This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Cooperative Data Scheduling in Hybrid Vehicular Ad Hoc Networks: VANET as a Software Defined Network

Kai Liu, *Member, IEEE*, Joseph K. Y. Ng, *Senior Member, IEEE, Member, ACM*, Victor C.S. Lee, *Member, IEEE, ACM*, Sang H. Son, *Fellow, IEEE*, and Ivan Stojmenovic, *Fellow, IEEE*

***Abstract*—This paper presents the first study on scheduling for cooperative data dissemination in a hybrid infrastructure-to-vehicle (I2V) and vehicle-to-vehicle (V2V) communication environment. We formulate the novel problem of cooperative data scheduling (CDS). Each vehicle informs the road-side unit (RSU) the list of its current neighboring vehicles and the identifiers of the retrieved and newly requested data. The RSU then selects sender and receiver vehicles and corresponding data for V2V communication, while it simultaneously broadcasts a data item to vehicles that are instructed to tune into the I2V channel. The goal is to maximize the number of vehicles that retrieve their requested data. We prove that CDS is NP-hard by constructing a polynomial-time reduction from the Maximum Weighted Independent Set (MWIS) problem. Scheduling decisions are made by transforming CDS to MWIS and using a greedy method to approximately solve MWIS. We build a simulation model based on realistic traffic and communication characteristics and demonstrate the superiority and scalability of the proposed solution. The proposed model and solution, which are based on the centralized scheduler at the RSU, represent the first known vehicular ad hoc network (VANET) implementation of software defined network (SDN) concept.**

***Index Terms*—Cooperative data dissemination, scheduling, software defined network, vehicular ad hoc networks.**

Manuscript received March 11, 2014; revised July 22, 2014 and January 19, 2015; accepted April 13, 2015; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor X. Liu. This work was supported in part by the Fundamental Research Funds for the Central Universities under Project No. 106112015CDJZR185518; the HKBU Research Centre for Ubiquitous Computing, the Institute of Computational and Theoretical Studies, and the HKBU Strategic Development Fund under Grant HKBU SDF 10-0526-P08; the Research Grants Council of the Hong Kong Special Administrative Region, China under Project No. CityU 115312; and the ICT R&D program of MSIP/IITP under Grant 14-824-09-013, (Resilient Cyber-Physical Systems Research) and the GRL Program under Grant 2013K1A1A2A02078326 through the NRF.

K. Liu is with the Key Laboratory of Dependable Service Computing in Cyber Physical Society (Chongqing University), Ministry of Education, and also with the College of Computer Science, Chongqing University, Chongqing 400030, China (e-mail: liukai0807@gmail.com).

J. K. Y. Ng is with the Computer Science Department, Hong Kong Baptist University, Hong Kong (e-mail: jng@comp.hkbu.edu.hk).

V. C. S. Lee is with the Computer Science Department, City University of Hong Kong, Hong Kong (e-mail: csvlee@cityu.edu.hk).

S. H. Son is with the Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu 711-873, Korea (e-mail: son@dgist.ac.kr).

I. Stojmenovic, deceased, was with the School of Information Technology (SIT), Deakin University, Melbourne, Burwood, Vic. 3125, Australia, and the School of Electrical Engineering and Computer Science (SEECS), University of Ottawa, Ottawa, ON K1N 6N5, Canada.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2015.2432804

I. INTRODUCTION

RECENT advances in vehicular communications have motivated increasing research interest in emerging applications in vehicular ad hoc networks (VANETs), such as collision avoidance [1], roadway reservation [2], and autonomous intersection management [3], to name but a few. The implementation of these systems imposes strict requirements on efficient data services in VANETs [4]. The dedicated short-range communication (DSRC) [5] is an unprecedented wireless technology intended to support both infrastructure-to-vehicle (I2V) and vehicle-to-vehicle (V2V) communications. In general, DSRC refers to a suite of standards including IEEE 802.11p, IEEE 1609.1/.2/.3/.4 protocol family and SAE J2735 message set dictionary [6]. In DSRC, the road-side unit (RSU) is a fixed infrastructure installed along the road to provide data services, while the on-board unit (OBU) is mounted on vehicles and enables them to communicate with the RSU and their neighboring vehicles.

Great endeavor has been made by automotive manufacturers, governments, and research universities toward efficient vehicular communications, which significantly boosts the development of innovative intelligent transportation systems (ITSs). In industry, the current generation of vehicles already equips with devices with certain computation and communication capacities, such as *MyFord Touch* from the Ford Motor Company, *Entune* from the Toyota Motor Company, *Mbrace2* from the Mercedes-Benz Company, etc. Meanwhile, the US Department of Transportation (USDOT) is actively collaborating with automotive manufacturers and universities on a variety of advanced ITS projects, including the *Connect Vehicle* program, the *Vehicle Infrastructure Integration* (VII) project, the *Berkeley PATH* project, etc.

This work is dedicated to the scheduling for cooperative data dissemination in a hybrid of I2V and V2V communication environments. We consider the delay-sensitive applications (e.g., the service has to be completed before vehicles leaving the RSU's coverage [7]). Application scenarios include intersection control systems [3], speed advisory systems [8], traffic management systems [9], etc. For example, considering the traffic management system for emergency situations, the emergency message on the road should be transmitted to vehicles around the RSU as file of interest nearby. The quality of such services highly depends on the efficiency of communications between the RSU and vehicles. In addition, it is desirable to enhance data dissemination performance by further exploiting the capacity of V2V communication. Sharing data items among vehicles has the potential to improve the bandwidth efficiency of the RSU, as it may reduce the redundancy of rebroadcasting the same data item via I2V communication. In addition, by appropriately exploiting the spatial reusability, multiple data items can be disseminated via V2V communication simultaneously without interference. In particular, the main contributions of this paper are outlined as follows.

- Vehicles are likely to request files of common interest, such as parking slots, road conditions, gas stations, etc. Inspired by this observation, we investigate the potential benefit of exploiting V2V communication to assist RSU-based data services and discuss the challenges on providing efficient data services in such a hybrid I2V/V2V communication environment. To the best of our knowledge, this is the first study on scheduling for cooperative data dissemination in VANETs with consideration of both communication constraints and application requirements.
- In accordance with IEEE 1609.4, this model consists of one control channel and two service channels. The control channel is used for disseminating management information, service advertisements and control messages from the RSU to vehicles. The two service channels are used for I2V and V2V data dissemination, respectively. Such channel coordination and simultaneous data dissemination have not been considered in literature so far.
- We formulate the novel problem of cooperative data scheduling (CDS). Vehicles can switch to either I2V or V2V communication at a time. In a scheduling period, vehicles in the I2V channel can receive a data item from the RSU, while other vehicles can simultaneously transmit or receive a data item over the V2V channel. Each vehicle maintains a set of requested data items. Only one data item can be transmitted or received for each vehicle in one scheduling period. Furthermore, vehicles cannot transmit and receive a data item at the same time. Each vehicle has a weight for being served. In our particular implementation, the weight is inversely proportional to the estimated remaining dwell time of the receiver vehicle, which captures the urgency of services. The objective of scheduling is to maximize the weighted gain, which is the summation of the weight of each served vehicle in a scheduling period.
- We prove that CDS is NP-hard by constructing a polynomial-time reduction from the maximum weighted independent set (MWIS) problem [10].
- For the first time, we propose a centralized RSU controlled data dissemination over I2V and V2V channels. The RSU delivers scheduling decisions to vehicles, instructing them which channel to tune to and which data to transmit/receive, which enables cooperative data dissemination. To facilitate this approach, we divide a scheduling period into three phases. In the first phase, all the vehicles are set to the V2V mode and broadcast their heartbeat messages, so that each vehicle is able to identify a list of its neighbors. In the second phase, all the vehicles switch to the I2V mode. Each vehicle informs the RSU the list of its current neighbors, and the identifiers of its retrieved and newly requested data items. In the third phase, each vehicle participates into either I2V or V2V communication based on the scheduling decision. Multiple instances of data dissemination may take place simultaneously in this phase.
- We solve CDS problem and present its detailed implementation via the hybrid of I2V and V2V communications. In the proposed online scheduling algorithm, scheduling decisions are computed by the RSU based on a greedy method, which approximately solves the MWIS problem.
- We describe the first application of Software defined network (SDN) concept [11] in VANETs. SDN is an emergent paradigm in computing and networking, which separates the control and data communication layers to simplify the network management and expedite system evolution. In SDN, the network intelligence is logically centralized in software-based controllers (i.e., control plane), and the nodes in the network will forward data packets based on the decisions made by the server. In our centralized implementation, vehicles do not need to maintain any control information. Logically centralized control is fully exercised by the RSU. SDN concept was not studied in VANETs so far. The closest studies are few implementations in wireless sensor networks [12] and mesh networks [13].
- We build the simulation model based on realistic traffic and communication characteristics and evaluate the performance of the proposed solution under a wide range of traffic workloads. The simulation results demonstrate the effectiveness of proposed solution.

The rest of this paper is organized as follows. Section II reviews the related work. Section III motivates the application scenario and introduces preliminary concepts. Section IV illustrates the data dissemination system based on the hybrid of I2V and V2V communications. In Section V, we formulate the cooperative data scheduling problem and prove that it is NP-hard. In Section VI, we propose a heuristic online scheduling algorithm. In Section VII, we build the simulation model and evaluate the algorithm performance. Finally, Section VIII concludes this work and discusses future research directions.

II. Related Work

Previous studies on data dissemination in VANETs mainly focused on improving communication quality and reliability, and the problems typically reside in MAC and PHY layers. A recent survey of data dissemination schemes in vehicular networks is given in [14]. Although there have been a few studies on cooperative I2V and V2V communications, they are dedicated to designing MAC protocols and routing strategies. Zhang *et al.* [15] proposed a vehicular cooperative media access control (VC-MAC). It considers the scenario where vehicles are expected to retrieve common information when passing through the RSU via I2V communication, whereas some of them cannot be successfully served due to unreliable wireless data transmission. The VC-MAC is proposed for vehicles to share information outside RSU's coverage via V2V communication and maximize the total throughput. Fujimura and Hasegawa [16] proposed a MAC protocol called Vehicle and Roadside Collaborative Protocol (VRCP) to support both I2V and V2V communications. It designs two channel access modes. One is ad-hoc mode (Mode-A) with nonpersistent CSMA scheme for decentralized V2V communication, and the other is infrastructure mode (Mode-I) with TDMA scheme for centralized I2V communication. Mak *et al.* [17] proposed a coordinated MAC mode in the presence of an RSU to compliment with *ad hoc* approaches when the RSU is not available. It aims to improve performance for both safety and nonsafety applications by designing a multichannel coordination mechanism. Unlike the above efforts on designing MAC protocols, this work focuses on data scheduling in the application layer and supports data services via the hybrid I2V and V2V communications.

A number of studies considered multihop routing via I2V and V2V communications in VANETs. Shan and Zhuang [18] proposed a cooperation-based multihop transmission scheme called path-based cooperative multihop relaying (PCMR), which aims at minimizing the end-to-end outage probability. When an outage occurs in the multihop transmission, PCMR reorganizes the delivery path adaptively based on the current channel condition and the network topology. Vegni and Little [19] proposed an I2V and V2V protocol switching decision algorithm for unicast routing. It considers the attributes of traffic density, message direction, network connectivity, and resource utilization for making routing decisions. The protocol is effective in enhancing data propagation performance based on opportunistic use of moving vehicles and available wireless network infrastructures. Wu *et al.* [20] presented a hybrid routing scheme for data dissemination in VANETs and proposed a protocol suite called Hybrid Routing in VANETs (HRV). It includes an online probabilistic localization algorithm ($\mathrm{HRV}_{\mathrm{retrieval}}$) and a network-coding-based multicast protocol ($\mathrm{HRV}_{\mathrm{multicast}}$), which are used for estimating the number and location of available RSUs and enhancing the robustness of data delivery, respectively. V2V communication is used to assist data forwarding. Compared to the above work, our problem statement is different. We focus on exploring the coordination between I2V and V2V communications to maximize data services, where the hybrid I2V and V2V data dissemination takes place simultaneously, which requires subtle coordination for satisfying particular communication constraints and application requirements.

