon Park, *Member, IEEE*

***Abstract*—This study investigates the synergy between centralized and decentralized (i.e., ad hoc) data scheduling in vehicular ad hoc networks (VANETs) for offloading and balancing the workloads of roadside units (RSU) in bidirectional road scenarios. In the centralized scheduling, an RSU schedules data dissemination using a hybrid of infrastructure-to-vehicle (I2V) and vehicle-to-vehicle (V2V) communications. Specifically, RSUs cooperate with each other by transferring unserved requests, and each RSU schedules the data services based on its locally received requests and the transferred requests. In the decentralized scheduling, vehicles driving in opposite directions share the cached data items via V2V communication when out of the coverage of RSU. The ad hoc scheduling can be benefited from the cooperation among RSUs since the chance for V2V data sharing could be enhanced when considering transferred requests into scheduling at each RSU. We formulate a hybrid of centralized and ad hoc data scheduling (HCA) problem, aiming at best exploiting the synergistic effects of I2V and V2V communication based on centralized data broadcast and ad hoc data sharing. On this basis, we propose an RSU Cooperation-based Adaptive Scheduling (RCAS) algorithm that consists of three mechanisms, including a centralized scheduling mechanism at each RSU, an ad hoc scheduling mechanism for vehicles, and a cluster management mechanism. Finally, we build the simulation model and give a comprehensive performance evaluation, which demonstrates the superiority of the proposed solution under a variety of circumstances.**

***Index Terms*—RSU cooperation, cooperative data dissemination, adaptive scheduling, cluster-based data sharing.**

## I. Introduction

RECENT advances in wireless communication technologies have paved the way for developing vehicular ad hoc networks (VANETs) and enabling emerging intelligent transportation systems. As a de facto communication protocol for VANETs, dedicated short range communication (DSRC) supports both infrastructure-to-vehicle (I2V) and vehicle-to-vehicle (V2V) communications. Specifically, vehicles equipped with on-board units (OBU) can communicate with roadside units (RSU) via I2V communication and can share information with neighboring vehicles via V2V communication.

Efficient data dissemination is critical for various VANET data services, e.g., to improve vehicle safety, transportation efficiency, and passenger comfort [1]–[3]. Although significant effort has been devoted to information services in vehicular networks, many existing studies have focused on improving the quality and reliability of I2V and V2V communication at the medium access control (MAC) and network layers [4]–[7]. In addition, a number of studies have been focused on developing vehicular network routing protocols [8]–[11]. Recently, many broadcasting protocols have been proposed for vehicular networks [12]–[14], primarily to enhance I2V bandwidth efficiency within the coverage range of an RSU. Some of the studies have investigated the cooperation of hybrid I2V/V2V communication [15]–[18]. To the best of our knowledge, none of previous studies have considered synergistic data services within and beyond the RSU coverage area to offload and balance RSU workloads in bidirectional road scenarios.

This work considers synergistic data service wherein vehicles may retrieve required data items via either I2V or V2V communication from RSUs or peer vehicles, respectively. Such a service paradigm has great potential to improve overall service performance because a) the service workload among different RSUs may be highly uneven, and hence it is important to balance the workload among RSUs via the transferring unserved requests to peer RSUs. b) Due to the high vehicle mobility, limited I2V bandwidth, as well as high demand on data services, it is likely that RSUs cannot complete the service within their coverage. Therefore, it is critical to exploit the V2V based data sharing outside of RSU's coverage. c) Since the performance of V2V based data sharing is highly related to the cached contents in vehicles, which are retrieved from RSUs. Clearly, it is important to best explore the synergetic effects between the RSU based and the ad hoc based scheduling. Considering the budget

Manuscript received January 17, 2019; revised August 15, 2019 and December 16, 2019; accepted December 18, 2019. Date of publication January 1, 2020; date of current version February 2, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61872049, in part by the Global Research Laboratory Program under Grant 2013K1A1A2A02078326 through the NRF, in part by the Daegu Gyeongbuk Institute of Science and Technology (DGIST) Research and Development Program of the Ministry of Science and ICT under Grant 18-EE-01, and in part by the Venture and Innovation Support Program for Chongqing Overseas Returnees under Project cx2018016. The Associate Editor for this article was X. Cheng. *(Corresponding author: Kai Liu.)*

B. Ko, S. H. Son, and K.-J. Park are with the Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu 42998, South Korea (e-mail: byungjin.ko@dgist.ac.kr; son@dgist.ac.kr; kjp@dgist.ac.kr).

K. Liu is with the College of Computer Science, Chongqing University, Chongqing 400030, China (e-mail: liukai0807@gmail.com).

Digital Object Identifier 10.1109/TITS.2019.2961705

and policy for RSU deployment, it is commonly believed that RSUs would not be deployed densely along the road [19]–[21]. In addition, with the ever-increasing data demands by vehicles and the limited I2V communication bandwidth, it is not sufficient to provide services solely based on RSUs. Therefore, it is important to have an efficient ad hoc mode to better exploit V2V communication bandwidth and enhance system scalability.

Nevertheless, it is non-trivial to achieve the above visions. First, the network topology is highly dynamic due to the high mobility of vehicles. This makes it challenging to design an efficient clustering mechanism. Second, considering that two vehicles driving in opposite directions may have high relative speed, designing an efficient V2V communication protocol is challenging due to the very short time window available for data sharing. Third, considering the communication constraints, such as bandwidth, the half-duplex transmission of OBUs, and concurrent transmission interference, it is difficult to make best utilization of the wireless bandwidth. Fourth, since the number of data items that can be retrieved by a vehicle is limited within the coverage of an RSU due to the high mobility of vehicles and the limited I2V bandwidth, it is non-trivial to serve both local requests as well as transferred requests efficiently. Fifth, the high and uneven service workload due to the nature of traffic features make it more challenging to design efficient scheduling algorithms.

To address the above issues, this study proposes a centralized scheduling algorithm within the RSU coverage area and an ad hoc data dissemination strategy outside the coverage area. Note that the designed solution focuses on the potential of V2V communication for data sharing among vehicles driving in opposite directions. To exploit the synergistic effect between the centralized and ad hoc data scheduling mechanisms, RSUs are cooperated by sharing the unserved requests information of requester vehicles. Based on the transferred information, each RSU can schedule data services for the vehicles within its coverage to cache the data items corresponding to the transferred requests of other vehicles which are approaching to it. In this manner, it has the potential to enhance V2V data sharing opportunity out of RSU's coverage. To enable the ad hoc data service, RSUs instruct vehicles to form clusters based on real-time traffic conditions. Initially, an RSU will assign a cluster head (CH) that manages the cluster by designating the next CH. After leaving the RSU coverage area, the CH designates a sender vehicle in the cluster to improve the spatial reusability of V2V communication bandwidth. The main contributions of this work are as follows.

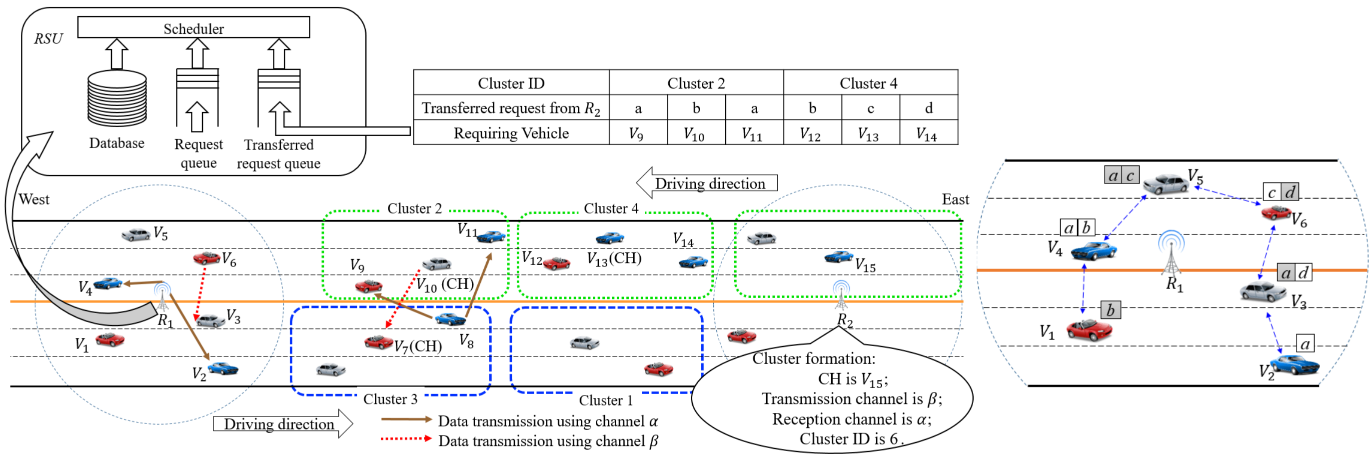
- We present a service architecture in bidirectional road scenarios via cooperative I2V/V2V communications, which coordinates both the centralized scheduling within RSUs' coverage and the ad hoc data scheduling out of the coverage. Specifically, such an architecture enables the cooperative service among RSUs, which not only enhances ad hoc data sharing opportunities among vehicles driving in opposite directions, but also realizes the workload balancing among RSUs.
- We formulate a hybrid of centralized and ad hoc data scheduling (HCA) problem, aiming at best exploiting the synergistic effects of I2V and V2V communication based on centralized data broadcast and ad hoc data sharing. In particular, it takes into consideration of the following factors to maximize overall service performance: a) the service efficiency of I2V based communication within RSUs' coverage; b) the cooperative caching efficiency via the sharing of service information among RSUs; c) the ad hoc based data sharing efficiency via V2V based communication outside RSUs' coverage.
- We propose an RSU Cooperation-based Adaptive Scheduling (RCAS) algorithm that consists of three mechanisms. First, for the centralized scheduling, we transform the scheduling problem to the maximum weighted independent set (MWIS) problem and apply a greedy method to solve the problem approximately. Second, for the ad hoc data scheduling, we propose a policy to schedule the sender vehicles and the corresponding data items to be shared via V2V communication. Third, to support the ad hoc data scheduling, we propose a cluster management mechanism that implements cluster generation, CH designation, cluster joining, and leaving for vehicles.

