Software-Defined Cooperative Data Sharing in Edge Computing Assisted 5G-VANET

Guiyang Luo, Haibo Zhou, Nan Cheng, *Member, IEEE*, Quan Yuan, JingLin Li, Fangchun Yang, and Xuemin Shen, *Fellow, IEEE*

Abstract—It is widely recognized that connected vehicles have the potential to further improve the road safety, transportation intelligence and enhance the in-vehicle entertainment. By leveraging the 5G enabled Vehicular Ad hoc NETworks (VANET) technology, which is referred to as 5G-VANET, a flexible software-defined communication can be achieved with ultra-high reliability, low latency, and high capacity. Many enabling applications in 5G-VANET rely on sharing mobile data among vehicles, which is still a challenging issue due to the extremely large data volume and the prohibitive cost of transmitting such data using 5G cellular networks. This article focuses on efficient cooperative data sharing in edge computing assisted 5G-VANET. First, to enable efficient cooperation between cellular communication and Dedicated Short-Range Communication (DSRC), we first propose a software-defined cooperative data sharing architecture in 5G-VANET. The cellular link allows the communications between OpenFlow enabled vehicles and the Controller to collect contextual information, while the DSRC serves as the data plane, enabling cooperative data sharing among adjacent vehicles. Second, we propose a graph theory based algorithm to efficiently solve the data sharing problem, which is formulated as a maximum weighted independent set problem on the constructed conflict graph. Specifically, considering the continuous data sharing, we propose a balanced greedy algorithm, which can make the content distribution more balanced. Furthermore, due to the fixed amount of computing resources allocated to this software-defined cooperative data sharing service, we propose an integer linear programming based decomposition algorithm to make full use of the computing resources. Extensive simulations in NS3 and SUMO demonstrate the superiority and scalability of the proposed software-defined architecture and cooperative data sharing algorithms.

Index Terms—5G-VANET, software-defined vehicular network, cooperative data sharing, graph theory, integer linear programming

1 Introduction

Automated driving is anticipated bring great benefit to current smart city and human society, including the potential reduction in traffic collisions, labor costs, road and parking spaces, and efficient transportation management [1], [2]. What comes with these benefits are a great number of applications and services, which require not only communication resources but also external computing and storage resources [3]. The former resources can endow vehicles with the ability to exchange a significant volume of data from the Internet or between adjacent vehicles. There exist tremendous vehicular applications that require the exchange of big-volume, Location-based and Latency-constrained Contents (LLC) [4], [5], [6], [7], e.g., High Definition (HD) maps, parking information, real-time traffic information, multimedia for entertainment, advertisements [8], etc. The latter resources can be applied to storing, analyzing, and mining knowledge from vehicular big data [3], [9]. Therefore, cooperation among communication, computing, and storage can boost the flourishing and prosperity of vehicular applications and services.

However, for both categories of resources, there exist challenges when current Vehicular Ad hoc NETworks (VANETs) cater to new requirements in automated driving and big data era. As for the communication resources, VANETs [10], where each vehicle could exchange information with neighboring vehicles through the Dedicated Short-Range Communication (DSRC) [11], significantly enrich the range of applications and services for transportation and entertainment. Nevertheless, there exist several challenges for the transmission of LLC: 1) Current VANETs cannot ensure an anywhere and anytime connection [12], 2) The transmission reliability and latency requirements cannot be fully satisfied, 3) Current VANETs cannot efficiently support the transmission of a large amount of vehicular big data [9]. As for the computing and storage resources, traditionally, Cloud Computing (CC) can be taken into consideration. CC enables ubiquitous access to a shared pool of configurable system resources and higher-level services that can be rapidly provisioned with minimal management effort. However, CC cannot meet the latency requirement and would congest the backbone network when current VANETs

- *G. Luo, Q. Yuan, J. Li, and F. Yang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: {luoguiyang, yuanquan, jlli, fcyang}@bupt.edu.cn.*
- *H. Zhou is with the Department of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China. E-mail: haibozhou@nju.edu.cn.*
- *N. Cheng is with the School of Telecommunications Engineering, State Key Lab of ISN, Xidian University, Xian 710071, China. E-mail: dr.nan.cheng@ieee.org.*
- *X. Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada. E-mail: sshen@uwaterloo.ca.*

Manuscript received 17 Mar. 2019; revised 28 Oct. 2019; accepted 3 Nov. 2019. Date of publication 12 Nov. 2019; date of current version 3 Feb. 2021. (Corresponding author: Jinglin Li.)
Digital Object Identifier no. 10.1109/TMC.2019.2953163

support applications and services for transportation and entertainment [13].

The fifth-generation (5G) mobile technology, combined with complementary communication techniques, i.e., Long Term Evolution (LTE), Super Wi-Fi [14], [15], are very promising to complement VANETs [16]. We refer to this 5G enabled VANETs as 5G-VANET, which guarantees a higher network capacity, an anytime and anywhere connection, and a more reliable and latency-sensitive connection [17]. By leveraging the Mobile Edge Computing (MEC) [18] technology, a significant amount of computing, storage, and control resources will be distributed near the edge in 5G-VANET. MEC is a network architecture concept that enables cloud computing capabilities and an IT service environment at the edge of the cellular network. It is emerging as a solution to the problems of capacity and latency, supplying vehicles with computing and storage resources. Consequently, computing, communication, and storage resources are distributed anywhere in this MEC assisted 5G-VANET system [19], i.e., storage resource can be provisioned by any vehicle and RoadSide Unit (RSU). To enable efficient cooperation and orchestration among these resources, Software-Defined Networking (SDN), with its centralized network programming capabilities and decoupling of the data plane from the control plane, stands out as a natural candidate technology to manage these resources. The control plane offers great convenience for the SDN Controller to collect context information, i.e., the communication capability of each node, the distribution of storage and computing resources. Consequently, a context-aware and throughput-optimal transmission can be scheduled based on the global network information, enabling cooperation among various communication modes, e.g., cooperation between 5G and DSRC. Meanwhile, the centralized network programming capability can reprogram the network to enable efficient cooperation and integration among various resources [20]. Besides, SDN brings flexibility and agility to this MEC assisted 5G-VANET system [21].