Sou [21] applied game theory to motivate incentives for intervehicle cooperation on data services. The work assumes that vehicles are selfish and economically rational. It considers local file sharing in a sparse vehicular network where vehicles intermittently connect to RSUs. A game-theoretic selfishness detection mechanism (SDM) is proposed to formulate selfish behaviors of vehicles and stimulate intervehicle cooperation to maximize the long-term utility of each vehicle. Although this work motivates the vehicles to participate into V2V communication, it does not consider the implementation and coordination issues on data dissemination in the hybrid I2V and V2V communication environment.

There have been extensive studies on data scheduling in conventional mobile computing systems [22]–[25], which primarily focused on improving the bandwidth efficiency at the server side. Nevertheless, in VANETs, vehicles retrieve information not only from the RSU via I2V communication, but also from neighboring vehicles via V2V communication. Meanwhile, the simultaneous I2V and V2V data dissemination brings us unique scheduling challenges including determination of I2V/V2V modes, data selection for I2V communication, vehicle coordination for V2V communication, etc. Therefore, none of the existing solutions is sufficient to exploit the synergy between I2V and V2V communications to enhance the data service. A few studies considered the data scheduling in I2V communication. Chang *et al.* [26] focused on the application scenario where RSUs form a seamless coverage along the road and provide passing vehicles with continuous file transmission services. A scheduling algorithm called Maximum Freedom Last (MFL) is proposed to minimize the handoff rate due to incompleteness of file transmission in the coverage of one RSU. Zhang *et al.* [27] focused on I2V data dissemination, where the file transmission (downloading) and the update installation (uploading) compete for a certain range of bandwidth spectrum. A two-step scheduling scheme is proposed to strike a balance between serving uploading and downloading requests. Our recent study [28] investigated temporal data dissemination in VANETs, which focused on delivering dynamic traffic information where the values of data items are updated periodically. An online scheduling algorithm is proposed to enhance the system performance by striking a balance between delivering temporal data items and serving real-time requests. The above studies only considered data scheduling in pure I2V communication environments. In contrast, this work is dedicated to the scheduling in the hybrid I2V and V2V communication environment.

III. Preliminaries

In this section, we first present an illustrative example to motivate the concerned application scenario. Then, we introduce preliminary concepts to facilitate problem description.

This work considers the hybrid of I2V and V2V communications based on DSRC, where data services are provided by the RSU. Although DSRC is exclusively developed for supporting emerging ITS applications in VANETs, there are still wide debates in regard to adopting other wireless techniques for vehicular communications, such as WiFi and cellular networks. Certainly, these alternatives can be used as supplementary mediums to enrich services in VANETs. Nevertheless, they cannot properly address some ITS applications that are delay-sensitive and location-dependent, as these applications have high demands on timely interactions between the RSU and vehicles. For example, in conventional WLAN data access model, clients have to go through association and authentication procedures to establish connections with the access point, which normally takes several seconds. In contrast, DSRC (802.11p) is designed to exchange data without the need of waiting for the association and authentication procedures, and thus vehicles may start to send or receive data as soon as they switch to the communication channel. In addition, frequent interaction with infrastructures in cellular networks may also incur high payments. Therefore, DSRC is the most appealing solution to enable applications requiring cooperative communications in VANETs.

We take the cooperative intersection control system [3] as an example to illustrate the feasibility of DSRC, which also helps to motivate the cooperative data dissemination model to be presented in Section IV. In such a system, vehicles are required to update their locations, velocities, and driving directions to the RSU in real time. With up-to-date information of vehicles, the RSU periodically computes the optimal movement for each vehicle and broadcasts the maneuver messages (e.g., advised velocities and accelerations) to them. With received instructions, all the vehicles will be able to cooperated to pass the intersection safely and smoothly. With DSRC, vehicles can piggyback their requests into the basic safety message (BSM) as defined in SAE J2735. The standard payload of a BSM is 39 B, including vehicle information such as locations, velocity, driving directions, etc. The payload of a request is small, and suppose it is 10 B. DSRC supports the data rate from 6 to 27 Mb/s [6], where 6 Mb/s is recommended for disseminating safety critical messages. Therefore, the time taken to submit a request is around $(39 + 10) * 8$ bits/6 Mb/s$\approx$ 0.06 ms. Suppose the data size is 500 kB, which is sufficient for typical ITS applications [29]. In such a case, the time for data transmission is around $500 * 8$ kb/6 Mb/s$\approx$ 0.65 s. Thus, compared to data transmission time, the time taken for submitting requests can be ignored. Adding the overhead of scheduling and disseminating control messages, it is reasonable to set the scheduling period as 1 s in such a system.

Due to the intrinsic nature of wireless communication, vehicles in close proximity may suffer transmission collisions when they are broadcasting data items simultaneously [15], [30]. We assume that vehicles can only switch to one of the communication modes (i.e., either I2V or V2V) at a time [16]. Furthermore, vehicles cannot transmit and receive data items at the same time due to the half-duplex transmission of OBUs specified by DSRC [5].

To prove that the formulated CDS problem is NP-hard, we will construct a polynomial time reduction from MWIS [10] to CDS. MWIS is a well-known NP-hard problem, which is briefly introduced as follows. Let $G = \{V, E\}$ be a weighted undirected graph, where V is the set of vertices and E is the set of edges. Each vertex $v \in V$ is associated with a weight $w(v)$. An independent set of G is a set of vertices, in which any two of them are not adjacent. That is, a subset $I \subseteq V$ is an independent set of G if for any two vertices $u, v \in I$, they satisfy $\{u, v\} \notin E$. The weight of an independent set I, denoted by $W(I)$, is the sum of $w(v)$, $\forall v \in I$. With the above knowledge, MWIS is the problem to find the independent set I^* in G, such that $W(I^*) \geq W(I), \forall I \in G$.

A greedy method presented in [31] is adopted to approximately solve MWIS, which operates as follows. First, it computes the value of $w(v)/(d(v) + 1)$ for each vertex v in G, where $w(v)$ and $d(v)$ represent the weight and the degree of v, respectively. Second, it selects the vertex v_{selected} with the maximum value of $w(v)/(d(v) + 1)$. Third, it updates G by removing the set of vertices $N^+(v_{\text{selected}})$, where $N^+(v_{\text{selected}})$ contains v_{selected} and all of its adjacent vertices. Forth, it repeats the above operations until there is no vertex remaining in G (i.e., $V(G) = \emptyset$). Denote $W(I)^*$ as the weight of a maximum independent set of G, and denote $W(I)$ as the weight of the independent set obtained by this greedy method. It has been proven that the performance ratio satisfies $\frac{W(I)}{W(I)^*} \geq \frac{1}{\Delta_G}$, where Δ_G is the maximum degree of any vertex in G.

IV. Data Dissemination System

Fig. 1 shows the data dissemination system in the hybrid vehicular communication environment. In accordance with IEEE 1609.4, we consider one control channel and two service channels in the system. Specifically, the control channel is used for disseminating management information, service advertisements, and control messages. One of the service channels is used for I2V data dissemination, while the other one is used for V2V data dissemination. We consider single radio OBUs as they are commonly adopted in VANETs due to both deployment and economic concerns. Therefore, vehicles can tune in to only one of the channels at a time [17]. The time unit adopted in this work refers to a *scheduling period*, which consists of three phases as introduced in the following.

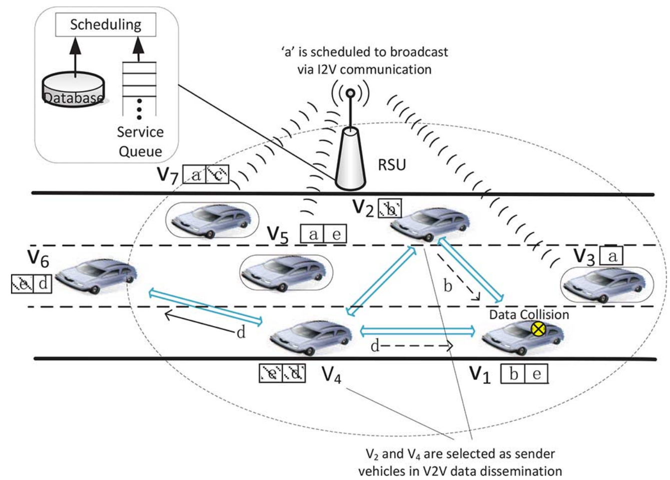


Fig. 1. Data dissemination via the hybrid of I2V and V2V communications.

In the first phase, all the vehicles are set to the V2V mode and broadcast their heartbeat messages (i.e., the *Basic Safety Message* as defined in SAE J2735), so that each vehicle is able to identify a list of its neighboring vehicles. For instance, by measuring the signal-to-noise ratio through the heartbeat messages received from other vehicles, a vehicle can recognize a set of vehicles, to which it can transmit and receive data items.

In the second phase, all the vehicles switch to the I2V mode and communicate with the RSU. Specifically, each vehicle informs the RSU with its updated information, including the list of its current neighbors, and the identifiers of the retrieved and newly requested data items. This information is piggybacked into the *Probe Vehicle Message* as defined in SAE J2735. Each request is made only for one data item, and the request is satisfied as long as the corresponding data item is retrieved via either I2V or V2V communication. Outstanding requests are pending in the service queue. According to a certain algorithm, the scheduling decisions are announced via the control channel (i.e., piggybacked into the WAVE service advertisements [32]).

In the third phase, each vehicle participates into either I2V or V2V communication based on the scheduling decisions. Multiple instances of data dissemination may take place simultaneously in this phase. Specifically, some vehicles will be instructed to tune in to the I2V mode and retrieve the data item transmitted from the RSU, while some others will be instructed to tune in to the V2V mode for data transmit or receive. Note that this work considers only one-hop V2V data dissemination.

To enable a collaborative I2V and V2V data dissemination, the algorithm is expected to make the following scheduling decisions. First, the algorithm divides the vehicles into two groups. One group is for I2V communication, while the other is for V2V communication. Second, the algorithm selects one data item to be transmitted from the RSU, so that vehicles in the I2V group can retrieve this data item via the I2V service channel. Third, for vehicles in the V2V group, the algorithm determines a set of sender vehicles and the corresponding data items disseminated by each sender vehicle, so that the neighbors of each sender vehicle may have chance to retrieve their requested data items via the V2V service channel. The vehicles are assumed to stay in the same neighborhood for a short period of time (i.e., during a scheduling period) [15].

TABLE I
SUMMARY OF NOTATIONS

Notations	Descriptions	Notes
D	set of data items in the database	$D=\{d_1,d_2,\cdots,d_{\|D\|}\}$
$V(t)$	set of vehicles at time t	$V(t)=\{V_1,V_2,...,V_{\|V(t)\|}\}$
$V_I(t)$	set of vehicles in the I2V mode	$V_I(t)\subseteq V(t)$
$V_V(t)$	set of vehicles in the V2V mode	$V_I(t)\cap V_V(t)=\emptyset$ and $V_I(t)\cup V_V(t)=V(t)$
$Q_{V_i}(t)$	set of requests submitted by V_i	$Q_{V_i}(t)=\{q^1_{V_i},q^2_{V_i},...q^{\|Q_{V_i}(t)\|}_{V_i}\}$
$q^j_{V_i}$	jth request of V_i	$1\le j\le \|Q_{V_i}(t)\|$ and $q^j_{V_i}\in D$
$SQ_{V_i}(t)$	set of satisfied requests of V_i	$SQ_{V_i}(t)\subseteq Q_{V_i}(t)$
$PQ_{V_i}(t)$	set of pending requests of V_i	$SQ_{V_i}(t)\cap PQ_{V_i}(t)=\emptyset$ and $SQ_{V_i}(t)\cup PQ_{V_i}(t)=Q_{V_i}(t)$
$N_{V_i}(t)$	set of neighboring vehicles of V_i	$N_{V_i}(t)\subset V_V(t)$
$d_I(t)$	data item transmitted from the RSU	$d_I(t)\in D$
$V_{RSU}(t)$	set of vehicles within the RSU's coverage	$V_{RSU}(t)\subseteq V(t)$
$SV(t)$	set of sender vehicles	$SV(t)=\{SV_1,SV_2,...,SV_{\|SV(t)\|}\}$ and $SV(t)\subseteq V_V(t)$
$d(SV_i)$	the data item disseminated by SV_i	$SV_i\in SV(t)$
$D(SV(t))$	set of data items disseminated by sender vehicles	$D(SV(t))=\{d(SV_1),d(SV_2),...,d(SV_{\|SV(t)\|})\}$
$RV(d_I(t))$	set of receiver vehicles for $d_I(t)$	$RV(d_I(t))\subseteq V_I(t)$
$RV(d(SV_i))$	set of receiver vehicles for $d(SV_i)$	$RV(d(SV_i))\subseteq N_{SV_i}(t)$
$RV(D(SV(t)))$	set of receiver vehicles in V2V communication	$RV(D(SV(t)))=\bigcup_{SV_i\in SV(t)} RV(d(SV_i))$
$TS_{I2V}(t)$	set of I2V tentative schedules	$\hat{d}\in PQ_{V_r}(t),\forall R\hat{d}V_r\in TS_{I2V}(t)$
$TS_{V2V}(t)$	set of V2V tentative schedules	$\hat{d}\in SQ_{V_s}(t)\wedge\hat{d}\in PQ_{V_r}(t),\forall V_s\hat{d}V_r\in TS_{V2V}(t)$
$G(t)$	gain of scalability	
$W_{V_i}(t)$	weight of V_i	
$G^w(t)$	weighted gain	