The remainder of this paper is organized as follows. Section II reviews related work. The proposed system is described in Section III. The HCA problem is formulated in Section IV, and the RCAS algorithm is proposed in Section V. In Section VI, we build the simulation model and give performance evaluation. Finally, we conclude the paper and discuss future work in Section VII.

## II. Related Work

Previous studies on data dissemination in VANETs have primarily focused on improving network throughput and enhancing packet transmission reliability, which are the problems resided in the MAC and network layers. Bharati and Zhuang [4] proposed a cooperative ad hoc MAC (CAH-MAC) scheme for VANETs, in which a neighboring node relays a packet to the destination node during an unserved time slot when detecting transmission failure between the sender and destination nodes. D.N.M. Dang *et al.* [5] proposed a multi-channel MAC for VANETs called HER-MAC, which supports time-division multiple access (TDMA) and CSMA multiple access schemes, and improves both safety message transmission opportunity and non-safety message transmission throughput. Prediction-based TDMA MAC (PTMAC) [6] was proposed to decrease the probability of packet collisions. Time slots are scheduled to eliminate potential collisions, which can be predicted based on real-time traffic conditions and vehicle information.

In addition, many studies have investigated multi-hop routing via V2V communications in VANETs. Wu *et al.* [8] proposed a broadcast protocol for a low message overhead and a high packet dissemination ratio. Fuzzy logic is used to select the next hop relay nodes based on inter-vehicle distance, vehicle speed, and link quality. Network coding is used to improve the packet dissemination ratio. To improve the packet delivery ratio and maintain low end-to-end delays, Rehman *et al.* [9]




(a) Overview of the service architecture (b) In the coverage of $R_1$

Fig. 1. Data dissemination in bidirectional scenarios via hybrid of I2V and V2V communications and multi-RSU cooperation.

proposed a relay node selection scheme, which is based on the link qualities of single-hop neighboring nodes, for multi-hop broadcasting over a platoon of vehicles. The scheme adaptively balances the estimated link qualities and the distance between the source broadcaster and potential forwarders when designating a relay node. Bi *et al.* [10] proposed a multi-hop broadcast protocol to disseminate emergency messages. According to the position of the emergency message sender in an urban area, a broadcast strategy is selected among bidirectional, multi-directional, and directional broadcasts. Hao *et al.* [11] proposed a protocol for cooperative data downloading without collisions and hidden terminal issues. For multi-hop data services, an RSU selects a vehicle to download a data item, which then acts as the data source and selects relay vehicles for multi-hop V2V based data dissemination.

For data scheduling at the application layer, Wang *et al.* [22] proposed a dynamic clustering and cooperative scheduling algorithm for data services out of the coverage of RSUs. To alleviate data collisions, clusters are activated based on time division policy. Meneguette and Boukerche [23] proposed a protocol to facilitate resource sharing in a mobile cloud based on V2V communication. Vehicles collaborate to share and search for feasible resources in the vehicular cloud by forwarding the message to neighboring vehicles until the message reaches the receiver vehicle. Zeng *et al.* [24] proposed a recursive least squares algorithm to realize large-scale channel prediction with low computational complexity, which is exploited for the centralized scheduling. K. Liu *et al.* [15] proposed an algorithm called SFR to effectively exploit the joint effects of I2V and V2V communication, which considers service scalability, fairness, and robustness in scheduling. Ko *et al.* [16] proposed maximum service (MS) scheduling algorithm to improve channel efficiency and the data service rate. Dai *et al.* [25] proposed a heuristic algorithm to schedule temporal data services in software-defined netwrok (SDN)-based heterogeneous vehicular networks. The algorithm adaptively schedules broadcast tasks of requests using the heterogeneous interfaces, to maximize the service ratio. Liu *et al.* [26] proposed a network-coding-assisted scheduling algorithm using the hybrid of I2V and V2V communications, which exploits both network coding and vehicular caching to increase the opportunity of data sharing among vehicles.

Unlike these previous studies, this work makes the first attempt on exploiting the synergistic effect on both centralized and ad hoc scheduling within and outside RSUs' coverage, respectively, by considering cooperation among RSUs, as well as cooperative caching among vehicles driving in opposite directions.

## III. System Architecture

In this section, we present the system architecture for data dissemination in a bidirectional road scenario via a hybrid of I2V and V2V communications. Specifically, to provide data services cooperatively, multiple RSUs are connected in the wired backbone network. Each vehicle is equipped with an OBU that supports both I2V and V2V communications. Data services are provided via centralized scheduling within an RSU coverage area and decentralized scheduling outside of the coverage area. To balance the workloads, some RSUs may transfer their unserved requests to neighboring RSUs by collecting both service information and driving status of vehicles within their coverage. With such a mechanism, RSUs can be cooperated to schedule data services so as to maximize overall system performance. According to the scheduling result, some vehicles are instructed to receive a data item via I2V communication and other vehicles may be instructed to tune to the V2V channel for data transmission or reception. Besides, to alleviate RSUs' workloads, when vehicles are out of RSU coverage, data services are provided in a decentralized manner via V2V communication. Note that vehicles can only retrieve their requested data items.

The system architecture is shown in Fig. 1 (a). Vehicles submit requests to an RSU via I2V communication. Besides, an RSU may receive information including unserved request ID, requester vehicle ID, and the vehicle's latest status, from neighboring RSUs. Thus, each RSU maintains two queues for outstanding requests, i.e., a pending request queue for requests submitted in its own coverage area and a transferred request queue for requests transferred from peer RSUs. The RSU makes scheduling decisions based on the two request queues, including designating each vehicle's communication mode (i.e., I2V or V2V), service role (i.e., sender or receiver), and the corresponding data item to be shared. To enable cooperative I2V and V2V communications, the system employs two SCHs and one CCH. The SCHs are used for I2V and V2V communication, and the CCH is exploited to transmit basic safety messages (BSMs) and auxiliary information for service scheduling.

A scheduling period is described as follows. First, each vehicle identifies its neighboring vehicles based on BSMs, which are broadcast periodically over the CCH. Second, each vehicle notifies the RSU of the ID information for cached and newly requested data items, as well as neighboring vehicles via probe vehicle data (PVD) messages, which are transmitted over the CCH via I2V communication. Third, the RSU makes scheduling decisions, which are appended in a WAVE service advertisement (WSA) message and broadcast to vehicles through the CCH via I2V communication. Fourth, each vehicle tunes to the corresponding SCH according to the scheduling result (i.e., the vehicle's communication mode), and then transmits or receives data items via I2V or V2V communication. Note that only one-hop V2V communication is considered in such a scheme.

We give an example to better illustrate the system characteristics and scheduling challenges. In Fig. 1 (a), the directional arrows indicate the data transmission using I2V or V2V communication. In Fig. 1 (b), the gray-shaded blocks represent the vehicles' cached data items. The non-shaded blocks represent the vehicles' pending requests. The dotted circles indicate the coverage of RSU. The dashed bidirectional arrows indicate that two vehicles can communicate via V2V communication. Assume that $V_9$-$V_{14}$ have not retrieved all requested data in the coverage of RSU $R_2$. Based on the received BSM and PVD messages, $R_2$ transmits the unserved requests to $R_1$. Then, to increase V2V data sharing opportunity outside RSU's coverage, $V_1$-$V_3$ need to cache the data items corresponding to transferred requests from $R_2$. Assume that the vehicles within the coverage of $R_1$ cannot retrieve their requested data items due to the mobility and limited I2V bandwidth, it is hard to schedule data services to satisfy both the requests from the vehicles and the transferred requests from $R_2$, simultaneously. In order to serve the outstanding requests and increase the V2V data sharing opportunity, simultaneously, one possible scheduling is illustrated as follows: $R_1$ designates the communication mode of $V_2$ and $V_4$ to I2V and broadcasts the data item '$a$'. Concurrently, $R_1$ schedules $V_6$ to transmit the data item '$d$' to $V_3$. Through the received data items, $V_2$ and $V_3$ not only receive data services and but also increase the opportunity of V2V data sharing out of $R_1$'s coverage. Clearly, it is non-trivial to search optimal solutions in real service scenarios with a large number of vehicles and requests.

For data service out of the RSU coverage, the system focuses on data sharing via V2V communication to serve vehicles driving in opposite directions. More efficient service could be achieved if the neighboring RSU could schedule some vehicles in its coverage area to cache some data items corresponding to the transferred requests. In Fig. 1 (a), assume that a) $V_7$ and $V_8$ were within $R_1$'s coverage, when $V_9$-$V_{11}$ left the $R_2$'s coverage. b) $V_8$ requested some data items including '$a$'. c) $V_9$ and $V_{11}$ are still requiring the data item '$a$'. Since the data service was scheduled based on the unserved requests of $V_9$-$V_{11}$, which had been transferred from $R_2$, $V_8$ retrieved the data item '$a$' by priority, through the centralized scheduling in $R_1$'s coverage. As shown in Fig. 1 (a), $V_8$ shares the data item '$a$' with $V_9$ and $V_{11}$ via V2V communication, after leaving $R_1$'s coverage. Again, in large-scale and complex traffic scenarios, it is challenging to have the best cooperation among RSUs and implement an efficient ad hoc data sharing to balance RSU workloads and maximize the spatial reusability of V2V communication.