This paper is dedicated to cooperative data sharing for context-aware transmission in edge computing assisted 5G-VANET, which is achieved by cooperation of DSRC and cellular communications, and integration of communication and storage resources. We first propose a software-defined cooperative data sharing architecture in 5G-VANET, which decouples the context sensing from data sharing. The cellular communication, with its broad coverage and unicast connection, is applied to collect contextual information, serving as the Control-Data-Plane Interface (CDPI). DSRC, with enormous advantages on data sharing among neighbors, is treated as the data plane. Then, we propose a three-phase algorithm for cooperative data sharing based on graph theory. In the first phase, CDPI is applied to collect contextual information, e.g., the requested and cached data segments for each vehicle, neighborhood relationship, and corresponding channel capacity, etc. In the following phase, based on the collected information, the edge server can schedule a context-aware and efficient content sharing utilizing the abundant computing resources. We aggregate the contextual information into an undirected neighbor graph, then transform it into a directed matched graph, and finally convert it into a conflict graph. The vertices of this conflict graph are tentative transmissions including both transceiver peers and the packets, and the edge reflects the conflicts between tentative transmissions that reflect the wireless communication constraints. Furthermore, we reformulate cooperative data sharing as a Maximum Weighted Independent Set (MWIS) problem based on the constructed conflict graph. In the third phase, vehicles transmit contents following the solution to the formulated MWIS problem. The transmission is carried out in the DSRC, enabling efficient cooperative data sharing among neighbors. In particular, the main contributions of this paper are summarized as follows:

- We propose a software-defined cooperative data sharing architecture in 5G-VANET, which enables efficient cooperation among various communication modes. We decouple context-aware information sensing and data sharing, achieving enhanced efficiency, robustness and agility.
- We model transmission conflicts as a graph and propose a low-complexity procedure to construct the conflict graph. Extensive simulations on NS3 and SUMO show a significant performance boost when the greedy algorithm is applied to solve the formulated MWIS problem, both in terms of complexity and efficiency, compared with the state-of-the-art methods.
- We propose two algorithms considering the characteristics of 5G-VANET to improve the solution obtained by the greedy algorithm. One is a balanced greedy algorithm, which enables a more balanced content distribution in each time slot considering the continuous data sharing. The other aims to fully utilize the computing resources allocated to this data sharing service. To this end, an Integer Linear Programming (ILP) based decomposition algorithm is proposed, by orchestrating branch-and-bound framework and ILP decomposition.

The remainder of this paper is organized as follows: Section 2 reviews related works. Section 3 introduces software-defined data cooperative sharing architecture in 5G-VANET. In Section 4, we reformulate the cooperative data sharing problem as an MWIS problem. In Section 5, we propose a balanced greedy algorithm and an ILP based decomposition algorithm. Section 6 builds the simulation model and evaluates the algorithm performance. Finally, Section 7 concludes this work.

2 Related Work

In this section, we review the existing related works from aspects of data dissemination in heterogeneous networks, SDN based dissemination and architecture of 5G-VANET.

2.1 Data Dissemination in Heterogeneous Networks

Data dissemination has become a hot research topic in vehicular communication, which has attracted much attention from both academic and industry communities [22]. Xing et al. [23] adopted a carry and forward mechanism in large-scale VANETs, with the aid of RSUs and drop-box, which was formulated as a utility-based maximization problem. Liu et al. [24] focused on the real-time cooperative data sharing for neighboring vehicles based on the

evolutionary fuzzy game. The evolution of mobile networks would lead to heterogeneous networks, e.g., LTE, 5G, WiFi, DSRC, etc. Therefore, data sharing in heterogeneous networks has been considered in [25], [26], [27], [28]. The combination of DSRC and LTE can offer vehicle a higher network capacity and an anytime and anywhere connection, provisioning vehicles with abundant communication resources. Considering the location-dependent content delivery, Yuan et al. [25] relied on the selected influential vehicles to effectively migrate cellular traffic to vehicular networks. Gu et al. [26] developed a content sharing approach in D2D based LTE-V2X networks. They jointly considered the data diversity and link quality when scheduling the vehicle to vehicle (V2V) and infrastructure to vehicle (V2I) link. Ucar et al. [27] proposed a hybrid architecture VMaSCLTE, combining IEEE 802.11p-based multihop clustering and the fourth-generation (4G) cellular system, which can achieve a high data packet delivery ratio and low delay while keeping the usage of the cellular architecture at a minimum level. Zhioua et al. [28] proposed a cooperative traffic transmission algorithm in a joint VANET-LTE advanced hybrid network architecture, which elects a gateway to connect the source vehicle to the LTE Advanced infrastructure under the scope of V2I communications. These schemes enable efficient cooperation among cellular communication and DSRC.

2.2 SDN Based Data Dissemination

Inspired by the great success of SDN technology in wireless networking, researchers resort SDN to a more effective and robust data sharing [21], [29], [30], [31], [32], [33]. Our previous work [34] proposed an SDN inspired MAC (sdnMAC) protocol to disseminate period safety massages, which was achieved by the cooperation of RSUs and On-Board Units (OBU). Chen et al. [30] adopted fog computing paradigm to implement software-defined vehicular networks. They developed a dynamic vehicular connection management approach to achieve a quality of service guarantee. Liu et al. [31] studied the network-coding-assisted data dissemination within the coverage of RSU, considering both communication constraints and application requirements in a pure vehicular network. They converted the cooperative data dissemination into an MWIS problem. In addition, they studied cooperative data scheduling under the same scenario, adopting the concept of SDN [32]. However, these approaches only considered the data sharing in a pure VANET, without consideration of heterogeneous networks and integration of storage and communication resources. Furthermore, Liu et al. [21] proposed an SDN-enabled network architecture assisted by MEC, focusing on quick-response by integrating different types of access technologies. Yu et al. [35] presented a new paradigm of 5G-VANET to improve network capacity and system computing capability. They extended the original cloud radio access network (C-RAN) to integrate local cloud services to provide a low-cost, scalable, self-organizing, and effective solution.

Our proposed algorithm is similar to the approach in [32], while there exist several challenges when the algorithm in [32] is applied to our scenario. 1) [32] considered the scheduling only within the coverage of RSU, which is not scalable and cannot be applied to large scale data sharing in VANETs. 2) [32] assumed that the channel capacity is the same for all links, which is impractical. Meanwhile, if the channel capacity is not the same, their approach would lead to a heavy computation overhead, which will be stated in Section 6.1. We have devised a series of schemes to reduce the complexity. 3) [32] shared the data solely based on DSRC, while our proposed approach enables efficient cooperation between cellular communication and DSRC, which is of high efficiency and scalability. 4) [32] solved the scheduling problem solely by a greedy algorithm, and did not consider how to make full use of the computing resources. Besides, this greedy algorithm can only achieve a performance ratio with only $1/\Delta(G)$, where $\Delta(G)$ is the maximum degree of conflict graph (more details can be found in Section 5). However, in this paper, considering the characteristics of 5G-VANET, we propose two algorithms to improve the solution obtained by the greedy algorithm and make full use of the computing resources.

In this paper, we consider the software-defined context-aware networking in edge computing assisted 5G-VANET. Such heterogeneous networks, with different communication modes (cellular communication and DSRC), different caching ability and different computing resources within each node, they should cooperate to enable efficient data sharing. We decouple contextual information sensing and data dissemination, by adopting the SDN concept. The cellular link is applied to collect context information while the DSRC is used to disseminate data among neighbors. This paper analyzes the characteristics of cooperative data sharing in such scenario, and designs a robust and efficient approach for context-aware networking, enabling effective cooperation.