Fig. 1 illustrates a toy configured example. Vehicles in the I2V group can retrieve the data item only when they are in the RSU's coverage, which is represented by the dotted ellipse. In this example, three vehicles are designated into the I2V group (i.e., V_3, V_5, and V_7). The data block without shadow represents that the corresponding data item has been requested by the vehicle but has not yet been retrieved. In contrast, the shadowed data block means that the corresponding data item has been retrieved and cached. Accordingly, when a is broadcast from the RSU, V_3, V_5, and V_7 can retrieve it via the I2V service channel. In V2V communication, the double-edged arrow represents that the two vehicles are neighbors. The data item to be disseminated by a sender vehicle has to be cached in advance. For instance, V_4 has cached e and d, and it could be selected as a sender for disseminating d with the target receiver of V_6. Similarly, V_2 has cached b, and it could be selected as another sender for disseminating b to serve V_1. However, due to the broadcast nature in wireless communications, simultaneously disseminating data items from the vehicles that are in the immediate or adjacent neighborhoods will lead to the data collision [15]. In this example, V_1 is in the neighborhood of both V_2 and V_4. Therefore, data collision happens at V_1 when V_2 and V_4 are disseminating data items at the same time. Thus, V_1 cannot receive b from V_2 because the interference caused by V_4. Note that for V_3, V_5, and V_7, as they are tuned in to the I2V service channel, they will not be influenced by V2V communication.

V. Problem Analysis

A. Cooperative Data Scheduling (CDS) Problem

1) Notations: For clear exposition, the primary notations throughout the problem description are summarized in Table I. The database $D=\{d_1,d_2,\ldots,d_{|D|}\}$ consists of $|D|$ data items. The set of vehicles is denoted by $V(t)=\{V_1,V_2,\ldots,V_{|V(t)|}\}$, where $|V(t)|$ is the total number of vehicles at time t. Depending on the communication mode of vehicles, $V(t)$ is divided into two sets: $V_{\mathrm{I}}(t)$ and $V_{\mathrm{V}}(t)$, where $V_{\mathrm{I}}(t)$ represents the set of vehicles in the I2V mode, and $V_{\mathrm{V}}(t)$ represents the set of vehicles in the V2V mode. Each vehicle stays in either I2V or V2V mode at a time, namely, $V_{\mathrm{I}}(t)\cap V_{\mathrm{V}}(t)=\emptyset$ and $V_{\mathrm{I}}(t)\cup V_{\mathrm{V}}(t)=V(t)$.

Each V_i $(1\le i\le |V(t)|)$ has a set of requests, which is denoted by $Q_{V_i}(t)=\{q^1_{V_i},q^2_{V_i},\ldots q^{|Q_{V_i}(t)|}_{V_i}\}$, where $|Q_{V_i}(t)|$ is the total number of requests submitted by V_i at time t. Each $q^j_{V_i}$ $(1\le j\le |Q_{V_i}(t)|)$ corresponds to a data item in the database, and it is satisfied once this data item is retrieved by V_i. According to the service status of requests (i.e., satisfied or not), $Q_{V_i}(t)$ is divided into two sets: $SQ_{V_i}(t)$ and $PQ_{V_i}(t)$, where $SQ_{V_i}(t)$ represents the set of satisfied requests, while $PQ_{V_i}(t)$ represents the set of pending requests. Then, we have $SQ_{V_i}(t)\cap PQ_{V_i}(t)=\emptyset$ and $SQ_{V_i}(t)\cup PQ_{V_i}(t)=Q_{V_i}(t)$. Since each request $q^j_{V_i}$ corresponds to a data item d_k, without causing ambiguity, the expression $d_k\in SQ_{V_i}(t)$ is adopted to represent that d_k is requested by V_i and it has been retrieved (i.e., $q^j_{V_i}$ has been satisfied).

For each V_i in the V2V mode, the set of its neighboring vehicles (i.e., the set of vehicles in the V2V mode and within the V2V communication range of V_i) is denoted by $N_{V_i}(t)$, where $N_{V_i}(t)\subset V_{\mathrm{V}}(t)$. The RSU maintains an entry in the service queue for each V_i, which is characterized by a 3-tuple: $\langle V_i,Q_{V_i}(t),N_{V_i}(t)\rangle$. The values of $Q_{V_i}(t)$ and $N_{V_i}(t)$ are updated in every scheduling period. To facilitate the formulation of CDS, relevant concepts are defined as follows.

2) Definitions: In I2V communication, the RSU broadcasts one data item in each scheduling period, which is denoted by $d_{\rm I}(t)$, where $d_{\rm I}(t) \in D$. Denote $V_{\rm RSU}(t)$ as the set of vehicles within the RSU's coverage. Only when $V_i \in V_{\rm RSU}(t)$, it can retrieve $d_{\rm I}(t)$ via the I2V service channel. Specifically, the receiver vehicle set in I2V communication is defined as follows.

Definition 1: Receiver Vehicle Set in I2V Communication: Given the data item $d_{\rm I}(t)$ transmitted from the RSU, the set of receiver vehicles for $d_{\rm I}(t)$, denoted by $RV(d_{\rm I}(t))$, consists of any vehicle V_i, which satisfies the following conditions: 1) V_i is in the RSU's coverage; 2) V_i is in the I2V mode; 3) $d_{\rm I}(t)$ is requested by V_i, and it has not yet been retrieved. That is

$$RV(d_{\rm I}(t)) = \{V_i | V_i \in V_{\rm RSU}(t) \wedge V_i \in V_{\rm I}(t) \wedge d_{\rm I}(t) \in PQ_{V_i}(t)\}. \quad (1)$$

In V2V communication, a set of sender vehicles is designated to disseminate data items, which is denoted by $SV(t) = \{SV_1, SV_2, \ldots, SV_{|SV(t)|}\}$, where $|SV(t)|$ is the number of designated sender vehicles. All sender vehicles are in the V2V mode. That is, $SV(t) \subseteq V_{\rm V}(t)$. The set of data items to be disseminated is denoted by $D(SV(t)) = \{d(SV_1), d(SV_2), \ldots, d(SV_{|SV(t)|})\}$, where $d(SV_i)$ $(1 \le i \le |SV(t)|)$ is the data item disseminated by SV_i. Note that $d(SV_i)$ has to be retrieved by SV_i in advance, namely, $d(SV_i) \in SQ_{SV_i}(t)$. Due to the broadcast effect, simultaneous data dissemination of multiple sender vehicles may cause data collision at receivers. Specifically, the set of receiver vehicles suffering from data collision is defined as follows.

Definition 2: Receiver Vehicle Set Suffering From Data Collision: Given the set of sender vehicles $SV(t)$, for any V_k in the V2V mode, if V_k is in the neighborhood of both SV_i and SV_j $(SV_i, SV_j \in SV(t))$, then data collision happens at V_k. Accordingly, the receiver vehicle set suffering from data collision is represented by $\{V_k | V_k \in V_{\rm V}(t) \wedge V_k \in N_{SV_i}(t) \wedge V_k \in N_{SV_j}(t)\}$ $(\forall SV_i, SV_j \in SV(t))$.

Considering data collision, given a sender vehicle SV_i with the transmitted data item $d(SV_i)$, the set of receiver vehicles for $d(SV_i)$ is defined as follows.

Definition 3: Receiver Vehicle Set for $d(SV_i)$: The receiver vehicle set for $d(SV_i)$, denoted by $RV(d(SV_i))$, consists of any vehicle V_j, which satisfies the following four conditions: 1) V_j is in the neighborhood of SV_i; 2) $d(SV_i)$ is requested by V_j but it has not yet been retrieved; 3) V_j is not in the sender vehicle set; 4) V_j is not in the neighborhood of any other sender vehicles excepting for SV_i. That is

$$RV(d(SV_i)) = \{V_j | V_j \in N_{SV_i}(t) \wedge d(SV_i) \in PQ_{V_j}(t) \wedge V_j \notin SV(t) \wedge V_j \notin N_{SV_k}(t), \forall SV_k \in \{SV(t) - SV_i\}. \quad (2)$$

The first two conditions are straightforward. The third condition means that a vehicle cannot be the sender and the receiver at the same time. The forth condition guarantees that no data collision happens at the receiver. On this basis, given the set of sender vehicles $SV(t)$ with the corresponding data items $D(SV(t))$, the receiver vehicle set in V2V communication is defined as follows.

Definition 4 Receiver Vehicle Set in V2V Communication: Given $D(SV(t))$, the receiver vehicle set in V2V communication, denoted by $RV(D(SV(t)))$, is the union of receiver vehicle sets for each $d(SV_i) \in D(SV(t))$. That is

$$RV(D(SV(t))) = \bigcup_{d(SV_i) \in D(SV(t))} RV(d(SV_i)). \quad (3)$$

In view of the dynamic traffic workload and the heavy data service demand, it is imperative to enhance the system scalability via cooperative data dissemination. Therefore, one of the primary objectives is to maximize the total number of vehicles that can be served via either I2V or V2V communication in each scheduling period. To this end, the gain of scalability is defined as follows.

Definition 5: Gain of Scalability: Given the data item $d_{\rm I}(t)$ transmitted from the RSU, the set of sender vehicles $SV(t)$, and the corresponding set of data items $D(SV(t))$ in V2V communication, the gain of scalability, denoted by $G(t)$, is the total number of vehicles that can be served via either I2V or V2V communication in a scheduling period, which is computed by

$$G(t) = |RV(d_{\rm I}(t))| + |RV(D(SV(t)))| \quad (4)$$

where $|RV(d_{\rm I}(t))|$ is the number of receiver vehicles in I2V communication, and $|RV(D(SV(t)))|$ is the number of receiver vehicles in V2V communication.

In practice, based on specific scheduling objective (to be elaborated in Section VI), serving different vehicles may have different impacts on overall system performance. For general purposes, we define the weighted gain as follows.