TABLE I
NOTATIONS

| Notations | Description |
|---|---|
| $R_i$ | RSU $i$ |
| $s(\cdot)$ | Vehicle speed |
| $rV2V$ | V2V communication range |
| $dis(\cdot,\cdot)$ | Distance between two vehicles |
| $dir(\cdot)$ | Driving direction of a vehicle |
| $V(t)$ | Set of vehicles at time $t$ |
| $C(t)$ | Set of clusters at time $t$ |
| $V_C^n(t)$ | Set of vehicles not in any cluster |
| $V_{R_i}(t)$ | Set of vehicles within the coverage of $R_i$ |
| $N_{V_j}(t)$ | Set of $V_j$'s neighboring vehicles at time $t$ |
| $N_{V_j}^S(t)$ | Set of neighboring vehicles moving in the same direction as $V_j$ |
| $N_{V_j}^O(t)$ | Set of neighboring vehicles moving in the opposite direction as $V_j$ |
| $N_{V_j}^n(t)$ | Set of vehicles that are in $N_{V_j}^S(t)$ but not in any cluster |
| $D$ | Database |
| $D_{V_j}(t)$ | Set of cached data items of $V_j$ |
| $Q_{V_j}(t)$ | Set of $V_j$'s pending requests at time $t$ |
| $U_{R_i}(t)$ | Set of unserved requests in $R_i$'s coverage |
| $T_{R_i}^I(t)$ | Set of tentative services using I2V communication in $R_i$'s coverage |
| $T_{R_i}^V(t)$ | Set of tentative services using V2V communication in $R_i$'s coverage |
| $V_S$ | Sender vehicle in tentative service |
| $V_{R'}$ | Receiver vehicle in tentative service |
| $S'_{R_i}(t)$ | Set of candidate scheduling result in $R_i$'s coverage |
| $S_{R_i}^c(t)$ | Centralized scheduling of $R_i$ |
| $S_{C_k}^a(t)$ | Ad hoc scheduling of $C_k$ |
| $A_{V_j}(t)$ | Service ability of $V_j$ |

## IV. Hybrid Centralized and Ad Hoc Data Scheduling (HCA) Problem

The primary notations are summarized in Table I. A set of vehicles at time $t$ is denoted by $V(t) = \{V_1, V_2, \ldots, V_j, \ldots, V_{|V(t)|}\}$, where $j$ and $|V(t)|$ are vehicle ID and the number of vehicles, respectively. The speed of $V_j$ is denoted by $s(V_j)$. The set of clusters at time $t$ is denoted by $C(t) = \{C_1, C_2, \ldots, C_k, \ldots, C_{|C(t)|}\}$, where $k$ denotes cluster ID and $|C(t)|$ is the number of clusters. Each cluster consists of a $CH$ and $CMs$. A set of vehicles not included in any cluster is denoted by $V_C^n(t)$. The set of vehicles within RSU $i$ ($R_i$)'s coverage is denoted by $V_{R_i}(t)$.

The set of $V_j$'s neighboring vehicles at time $t$ is denoted by $N_{V_j}(t) = \{V_h \mid V_h \in V(t) \wedge dis(V_j, V_h) \leq rV2V\}$, where $V_h$ $(1 \leq h \leq |V(t)|)$ represents a neighboring vehicle of $V_j$ and $rV2V$ is the V2V communication range. $dis(\cdot,\cdot)$ is a distance between two vehicles. According to the driving direction of the neighboring vehicles, $N_{V_j}(t)$ is represented by $N_{V_j}(t) = \{N^S_{V_j}(t) \cup N^O_{V_j}(t)\}$, where $N^S_{V_j}(t)$ and $N^O_{V_j}(t)$ are the sets of neighboring vehicles moving in the same/opposite direction as $V_j$, respectively. $N^S_{V_j}(t)$ is expressed as $N^S_{V_j}(t) = \{V_h \mid V_h \in N_{V_j}(t) \wedge dir(V_h) = dir(V_j)\}$, where $dir(\cdot)$ denotes the driving direction of a vehicle. $N^O_{V_j}(t)$ is represented by $N^O_{V_j}(t) = \{V_h \mid V_h \in N_{V_j}(t) \wedge dir(V_h) \neq dir(V_j)\}$. The set of vehicles that are in $N^S_{V_j}(t)$ but not in any cluster is expressed as $N^n_{V_j}(t) = \{V_h \mid V_h \in N^S_{V_j}(t) \wedge V_h \in V^n_C(t)\}$.

The database is represented by $D = \{d_1, d_2, \ldots, d_{|D|}\}$, where $|D|$ denotes the number of data items (i.e., database size). Vehicles may request data items in the $D$ based on specific applications with different deadlines. The set of $V_j$'s pending requests at time $t$ is expressed as $Q_{V_j}(t) = \{q^1_{V_j}, q^2_{V_j}, \ldots, q^x_{V_j}, \ldots, q^{|Q_{V_j}(t)|}_{V_j}\}$, and each request corresponds to a data item in the $D$ (i.e., $q^x_{V_j} \in D$), where $x$ $(1 \leq x \leq |Q_{V_j}(t)|)$ is $V_j$'s $x$th pending request and $|Q_{V_j}(t)|$ is the number of pending requests. The set of cached data items of $V_j$ is expressed as $D_{V_j}(t) = \{d^1_{V_j}, d^2_{V_j}, \ldots, d^m_{V_j}, \ldots, d^{|D_{V_j(t)}|}_{V_j}\}$, where $m$ $(1 \leq m \leq |D_{V_j}(t)|)$ is $V_j$'s $m$th cached data item and $|D_{V_j}(t)|$ is the total number of cached data items. Each $d^m_{V_j} \in D$.

Within an RSU's coverage, pending requests can be served by the RSU via I2V communication or by a neighboring vehicle through V2V communication. Based on neighboring vehicles' cached data items, a data item corresponding to a request can be retrieved in multiple ways when multiple neighboring vehicles can provide the service. We refer to such viable data services via I2V or V2V communication as tentative services (TS). According to the communication mode (i.e., I2V or V2V communication), there are two types of TSs: $T^I_{R_i}(t)$ and $T^V_{R_i}(t)$. The set of TSs using I2V communication in $R_i$'s coverage is expressed as $T^I_{R_i}(t) = \{(R_i, q^x_{V_{R'}}, V_{R'}) \mid V_{R'} \in V_{R_i}(t) \wedge q^x_{V_{R'}} \in Q_{V_{R'}}(t)\}$, and the set of TSs using V2V communication in $R_i$'s coverage is expressed as $T^V_{R_i}(t) = \{(V_S, q^x_{V_{R'}}, V_{R'}) \mid V_S, V_{R'} \in V_{R_i}(t) \wedge q^x_{V_{R'}} \in Q_{V_{R'}}(t) \wedge V_S \in N_{V_{R'}}(t) \wedge q^x_{V_{R'}} \in D_{V_S}(t)\}$. Each TS is represented by a tuple comprising the sender $(TX)$, the data item to transmit, and the receiver $(V_{R'})$ in order, where $TX$ is either a sender vehicle $(V_S)$ or $R_i$. Considering hardware limitations and the characteristics of wireless communication (i.e., multiple data items cannot be transmitted or received simultaneously; a data item cannot be transmitted and received simultaneously; data collision can occur at the receiver due to the existence of multiple senders), each TS may have different incompatible TSs. Therefore, several candidate scheduling results can be generated based on the TSs. The set of candidate scheduling results within the coverage of $R_i$ is denoted by $S'_{R_i}(t) = \{s^1_{R_i}, s^2_{R_i}, \ldots, s^w_{R_i}, \ldots, s^{|S'_{R_i}(t)|}_{R_i}\}$, where $|S'_{R_i}(t)|$ is the number of candidate scheduling results. The $w$th $\left(1 \leq w \leq |S'_{R_i}(t)|\right)$ candidate scheduling result is represented by $s^w_{R_i} = \{(TX, q^x_{V_{R'}}, V_{R'}) \mid (TX, q^x_{V_{R'}}, V_{R'}) \in T^I_{R_i}(t) \cup T^V_{R_i}(t)\}$, where $TX$ is either $V_S$ or $R_i$. The set of unserved requests in $R_i$'s coverage is expressed as $U_{R_i}(t) = \bigcup_{V_j \in V_{R_i}(t') \wedge V_j \notin V_{R_i}(t)} Q_{V_j}(t)$, where $t'$ is the latest time in the coverage of $R_i$ before $V_j$ left the coverage. The unserved request information, i.e., the tuple $(V_j, q^x_{V_j}, s(V_j), C_k)$, is transferred to the RSU to which the vehicle is heading. Here, $C_k$ represents the cluster including $V_j$.

Given the above definitions, the HCA problem is formulated as follows. First, for centralized scheduling within an RSU's coverage area, HCA attempts to serve as many requests as possible in the coverage of RSUs via a hybrid of I2V and V2V communications. Besides, to maximize the chance of V2V data sharing in bidirectional road scenarios outside an RSU's coverage area, HCA also considers requests transferred from neighboring RSUs. The centralized scheduling of $R_i$ is expressed as follows:

$$S^c_{R_i}(t) = \max_{1 \leq w \leq |S'_{R_i}(t)|} \sum_{(TX, q^x_{V_{R'}}, V_{R'}) \in s^w_{R_i}} \Big\{ n\left((TX, q^x_{V_{R'}}, V_{R'})\right) + p\left(q^x_{V_{R'}} \mid q^x_{V_{R'}} \in U_{R_{i-1}}(t) \cup U_{R_{i+1}}(t)\right) \Big\}, \quad (1)$$

where $n(\cdot)$ indicates the number of TS receivers (i.e., $n(\cdot) = 1$); $p(\cdot)$ $(0 \leq p(\cdot) \leq 2)$ is the degree that a data item corresponding to $q^x_{V_{R'}}$ can be shared via ad hoc scheduling out of the RSU coverage; $R_{i-1}$ and $R_{i+1}$ are the peer RSUs of $R_i$.