2.3 Architecture of 5G-VANET

Zheng et al. [36] integrated SDN and radio resource virtualization into an LTE system for vehicular networks, proposing a delay-optimal virtualized radio resource scheduling scheme via stochastic learning to reduce the operating expense and capital expenditure. In order to support the increasing traffic and improve HetNet management, an SDN enabled 5G VANET was proposed in [37], where neighboring vehicles are clustered adaptively according to real-time road conditions using SDN's global information gathering and network control capabilities. The most relevant literature of our work is [33]. They proposed a centralized scheduling scheme in 5G-enabled software-defined vehicular networks, where the SDN-based controller makes decisions for data offloading by using the priority manager and load balancer. They focused on centralized management of the traffic routing to balance the load. However, this paper aims at cooperative data sharing through the cooperation of DSRC and cellular communications, and the integration of communication and storage resources.

Our proposed architecture is inspired by SDN. There is a little resemblance between our architecture and the one in [21], [33]. They all propose software-defined vehicular networking methods. However, in [21], they abstracted all the underlying network resources, such as RSUs, vehicles, Base Station (BS) transceivers, and Ethernet interfaces, as SDN switches, and regarded all the wireless interface as the DCPI interface. In [33], only the base stations were abstracted as SDN switches. Meanwhile, they did not

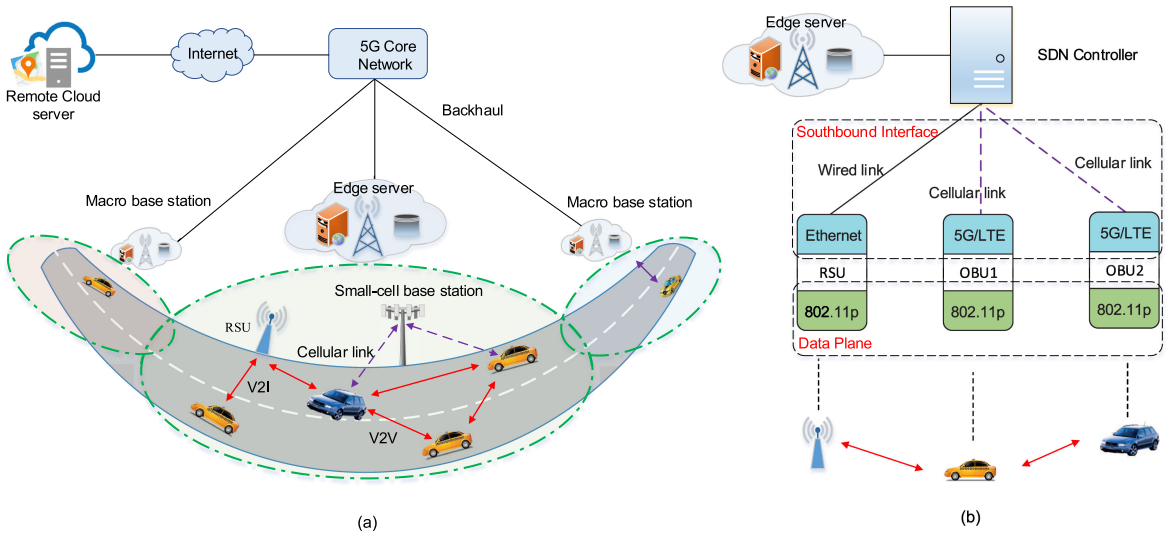


Fig. 1. Software-defined cooperative data dissemination in edge computing assisted 5G-VANET.

consider the cooperative data sharing between vehicles through D2D communication and neglect optimizing the computing resources. In this paper, we abstract OBUs and RSUs as SDN switches, and regard cellular communication as DCPI interface. This abstraction enables efficient cooperation between cellular communication and DSRC. Besides, the architecture in [21] focuses on scalable and quick-response and that in [33] focuses on priority management and load balancer. However, we concentrate on efficient data sharing and cooperation among various communication modes. These different objects further differ details of the proposed architecture.

3 Software-Defined 5G-VANET Architecture and Overview of Methodology

3.1 Software-Defined 5G-VANET Architecture

Consider the scenario of edge computing assisted 5G-VANET, where resources of computing, networking, control and services are distributed at the edge, as shown in Fig. 1a. The set of OBUs and RSUs within the coverage of edge server in the scheduling period are denoted by $\mathcal{V} = \{V_1, V_2, \ldots, V_m\}$ and $\mathcal{R} = \{R_1, R_2, \ldots, R_n\}$, where m and n are the number of OBUs and RSUs, respectively. The set of nodes is denoted by $\mathcal{N} = \{N_1, N_2, \ldots, N_{m+n}\} = \mathcal{V} \cup \mathcal{R}$. The LLC contents need to be disseminated to all vehicles. The popular contents are equally divided into L_0 items of the same length C_0, which is denoted by $\mathcal{D} = \{d_1, d_2, \ldots, d_{L_0}\}$. This dissemination is cooperatively achieved by both cellular communication and DSRC. The V2V and V2I are scheduled in the same service channel for higher spectrum efficiency. RSUs and BSs are interconnected through a wired connection. Each OBU is equipped with a cellular interface and a DSRC interface. Dedicated spectrum has been allocated to DSRC to provision vehicle with information exchange ability. The cellular network can complement DSRC, offering vehicle with an anytime and anywhere connection to the Internet or remote vehicles. Based on this hybrid network, we propose a three-phase context-aware data cooperative dissemination mechanism, which enables efficiently cooperation between cellular communication and DSRC, and integration of communication and storage resources. This architecture is based on SDN. Each vehicle or RSU is regarded as an OpenFlow switch and the SDN Controller is located beside the BS, with a wired connection to computing resources in edge server, which is shown in Fig. 1b. Under this architecture, the data plane is scheduled in DSRC, and the control plane is achieved by cellular communication. The southbound interface is achieved by the cellular interface for OBU and the Ethernet interface for RSU. Based on this software-defined 5G-VANET architecture, we propose a novel three-phase scheme to manage and schedule the communication and storage resources, which is introduced as follows:

Phase I-Context Sensing. RSUs and vehicles are set to the DSRC mode and broadcast their beacons, so that each node is able to sense contextual information. The contextual information includes cached data items, required data items, the set of nodes to which it can transmit and receive data items, and the corresponding channel capacity, by measuring the SIgnal-to-Noise Ratio (SINR) of the beacons. We suppose the channel is symmetric. The channel capacity can be calculated by

$$C_{i,j} = W \log(1 + SINR_{i,j}), \tag{1}$$

where W is bandwidth and $SINR_{i,j}$ is the SINR between node N_i and N_j, $N_i,\ N_j \in \mathcal{N}$. Therefore, the maximum number of data items that can be transmitted between N_i and N_j is denoted by $M_{i,j}$ in the scheduled period T_0, which can be calculated by

$$M_{i,j} = \left\lfloor \frac{C_{i,j} T_0}{C_0} \right\rfloor, \tag{2}$$

where $\lfloor b \rfloor$ is the maximum integer that is no more than b.