Definition 6: Weighted Gain: Denote $W_{V_i}(t)$ as the weight of serving V_i at t. The weighted gain, denoted by $G^w(t)$, is the summation of the weight for each served vehicle in a scheduling period, which is computed by

$$G^w(t) = \sum_{V_i \in RV(d_{\rm I}(t)) \cup RV(D(SV(t)))} W_{V_i}(t). \quad (5)$$

3) CDS: With the above knowledge, CDS is formulated as follows. Given the database $D = \{d_1, d_2, \ldots, d_{|D|}\}$, the set of vehicles $V(t) = \{V_1, V_2, \ldots, V_{|V(t)|}\}$, the set of requests $Q(t) = \{Q_{V_1}, Q_{V_2}, \ldots, Q_{V_{|V(t)|}}\}$, and the set of weights for serving each vehicle $W(t) = \{W_{V_1}, W_{V_2}, \ldots, W_{V_{|V(t)|}}\}$, the algorithm makes the following scheduling decisions. First, it divides the vehicles into I2V and V2V sets, namely, $V_{\rm I}(t)$ and $V_{\rm V}(t)$. Then, a data item $d_{\rm I}(t)$ is selected to broadcast via I2V communication. Meanwhile, a set of sender vehicles $SV(t)$ together with the corresponding set of data items $D(SV(t))$ are selected in V2V communication. Given D, V, Q, and W, let $\Lambda(D, V, Q, W)$ be the set of scheduling decisions for $V_{\rm I}$, $V_{\rm V}$, $d_{\rm I}$, SV, and $D(SV)$. CDS is to find an optimal scheduling decision, denoted by $(V_{\rm I}, V_{\rm V}, d_{\rm I}, SV, D(SV))^*$, such that the weighted gain $G^w(t)$ is maximized. That is:

$$(V_{\rm I}, V_{\rm V}, d_{\rm I}, SV, D(SV))^* = \arg \max_{(V_{\rm I}, V_{\rm V}, d_{\rm I}, SV, D(SV)) \in \Lambda(D,V,Q,W)} G^w(t). \quad (6)$$

4) Example: Fig. 2 shows an example of CDS. The vehicle set $V = \{V_1, V_2, \ldots, V_6\}$ and requests of each vehicle are represented by data blocks. Specifically, the shadowed data block

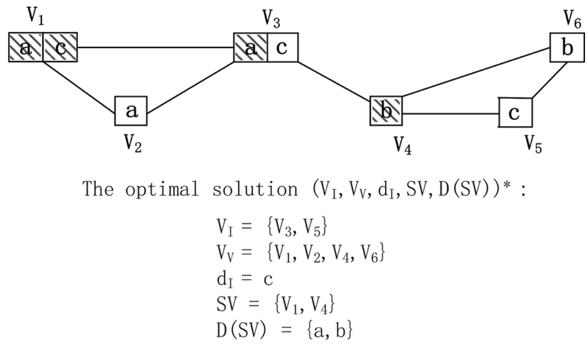


Fig. 2. Example of CDS problem.

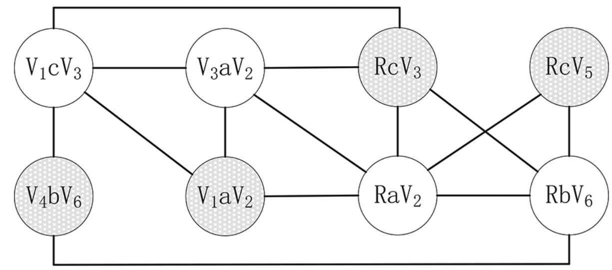


Fig. 3. Example of reduction from MWIS to CDS.

represents that the corresponding data item has been retrieved. In contrast, the data block without shadow represents the outstanding request. The edge represents the neighborhood relationship of vehicles. Assuming all the vehicles are within the RSU's coverage (i.e., $V = V_{\mathrm{RSU}}$), and assuming the gain of serving each vehicle is 1 (i.e., $W_{V_i} = 1, i = 1, 2, \ldots 6$), then it is not difficult to observe the following optimal solution: 1) c is scheduled to be transmitted from the RSU (i.e., $d_{\mathrm{I}} = c$). 2) V_3 and V_5 are set to the I2V mode (i.e., $V_{\mathrm{I}} = \{V_3, V_5\}$), while the other four vehicles are set to the V2V mode (i.e., $V_{\mathrm{V}} = \{V_1, V_2, V_4, V_6\}$). 3) V_1 and V_4 are designated as sender vehicles (i.e., $SV = \{V_1, V_4\}$) for disseminating a and b, respectively. Accordingly, the set of data items in V2V communication is $D(SV) = \{a, b\}$. Given such a schedule, the receiver vehicle sets in I2V and V2V communications are $RV(d_{\mathrm{I}}) = \{V_3, V_5\}$ and $RV(D(SV)) = \{V_2, V_6\}$, respectively. As shown, all the outstanding requests can be served in this scheduling period, and $G^w = |RV(d_{\mathrm{I}})| + |RV(D(SV))| = 4$.

B. NP-Hardness

We prove that CDS is NP-hard by constructing a polynomial-time reduction from a well-known NP-hard problem MWIS [10]. Before presenting the formal proof, a sketch of the idea is outlined as follows. First, we introduce a set of operations (to be defined as the "tentative schedule" formally), which forms the basis of finding an optimal solution of CDS. Second, based on certain constraints on cooperative data dissemination in VANETs, we establish a set of rules to identify conflicting operations such that any pair of conflicting operations cannot coexist in an optimal solution of CDS. Third, we construct an undirected graph G by creating a vertex for each operation and adding an edge between any two conflicting operations. The weight of each vertex is set as the weight of the corresponding operation. With the above mapping, we demonstrate that the optimal schedule of CDS is derived if and only if the MWIS of G is computed. Therefore, CDS is NP-hard. To have clearer exposition, we further illustrate the idea with an example.

As shown in Fig. 3, an undirected graph G is constructed based on the example shown in Fig. 2. The identifier of each vertex represents a viable operation. For instance, the vertex V_1cV_3 represents that V_1 transmits c to V_3. Referring to Fig. 2, it is viable because V_1 has cached c, while its neighbor V_3 is requesting for c. Therefore, this operation has the potential to serve V_3. Thus, there is a one-to-one mapping between each operation and each vertex. An edge between two vertices represents that the two corresponding operations are in conflict with each other. For instance, the edge between V_1cV_3 and V_3aV_2 means that the two operations (i.e., V_1 transmits c to V_3 and V_3 transmits a to V_2) cannot be scheduled at the same time. This is due to the constraint that V_3 cannot be the sender and the receiver simultaneously. We will define a set of rules to capture all the constraints on cooperative data dissemination so that very pair of conflicting operations can be identified. In accordance with the assumption in Fig. 2 (i.e., $W_{V_i} = 1$), the weight of each vertex is set to 1. Given the constructed G, we can check that the four shadowed vertices shown in Fig. 3, namely RcV_3, RcV_5, V_1aV_2, and V_4bV_6 are the MWIS of G. Accordingly, the total weight is 4, which is consistent with the maximum weighted gain derived from Fig. 2. The formal description of the above idea is presented as follows.

1) Tentative Schedules (TS): A TS refers to an operation that has the potential to serve one pending request via either I2V or V2V communication. In this regard, we classify the TS into two sets, $\mathrm{TS}_{\mathrm{I2V}}(t)$ and $\mathrm{TS}_{\mathrm{V2V}}(t)$, where $\mathrm{TS}_{\mathrm{I2V}}(t)$ is the set of TSs serving requests via I2V communication, and $\mathrm{TS}_{\mathrm{V2V}}(t)$ is the set of TSs serving requests via V2V communication. To facilitate the analysis, a TS in $\mathrm{TS}_{\mathrm{I2V}}(t)$ is parsed by $R\hat{d}V_r$, where R represents the RSU; $\hat{d}$ represents the data item transmitted from the RSU; and V_r represents the receiver vehicle for $\hat{d}$. Note that $\hat{d}$ has to be in the pending request set of V_r (i.e., $\hat{d} \in PQ_{V_r}(t)$). Similarly, a TS in $\mathrm{TS}_{\mathrm{V2V}}(t)$ is parsed by $V_s\hat{d}V_r$, where V_s represents the sender vehicle; $\hat{d}$ represents the data item to be disseminated by V_s; and V_r represents the receiver vehicle for $\hat{d}$. Note that $\hat{d}$ has to be in the satisfied request set of V_s. In the meantime, it has to be in the pending request set of V_r (i.e., $\hat{d} \in SQ_{V_s}(t) \wedge \hat{d} \in PQ_{V_r}(t)$). As specified, each TS has the potential to serve an outstanding request via either I2V or V2V communication. For instance, as shown in Fig. 2, V_1aV_2 is a TS, which can be interpreted as the potential service by assigning V_1 as the sender and V_2 as the receiver with respect to the data item a. In contrast, V_1aV_3 is not a TS, because V_3 has already received a, and this schedule cannot serve any outstanding request.

2) Conflicting TSs: Different TSs may be in conflict with each other due to the following constraints on cooperative data dissemination. 1) The RSU can only broadcast one data item at a time. 2) Each sender vehicle can only disseminate one data item at a time. 3) A vehicle cannot be both the sender and the receiver at the same time. 4) Data collision happens at receivers. 5) Each vehicle can be only in one of the modes (i.e., I2V or V2V) at a time. The corresponding five rules for identifying conflicting TSs are introduced as follows.

1) If the two TSs are both for I2V communication (i.e., $R\hat{d}V_r \in \mathrm{TS}_{\mathrm{I2V}}(t)$ and $R\hat{d}'V_r' \in \mathrm{TS}_{\mathrm{I2V}}(t)$), but they specify different data items to broadcast (i.e., $\hat{d} \neq \hat{d}'$),

then $R\hat{d}V_r$ is in conflict with $R\hat{d}'V_r{}'$ because the RSU can only broadcast one data item at a time. For example, RcV_3 and RaV_2 are in conflict with each other.

2) If the two TSs are both for V2V communication (i.e., $V_s\hat{d}V_r \in \text{TS}_{\text{V2V}}(t)$ and $V_s{}'\hat{d}'V_r{}' \in \text{TS}_{\text{V2V}}(t)$), but they designate the same sender vehicle to disseminate different data items (i.e., $V_s = V_s{}'$ and $\hat{d} \neq \hat{d}'$), then $V_s\hat{d}V_r$ is in conflict with $V_s{}'\hat{d}'V_r{}'$ because each sender vehicle can only disseminate one data item at a time. For example, V_1cV_3 and V_1aV_2 are in conflict with each other.
3) If the two TSs are both for V2V communication (i.e., $V_s\hat{d}V_r \in \text{TS}_{\text{V2V}}(t)$ and $V_s{}'\hat{d}'V_r{}' \in \text{TS}_{\text{V2V}}(t)$), where one TS designates a vehicle as the sender, while the other TS designates the same vehicle as the receiver (i.e., $V_s = V_r{}'$ or $V_r = V_s{}'$), then $V_s\hat{d}V_r$ is in conflict with $V_s{}'\hat{d}'V_r{}'$ because a vehicle cannot be both the sender and the receiver at the same time. For example, V_1cV_3 and V_3aV_2 are in conflict with each other.
4) If the two TSs are both for V2V communication (i.e., $V_s\hat{d}V_r \in \text{TS}_{\text{V2V}}(t)$ and $V_s{}'\hat{d}'V_r{}' \in \text{TS}_{\text{V2V}}(t)$), but a receiver is the neighbor of both the senders (i.e., $V_r \in N_{V_s{}'}(t)$ or $V_r{}' \in N_{V_s}(t)$), then $V_s\hat{d}V_r$ is in conflict with $V_s{}'\hat{d}'V_r{}'$ because data collision happens at one of the receivers. For example, V_1cV_3 and V_4bV_6 are in conflict with each other.
5) If one TS is for I2V communication (i.e., $R\hat{d}V_r \in TS_{\text{I2V}}(t)$) and the other TS is for V2V communication (i.e., $V_s{}'\hat{d}'V_r{}' \in TS_{\text{V2V}}(t)$), but the receiver in I2V communication is the same with either the sender or the receiver in V2V communication (i.e., $V_r = V_s{}'$ or $V_r = V_r{}'$), then $R\hat{d}V_r$ is in conflict with $V_s{}'\hat{d}'V_r{}'$ because a vehicle can be only in one of the modes (i.e., I2V or V2V) at a time. For example, RcV_3 and V_1cV_3 are in conflict with each other.

3) Constructing the Graph: First, we find the set of TSs by traversing both the retrieved and outstanding data items for each vehicle. Then, the undirected graph G can be constructed by the following procedures: 1) Create a vertex for each TS. 2) Set the weight of each vertex to the weight of the receiver vehicle in the corresponding TS. 3) For any two conflicting TSs, add an edge between the two corresponding vertices. Apparently, G can be constructed in polynomial time. There is a one-to-one mapping between each vertex and each TS. Any two nonadjacent vertices in G represent that the two corresponding TSs are not in conflict with each other, and hence their weighted gain can be accumulated, which is equivalent to the summation of the weight of the two corresponding vertices. Overall, the maximum weighted gain of CDS is achieved if and only if the MWIS of G is computed. The above proves that CDS is NP-hard.