Second, for ad hoc scheduling out of RSU coverage, HCA attempts to maximize V2V data sharing by constructing clusters dynamically, selecting CHs, and determining the sender vehicles and data items for broadcasting in each cluster. Therefore, ad hoc scheduling is expressed as follows:

$$S^a_{C_k}(t) = \max_{V_j \in C_k} A_{V_j}(t), \quad (2)$$

where $A_{V_j}(t)$ denotes the service ability of $V_j$ in $C_k$. In other words, $A_{V_j}(t)$ indicates the maximum number of target receivers among the vehicles driving in the opposite direction. Based on (1) and (2), to maximize the complete set of data service opportunities in bidirectional road scenarios, the overall objective of HCA $(HCA(t))$ is expressed as follows.

$$HCA(t) = \sum_{\forall R_i} S^c_{R_i}(t) + \sum_{\forall C_k} S^a_{C_k}(t), \quad (3)$$

## V. RSU Cooperation-Based Adaptive Scheduling (RCAS) Algorithm

In this section, we propose an online scheduling algorithm called RCAS. Fig. 2. shows the flow chart of the overall framework of RCAS, which includes both centralized scheduling and ad hoc scheduling mechanisms depending on vehicle locations, as well as a mechanism for clustering mechanism. Within the RSU's coverage, data items are disseminated using cooperative I2V and V2V communications based on centralized scheduling. In contrast, when vehicles are out of RSUs'

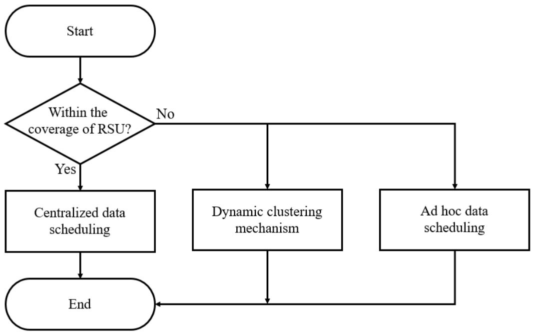



Fig. 2. Overall framework of RCAS.

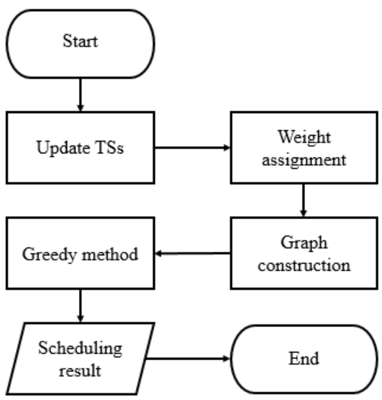



Fig. 3. Flow chart of the centralized data scheduling.

coverage, the ad hoc scheduling is exploited to share cached data items via V2V based on the dynamic construction of clusters.

### A. Centralized Data Scheduling

The flow chart of the centralized data scheduling is shown in Fig. 3. RSUs update TSs based on the vehicle information within their coverage. After that, they assign weight values to the TSs. We construct an undirected graph $G$ as shown in Fig. 4. Each vertex denotes a TS and each edge means that the connected vertices cannot be implemented simultaneously due to hardware limitations or wireless communication constraints. We transform the problem to MWIS, which is a well-known NP-hard problem [27], and a greedy algorithm is adopted to search viable solutions. Details of the centralized scheduling algorithm are described below.

*1) Weight Assignment:* Due to highly dynamic and bias of service workload distribution among different RSUs, many requests are expected to be served via V2V data sharing outside the RSU's coverage based on the cached contents of peer vehicles. As described, we transform the scheduling problem to the MWIS problem. With that, to improve data services within the coverage of RSU and increase the V2V data sharing opportunity simultaneously, it is critical to assign a proper weight to each TS. To this end, the following factors are considered for the weight assignment.

- Service within RSU coverage ($sw$)
  $sw(TS)$ represents how many vehicles can receive a data item with the given TS. According to the definition, the TS means one data service for one receiver via either I2V or V2V communication, and hence, we have:

$$sw(TS) = 1. \tag{4}$$

- Influence on ad hoc data scheduling ($ia$)
  $ia(TS)$ is the influence of a TS on ad hoc scheduling out of RSUs' coverage. Clearly, a data item has higher influence if it has higher potential to serve transferred requests via V2V communication out of RSUs' coverage. The set of clusters approaching $R_i$ with the opposite direction with the receiver in TS is indicated by $C^O_{R_i}(V_{R'})$. The set of unserved requests of $C_k$ in $C^O_{R_i}(V_{R'})$, which are transferred to $R_i$, is denoted by $U_{C_k}(t) = \{q^x_{V_j} \mid q^x_{V_j} \in U_{R_{i-1}}(t) \cup U_{R_{i+1}}(t) \wedge V_j \in C_k \wedge C_k \in C^O_{R_i}(V_{R'})\}$, where $q^x_{V_j}$ is $V_j$'s $x$th pending request. The frequency of a request in $U_{C_k}(t)$ is denoted by $fr(request\ ID,\ C_k)$. To assign the highest weight to the TS that is most influential in ad hoc data scheduling, $ia(TS)$ is expressed as follows:

$$ia(TS) = 1 - \sum_{\substack{k\ s.t.\\ C_k \in C^O_{R_i}(V_{R'})}} \sum_{\substack{j\ s.t.\\ V_j \in C_k}} \sum_{\substack{x\ s.t.\\ q^x_{V_j} \in U_{C_k}(t) \wedge\\ fr(q_{V_{R'}}, C_k) < fr(q^x_{V_j}, C_k)}} \frac{fr(q^x_{V_j}, C_k)}{|U_{C_k}(t)||C^O_{R_i}(V_{R'})|}, \tag{5}$$

  where $q_{V_{R'}}$ is the required data item ID by $V_{R'}$ in the TS. Note that $0 \le ia(TS) \le 1$. With (5), we find the probabilities by which the other data items with greater frequency than $q_{V_{R'}}$ are selected for data services relative to each cluster approaching the RSU first. Then, we calculate the mean of the probabilities to find the mean transmission probability of $q_{V_{R'}}$. Here, a greater $ia(TS)$ value means that $q_{V_{R'}}$ has higher probability to provide data services to the clusters.
- Expected connection time ($ct$)
  For success data sharing via V2V communication, sufficient connection time is required between two vehicles. Therefore, it is effective to select the TS that has a long expected connection time between a sender (i.e., the receiver of the TS) and a tentative receiver (i.e., a vehicle in a cluster approaching the RSU). To this end, we calculate the expected relative connection time between a receiver of the TS and vehicles requiring $q_{V_{R'}}$ in each cluster.
  The set of expected connection times between a receiver of TS ($V_{R'}$) in $R_i$'s coverage and each vehicle in a cluster ($C_k$) approaching $R_i$ is expressed as:

$$E_{C_k} = \left\{ e(V_{R'}, V_j) \mid e(V_{R'}, V_j) = \frac{2rV2V}{s(V_{R'}) + s(V_j)}, V_j \in C_k \wedge C_k \in C^O_{R_i}(V_{R'}) \right\}, \tag{6}$$

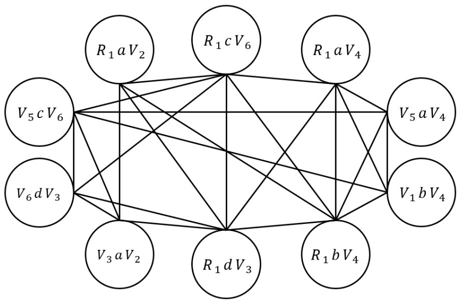



Fig. 4. Example of constructed graph based on Fig. 1 (b).

where $V_j$ indicates a vehicle in $C_k$. The set of expected connection times between $V_{R'}$ and the vehicle that requires the same data service in $C_k$ is expressed as

$$E_{C_k}^{V_{R'}} = \{e(V_{R'}, V_j) \mid e(V_{R'}, V_j) \in E_{C_k} \wedge q_{V_{R'}} \in Q_{V_j}(t) \wedge V_j \in C_k\}. \quad (7)$$

Therefore, $ct(TS)$ is expressed as:

$$ct(TS) = \sum_{\substack{k \ s.t. \\ C_k \in C_{R_i}^O(V_{R'})}} \frac{\overline{E_{C_k}^{V_{R'}}}}{\max\left(E_{C_k}\right) |C_{R_i}^O(V_{R'})|}, \quad (8)$$

where $\overline{E_{C_k}^{V_{R'}}}$ is average value of set $E_{C_k}^{V_{R'}}$. Note that $ct(TS)$ is $0 \le ct(TS) \le 1$ and a greater $ct(TS)$ value indicates that the connection time between vehicles is more stable. Here, the latest speed information is used by the RSU to calculate the connection time.

Based on the above three factors, the weight TS ($W(TS)$) is expressed as:

$$W(TS) = sw(TS) + ia(TS) + ct(TS). \quad (9)$$

*2) Graph Construction:* We represent the TSs via vertices with the identifiers of the sender, the data item to transmit, and the receiver shown in Fig. 4. Then, we connect those TSs which cannot be implemented simultaneously, which are considered as conflicting TSs. Conflicting TSs are caused by hardware limitations and communication constraints. Specifically, OBUs and RSUs can only transmit a single data item at a time, and they cannot send and receive data items simultaneously. Moreover, data collision can occur at a receiver when it receives data items from multiple senders simultaneously through the same channel. To give clearer illustration, we define conflicting TSs based on the example shown in Fig. 1 (b).

- Two TSs are in conflict if they represent I2V communication because an RSU can broadcast only one data item at a time (e.g., $R_1aV_2$ and $R_1dV_3$ are in conflict).
- Two TSs are in conflict if the receiver of the TS using I2V communication is also assigned as either the sender or receiver of the TS using V2V communication, because an OBU cannot receive data services from multiple senders simultaneously (e.g., $R_1aV_4$ and $V_1bV_4$ are in conflict).
- Two TSs are in conflict if two senders are the same and they are assigned to transmit different data items, because OBUs can only transmit a single data item at a time (e.g., $V_5cV_6$ and $V_5aV_4$ are in conflict).
- Two TSs are in conflict if a sender of one TS is a receiver of the other TS, because OBUs cannot be a sender and receiver simultaneously (e.g., $V_6dV_3$ and $V_3aV_2$ are in conflict).
- Two TSs are in conflict if a receiver of one TS is a neighboring vehicle of a sender of the other TS, because data collision will occur at the receiver (e.g., $V_5cV_6$ and $V_1bV_4$ are in conflict).