Phase II-Southbound Communication and Cooperative Data Sharing. All the OBUs communicate with BS through the cellular link. Specifically, each vehicle informs the edge server with its sensed contextual information, e.g., the list of its current neighbors, the channel capacity of each neighbor's

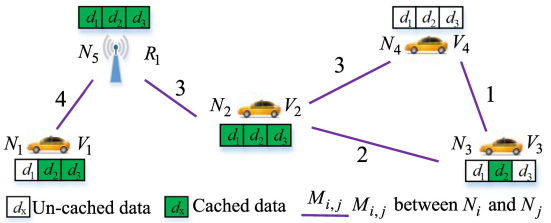


Fig. 2. Example of undirected neighbor graph G_u.

link, the identifiers of the cached and un-retrieved data items, etc. RSUs are connected to edge server via a wired link, and therefore RSUs have the latest and entire data items, serving as the source of data.

By integrating the updated contextual information from $\mathcal{N}$, the edge server can obtain an undirected neighbor graph $G_u = (\mathcal{N}, E_u)$, as shown in Fig. 2. E_u stands for the neighbor relationship among nodes. $(N_i, N_j) \in E_u$ stands for that N_i can transmit and receive data items from N_j, $N_i, N_j \in \mathcal{N}$. The weight on each edge is the maximum number of packets that can be transmitted ($M_{i,j}$). Each node N_i is associated with two sets, one is the set of cached data items, which is denoted by α_i and is showed in green box, the other is the set of data items that have not been cached, which is denoted by β_i and is exhibited in white box, where $\alpha_i \cap \beta_i = \emptyset$. Each node works in a half-duplex mode, i.e., it cannot be a transmitter and receiver simultaneously. Therefore, based on the updated contextual information G_u, the SDN controller needs to schedule the cooperative data sharing. This sharing scheme needs to find out the Two Scheduling Results (TSR), which are as follows:

- The set of nodes to transmit;
- The set of data items that should be transmitted for each sender node.

Based on the scheduling algorithm, the decisions are announced via the cellular link. These scheduling results are transmitted through the cellular link due to the following reasons. On the one hand, RSUs have a shorter communication range and cannot fully cover the roads, while the cellular link has nearly full coverage. On the other hand, the updated contextual information is unicast, which is quite suitable for the attributes of cellular links.

Phase III-Data Plane. Each node participates in either V2I or V2V communication based on the scheduling decisions. Multiple instances of data dissemination may take place simultaneously in this phase.

This three-phase algorithm is processed during a scheduling period (T_0), which we refer to as time slot. The contextual information for each node in Phase II is very small, and can be transmitted through 5G, treating it as ultra Reliable and Low Latency Communication (uRLLC) service. 5G supports a 0.5 ms latency for uRLLC services [38]. Therefore, the time required for Phase II is at most 0.5 ms. we set T_0 to 0.5 seconds. Then, the minimum size of packets can be transmitted in a time slot is 12 Mb/s $*(0.5\text{ s} - 0.5*\ 2\text{ ms}) \approx$ 6 Mb (DSRC supports the data rate from 6 to 27 Mb/s [39], and 12 Mb/s can be achieved for transmissions between neighbors), which is sufficient for typical ITS non-safety applications [40][1] Thus, compared to cooperative data sharing time, the time taken for uploading contextual information via 5G can be ignored. All the nodes are assumed to stay in the same neighborhood in a time slot and the channel capacity in this period is invariant. The proposed three-phase algorithm should be executed periodically to support efficient and ceaseless cooperative data dissemination.

3.2 Overview of Proposed Scheduling Algorithm

The SDN controller schedules the cooperative data sharing based on the sensed contextual information, which maximizes the number of total valid received items. To efficiently solve this scheduling problem, we adopt the graph theory to transform this problem. We first transform the undirected neighbor graph into a directed matched graph, by which a lot of impractical data transmissions can be removed. Then, the constructed directed matched graph is further transformed into an undirected conflict graph, by which communication constraints can be indicated by edges. These two steps can not only reduce the search space but also regulate the constraints. We reformulate the scheduling problem as an MWIS problem based on the constructed conflict graph. Due to the NP-hardness of this problem, we propose a balanced greedy algorithm to solve this problem. Furthermore, we propose an ILP based decomposition algorithm, adopting a branch-and-bound framework to make full use of computing resources at the edge server. The methodology of this paper is shown in Fig. 3.

4 Constructing a Low Complexity Conflict Graph

We first transform the undirected neighbor graph into a directed matched graph to remove impractical transmissions, then into a conflict graph to represent wireless communication constraints. After that, we formulate the cooperative data sharing problem as an MWIS problem. The Context-aware Cooperative Data sharing in Edge Computing assisted 5G-VANET is referred to as CCDEC. Table 1 is a summary of abbreviations and notations used in this paper.

4.1 Conflict Graph Construction

For each sender node $N_i \in \mathcal{N}$, the set of tentative transmit data items is denoted by $\mathcal{T}(N_i)$. For each tentative transmit data items T_k^i $(\in \mathcal{T}(N_i))$, the set of receiver vehicles is denoted by $R(T_k^i)$, $0 \le k < |\mathcal{T}(N_i)|$. As for the data dissemination link N_i transmitting T_k^i to N_j, $N_j \in R(T_k^i)$, the set of valid items can be retrieved by N_j, which is denoted by $V_{i,j}(T_k^i)$ and is calculated by

$$V_{i,j}(T_k^i) = \begin{cases} T_k^i \cap \beta_j & (N_i, N_j) \in E_u \text{ and } |T_k^i| \leqslant M_{i,j}, \\ \emptyset & \text{Otherwise.} \end{cases} \quad (3)$$

1. 5G is anticipated to support three generic services with vastly heterogeneous requirements: enhanced mobile broadband, massive machine-type communications, and ultra-reliable low-latency communications. Service heterogeneity can be accommodated by network slicing, through which each service is allocated resources to provide performance guarantees and isolation from the other services. This paper provides an efficient solution for enhanced mobile broadband applications in 5G-VANET.

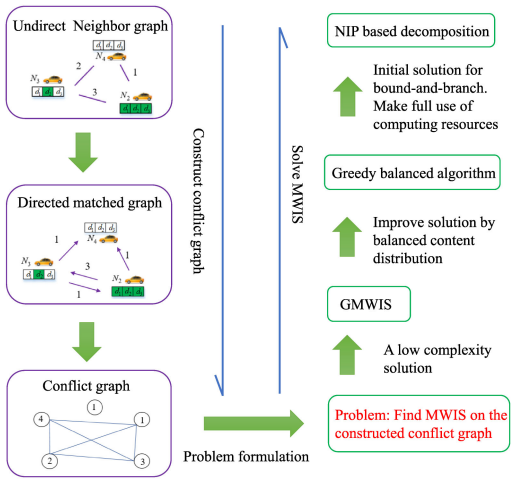


Fig. 3. Overview of methodology.