VI. Proposed Scheduling Algorithm

We propose an online scheduling algorithm, which is called CDD, for cooperative data dissemination. To enhance overall system performance on data services, simply maximizing the gain of scalability as defined in Section V-A.2 in each scheduling period cannot guarantee global optimal scheduling performance, as it cannot distinguish the urgency of different served vehicles in a particular scheduling period. In view of this, to capture the service urgency and improve the overall scheduling performance, it is expected to give a higher priority for vehicles that have shorter remaining dwell time in the service region. Fig. 4 illustrates an example to give an insight into the algorithm design.

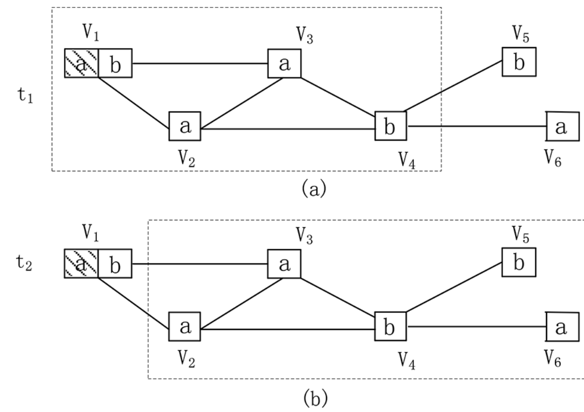


Fig. 4. Scheduling scenarios. (a) At t_1, V_1, V_2, V_3, and V_4 are in the service region. (b) At t_2, V_1 has left the service region, while V_5 and V_6 have arrived.

Fig. 4(a) shows the scheduling scenario at t_1, in which V_1, V_2, V_3, and V_4 ask for b, a, a, and b, respectively. Moreover, V_1 has cached a. The service region is represented by the dotted box. Accordingly, as shown in Fig. 4(b), V_5 and V_6 drive into the service region at t_2, and they ask for b and a, respectively. Meanwhile, V_1 has left. When considering to maximize the number of served vehicles at t_1, the optimal schedule is to set $V_{\text{I}}(t_1) = \{V_4\}$, $V_{\text{V}}(t_1) = \{V_1, V_2, V_3\}$, $d_{\text{I}}(t_1) = b$, $SV(t_1) = \{V_1\}$, and $D(SV(t_1)) = \{a\}$. With such a schedule, V_2 and V_3 retrieve a from V_1, and V_4 retrieves b from the RSU. Nevertheless, V_1 cannot retrieve b at t_1. When V_5 and V_6 arrive at t_2, the optimal schedule in this round is to let V_4 disseminate b to V_5 via V2V communication, and let the RSU disseminate a to V_6 via I2V communication. In this case, V_1 cannot be served since it has left the service region at t_2.

Although the above schedule maximizes the number of served vehicles in each scheduling period, it does not distinguish the service urgency for different vehicles, which results in the failure of serving V_1. In contrast, we consider the following schedule at t_1: $V_{\text{I}}(t_1) = \{V_1, V_4\}$, $d_{\text{I}}(t_1) = \{b\}$, and no vehicles participate into the V2V communication. Although only two vehicles (i.e., V_1 and V_4) are served at t_1, all the remaining vehicles can be served with the following schedule at t_2: $V_{\text{I}}(t_2) = \{V_2, V_3, V_6\}$, $V_{\text{V}}(t_2) = \{V_4, V_5\}$, $d_{\text{I}}(t_2) = a$, $SV(t_2) = \{V_4\}$, and $D(SV(t_2)) = \{b\}$. With such a schedule, V_2, V_3, and V_6 will retrieve a from the RSU, and V_5 will retrieve b from V_4 (note that V_4 has cached b at t_1).

With the above observations, it is expected to estimate the remaining time-slots of vehicles ($Slack_{V_i}(t)$) in the service region, which is computed by $\text{Slack}_{V_i}(t) = \frac{\text{Dis}_{V_i}(t)}{\text{Vel}_{V_i}(t)}$, where $\text{Dis}_{V_i}(t)$ is the current distance to the exit of the service region and $\text{Vel}_{V_i}(t)$ is the current velocity of V_i. Note that there could be a feasible schedule only when $\text{Slack}_{V_i}(t) \geq 1$. To give higher priority on serving more urgent vehicles, the weight of V_i is inversely proportional to its slack time, which is defined as $W_{V_i}(t) = \frac{1}{\text{Slack}_{V_i}(t)^{\alpha}}$, where α ($\alpha > 0$) is the tuning parameter to weight the urgency factor. To give an overview, CDD schedules with the following three steps.

- First, CDD examines all the TSs in both $\mathrm{TS}_{\mathrm{I2V}}(t)$ and $\mathrm{TS}_{\mathrm{V2V}}(t)$ to find every pair of conflicting TSs and compute the weight of each TS.
- Second, CDD constructs the graph G and transforms CDS to MWIS. Then, it selects a subset of TSs based on a greedy method.
- Third, CDD generates the following outputs: 1) the data item $d_{\mathrm{I}}(t)$ to be transmitted from the RSU; 2) the set of receiver vehicles $RV(d_{\mathrm{I}}(t))$ for $d_{\mathrm{I}}(t)$; 3) the set of sender vehicles $SV(t)$ in V2V communication; 4) the set of data items $D(SV(t))$ for each sender vehicle; 5) the set of receiver vehicles $RV(D(SV(t)))$ in V2V communication.

Details of each step are presented as follows, and the pseudocode of CDD is attached in the Appendix.

A. Identify Conflicting TSs and Compute the Weight

This step consists of four operations. First, CDD determines the set of vehicles $V_{\mathrm{RSU}}(t)$, which are in the RSU's coverage (lines1–6 in CDD Step 1). Second, it finds all the TSs in both I2V and V2V communications and constructs $\mathrm{TS}_{\mathrm{I2V}}(t)$ and $\mathrm{TS}_{\mathrm{V2V}}(t)$ (lines 7–22 in CDD Step 1). Third, it identifies any pair of conflicting TSs (lines 23–43 in CDD Step 1). Finally, it computes the weight for each TS (lines 44–50 in CDD Step 1). The implementation is elaborated as follows.

In order to determine $V_{\mathrm{RSU}}(t)$, CDD checks each entry $\langle V_i, Q_{V_i}(t), N_{V_i}(t)\rangle$ maintained in the service queue. Since every vehicle within the RSU's coverage shall update its information in each scheduling period, in case there is no update received for an entry $\langle V_i, Q_{V_i}(t), N_{V_i}(t)\rangle$, it implies that V_i has left the RSU's coverage. In order to construct $TS_{\mathrm{I2V}}(t)$ and $TS_{\mathrm{V2V}}(t)$, CDD examines the entry $\langle V_j, Q_{V_j}(t), N_{V_j}(t)\rangle$ for each $V_j \in V_{\mathrm{RSU}}(t)$. Specifically, for each pending request of V_j (i.e., $\forall q^m_{V_j} \in PQ_{V_j}(t)$), there is a TS: $\{Rq^m_{V_j}V_j\} \in TS_{\mathrm{I2V}}(t)$, which represents that the RSU disseminates $q^m_{V_j}$ to V_j via I2V communication. On the other hand, for each satisfied request of V_j (i.e., $\forall q^m_{V_j} \in SQ_{V_j}(t)$), if $q^m_{V_j}$ is a pending request of V_k (i.e., $q^m_{V_j} \in PQ_{V_k}(t)$) and V_k is the neighbor of V_j (i.e., $V_k \in N_{V_j}(t)$), then there is a TS: $\{V_j q^m_{V_j} V_k\} \in TS_{\mathrm{V2V}}(t)$, which represents that V_j disseminates $q^m_{V_j}$ to V_k via V2V communication. Note that V_k may not necessarily in the RSU's coverage. In order to identify each pair of conflicting TSs, CDD follows the five rules as specified in Section V-B. Finally, for each $V_i \in V_{\mathrm{RSU}}(t)$, CDD updates its the current velocity $Vel_{V_i}(t)$ and its current distance to the exit $\mathrm{Dis}_{V_i}(t)$, so that the remaining dwell time is estimated by $\mathrm{Slack}_{V_i}(t) = \frac{\mathrm{Dis}_{V_i}(t)}{\mathrm{Vel}_{V_i}(t)}$. Given any TS with a receiver vehicle V_r, its weight is computed by $\frac{1}{\mathrm{Slack}_{V_r}(t)^{\alpha}}$.

B. Construct G and Select TSs

This step consists of three operations. First, CDD constructs the graph G based on the mapping rules (lines 1–10 in CDD Step 2). Second, it approximately solves the weighted independent set problem using a greedy method (lines 11–22 in CDD Step 2). Third, it constructs the set of selected TSs (lines 23–27 in CDD Step 2). The implementation is elaborated as follows.

In order to construct G, CDD creates a vertex v for each TS derived from Step 1, and it sets the weight of the corresponding TS to $w(v)$. For each pair of the identified conflicting TSs, they are mapped to the corresponding vertices (e.g., v_i and v_j), and then an edge e_{ij} is added between v_i and v_j. In order to select independent vertices, CDD adopts the greedy method [31], which has been elaborated in Section III.

C. Generate Outputs and Update Service Queue

This step consists of two operations. First, CDD parses each selected TS to make the scheduling decisions, including the determination of $d_{\mathrm{I}}(t)$, $RV(d_{\mathrm{I}}(t))$, $SV(t)$, $D(SV(t))$, and $RV(D(SV(t)))$ (lines 1–11 in CDD Step 3). Second, it updates the service queue by adding the entries for newly arrived vehicles and removing the entries for left vehicles (lines 12–21 in CDD Step 3). The implementation is elaborated as follows.

In order to generate the outputs, CDD parses each TS in $\mathrm{TS}_{\mathrm{selected}}(t)$. Specifically, for any selected TS belonging to I2V communication (i.e., $\forall R\hat{d}V_r \in \mathrm{TS}_{\mathrm{selected}}(t)$), the data item $\hat{d}$ will be the same, because all the selected $R\hat{d}V_r$ are not in conflict with each other. Accordingly, $\hat{d}$ is scheduled to be transmitted from the RSU, which determines $d_{\mathrm{I}}(t)$. Then, the union of V_r from each $R\hat{d}V_r$ is selected as the set of receiver vehicles in I2V communication, which determines $RV(d_{\mathrm{I}}(t))$. On the other hand, for any selected TS belonging to V2V communication (i.e., $\forall V_s\hat{d}V_r \in \mathrm{TS}_{\mathrm{selected}}(t)$), the union of V_s from each $V_s\hat{d}V_r$ forms the set of sender vehicles so that $SV(t)$ is determined. Meanwhile, the union of $\hat{d}$ from each $V_s\hat{d}V_r$ forms the set of data items to be disseminated via V2V communication, and thus $D(SV(t))$ is determined. Last, the union of V_r from each $V_s\hat{d}V_r$ forms the set of receiver vehicles so that $RV(D(SV(t)))$ is determined. In order to maintain the service queue, the system needs to add an entry for each newly arrived vehicle and remove the entry for each left vehicle. Note that for a left vehicle V_i, its entry is removed only when V_i is out of the RSU's coverage (i.e., $V_i \notin V_{\mathrm{RSU}}(t)$) and V_i is not in the neighborhood of any vehicle within the RSU's coverage (i.e., $V_i \notin N_{V_k}(t), \forall V_k \in V_{\mathrm{RSU}}(t)$). This is because if $V_k \in V_{\mathrm{RSU}}(t)$ and $V_i \in N_{V_k}(t)$, then V_i may still have chance to retrieve the data item from V_k via V2V communication, even though $V_i \notin V_{\mathrm{RSU}}(t)$.