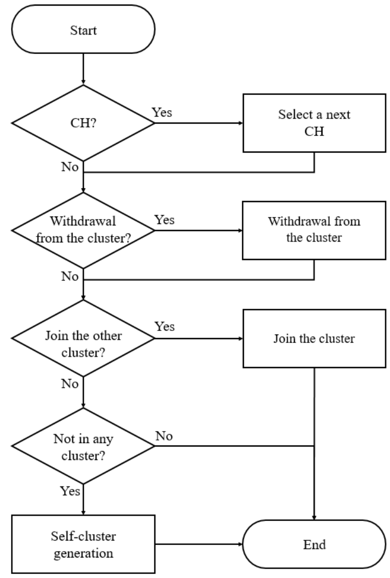



Fig. 5. Flow chart of the dynamic clustering mechanism.

*3) Greedy Method:* With the construction of $G$, we adopt a greedy method [28] to solve the MWIS approximately. According to the selected TSs, data services are implemented using a hybrid of I2V and V2V communications. The main procedures of the adopted greedy method are outlined as follows. First, it calculates the $W(TS)/(d(TS)+1)$ value of each vertex, where $d(TS)$ means the degree of vertex indicating $TS$. Second, it selects the vertex with the maximum $W(TS)/(d(TS)+1)$ value. Third, it removes the selected vertex, the vertices connected to the selected vertex, and the edges of the removed vertices. Fourth, it repeats the above procedures in the remaining $G$ until no vertex remains.

### B. Dynamic Clustering Mechanism

A dynamic clustering mechanism is designed for supporting ad hoc data sharing out of RSUs' coverage. As shown in the flow chart of Fig. 5, CH selects the next CH, and then, vehicles decide whether to withdraw from the cluster. For ad hoc scheduling, the vehicles need to be affiliated to clusters. Therefore, the vehicles determine whether to join the neighboring cluster, after the withdrawal decision. For vehicles to join a cluster, they follow join procedures to be elaborated below. Otherwise, the vehicles check whether they are included in a cluster. If they are CH or CM of a cluster, they remain in the cluster. Otherwise, the vehicles create a new cluster by themselves. The detailed CH designation and cluster management consist of four phases:

- Phase 1: CMs broadcast the cluster management information to the CH, including ID of the affiliated cluster, location, and speed.
- Phase 2: based on the received message from the CMs, the CH updates the number of CMs, the average speed of the cluster, and the locations of front-most ($V_f$) and rear-most ($V_r$) vehicles in the cluster. Then, the CH broadcasts the updated control message.
- Phase 3: the CH and the CMs calculate their weights for CH designation based on the received message in the second phase. After that, CMs broadcast the control message including the weight values.
- Phase 4: the CH selects the vehicle with the highest weight as the next CH. Then, the CH announces the next CH through the control message to the CMs.

Note that the communication overhead will not be the hurdle of cluster management and ad hoc scheduling because DSRC supports 6∼27 Mbps data transmission rate and the size of control message is in the order of hundred bytes.

To calculate the weight for CH designation, we consider the following three factors: speed difference between the average speed of the cluster and a vehicle ($ws(V_j)$), the vehicle's location in the cluster ($wc(V_j)$), and the number of neighboring vehicles in the same cluster ($wn(V_j)$). The weight of the vehicle $V_j$ is calculated by $W(V_j) = ws(V_j) + wc(V_j) + wn(V_j)$.

- To prevent frequent CH changes, we assign a higher weight to a vehicle whose speed is closer to the average speed of the cluster ($as(C_k)$), where the vehicle is in $C_k$. The weight is denoted by $ws(V_j)$, which is computed by

$$ws(V_j) = max\left(0, 1 - \left|\frac{as(C_k) - s(V_j)}{as(C_k)}\right|\right), \quad (10)$$

  where $V_j \in C_k$ and $0 \le ws(V_j) \le 1$.
- To further enhance cluster stability, we assign a higher weight to a vehicle which is closer to the center point of the cluster. The weight is denoted by $wc(V_j)'$ and it is computed by

$$\begin{aligned} wc(V_j)' &= \frac{1}{\frac{1}{dis(V_f,V_j)} + \frac{1}{dis(V_r,V_j)}} \\ max\left(wc(V_j)'\right) &= \frac{1}{\frac{2}{dis(V_f,V_r)} + \frac{2}{dis(V_f,V_r)}} \\ wc(V_j) &= \frac{wc(V_j)'}{max\left(wc(V_j)'\right)} \end{aligned} \quad (11)$$

  where $V_j, V_f, V_r \in C_k$. Since $wc(V_j)' \ge 0$, $wc(V_j)'$ is nomalized to the range [0,1] using $max\left(wc(V_j)'\right)$, which is the maximum value of $wc(V_j)'$ when the vehicle is at the center of cluster. The normalized value is represented by $wc(V_j)$ as shown in (11).
- To maintain cluster size, i.e., the number of vehicles in the cluster, we assign a higher weight to a vehicle that can cover more vehicles in the same cluster. In other words, the next CH must cover the most CMs. Therefore, $wn(V_j)$ is expressed as:

$$wn(V_j) = \frac{\left|N_{V_j}^{C_k}(t)\right|}{|C_k|}, \quad (12)$$

  where $0 \le wn(V_j) \le 1$ and $N_{V_j}^{C_k}(t) = \{V_h \mid V_j, V_h \in C_k \wedge V_h \in N_{V_j}(t)\}$, which indicates the set of $V_j$'s neighboring vehicles in the cluster $C_k$.

Based on the $W(V_j)$ received from the CMs, the CH selects the next CH with the highest weight, and the other vehicles become CMs. In the following, we define the cluster management schemes with three possible options.

- Withdrawal from the cluster: it occurs when a CM is no longer within the CH's coverage. Namely, when $V_j$ is a CM of $C_k$ but cannot communicate with CH of $C_k$ ($CH_k$) via one-hop V2V communication. Therefore, withdrawal from a cluster is implemented when the CM satisfies $dis(CH_k, V_j) > rV2V$, where $dis(CH_k, V_j)$ is the distance between the CH and the given CM (i.e., $V_j$).
- Joining a cluster: it is implemented when a vehicle is within the coverage of another cluster's CH. There are two cases according to the state of the vehicle: a) when a vehicle not in any cluster attempts to join the nearest cluster. Assuming that $V_j$ is not in a cluster and the nearest cluster is $C_k$, $V_j$ must be within $CH_k$'s coverage to join that cluster. In other words, $V_j$ must satisfy $dis(CH_k, V_j) \le rV2V$ to join $C_k$. b) when a CM is within the coverage of another cluster's CH. To prevent frequent cluster changes, joining a cluster is only allowed when another cluster is larger than the currently joined cluster. Assuming that $V_j$ that is a CM of $C_k$ is within the coverage of the $CH_{k+1}$ of $C_{k+1}$, the vehicle must satisfy the condition $dis(V_j, CH_{k+1}) \le rV2V \wedge |C_k| < |C_{k+1}|$ to join cluster $C_{k+1}$. Note that a CH can also join a neighboring cluster. In this case, the CH disbands the cluster through a control message before joining the other cluster. The disbanded CMs attempt to join the neighboring cluster or generate a new cluster if there are no neighboring clusters within their coverage.
- Self-cluster generation: it is executed when a vehicle that cannot be a CM of any cluster. If multiple vehicles are not included in any cluster, the vehicle with the highest weight becomes the CH. To this end, vehicles broadcast control messages and calculate their weight based on the received information. Specifically, if $V_j$ satisfies the condition, $ws(V_j)$ is computed with the average speed of $N_{V_j}^{n}(t)$. For $wc(V_j)$, the locations of $V_f$ and $V_r$ of $N_{V_j}^{n}(t)$ are used, and $wn(V_j)$ is calculated as $|N_{V_j}^{n}(t)|/|N_{V_j}^{S}(t)|$, where $|N_{V_j}^{S}(t)|$ is the number of vehicles moving in the same direction in $N_{V_j}(t)$. After weight calculation, the vehicles broadcast their weight value. Then, the vehicle with the highest weight among its neighboring vehicles becomes the CH and announces its role in the cluster and its new cluster ID through a control message.

TABLE II
SIMULATION STATISTICS UNDER DIFFERENT TRAFFIC SCENARIOS

| Traffic Volume Level | Mean Arrival Rate (vehicles/h) | | | Mean Speed (km/h) | | | Mean Density (vehicles/km) | | |
|---|---|---|---|---|---|---|---|---|---|
| | First Lane | Second Lane | Third Lane | First Lane | Second Lane | Third Lane | First Lane | Second Lane | Third Lane |
| 1 | 1000 | 900 | 800 | 109.31 | 100.26 | 91.23 | 8.91 | 8.85 | 8.77 |
| 2 | 1500 | 1400 | 1300 | 102.42 | 93.60 | 84.84 | 14.64 | 14.91 | 15.15 |
| 3 | 2000 | 1900 | 1800 | 94.36 | 86.27 | 77.04 | 21.36 | 21.57 | 22.95 |
| 4 | 2500 | 2400 | 2300 | 84.75 | 76.08 | 65.46 | 29.37 | 30.83 | 34.54 |
| 5 | 3000 | 2900 | 2800 | 70.72 | 53.18 | 42.03 | 41.06 | 51.65 | 57.96 |

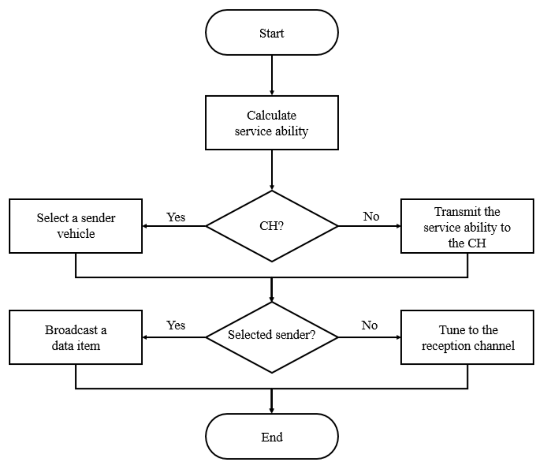



Fig. 6. Flow chart of the ad hoc data scheduling.