For a data dissemination link $N_i \rightarrow N_j$, $\mathcal{T}(N_i)$ should be matched with β_j and $M_{i,j}$. As shown in Fig. 2, for the link $N_2 \rightarrow N_5$, $\alpha_2 \cap \beta_5 = \emptyset$, this link is futile, since no valid data items can be transmitted, due to the unmatch between $\mathcal{T}(N_i)$ and β_j. As for the link $N_5 \rightarrow N_1$, $M_{5,1} = 4$, while $|\alpha_5| = 3$, $|T_k^5|$ should not be greater than $|\alpha_5|$. The actual maximum number of packets that can be transmitted from N_5 to N_1 is 3. This is due to the unmatch between T_k^i and $M_{i,j}$ for the link from N_i to N_j.

We denote the maximum number of valid data items that can be transmitted for link $N_i \rightarrow N_j$ as $MV_{i,j}$, which can be calculated by

$$MV_{i,j} = \begin{cases} 0 & \alpha_i \cap \beta_j = \emptyset \text{ or } (N_i, N_j) \notin E_u, \\ \min\{|\alpha_i|, M_{i,j}\} & \text{Otherwise.} \end{cases} \tag{4}$$

By correcting the unmatch in the undirected neighbor graph (G_u), we could transform the undirected neighbor graph

TABLE 1
Summary of Notations

Notations	Descriptions
m, n, L_0	The number of OBUs, RSUs, and items, respectively.
T_0	Scheduling period of the CCDEC algorithm.
ρ_0	Initialization rate of data items.
$\mathcal{V}, \mathcal{R}, \mathcal{N}$	The set of OBUs $\mathcal{V} = \{V_1, \ldots, V_m\}$, RSUs $\mathcal{R} = \{R_1, \ldots, R_n\}$ and nodes $\mathcal{N} = \mathcal{V} \cup \mathcal{R}$, respectively.
$\mathcal{D}$	Items of the popular contents. $\mathcal{D} = \{d_1, \ldots, d_{L_0}\}$
$C_{i,j}$	The channel capacity between N_i and N_j.
$M_{i,j}$	The maximum number of data items that can be transmitted between N_i and N_j.
α_i, β_i	The set of cached data items and required data items, respectively.
$\mathcal{T}(N_i)$	The set of tentative transmit data items for N_i.
$V_{i,j}(T_k^i)$	The set of valid items that can be retrieved by N_j.
$MV_{i,j}$	The maximum number of valid data items that can be transmitted for link $N_i \rightarrow N_j$.
$R(T_k^i)$	The set of neighbors that could successful receive T_k^i.
$\Omega(N_i, G)$	The set of neighbors for node N_i in the graph G.
$V(b)$	The set of valid data items that can be received via $b = N_i \xrightarrow{T_{k^*}^i} \{a\}$.

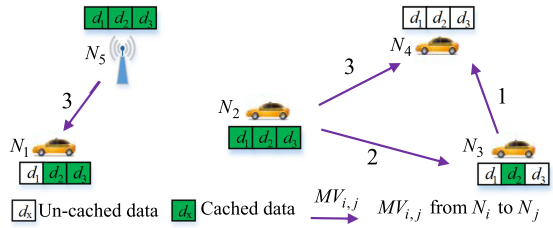


Fig. 4. Example of directed matched graph G_d.

$G_u = (\mathcal{N}, E_u)$ into a directed matched graph $G_d = (\mathcal{N}, E_d)$. This transformation can remove a lot of impractical transmissions, reducing the complexity. The nodes in G_d is the same as G_u, while the weight for the edge $(N_i, N_j) \in E_d$ is $MV_{i,j}$. Therefore, a directed matched graph G_d can be constructed, as shown in Fig. 4.

Based on the directed matched graph, we further transform it into a conflict graph, where the interference and communication constraints are indicated by the edges.

For a node N_i in the graph G, the set of neighbors is denoted by $\Omega(N_i, G)$, which can be calculated by

$$\Omega(N_i, G) = \{N_j | WE_{i,j} > 0, N_j \in \mathcal{N}\}, \tag{5}$$

where $WE_{i,j}$ is the weight for the edge $N_i \rightarrow N_j$ on graph G. $\mathcal{T}(N_i)$ is related to $\Omega(N_i, G_d)$, $MV_{i,j}$, and α_j. T_k^i ($\in \mathcal{T}(N_i)$) should satisfy the following inequalities:

$$\begin{cases} T_k^i \subseteq \alpha_i \\ |T_k^i| \in \{MV_{i,j} \mid N_j \in \Omega(N_i, G_d)\}. \end{cases} \tag{6}$$

For each tentative transmitted data items T_k^i ($\in \mathcal{T}(N_i)$), only the set of neighbors whose channel capacity are greater than or equal to $|T_k^i|$ could successfully receive the transmitted data items. This set of neighbors is denoted by $R(T_k^i)$, which is calculated by

$$R(T_k^i) = \{N_j | MV_{i,j} \geq |T_k^i|, N_j \in \Omega(N_i, G_d)\}. \tag{7}$$

As shown in Fig. 4, for N_2, $\Omega(N_2, G_d) = \{N_3, N_4\}$. According to (6), $T_k^2 \subseteq \{d_1, d_2, d_3\}$ and $|T_k^2| \in \{2, 3\}$, $\mathcal{T}(N_2) = \{\{d_1, d_2\}, \{d_1, d_3\}, \{d_2, d_3\}, \{d_1, d_2, d_3\}\}$. For $T_0^2 = \{d_1, d_2\}$, both N_3 and N_4 can successfully receive the transmitted data items, that is, $R(T_0^2) = \{N_3, N_4\}$. While for $T_3^2 = \{d_1, d_2, d_3\}$, only N_4 can successfully receive the transmitted data items, that is, $R(T_3^2) = \{N_4\}$.

A multicast Tentative Transmission (TT) consists of three indispensable parts: a sender node N_i ($\in \mathcal{N}$), transmitted data items $T_{k^*}^i$ ($\in \mathcal{T}(N_i)$), and a set of receiver nodes a ($a \subseteq \Omega(N_i, G_d)$ and $a \neq \emptyset$). The transmitted data items is $T_{k^*}^i = \arg\max_{T_k^i} \{\sum_{j \in a} |V_{i,j}(T_k^i)|\}$. For each N_i, $T_{k^*}^i$ and a, there exists a multicast TT, which can be denoted as $N_i \xrightarrow{T_{k^*}^i} \{a\}$. For example, as shown in Fig. 4, N_2 has three multicast TTs, which are $N_2 \xrightarrow{\{d_1, d_3\}} \{N_3\}$, $N_2 \xrightarrow{\{d_1, d_2, d_3\}} \{N_4\}$, and $N_2 \xrightarrow{\{d_1, d_3\}} \{N_3, N_4\}$.