To demonstrate the scalability of scheduling, we analyze the algorithm complexity. Suppose there are $|V|$ vehicles and the maximum number of requests submitted by a vehicle is a constant $|Q|$. In Step 1, the upper bound for finding $\mathrm{TS}_{\mathrm{I2V}}$ is $|V| \cdot |PQ|$, where $|PQ|$ represents the number of pending requests of a vehicle and $|PQ| \leq |Q|$. Meanwhile, the upper bound for finding $\mathrm{TS}_{\mathrm{V2V}}$ is $|V| \cdot |SQ| \cdot (|V| - 1) \cdot |PQ|$, where $|SQ|$ represents the number of satisfied requests of a vehicle and $|SQ| \leq |Q|$. Note that $(|V| - 1) \cdot |PQ|$ represents that in the worst case, a vehicle is in the neighborhood of all other vehicles and it requires to check all the neighbors' pending requests to find valid TSs. Therefore, the complexity in Step 1 is $O(|V|^2)$. In Step 2, suppose there are n vertices in the constructed graph. According to the Greedy method, the worst case to find the independent set requires to check all the vertices when all of them are independent with each other, giving the complexity of $O(n)$. On the other hand, according to the mapping principle, we have $n = |TS_{\mathrm{I2V}}| + |TS_{\mathrm{V2V}}|$. In addition, since $|TS_{\mathrm{I2V}}| \leq |V| \cdot |Q|$ and $|TS_{\mathrm{V2V}}| \leq |V| \cdot |Q| \cdot (|V| - 1)$, the complexity in Step 2 is $O(|V|^2)$. In Step 3, it requires to check $|TS_{\mathrm{selected}}|$ vertices to generate the outputs. Since $|TS_{\mathrm{selected}}| \leq |TS_{\mathrm{I2V}}| + |TS_{\mathrm{V2V}}|$, the complexity is $O(|V|^2)$. To sum up, the complexity of CDD

TABLE II
SIMULATION STATISTICS UNDER DIFFERENT TRAFFIC SCENARIOS

Traffic Scenarios	**Mean Arrival Rate** (vehicles / h)			**Mean Velocity** (km / h)			**Mean Density** (vehicles / km)		
	Lane 1	Lane 2	Lane 3	Lane 1	Lane 2	Lane 3	Lane 1	Lane 2	Lane 3
1	1200	1000	800	104.32	86.83	70.59	13.06	13.17	11.79
2	1600	1400	1200	98.75	81.17	65.03	17.69	18.82	18.71
3	2000	1800	1600	91.31	74.64	56.03	23.89	25.34	29.95
4	2400	2200	2000	83.43	60.99	40.37	30.44	38.96	49.50
5	2800	2600	2400	64.77	39.30	28.14	45.96	60.66	64.80

is $O(|V|^2)$, which is reasonable and will not be the hurdle of scheduling scalability.

VII. PERFORMANCE EVALUATION

A. Setup

The simulation model is built based on the system architecture described in Section IV, and it is implemented by CSIM19 [33]. The traffic characteristics are simulated according to the Greenshield's model [34], which is widely adopted in simulating macroscopic traffic scenarios [35]. Specifically, the relationship between the vehicle velocity (v) and the traffic density (k) is represented by $v = V^f - \frac{V^f}{K^j} \cdot k$, 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 that causes the traffic jam). Three lanes are simulated, and the free-flow speeds of the three lanes are set to $V_1^f = 120$ k/h $V_2^f = 100$ k/h, and $V_3^f = 80$ k/h, respectively. The same jam density K^j is set for each lane, which is 100 vehicles/km. Consider that all the vehicles drive in the same direction and the arrival of vehicles in each lane follows the Poisson process. In order to evaluate the system performance under different traffic workloads, a wide range of vehicle arrival rates is simulated. Given a specific vehicle arrival rate on each lane, the corresponding vehicle velocities and vehicle densities are also collected. Detailed traffic statistics are summarized in Table II.

The communication characteristics are simulated based on DSRC. In particular, the radius of RSU's coverage is set to 300 m, and the V2V communication range is set to 150 m. We do not specify absolute values of the data size and the wireless bandwidth, but setting the scheduling period as 1 s. The feasibility of such a setting has been discussed in Section III. The database size is set to 100. Each vehicle may submit a request at random time when passing through the RSU. The total number of submitted requests is uniformly distributed from 1 to 7. The data access pattern follows the Zipf distribution [36] with the parameter $\theta = 0.7$. Specifically, the access probability of a data item d_i is computed by $\frac{(1/i)^\theta}{\sum_{j=1}^{|D|} (1/j)^\theta}$, where $|D|$ is the size of the database.

We implement two well-known algorithms for performance comparison. One is First Come First Served (FCFS) [37], which broadcasts data items according to the arrival order of requests. The other is Most Requested First (MRF) [38], which broadcasts the data item with the maximum number of pending requests. As discussed in Section II, none of previous studies have considered the scheduling for cooperative data dissemination, which can coordinate between I2V and V2V communication at the same time. Therefore, although FCFS and MRF can be only applied for I2V communication, they are the closest solutions for comparison. In addition, to be elaborated in performance analysis, they are also competitive solutions to the stated problem. Note that all the algorithms are compared with the same timeline (i.e., the same time-slot of each scheduling period) regardless whether they can support the hybrid of I2V and V2V communications. The tuning parameter α of CDD is set to 4%, which gives it the best performance in the default setting.

B. Metrics

We design the following metrics to quantitatively analyze the algorithm performance.

- Gain of scalability: It is the total number of vehicles that are served via either I2V or V2V communication in a scheduling period.
- I2V broadcast productivity: Given the number of data items broadcast from the RSU (n_r), and the total number of served requests via I2V communication (n_{rs}), the I2V broadcast productivity is computed by n_{rs}/n_r. A high I2V broadcast productivity implies that the algorithm is good at exploiting the broadcast effect and it is able to utilize the RSU's bandwidth more efficiently.
- Distribution of gains: This metric partitions the served requests into two sets. One set contains those requests served by the RSU, and the other set contains the requests served by neighboring vehicles. The proportion of each set reflects the contribution of I2V and V2V communications to the overall performance.
- Service ratio: Given the total number of served requests (n_s) and the total number of submitted requests (n) by all vehicles, the service ratio is computed by n_s/n.
- Service delay: It measures the waiting time of served requests, which is the duration from the instance when the request is submitted to the time when the corresponding data item is retrieved.

C. Simulation Results

Fig. 5 shows the gain of scalability of algorithms under different traffic scenarios. The ID of each traffic scenario (x-axis) corresponds to the index number in Table II. As shown, a larger traffic scenario ID corresponds to a higher vehicle arrival rate on each lane. In other words, the traffic workload is the lightest in scenario 1, and it is getting heavier in the subsequent scenarios. For FCFS and MRF, which schedule via pure I2V communication, their gain of scalability is the average number of vehicles served via I2V communication in each scheduling period. In contrast, CDD is dedicated to striking a balance between I2V and V2V communications for data services, so that

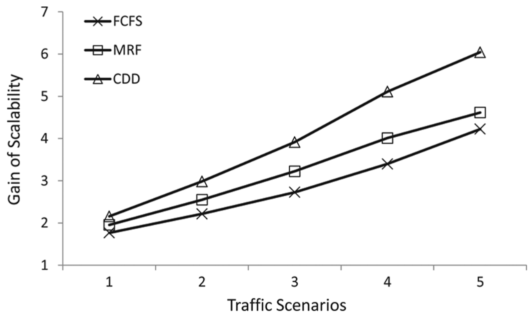


Fig. 5. Gain of scalability under different traffic scenarios.

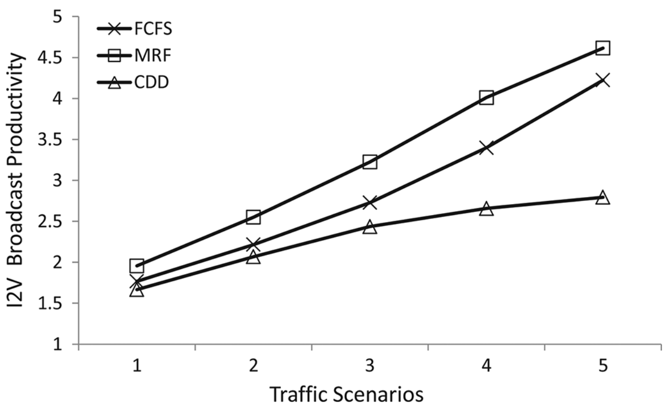


Fig. 6. I2V broadcast productivity under different traffic scenarios.

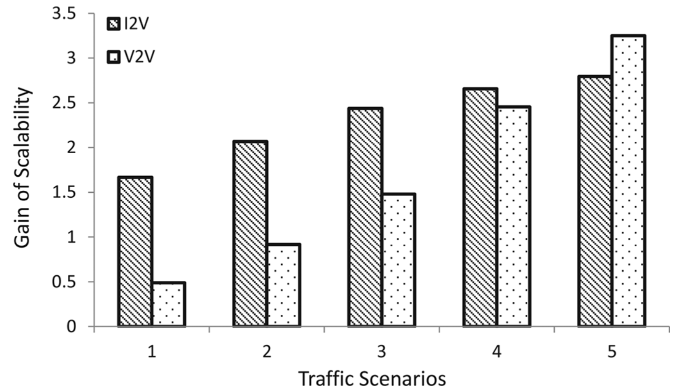


Fig. 7. Distribution of gains under different traffic scenarios.

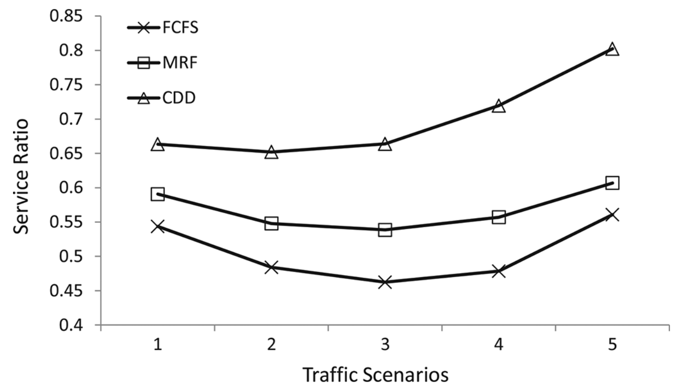


Fig. 8. Service ratio under different traffic scenarios.

the total number of served vehicles can be maximized. As shown in Fig. 5, CDD outperforms other algorithms significantly, especially in heavier traffic workload scenarios. This result demonstrates the superiority of CDD on enhancing system scalability.

Fig. 6 shows the I2V broadcast productivity of algorithms under different traffic scenarios. As demonstrated in previous work [24], the effectiveness of broadcast effect increases by giving preference of scheduling hot data items. MRF always schedules the data item with the most pending requests. Accordingly, it achieves the highest I2V broadcast productivity. CDD also considers the popularity of data items. However, it even ranks behind FCFS, which does not consider data popularity in scheduling. This is because CDD considers not only the number of vehicles that can be served via I2V communication, but also the cooperation of vehicles on data sharing via V2V communication. Vehicles in the V2V mode cannot retrieve any data item from the RSU at the same time, which disperses the I2V broadcast productivity. The result demonstrates that pure I2V communication algorithms can better exploit the RSU's broadcast effect, and CDD would not be able to add significant benefits to the performance without proper coordination between I2V and V2V data dissemination.

Fig. 7 examines the distribution of gains contributed by I2V and V2V communications for CDD. As observed, when the traffic workload is light, most vehicles are served via I2V data dissemination. This is because the vehicle density on each lane is low (i.e., as shown in Table II, only around 13 vehicles/km on each lane in scenario 1). Therefore, with few neighbors of each vehicle, the chance to retrieve an interested data item via V2V communication is slim. With an increase of the traffic workload, the contribution of V2V data dissemination increases notably. This makes sense due to the following reasons. First, the vehicle density is getting higher in a heavier traffic workload environment, resulting more neighbors of each vehicle. Accordingly, the chance to have common requests among neighboring vehicles is higher. Note that these common requests of different vehicles may be submitted at different times. Therefore, it is not likely to serve all of them via a single I2V broadcast, leaving the rest of the requests to have higher possibility to be served via V2V communication. Second, a higher traffic workload also causes longer dwell time of vehicles in the service region. This gives a higher chance for each vehicle to retrieve more requested data items, which gives higher opportunity for V2V data sharing. Last, there are more requests submitted by vehicles asking for different data items when the traffic workload is getting heavier, but only one data item can be broadcast from the RSU in each scheduling period. This limits the portion of contribution via I2V communication. In contrast, by appropriately exploiting the spatial reusability, multiple data items can be disseminated via V2V communication simultaneously without conflicting, which enhances the portion of V2V contribution in a heavy traffic scenario. To sum up, CDD is able to strike a balance between I2V and V2V communications on data services.