### C. Ad hoc Data Scheduling via V2V Communication

The flow chart of ad hoc data scheduling is shown in Fig. 6. As shown, vehicles calculate their service abilities first. Then, CMs transmit their service ability values to the CH. Based on the received information, the CH selects a sender vehicle. The selected vehicle broadcasts a data item, and the other vehicles tune to the reception channel to receive data services. The detailed description of ad hoc data scheduling is described as follows.

- First, CMs advertise their outstanding requests.
- Second, a CH advertises its unserved requests.
- Third, the CMs and the CH calculate their service ability based on their cached data items and the unserved requests of neighboring vehicles driving in the opposite direction, and then the CMs broadcast their service ability.
- Fourth, the CH designates a sender vehicle based on the received information. The CH assigns the ID of the sender vehicle, which has the highest service ability in the cluster.
- Fifth, the designated sender vehicle searches its cached data items and broadcasts the one with most pending requests.

Note that from Step 1 to 4, the parameters are included in the control message. The computation of the service ability is described as follows.

Assume that $V_j$ is in $C_k$, and its neighboring vehicles driving in the opposite direction is denoted by $N^O_{V_j}(t)$. The set of vehicles in $N^O_{V_j}(t)$ requiring $d^m_{V_j}$ is expressed as $RV(d^m_{V_j}) = \{V_h \mid V_h \in N^O_{V_j}(t) \wedge d^m_{V_j} \in Q_{V_h}(t)\}$. Then, the service ability of a vehicle is determined according to the cached data item corresponding to the data item most requested by vehicles in $N^O_{V_j}(t)$, which is expressed as $A_{V_j}(t) = \max_{1 \le m \le |D_{V_j}(t)|}(|RV(d^m_{V_j})|)$, where $|RV(d^m_{V_j})|$ is the number of vehicles in $RV(d^m_{V_j})$.

Based on the above description, we analyze the complexity of the ad hoc mode as follows. First, when calculating the service ability, the vehicles need to compare their cached data items and the unserved requests of neighboring vehicles, and the complexity is $O\left(|N^O_{V_j}(t)| \cdot |D_{V_j}(t)| \cdot |Q_{V_h}(t)|\right)$, where $V_h \in N^O_{V_j}(t)$. Second, when designating a sender vehicle, the CH scans the service ability values of the CMs, and the complexity is $O(|C_k|)$. Therefore, the overall complexity is $O\left(|N^O_{V_j}(t)| \cdot |D_{V_j}(t)| \cdot |Q_{V_h}(t)| + |C_k|\right)$. Since the number of cached data items (i.e., $|D_{V_j}(t)|$) and the number of outstanding requests (i.e., $|Q_{V_h}(t)|$) are bounded by the size of database $|D|$, the complexity of the ad hoc mode can be expressed as $O\left(|N^O_{V_j}(t)| \cdot |D|^2 + |C_k|\right)$. Note that in practice, $|N^O_{V_j}(t)|$ and $|C_k|$ are typically very small due to the high mobility of vehicles and the short V2V communication range. Hence, the computational overhead of ad hoc mode will not a hindrance in the feasibility and the scalability.

## VI. PERFORMANCE EVALUATION

### A. Setup

We build the simulation model with CSIM 19 [29] based on the system architecture described in Section III. Specifically, there are five RSUs at intervals of 3 km on a bidirectional straight road (three lanes in each direction). The arrival pattern of vehicles follows the Poisson process. The default vehicle arrival rate is set to traffic volume level 3 described in Table II. The traffic flow is simulated based on the Greenshield's model, which is commonly adopted when simulating macroscopic traffic scenarios [30]. Accordingly, vehicle speed ($\upsilon$) is derived by $\upsilon = V^f - \frac{V^f}{K^j} \cdot k$, where $V^f$ is the free flow speed (i.e., the maximum speed limit) and $K^j$ is the jam density (i.e., the density leading to zero velocity of the vehicles). $k$ is the current density of the vehicles in each lane. For each driving direction, the free flow speeds of the three lanes are set to $\upsilon^f_1 =$ 120 km/h, $\upsilon^f_2 = 110$ km/h, and $\upsilon^f_3 = 100$ km/h, respectively. $K^j$ is set to 100 vehicles/km. The communication ranges of the RSU and OBU are set to 300m and 150m, respectively. The scheduling period is 1 s. Each vehicle generates one to seven requests within each RSU's coverage area, and the data

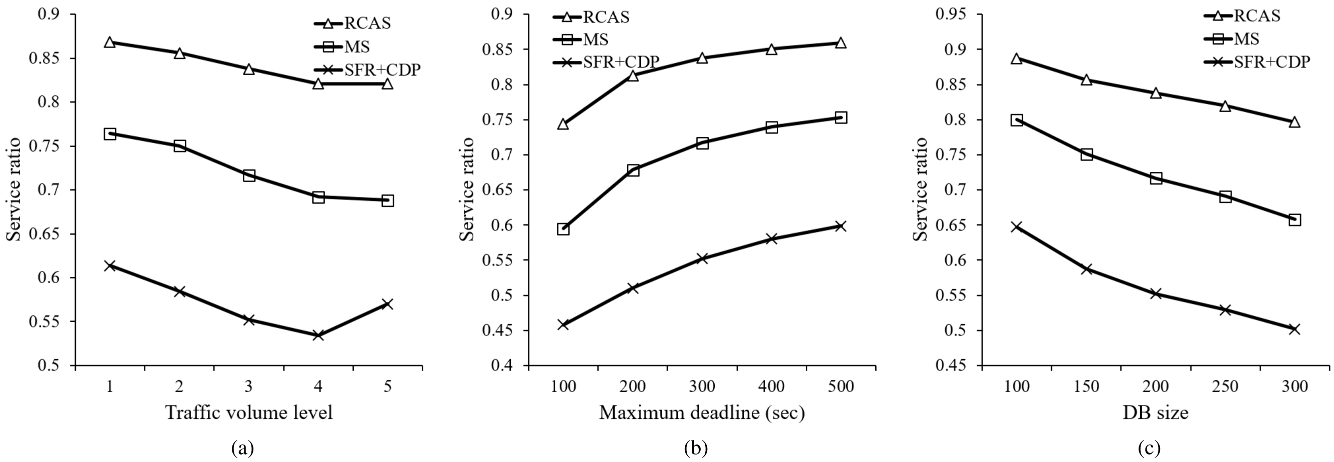



Fig. 7. Overall service ratios under different traffic workloads (a), maximum deadline (b), and DB size (c).

access pattern follows the Zipf distribution [31]. Specifically, for $d_g$ ($d_g \in D$ where $1 \le g \le |D|$), the access probability of $d_g$ is computed as $\frac{\left(\frac{1}{g}\right)^{\theta}}{\sum_{u=1}^{|D|}\left(\frac{1}{u}\right)^{\theta}}$, where $|D|$ is the database (DB) size. As the default settings, DB size and parameter $\theta$ are set to 200 data items and 0.6, respectively.

We simulate various traffic volume levels to evaluate algorithm performance under different traffic workload environments. The detailed statistics are summarized in Table II. Also, we evaluate the algorithm performance under various deadline ranges and DB sizes. As the default settings, the minimum and maximum deadlines are set to 10s and 300s, where the deadline of each request is distributed uniformly.

We compare the proposed algorithm to MS [16] and a combined SFR [15] and the Cooperative Downloading Protocol (CDP) [11] algorithm. MS is the most comparable algorithm to our problem in terms of service strategy. SFR is the closest solution in the literature to our centralized data scheduling, but it does not consider the V2V data sharing out of RSUs' coverage. For a fair comparison, we combine SFR with CDP.

We evaluate the performance of the proposed algorithm using the following metrics.

- Overall service ratio: it is calculated by the total number of served requests over the total number of submitted requests. The overall service ratio is the primary metric for evaluating system performance.
- Service ratio by V2V communication: it is calculated by the number of requests served via V2V communication over the total number of submitted requests. This metric evaluates algorithm performance relative to the benefits of V2V communication.
- Average RSU workload: it is the average number of pending requests in an RSU in each scheduling period, which is calculated as $\sum_{R_i \in \forall RSU} \sum_{t=0}^{T} \frac{|Q_{R_i}(t)|}{N(RSU) \cdot T}$, where $|Q_{R_i}(t)|$ is the number of pending requests in every time $t$ at $R_i$, $N(RSU)$ is the number of RSUs, and $T$ represents the simulation time. This metric evaluates the effectiveness of the algorithm in terms of offloading an RSU's workload.
- Data collision ratio: it is the ratio of V2V data sharing out of the RSU coverage, which is calculated as $\frac{DC}{\sum_{\forall SV} A_{SV}}$, where $SV$ is a sender vehicle out of RSU coverage, $A_{SV}$ is the service ability of $SV$, and $DC$ is the number of data collisions at receivers.