Different TTs may be in conflict with each other due to the following constraints on data dissemination: Constraint 1: each node can broadcast only a set of data items at a time. Constraint 2: a node cannot be both the sender and the receiver in the same scheduling period. Constraint 3: data

should not collide at receivers. The corresponding three rules for identifying conflicting TTs are introduced as follows:

Constraint 1. *Two TTs* $N_i \xrightarrow{T^i_{k1}} \{a_{j1}\}$ *and* $N_i \xrightarrow{T^i_{k2}} \{a_{j2}\}$, $T^i_{k1}, T^i_{k2} \in \mathcal{T}(N_i)$, *if* $T^i_{k1} \neq T^i_{k2}$, *this means that* N_i *broadcasts two different packets simultaneously, which is impossible to come true. For example,* $N_2 \xrightarrow{\{d_1,d_3\}} \{N_3\}$ *and* $N_2 \xrightarrow{\{d_1,d_2,d_3\}} \{N_4\}$ *are in conflict with each other.*

Constraint 2. *Two TTs* $N_{i1} \xrightarrow{T^{i1}_{k1}} \{a_{j1}\}$ *and* $N_{i2} \xrightarrow{T^{i2}_{k2}} \{a_{j2}\}$, $T^{i1}_{k1} \in \mathcal{T}(N_{i1})$, $T^{i2}_{k2} \in \mathcal{T}(N_{i2})$, *if* $N_{i1} \in \{a_{j2}\}$ *or* $N_{i2} \in \{a_{j1}\}$, *this means that the node* (N_{i1} *or* N_{i2}) *acts as both the sender and the receiver simultaneously. For example,* $N_2 \xrightarrow{\{d_2,d_3\}} \{N_3\}$ *and* $N_3 \xrightarrow{\{d_2\}} \{N_4\}$ *are in conflict with each other.*

Constraint 3. *Data collision happens when a receiver is the neighbor of more than two senders. Two TTs* $N_{i1} \xrightarrow{T^{i1}_{k1}} \{a_{j1}\}$ *and* $N_{i2} \xrightarrow{T^{i2}_{k2}} \{a_{j2}\}$, $T^{i1}_{k1} \in \mathcal{T}(N_{i1})$, $T^{i2}_{k2} \in \mathcal{T}(N_{i2})$, *if for* $\exists N_{i3} \in \{a_{j1}\}$, $N_{i3} \in \Omega(N_{i2}, G_u)$ *or for* $\exists N_{i3} \in \{a_{j2}\}$, $N_{i3} \in \Omega(N_{i1}, G_u)$, *these two TTs are conflict with each other because data collision happens at one of the receivers. For example,* $N_2 \xrightarrow{\{d_1,d_2,d_3\}} \{N_4\}$ *and* $N_3 \xrightarrow{\{d_2\}} \{N_4\}$ *are in conflict with each other.*

For each TT $N_i \xrightarrow{T^i_{k^*}} \{a\}$, the set of valid data items that can be received is $\sum_{N_j \in a} V_{i,j}(T^i_{k^*})$. The weight for this TT is denoted by $V(N_i \xrightarrow{T^i_{k^*}} \{a\}) = \sum_{N_j \in a} |V_{i,j}(T^i_{k^*})|$. For example, for TT $N_2 \xrightarrow{\{d_1,d_3\}} \{N_3, N_4\}$, its weight $V(N_2 \xrightarrow{\{d_1,d_3\}} \{N_3, N_4\})$ is $|V_{2,3}(\{d_1,d_3\})| + |V_{2,4}(\{d_1,d_3\})| = 4$.

The undirected conflict graph (G_c) can be constructed based on the directed matched graph (G_d). The constructing procedures is as follows: 1) For each node N_i in G_d, find the $\mathcal{T}(N_i)$ and $\Omega(N_i, G_d)$. The multicast TTs for N_i are $N_i \xrightarrow{T^i_{k^*}} \{a\}$, ($a \subseteq \Omega(N_i, G_d)$ and $a \neq \emptyset$), and $T^i_{k^*} = \arg\max_{T^i_k} \{\sum_{j \in a} |V_{i,j}(T^i_k)|\}$. Create a vertex for each multicast TT. 2) Set the weight of each vertex $N_i \xrightarrow{T^i_k} \{N_j\}$ to the number of valid packets that can be received, which is $V(N_i \xrightarrow{T^i_{k^*}} \{a\})$. 3) For any two conflicting TTs, add an edge between the two corresponding vertices. After the above three procedures, the corresponding undirected conflict graph $G_c = (U_c, E_c)$ can be constructed from the directed matched graph G_d, which is shown in Fig. 5.

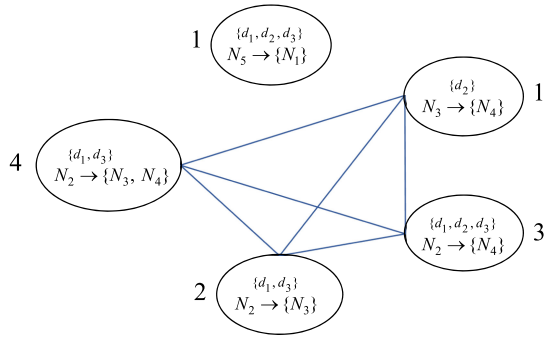


Fig. 5. Example of undirected conflict graph.

4.2 Cooperative Data Sharing

Based on the constructed undirected conflict graph ($G_c = (U_c, E_c)$), we could reformulate the problem as an MWIS problem. For the constructed weighted graph $G_c = (U_c, E_c)$, where U_c is the set of vertices and E_c is the set of edges. Each vertex u_m is associated with a weight, which is denoted by $w(u_m) = V(u_m)$, where $u_m = N_i \xrightarrow{T^i_{k^*}} \{a\}$. An independent set of G_c is a set of vertices, in which any two of them are not adjacent. That is, a subset $X \subseteq U_c$ is an independent set of G_c if for any two vertices $u_i, u_j \in X$, they satisfy $(u_i, u_j) \notin E_c$. The weight of an independent set X, denoted by $W(X)$, is the sum of $w(u_i)$, $u_i \in X$. MWIS is the problem to find the independent set X^* in G_c, such that $W(X^*) \geq W(X), \forall X \subseteq U_c$, which can be denoted by

$$\max \sum_{u_k \in X} w(u_k) \quad s.t. \begin{cases} X \subseteq U_c \\ (u_n, u_m) \notin E_c, \forall u_n, \forall u_m \in X, u_n \neq u_m. \end{cases} \tag{8}$$

After the maximum independent set $X^* \subset U_c$ is derived, we could easily obtain TSR: 1) The set of sender nodes is the union of N_i, where $(N_i \xrightarrow{T^i_{k^*}} N_j) \in X^*$; 2) The set of data items that will be transmitted for sender node N_i is $T^i_{k^*}$.