Fig. 8 shows the service ratio of algorithms under different traffic scenarios. As observed, the service ratio all the algorithms decline to a certain extent when the traffic workload starts to get heavier. When the traffic workload keeps increasing, the service ratio of all the algorithms is getting higher. The reasons are explained as follows. At the beginning (i.e., in scenario 1), although vehicles pass through the service region with pretty high velocities due to the low density, the system can still achieve reasonable good performance due to the small number

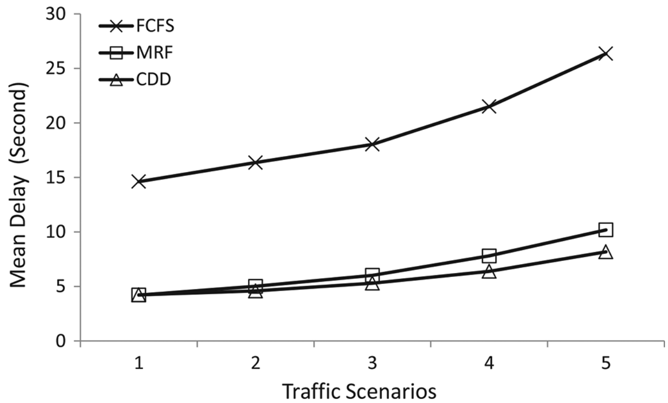


Fig. 9. Service delay under different traffic scenarios.

of total submitted requests. When the vehicle arrival rate starts to increase (i.e., in scenario 2), although the velocity drops accordingly, the increased number of requests dominates the performance, which results in the decline of the service ratio. When the vehicle velocity keeps dropping in a heavier traffic workload environment, the long dwell time of vehicles gradually dominates the performance. Accordingly, the service ratio is getting higher. As shown, CDD outperforms other algorithms significantly in all scenarios. Note that although this work only focuses on data dissemination from a single RSU, it is straightforward to extend the solution and further enhance system performance when multiple RSUs can cooperate to provide data services.

Fig. 9 shows the service delay of algorithms under different traffic scenarios. Although CDD performs closely to MRF in light traffic scenarios, it gradually achieves shorter service delay when the traffic workload is getting heavier. This is because as analyzed in Fig. 7, the benefit of V2V communication achieved by CDD is more significant in a heavy traffic environment. Furthermore, note that the mean service delay is derived from all the served requests. As demonstrated in Fig. 8, CDD serves more requests than both MRF and FCFS in all scenarios, which further demonstrates the superiority of CDD as it is not trivial to achieve both shorter service delay and higher service ratio.

VIII. Conclusion and Future Work

In this work, we present a data dissemination system via the hybrid of I2V and V2V communications and discuss the scheduling challenges arising in such an environment. We give an intensive analysis on both the requirement and the constraint on data dissemination in hybrid vehicular communication environments. On this basis, we give a formal description of the cooperative data scheduling problem, CDS. We prove that CDS is NP-hard from the reduction of MWIS. An online scheduling algorithm CDD is proposed to enhance the data dissemination performance by best exploiting the synergy between I2V and V2V communications. In particular, CDD makes scheduling decisions by transforming CDS to MWIS and approximately solving MWIS using a greedy method. It enables the centralized scheduling at the RSU, which resembles the concept of software defined network in VANETs. We build a simulation model based on realistic traffic and communication characteristics. The simulation results under a wide range of traffic workloads demonstrate the superiority of CDD.

In our future work, we will extend application scenarios by considering the coordination of a set of RSUs, which resembles the SDN concept of "logically centralized" control in a distributed network. Also, the current centralized RSU model does not support multihop V2V communication. In future solutions, RSUs can direct multihop communication among vehicles in several consecutive steps. Furthermore, to better exploit the benefit of V2V communication, the current model can be further enhanced by scheduling vehicles for retrieving their unrequested data items so that it has higher potentiality of data sharing among vehicles. Finally, the impacts of data dissemination at MAC and physical layers are expected to be examined to validate the model in realistic vehicular communication environments.

Appendix
Pseudocode of CDD

CDD Step 1: Identify Conflicting TSs and Compute the Weight

$V_{\mathrm{RSU}}(t) \leftarrow \emptyset$;
for each $V_i \in V(t)$ **do**
  if RSU receives the periodical update from V_i **then**
    $V_{\mathrm{RSU}}(t) \leftarrow V_{\mathrm{RSU}}(t) \cup \{V_i\}$;
  end if
end for
$TS_{\mathrm{I2V}}(t) \leftarrow \emptyset$;
$TS_{\mathrm{V2V}}(t) \leftarrow \emptyset$;
for each $V_j \in V_{\mathrm{RSU}}(t)$ **do**
  for each $q_{V_j}^m \in Q_{V_j}(t)$ **do**
    if $q_{V_j}^m \in PQ_{V_j}(t)$ **then**
      $TS_{\mathrm{I2V}}(t) \leftarrow TS_{\mathrm{I2V}}(t) \cup \{Rq_{V_j}^m V_j\}$;
    end if
    if $q_{V_j}^m \in SQ_{V_j}(t)$ **then**
      for each $V_k \in N_{V_j}(t)$**do**
        if $q_{V_j}^m \in PQ_{V_k}(t)$ **then**
          $TS_{\mathrm{V2V}}(t) \leftarrow TS_{\mathrm{V2V}}(t) \cup \{V_j q_{V_j}^m V_k\}$;
        end if
      end for
    end if
  end for
end for
for any $R\hat{d}V_r \in TS_{\mathrm{I2V}}(t)$ and $R\hat{d}'V_r' \in TS_{\mathrm{I2V}}(t)$ **do**
  if $\hat{d} \neq \hat{d}'$ **then**
    $R\hat{d}V_r$ is in conflict with $R\hat{d}'V_r'$;
  end if
end for
for any $V_s\hat{d}V_r \in TS_{\mathrm{V2V}}(t)$ and $V_s'\hat{d}'V_r' \in TS_{\mathrm{V2V}}(t)$ **do**
  if $V_s = V_s'$ and $\hat{d} \neq \hat{d}'$ **then**
    $V_s\hat{d}V_r$ is in conflict with $V_s'\hat{d}'V_r'$;
  end if
  if $V_s = V_r'$ or $V_r = V_s'$ **then**
    $V_s\hat{d}V_r$ is in conflict with $V_s'\hat{d}'V_r'$;
  end if
  if $V_r \in N_{V_s'}(t)$ or $V_r' \in N_{V_s}(t)$ **then**
    $V_s\hat{d}V_r$ is in conflict with $V_s'\hat{d}'V_r'$;
  end if
end for
for any $R\hat{d}V_r \in TS_{\mathrm{I2V}}(t)$ and $V_s'\hat{d}'V_r' \in TS_{\mathrm{V2V}}(t)$**do**
  if $V_r = V_s'$ or $V_r = V_r'$ **then**

$R\hat{d}V_r$ is in conflict with $V_s'\hat{d}'V_r'$
end if
end for
for each $V_i \in V_{\mathrm{RSU}}(t)$ **do**
Update $Dis_{V_i}(t)$ and $Vel_{V_i}(t)$;
end for
for each TS with receiver vehicle V_r **do**
$Slack_{V_r}(t) \leftarrow \frac{Dis_{V_r}(t)}{Vel_{V_r}(t)}$;
$weight_{TS} \leftarrow \frac{1}{Slack_{V_r}(t)^{\alpha}}$;
end for

CDD Step 2: Construct G and select TSs

for each $TS \in TS_{\mathrm{I2V}}(t) \cup TS_{\mathrm{V2V}}(t)$ **do**
Create a vertex v in G;
$w(v) \leftarrow weight_{TS}$;
end for
for each pair of conflicting TSs **do**
Map the two TSs to the corresponding vertices v_i and v_j;
Add an edge e_{ij} in G;
$d(v_i) \leftarrow d(v_i) + 1$;
$d(v_j) \leftarrow d(v_j) + 1$;
end for
$V_{\mathrm{selected}}(t) \leftarrow \emptyset$;
$max \leftarrow 0$;
while $V(G) \neq \emptyset$ **do**
for each $v_i \in V(G)$ **do**
if $max < w(v_i)/(d(v_i)+1)$ **then**
$max \leftarrow w(v_i)/(d(v_i)+1)$;
$v_{\mathrm{selected}} \leftarrow v_i$;
end if
end for
$V_{\mathrm{selected}}(t) \leftarrow V_{\mathrm{selected}}(t) \cup \{v_{\mathrm{selected}}\}$;
$G \leftarrow G[V - N^{+}(v_{\mathrm{selected}})]$;
end while
$TS_{\mathrm{selected}}(t) \leftarrow \emptyset$
for each $v \in V_{\mathrm{selected}}(t)$ **do**
Map v to the corresponding TS;
$TS_{\mathrm{selected}}(t) \leftarrow TS_{\mathrm{selected}}(t) \cup \{TS\}$
end for

CDD Step 3: Generate outputs and update service queue

for each $R\hat{d}V_r \in TS_{\mathrm{selected}}(t)$ **do**
if $d_{\mathrm{I}}(t)$ has not yet been determined **then**
$d_{\mathrm{I}}(t) \leftarrow \hat{d}$;
end if
$RV(d_{\mathrm{I}}(t)) \leftarrow RV(d_{\mathrm{I}}(t)) \cup \{V_r\}$;
end for
for each $V_s\hat{d}V_r \in TS_{\mathrm{selected}}(t)$ **do**
$SV(t) \leftarrow SV(t) \cup \{V_s\}$;
$D(SV(t)) \leftarrow D(SV(t)) \cup \{\hat{d}\}$;
$RV(D(SV(t))) \leftarrow RV(D(SV(t))) \cup \{V_r\}$;
end for
if RSU receives an update from a newly arrived V_i **then**
$V(t) \leftarrow V(t) \cup \{V_i\}$;
end if
for each $V_i \in V(t)$ **do**
if no periodical update is received from V_i **then**
if $V_i \notin N_{V_k}(t), \forall V_k \in V_{\mathrm{RSU}}(t)$ **then**
$V(t) \leftarrow V(t) - \{V_i\}$;
end if
end if
end for

Acknowledgment

The authors wish to express their deepest and most sincere gratitude to Prof. Ivan Stojmenovic for his invaluable help and profound influence on this work. They were deeply saddened by his sudden passing away during the final stage of publishing this work. Prof. Ivan Stojmenovic will forever be remembered in their hearts, and this work is dedicated to him.

References

[1] S. K. Gehrig and F. J. Stein, "Collision avoidance for vehicle-following systems," *IEEE Trans. Intell. Transport. Syst.*, vol. 8, no. 2, pp. 233–244, Jun. 2007.

[2] K. Liu, S. H. Son, V. C. Lee, and K. Kapitanova, "A token-based admission control and request scheduling in lane reservation systems," in *Proc. 14th IEEE ITSC*, 2011, pp. 1489–1494.

[3] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *IEEE Trans. Intell. Transport. Syst.*, vol. 13, no. 1, pp. 81–90, Mar. 2012.

[4] W. Kang, K. Kapitanova, and S. H. Son, "RDDS: A real-time data distribution service for cyber-physical systems," *IEEE Trans. Ind. Informatics*, vol. 8, no. 2, pp. 393–405, May 2012.

[5] FCC, "Amendment of the Commission's rules regarding dedicated short-range communication services in the 5.850–5.925 GHz band," FCC report and order 06-110, July 20, 2006.

[6] Y. L. Morgan, "Notes on dsrc & wave standards suite: Its architecture, design, and characteristics," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 4, pp. 504–518, 4th Quart., 2010.

[7] K. Liu and V. C. Lee, "RSU-based real-time data access in dynamic vehicular networks," in *Proc. 13th IEEE ITSC*, 2010, pp. 1051–1056, IEEE.

[8] E. Adell, A. Várhelyi, and M. D. Fontana, "The effects of a driver assistance system for safe speed and safe distance—A real-life field study," *Transport. Res. Part C, Emerging Technol.*, vol. 19, no. 1, pp. 145–155, 2011.

[9] A. Bermejo *et al.*, "Ontology based road traffic management in emergency situations," *Ad Hoc Sensor Wireless Netw.*, vol. 20, no. 1–2, pp. 47–69, 2014.

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

[11] ONF, "Software-defined networking: The new norm for networks," Open Networking Foundation White Paper, 2012.