## B. Simulation Results

Fig. 7 (a) shows the overall service ratios under different traffic workloads. A greater volume level means a higher vehicle density. As shown, RCAS has a higher service ratio than MS and SFR+CDP. The service ratios of the three algorithms decrease until level 4 because the traffic workloads have increased such that the vehicles cannot retrieve the required data items at relatively high speed within RSU coverage. After level 4, the service ratio of SFR+CDP increases because the vehicles stay within RSU coverage for sufficient time to retrieve the required data items. In contrast, the service ratios of RCAS and MS decrease slightly because V2V data sharing decreases when out of RSU coverage due to increased interference, as shown in Fig. 9 (a). By comparing the gradient of each algorithm, it can be seen that the proposed RCAS has a stable service ratio for each traffic volume level, which means it is less affected by the volume of traffic. Fig. 7 (b) and (c) show the overall service ratios under various deadline ranges and DB sizes, respectively. The service ratios of the three algorithms are proportional to the maximum deadline. However, they are inversely proportional to DB size because the opportunities for V2V data services decrease. In Fig. 7 (c), the proposed RCAS has a lesser gradient than the other algorithms, which means it is more adaptive to DB size (i.e., diversity of service applications). Note that none of the algorithms can guarantee the timeliness and completeness of data service due to various service workloads, fast vehicle mobility, and different request deadlines.

Fig. 8 (a), (b), and (c) show the service ratios by V2V communication of RCAS, MS, and SFR+CDP, respectively. The three algorithms have similar performance within

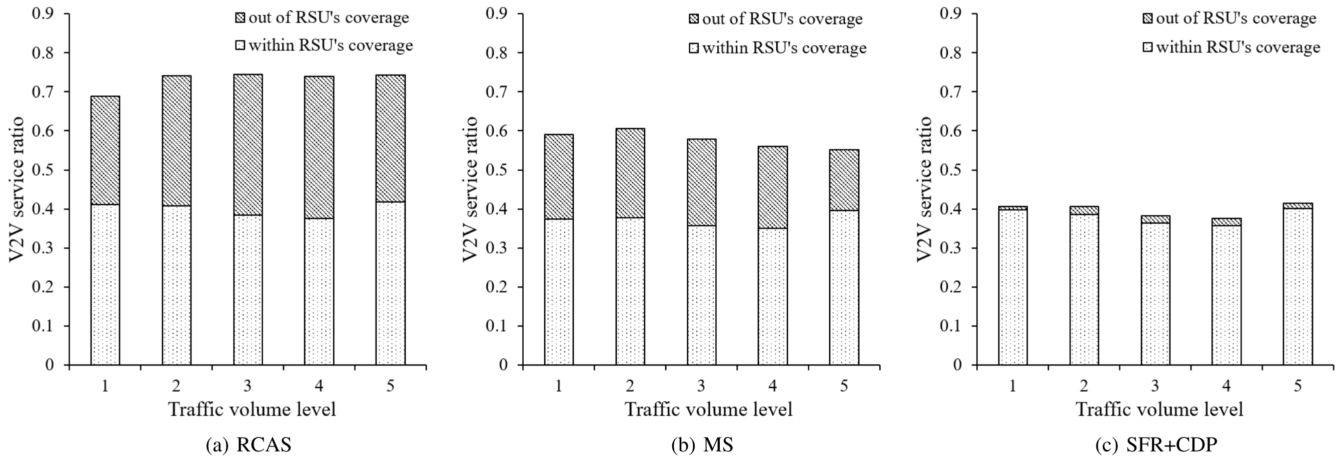



Fig. 8. Service ratio by V2V communication under different traffic workloads.

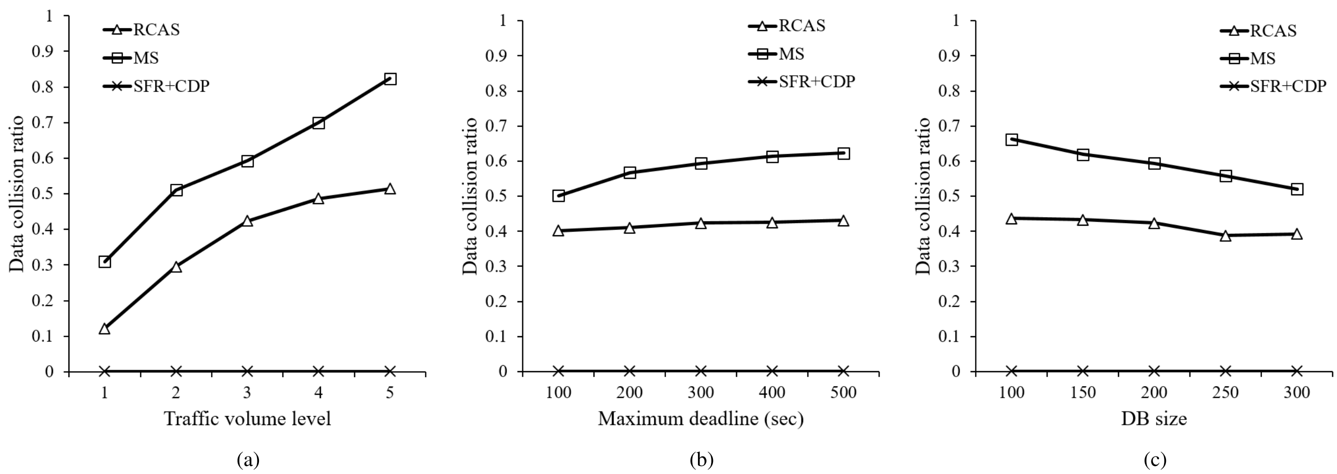



Fig. 9. Data collision ratio under different traffic workloads (a), maximum deadline (b), and DB size (c).

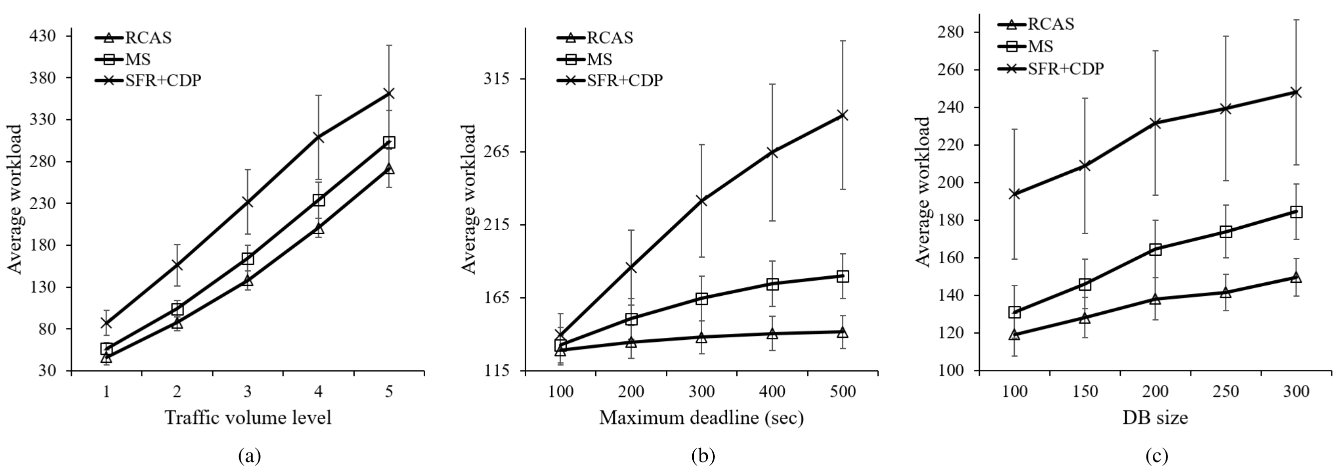



Fig. 10. Average workload at RSU under different traffic workloads (a), maximum deadline (b), and DB size (c).

RSU coverage. However, RCAS has the best V2V data sharing performance out of RSU coverage due to its ad hoc data scheduling policy. Therefore, it can better exploit V2V communication relative to offloading an RSU's workload.

Fig. 9 shows the data collision ratios of RCAS and MS. Note that there is no data collision for SFR+CDP because SFR and CDP are designed to avoid data collision. As noted from Figs. 9 (a) and (b), the data collision ratio of both RCAS and MS are proportional to the traffic workload and deadline range, because the opportunities of V2V data sharing increase with increasing vehicle density and request deadline. Note that RCAS is less affected by traffic workloads and the maximum deadline because the proposed dynamic clustering mechanism reduces data collisions. In Fig. 9 (c), the collision ratio of both RCAS and MS are inversely proportional to DB size due to the decreased opportunities for V2V data sharing because it is difficult for vehicles to find neighboring vehicles that have cached the data items corresponding to their pending requests. Here, even though it has a higher service rate (Fig. 7 (c)), RCAS is less affected by DB size due to the proposed ad hoc scheduling mechanism.

Fig. 10 shows the average and the standard deviation of RSUs' workloads. The standard deviations are represented by error bars in the figure. As shown, RCAS has the lowest average workload under all the scenarios. The average workload of each algorithm increases with an increasing traffic workload, as shown in Fig. 10 (a). In Figs. 7 (b) and (c), the average workload of RCAS increases slowly with an increase of maximum deadline and DB size. When the maximum deadline is 100s, RCAS has a similar average workload with MS. This is because most of the unserved requests expire before the requester vehicles reach the coverage of next RSU. In terms of balancing the workloads, RCAS has the smallest standard deviation value, in general. Therefore, it demonstrates that RCAS is effective in terms of adaptively alleviating and balancing RSUs' workloads.

## VII. Conclusion

In this paper, we presented a data dissemination system in bidirectional road scenarios based on I2V/V2V vehicular communications. Specifically, we formulated a hybrid of centralized and ad hoc data scheduling (HCA) problem, which aims to better exploit the synergetic effects between the centralized scheduling within the coverage of RSUs and the ad hoc data scheduling out of the RSUs coverage. To solve this problem, we proposed an RSU cooperation-based adaptive scheduling (RCAS) algorithm that consists of three mechanisms: centralized scheduling, ad hoc data scheduling, and cluster management. Finally, we developed a simulation model and gave a comprehensive performance, which demonstrated the superiority of our proposed algorithm in terms of improving service ratio and alleviating and balancing the workloads of RSUs.