The proposed scheduling algorithm focuses on software-defined cooperative data dissemination in edge computing assisted 5G-VANET. CCDEC maximizes the total number of received valid data items, enabling cooperative data dissemination between RSUs and OBUs, cooperation between cellular communication and DSRC, and cooperation between communication and computing. To give an overview, CCDEC is proceeded with the following steps:

Step 1. Construct the undirected neighbor graph G_u. By integrating the updated contextual information, including the list of its current neighbors, the channel capacity of each neighbor's link and the identifiers of the cached and un-cached data items, G_u can be constructed in the edge server.

Step 2. Construct the directed matched graph G_d. For each link $N_i \rightarrow N_j$ in G_u, by matching T^i_k with β_j and $M_{i,j}$, G_d can be constructed.

Step 3. Construct the undirected conflict graph G_c. For each node N_i in G_d, obtain all multicast TTs and the weight of each TT. Add an edge if two TTs are in conflict with each other, according to the constraints, as described in Section 4.1.

Step 4. Reformulate the scheduling problem as an MWIS problem on the constructed conflict graph. Then, it selects a subset of TTs X based on the proposed algorithm, which will be narrated in Section 5.

Step 5. Construct TSR from the selected subset X.

We analyze the algorithm complexity of CCDEC. Furthermore, we prove the NP-hard of CCDEC. The number of nodes is $m + n$. In *Step 1*, it is a one-to-one mapping between each updated contextual information and each vertex, the

complexity is $O(m+n)$. In *Step 2*, the upper bound for constructing G_d is $O((m+n)(m+n-1))$, since in the worst case, a node is in the neighbor of all other nodes and it requires to correct the unmatch with all neighbors. In *Step 3*, each node has at most $2^{\Delta_0}-1$ multicast TTs, where $\Delta_0 = \max_{N_i \in \mathcal{N}} \Omega(N_i, G_d)$. Therefore, the number of vertices in G_c is at most $(2^{\Delta_0}-1)(m+n)$. In the worst case, a TT has to check all other TTs to find conflicting TTs, giving the complexity of $O(2^{2\Delta_0}(m+n)^2)$. Due to the topology constraint, the number of neighbor for each node is very small. Besides, the line-of-sight channel is required for this data sharing, especially for transmission of a large amount of contents [41], [42]. Δ_0 can be regarded as a constant and therefore the complexity of *Step 3* is $O(|m+n|^2)$. The above three steps can convert the cooperative data sharing algorithm into an MWIS problem within polynomial time. Overall, the optimal scheduling is achieved if and only if the MWIS is computed. The above proves that cooperative data sharing problem with different channel capacity in 5G-VANET is NP-hard.

4.3 Reducing Complexity of Scheduling

Furthermore, a part of these multicast TTs can be eliminated, as they could not provide a better solution. For example, node N_0 and its neighbors are N_1 and N_2, $\alpha_0 = \{d_1, d_2, d_3\}$, $\beta_1 = \{d_1\}$, $\beta_2 = \{d_1, d_2, d_3\}$, $M_{0,1} = 1$ and $M_{0,2} = 3$. Based on above schemes, there exist three multicast TTs, which are $N_0 \xrightarrow{\{d_1\}} \{N_1\}$ (weight: 1), $N_0 \xrightarrow{\{d_1\}} \{N_1, N_2\}$ (weight: 2), and $N_0 \xrightarrow{\{d_1, d_2, d_3\}} \{N_2\}$ (weight: 3). For these three TTs, the second one could be removed. The reason is that the last multicast TT $N_0 \xrightarrow{\{d_1, d_2, d_3\}} \{N_2\}$ can generate a better solution than the second multicast TT $N_0 \xrightarrow{\{d_1\}} \{N_1, N_2\}$, as the weight of second multicast TT is 2, which is less than the weight (3) of last multicast TT. Removing such multicast TTs could reduce the total number of multicast TTs, reducing the complexity of scheduling without lowering down the performance. A definition is given and then a formal description to this scheme is presented.

Definition 1 (Preference Order). *The scheduling order relation $\succeq_i$ is defined as a complete, reflexive and transitive binary relation over all multicast TTs for each sender.*

Hence, for two multicast TTs of the same node, $N_i \xrightarrow{T^i_{k_1}} \{a_1\} \succeq_i N_i \xrightarrow{T^i_{k_2}} \{a_2\}$ implies that if N_i is selected as the final sender node, N_i prefers transmitting $T^i_{k_1}$ over $T^i_{k_2}$, or at least, N_i prefers transmitting these two data items equally. The preferences of the players could be quantified differently in different applications. In this paper, for any two multicast TTs $N_i \xrightarrow{T^i_{k_1}} \{a_1\} \succeq_i N_i \xrightarrow{T^i_{k_2}} \{a_2\}$, we propose the following preference:

$$N_i \xrightarrow{T^i_{k_1}} \{a_1\} \quad \succeq_i N_i \xrightarrow{T^i_{k_2}} \{a_2\}$$

$$\Leftrightarrow \begin{cases} \sum_{N_j \in a_1} |V_{i,j}(T^i_{k_1})| \geq \sum_{N_j \in a_2} |V_{i,j}(T^i_{k_2})| \\ a_1 \subseteq a_2, a_1, a_2 \subseteq \Omega(N_i). \end{cases} \tag{9}$$

The above preference implies that N_i transmitting $T^i_{k_1}$ can make sure that more valid data items are successfully received than transmitting $T^i_{k_1}$, or at least, equal number of valid data items can be successfully received. The reason is that, although a_2 has a large number of receivers, the number of transmitted data items $|T^i_{k_2}|$ is no larger than $|T^i_{k_1}|$. More receivers will lead to more constraints on transmitted data items. Therefore, more receivers will not always ensure a larger number of successfully received valid data items.

For each N_i, we could get all the multicast TTs $\mathcal{T}(N_i)$, where the number of multicast TTs for N_i is at most $2^{|\Omega(N_i)|}-1$. For any two TTs, e.g., $TT_1(N_i)$ and $TT_2(N_i)$, if $TT_1(N_i) \succeq_i TT_2(N_i)$, $TT_2(N_i)$ should be removed, as $TT_2(N_i)$ will not lead to a better solution. This scheme can decrease the number of multicast TTs, thus reducing the complexity, without degrading the scheduling performance.

The big data volume and high transmission rate for data dissemination have a strong requirement on channel quality. Therefore, the dissemination can be processed in the Line-Of-Sight (LOS) link, which has a higher quality [41], [43]. The number of neighbors that have the LOS link is very small, i.e., $\Omega(N_i, G_d)$ is very small. A small $\Omega(N_i, G_d)$ leads to a small Δ_o and less number of TTs, which guarantees the scalability of the scheduling algorithm.