[12] T. Luo, H.-P. Tan, and T. Q. Quek, "Sensor openflow: Enabling software-defined wireless sensor networks," *IEEE Commun. Lett.*, vol. 16, no. 11, pp. 1896–1899, Nov. 2012.

[13] X. N. Nguyen, "Software defined networking in wireless mesh networks," M.Sc. Thesis, INRIA, UNSA, Sophia-Antipolis, France, Aug. 2012.

[14] Y.-A. Daraghmi, C.-W. Yi, and I. Stojmenovic, "Forwarding methods in data dissemination and routing protocols for vehicular ad hoc networks," *IEEE Netw.*, vol. 27, no. 6, pp. 74–79, Nov.–Dec. 2013.

[15] J. Zhang, Q. Zhang, and W. Jia, "VC-MAC: A cooperative MAC protocol in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 3, pp. 1561–1571, Mar. 2009.

[16] K. Fujimura and T. Hasegawa, "A collaborative MAC protocol for inter-vehicle and road to vehicle communications," in *Proc. 7th IEEE ITSC*, 2004, pp. 816–821.

[17] T. K. Mak, K. P. Laberteaux, and R. Sengupta, "A multi-channel vanet providing concurrent safety and commercial services," in *Proc. 2nd ACM VANET*, 2005, pp. 1–9.

[18] H. Shan and W. Zhuang, "Multihop cooperative communication for vehicular ad hoc networks," in *Proc. 6th IEEE ICST CHINACOM*, 2011, pp. 614–619.

[19] A. M. Vegni and T. D. Little, "Hybrid vehicular communications based on V2V-V2I protocol switching," *Int. J. Veh. Inf. Commun. Syst.*, vol. 2, no. 3, pp. 213–231, 2011.

[20] D. Wu, Y. Zhang, L. Bao, and A. C. Regan, "Location-based crowdsourcing for vehicular communication in hybrid networks," *IEEE Trans. Intell. Transport. Syst.*, vol. 14, no. 2, pp. 837–846, Jun. 2013.

[21] S.-I. Sou, "Advanced detection of selfish vehicles for local file sharing in sparse vehicular networks," *IEEE Commun. Lett.*, vol. 17, no. 5, pp. 880–883, May 2013.

[22] D. Aksoy and M. Franklin, "R×w: A scheduling approach for large-scale on-demand data broadcast," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 846–860, Dec. 1999.

[23] J.-Y. Ng, V. Chung-Sing, and C. Y. Hui, "Client-side caching strategies and on-demand broadcast algorithms for real-time information dispatch systems," *IEEE Trans. Broadcast.*, vol. 54, no. 1, pp. 24–35, Mar. 2008.

[24] J. Chen, V. Lee, K. Liu, G. M. N. Ali, and E. Chan, "Efficient processing of requests with network coding in on-demand data broadcast environments," *Inf. Sci.*, vol. 232, pp. 27–43, 2013.

[25] G. M. N. Ali *et al.*, "Admission control-based multichannel data broadcasting for real-time multi-item queries," *IEEE Trans. Broadcast.*, vol. 60, no. 4, pp. 589–605, Dec. 2014.

[26] C.-J. Chang, R.-G. Cheng, H.-T. Shih, and Y.-S. Chen, "Maximum freedom last scheduling algorithm for downlinks of DSRC networks," *IEEE Trans. Intell. Transport. Syst.*, vol. 8, no. 2, pp. 223–232, Jun. 2007.

[27] Y. Zhang, J. Zhao, and G. Cao, "On scheduling vehicle-roadside data access," in *Proc. 4th ACM VANET'*, 2007, pp. 9–18.

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

[29] K. Liu, H. B. Lim, E. Frazzoli, H. Ji, and V. Lee, "Improving positioning accuracy using GPS pseudorange measurements for cooperative vehicular localization," *IEEE Trans. Veh. Technol.*, vol. 63, no. 6, pp. 2544–2556, Jul. 2013.

[30] F. J. Ros, P. M. Ruiz, and I. Stojmenovic, "Acknowledgment-based broadcast protocol for reliable and efficient data dissemination in vehicular ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 1, pp. 33–46, Jan. 2012.

[31] 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.

[32] C. Campolo, A. Vinel, A. Molinaro, and Y. Koucheryavy, "Modeling broadcasting in IEEE 802.11 p/wave vehicular networks," *IEEE Commun. Lett.*, vol. 15, no. 2, pp. 199–201, Feb. 2011.

[33] H. Schwetman, "Csim19: A powerful tool for building system models," in *Proc. 33rd IEEE WSC*, 2001, pp. 250–255.

[34] C. F. Daganzo*, Fundamentals of Transportation and Traffic Operations*. Bingley, U.K.: Emerald Group, 1997.

[35] P. Edara and D. Teodorović, "Model of an advance-booking system for highway trips," *Transport. Res. Part C, Emerging Technol.*, vol. 16, no. 1, pp. 36–53, 2008.

[36] G. Zipf*, Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Cambridge, MA, USA: Addison-Wesley, 1949.

[37] J. Wong and M. H. Ammar, "Analysis of broadcast delivery in a videotex system," *IEEE Trans. Comput.*, vol. C-100, no. 9, pp. 863–866, Sep. 1985.

[38] J. W. Wong, "Broadcast delivery," *Proc. IEEE*, vol. 76, no. 12, pp. 1566–1577, Dec. 1988.

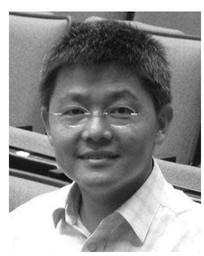
Kai Liu (S'07–M'12) received the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2011.

He is currently an Assistant Professor with the College of Computer Science, Chongqing University, Chongqing, China. From 2010 to 2011, he was a Visiting Scholar with the Department of Computer Science, University of Virginia, Charlottesville, VA, USA. From 2011 to 2014, he was a Postdoctoral Fellow with Nanyang Technological University, Singapore; City University of Hong Kong; and Hong Kong Baptist University, Hong Kong. His research interests include mobile computing, intelligent transportation systems, and vehicular cyber-physical systems.

Joseph Kee-Yin Ng (M'91–SM'00) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 1986, 1988, and 1993, respectively.

He joined Hong Kong Baptist University (HKBU), Hong Kong, in 1993, and is a Professor and the Director of the Research Centre for Ubiquitous Computing in the Department of Computer Science. He is also the programme coordinator of the Computer Science degree program and introduced Health Information Technology and Health Informatics into the undergraduate as well as the graduate programs in HKBU. He has obtained two patents and published over 150 technical papers in journals and conferences. His current research interests include real-time and embedded systems, multimedia communications, and ubiquitous/pervasive computing.

Prof. Ng served as Steering Chair, Program Chair, and General Chair for numerous International Conferences, as well as an Associate Editor and member of the Editorial Board of international journals. He served as the Region 10 Coordinator for the Chapter Activities Board of the IEEE Computer Society and was the Coordinator of the IEEE Computer Society Distinguished Visitors Program (Asia/Pacific). He has been a member of the IEEE Computer Society since 1991. He has been an Exco-member, General Secretary, Vice-Chair, Chair, and now a past Chair and Exco-member for the IEEE Hong Kong Section Computer Chapter. He is a member of the IEEE Communication Society, the Association for Computing Machinery (ACM), Hong Kong Computer Society, and the Founding Member and Exco-Member, Treasurer, and Vice-Chair for the Internet Society (ISOC)-Hong Kong Chapter. He has also been a Director of the Hong Kong Internet Registration Corporation Limited (HKIRC). He received numerous awards and certificates of appreciation from the IEEE, IEEE Region 10, IEEE Computer Society, and from the IEEE Hong Kong Section for his leadership and services to the ICT industry.



Victor C. S. Lee (M'92) received the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 1997.

He is currently an Assistant Professor with the Department of Computer Science, City University of Hong Kong. His research interests include data dissemination in vehicular networks, real-time databases, and performance evaluation.

He is a member of the Association for Computing Machinery (ACM) and the IEEE Computer Society. He has been the Chairman of the IEEE Hong Kong Section Computer Chapter in 2006–2007.

Sang Hyuk Son (M'85–SM'98–F'13) received the B.S. degree in electronics engineering from Seoul National University, Seoul, Korea, in 1976, the M.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1978, and the Ph.D. degree in computer science from the University of Maryland, College Park, MD, USA, in 1986.

He is the Dean of the Graduate School and Director of CPS Global Center, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, Korea. He has been a Professor with the Computer Science Department, University of Virginia, Charlottesville, VA, USA, and a WCU Chair Professor with Sogang University, Seoul, Korea. He has been a Visiting Professor with KAIST; City University of Hong Kong, Hong Kong; Ecole Centrale de Lille, Villeneuve-d'Ascq, France; Linkoping University, Linkoping, Sweden; and the University of Skovde, Skovde, Sweden. His research has been funded by the Korean Government, National Research Foundation, US National Science Foundation, DARPA, Office of Naval Research, Department of Energy, National Security Agency, and IBM. His research interests include cyber-physical systems, real-time and embedded systems, database and data services, and wireless sensor networks. He has written or coauthored over 300 papers and edited/authored four books in these areas.

Prof. Son is a member of both the Korean Academy of Science and Technology and the National Academy of Engineering of Korea. He has served as the Chair of the IEEE Technical Committee on Real-Time Systems during

2007–2008. He is serving as an Associate Editor for *Real-Time Systems* and the *Journal on Self Computing*, and has served on the Editorial Board of the IEEE TRANSACTIONS ON COMPUTERS and the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. He is a founding member of the ACM/IEEE CPS Week and is serving as a member of the steering committee for RTCSA, Cyber Physical Systems Week, and SEUS. He received the Outstanding Contribution Award from the Cyber Physical Systems Week in 2012.

Ivan Stojmenovic (A'88–M'04–SM'05–F'08) received the Ph.D. degree in mathematics from the University of Zagreb, Zagreb, Croatia, in 1985.

He held regular and visiting positions with the King Abdulaziz University, Jeddah, Saudi Arabia (Distinguished Adjunct Professor); Tsinghua University, Beijing, China; Dalian University of Technology, Dalian, China; Beihang University, Beijing China; Singidunum University, Belgrade, Serbia; University of Ottawa, Ottawa, ON, Canada (Professor); University of Birmingham, Edgbaston, U.K. (Chair in Applied Computing, 2005–2006); School of Information Technology, Deakin University, Melbourne, Australia (Professor); as well as in Japan, USA, France, Mexico, Spain, Hong Kong, Brazil, and Taiwan. He published over 300 different papers and edited seven books on wireless, ad hoc, sensor and actuator networks, and applied algorithms with Wiley.

Prof. Stojmenovic was a Fellow of the IEEE Communications Society class of 2008, a Fellow of the Canadian Academy of Engineering since 2012, and a member of the Academia Europaea (The Academy of Europe) Informatics Section since 2012. He was an IEEE CS Distinguished Visitor 2010–2011 and received a 2012 Distinguished Service Award from the IEEE ComSoc Communications Software TC. He chaired and/or organized more than 60 workshops and conferences and served in over 200 program committees. He was Program Co-Chair at IEEE PIMRC 2008, IEEE AINA 2007, and IEEE MASS 2004 and 2007, was Workshop Chair at IEEE ICDCS 2013, IEEE INFOCOM 2011, IEEE MASS 2009, and ACM MobiHoc 2007 and 2008, and founded several workshop series. He was Editor-in-Chief of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS from 2010 to 2013, and was founder and Editor-in-Chief of three journals. He was Associate Editor-in-Chief of the *Tsinghua Journal of Science and Technology*, a steering committee member of the IEEE TRANSACTIONS ON EMERGENT TOPICS IN COMPUTING, and an Editor of *IEEE Network*, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON COMPUTERS, *Wireless Networks*, and some other journals. He was on the Thomson Reuters list of highly cited researchers for 2013, having an h-index of 63, a top h-index in Canada for mathematics and statistics, and more than 16 000 citations. He received five Best Paper awards and the Fast Breaking Paper for October 2003 by Thomson ISI ESI. He received the Royal Society Research Merit Award, U.K., in 2006, and the Humboldt Research Award, Germany, in 2012. He was a Tsinghua 1000 Plan Distinguished Professor from 2012 to 2015. He received the Excellence in Research Award of the University of Ottawa in 2009.