In the future work, the timing constraint of services will be further considered for real-time services. Also, multi-hop V2V communications will be investigated in such a bidirectional road scenario.

## References

[1] K. Dar, M. Bakhouya, J. Gaber, M. Wack, and P. Lorenz, "Wireless communication technologies for its applications [topics in automotive networking]," *IEEE Commun. Mag.*, vol. 48, no. 5, pp. 156–162, May 2010.

[2] X. Cheng, R. Zhang, S. Chen, J. Li, L. Yang, and H. Zhang, "5G enabled vehicular communications and networking," *China Commun.*, vol. 15, no. 7, pp. 3–6, 2018.

[3] X. Cheng, C. Chen, W. Zhang, and Y. Yang, "5G-enabled cooperative intelligent vehicular (5GenCIV) framework: When Benz meets Marconi," *IEEE Intell. Syst.*, vol. 32, no. 3, pp. 53–59, May/Jun. 2017.

[4] S. Bharati and W. Zhuang, "CAH-MAC: Cooperative adhoc MAC for vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 470–479, Sep. 2013.

[5] D. N. M. Dang, H. N. Dang, V. Nguyen, Z. Htike, and C. S. Hong, "HER-MAC: A hybrid efficient and reliable MAC for vehicular ad hoc networks," in *Proc. IEEE 28th Int. Conf. Adv. Inf. Netw. Appl.*, May 2014, pp. 186–193.

[6] X. Jiang and D. H. C. Du, "PTMAC: A prediction-based TDMA MAC protocol for reducing packet collisions in VANET," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 9209–9223, Nov. 2016.

[7] X. Cheng, R. Zhang, and L. Yang, "Wireless toward the era of intelligent vehicles," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 188–202, Feb. 2019.

[8] C. Wu, S. Ohzahata, Y. Ji, and T. Kato, "Joint fuzzy relays and network-coding-based forwarding for multihop broadcasting in VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1415–1427, Jun. 2015.

[9] O. Rehman, M. Ould-Khaoua, and H. Bourdoucen, "An adaptive relay nodes selection scheme for multi-hop broadcast in vanets," *Comput. Commun.*, vol. 87, pp. 76–90, Aug. 2016.

[10] Y. Bi, H. Shan, X. S. Shen, N. Wang, and H. Zhao, "A multi-hop broadcast protocol for emergency message dissemination in urban vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 736–750, Mar. 2016.

[11] Y. Hao, J. Tang, and Y. Cheng, "Secure cooperative data downloading in vehicular ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 523–537, Sep. 2013.

[12] M. A. Rahman, G. G. M. N. Ali, P. H. J. Chong, S. K. Samantha, M. F. Muntasir, and C. Chen, "On scheduling real-time multi-item query with network coding in multi-RSU vehicular networks," in *Proc. IEEE 22nd Int. Conf. Embedded Real-Time Comput. Syst. Appl. (RTCSA)*, Aug. 2016, pp. 218–227.

[13] K. Liu, V. C. S. Lee, J. K. Ng, J. Chen, and S. H. Son, "Temporal data dissemination in vehicular cyber–physical systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2419–2431, Dec. 2014.

[14] S. Cao and V. C. S. Lee, "A novel adaptive TDMA-based MAC protocol for VANETs," *IEEE Commun. Lett.*, vol. 22, no. 3, pp. 614–617, Mar. 2018.

[15] K. Liu, J. K. Y. Ng, V. C. S. Lee, W. Wu, and S. H. Son, "Towards scalable, fair and robust data dissemination via cooperative vehicular communications," in *Proc. IEEE 20th Int. Conf. Embedded Real-Time Comput. Syst. Appl.*, Aug. 2014, pp. 1–9.

[16] B. Ko, K. Liu, and S. H. Son, "Towards efficient data services in vehicular networks via cooperative infrastructure-to-vehicle and vehicle-to-vehicle communications," in *Proc. Int. IEEE Conf. Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People, Smart World Congr. (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, Jul. 2016, pp. 82–89.

[17] X. Cheng, L. Yang, and X. Shen, "D2D for intelligent transportation systems: A feasibility study," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1784–1793, Aug. 2015.

[18] K. Liu, L. Feng, P. Dai, V. C. Lee, S. H. Son, and J. Cao, "Coding-assisted broadcast scheduling via memetic computing in SDN-based vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2420–2431, Aug. 2018.

[19] T. G. McGiffen, S. Beiker, and A. Paulraj, "Motivating network deployment: Vehicular communications," *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 22–33, Sep. 2017.

[20] H. Yang, Z. Jia, and G. Xie, "Delay-bounded and cost-limited RSU deployment in urban vehicular ad hoc networks," *Sensors*, vol. 18, no. 9, p. 2764, 2018.

[21] Z. Gao, D. Chen, S. Cai, and H.-C. Wu, "Optdynlim: An optimal algorithm for the one-dimensional RSU deployment problem with nonuniform profit density," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1052–1061, Feb. 2019.

[22] J. Wang *et al.*, "Dynamic clustering and cooperative scheduling for vehicle-to-vehicle communication in bidirectional road scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 1913–1924, Jun. 2018.

[23] R. I. Meneguette and A. Boukerche, "Peer-to-peer protocol for allocated resources in vehicular cloud based on v2v communication," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–6.

[24] F. Zeng, R. Zhang, X. Cheng, and L. Yang, "Channel prediction based scheduling for data dissemination in VANETs," *IEEE Commun. Lett.*, vol. 21, no. 6, pp. 1409–1412, Jun. 2017.

[25] P. Dai, K. Liu, X. Wu, Z. Yu, H. Xing, and V. C. S. Lee, "Cooperative temporal data dissemination in SDN-based heterogeneous vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 72–83, Feb. 2019.

[26] K. Liu, J. K.-Y. Ng, J. Wang, V. C. Lee, W. Wu, and S. H. Son, "Network-coding-assisted data dissemination via cooperative vehicle-to-vehicle/-infrastructure communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1509–1520, Jun. 2016.

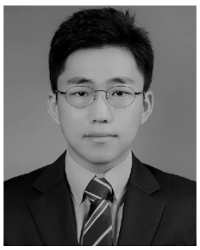
[27] D. S. Hochba, "Approximation algorithms for NP-hard problems," *ACM Sigact News*, vol. 28, no. 2, pp. 40–52, 1997.

[28] S. Sakai, M. Togasaki, and K. Yamazaki, "A note on greedy algorithms for the maximum weighted independent set problem," *Discrete Appl. Math.*, vol. 126, no. 2, pp. 313–322, 2003.

[29] H. Schwetman, "Csim19: A powerful tool for building system models," in *Proc. Winter Simulation Conf.*, vol. 1, Dec. 2001, pp. 250–255.

[30] K. Liu, E. Chan, V. Lee, K. Kapitanova, and S. H. Son, "Design and evaluation of token-based reservation for a roadway system," *Transp. Res. C, Emerg. Technol.*, vol. 26, pp. 184–202, Jan. 2013.

[31] G. K. Zipf, *Human Behavior Principle Least Effort: An Introduction to Human Ecology*. Cambridge, U.K.: Ravenio Books, 2016.



**Byungjin Ko** (S'15) received the B.S. degree in information and communication engineering from Yeungnam University, Gyeongsan-si, South Korea, in 2014, and the M.S. degree in information and communication engineering from the Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, South Korea, in 2016. He is currently pursuing the Ph.D. degree with the Department of Information and Communication Engineering, DGIST. His research interests include intelligent transportation systems and vehicular networks.

**Kai Liu** (S'07–M'12–SM'19) received the Ph.D. degree in computer science from the City University of Hong Kong in 2011. From December 2010 to May 2011, he was a Visiting Scholar with the Department of Computer Science, University of Virginia, USA. From 2011 to 2014, he was a Post-Doctoral Fellow with Singapore Nanyang Technological University, City University of Hong Kong, and Hong Kong Baptist University. He is currently a Professor with the College of Computer Science, Chongqing University, China. His research interests include the Internet of Vehicles, mobile computing, pervasive computing, and big data.

**Sang Hyuk Son** (M'85–SM'98–F'13) received the B.S. degree in electronics engineering from Seoul National University, the M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), and the Ph.D. degree in computer science from the University of Maryland, College Park, MD, USA. He has been a Professor with the Computer Science Department, University of Virginia, and the WCU Chair Professor with Sogang University. He has been a Visiting Professor with KAIST, the City University of Hong Kong, the Ecole Centrale de Lille, France, Linköping University, and the University of Skövde, Sweden. He is currently with the Daegu Gyeongbuk Institute of Science and Technology. He has authored or coauthored over 340 articles and edited/authored four books in his research areas. His research interests include cyber-physical systems, real-time and embedded systems, database and data services, and wireless sensor networks. He is a member of the Korean Academy of Science and Technology and the National Academy of Engineering of Korea. He is a Founding Member of the ACM/IEEE CPS Week. He received the Outstanding Contribution Award from the Cyber Physical Systems Week in 2012. He has served on the Editorial Board of the *ACM Transactions on Cyber Physical Systems*, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and *Real-Time Systems* journal.

**Kyung-Joon Park** (M'05) received the B.S. and M.S. degrees in electrical engineering, and the Ph.D. degree in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 1998, 2000, and 2005, respectively. From 2005 to 2006, he was a Senior Engineer with Samsung Electronics, Suwon, South Korea. From 2006 to 2010, he was a Postdoctoral Research Associate with the Department of Computer Science, University of Illinois at Urbana–Champaign, Champaign, IL, USA. He is currently an Associate Professor with the Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology, Daegu, South Korea. His research interests include resilient cyber-physical systems and smart production systems.