5 Solving the MWIS Problem

A greedy method presented in [44] is adopted to approximately solve MWIS, which is GMWIS. This greedy algorithm operates as follows. First, it computes the value of $w(u_i)/(d(u_i)+1)$ for each vertex u_i in $G_c = (U_c z, E_c)$, where $d(u_i)$ represents the degree of u_i. Second, it selects the vertex $u_{selected}$ with the maximum value of $w(u_i)/(d(u_i)+1)$. Third, it updates G_c by removing the set of vertices $RE(u_{selected})$, where $RE(u_{selected})$ contains $u_{selected}$ and all of its adjacent vertices. Fourth, it repeats the above operations until there is no vertex remaining in G_c. These selected vertices make up a feasible independent set. Proposition 1 gives the lower bound of GMWIS [44], which proves the feasibility of this algorithm.

Proposition 1 (Lower bound). *GMWIS outputs an independent set of weight at least $\sum_{u_i \in U_c} w(u_i)/(d(u_i)+1)$.*

We denote the weight of a maximum independent set of graph G by $\alpha(G)$. For an independent set algorithm A, $A(G)$ is the weight of the solution obtained by A on graph G. The performance ratio $\rho(A)$ of A is defined by

$$\rho(A) = inf\left\{\frac{A(G)}{\alpha(G)}\right\}, \tag{10}$$

where $inf\{b\}$ is the infimum over b.

The performance ratio indicates the efficiency of an algorithm. The efficiency of GMWIS can be ensured by Proposition 2 [44].

Proposition 2 (Performance ratio). $\rho(GMWIS) = 1/\Delta(G)$, *where $\Delta(G)$ is the maximum degree of graph G.*

Such algorithm could solve the MWIS in each scheduling separately and does not consider the inter-influence of continues scheduling results. For example, in each scheduling, the data items should be uniformly distributed among vehicles, which will be of higher efficiency for the next

scheduling. Based on this consideration, we propose a balanced greedy algorithm.

5.1 Balanced Greedy Algorithm

The requested popular content is big-volume, and with the mobility of vehicles, each vehicle will request new popular content. For example, when each vehicle reaches to a new location, they may need the parking information and the high definition map for that area. Therefore, the scheduling algorithm should be executed periodically. The scheduling decision in current time slot may impact future scheduling. For example, if cached data items between node N_i and its neighbors $\Omega(N_i, G_d)$ are the same, then no data can be shared among these nodes. The reason is that the distribution of data items is unbalanced. The balanced rate for the set of vehicles a caching the set of item μ is defined as $f(a, \mu)$, which can be denoted by

$$f(a,\mu) = \sum_{N_i \in a} \sum_{N_j \in \Omega(N_i, G_d)} \sum_{d_k \in \mu} g(N_j, d_k), \tag{11}$$

where $g(N_j, d_k)$ is defined as

$$g(N_j, d_k) = \sum_{N_l \in \Omega(N_j, G_d)} |d_k \cap \beta_l|. \tag{12}$$

After the greedy algorithm has decided the set of sender nodes and corresponding transmitted items, for each multicast TT $N_i \xrightarrow{T_k^i} \{a\}$, we should replace the transmitted items T_k^i with the set of data items T_k^{i*} that has a higher balanced rate, which can be denoted by

$$T_k^{i*} = \arg\max_{T_k^i} f(a, T_k^i). \tag{13}$$

At the same time, this set of data items cannot reduce the number of valid data items that can be received, i.e., $\sum_{N_j \in a} |V_{i,j}(T_k^i)| \leq \sum_{N_j \in a} |V_{i,j}(T_k^{i*})|$. For example, RSUs cache all the data items, while vehicles cache a very small number of data items. At the beginning of the scheduling, the RSU may have a lot of data items to be disseminated into vehicles. The T_k^i may not be unique, while T_k^{i*} is the set of data items that has the highest balanced rate.

5.2 Integer Linear Programming Based Decomposition

In the practical deployment of CCDEC as a service, a fixed amount of computing resources are allocated to this service. The upper execution time for this algorithm is fixed, as the proposed approach is executed periodically. The number of nodes is constantly in flux. Therefore, when the number of nodes is small, the above two algorithms may lead to a sub-optimal solution and the computing power is wasted. We hope that an algorithm can make full use of the computing power allocated to it. The algorithm can explore a better solution with execution goes on and can be stopped at any time.

For the constructed conflict graph $G_c = (U_c, E_c)$, the weight of vertex $u_i = N_i \xrightarrow{T_k^i} \{a\}$ is the total number of received valid data items, $w(u_i) = \sum_{N_j \in a} |V_{i,j}(T_k^i)|$ and the edge (u_i, u_j), $u_i, u_j \in U_c$ means that u_i is in conflict with u_j. We can formulate the MWIS problem as an ILP problem,

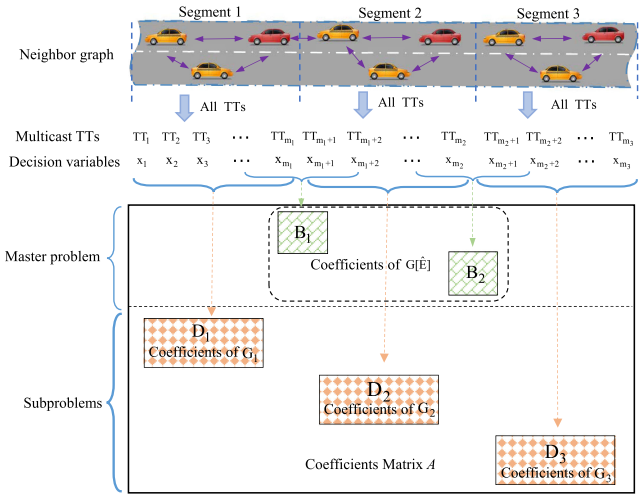


Fig. 6. Illustration of the block-diagonal structure for the sparse coefficient matrix A. A is very sparse, and the value is concentrated on B_i and D_i.

where we have a decision variable $x(u_i)$ for each vertex $u_i \in U_c$ indicating whether u_i is selected as the independent set X or not. $x(u_i) = 1$ means that u_i is selected and $x(u_i) = 0$ means that this vertex is not selected. We have constraints which state that for each edge (u_i, u_j), only one of vertex u_i or u_j can be elected. Therefore, the formulated ILP is as follows:

$$\begin{aligned} &\max \sum_{i=0}^{|U_c|} w(u_i) x(u_i) \\ &s.t. \begin{cases} x(u_i) + x(u_j) \leq 1 \ \text{ if } (u_i, u_j) \in E_c \\ x(u_i) \in \{0, 1\}. \end{cases} \end{aligned} \tag{14}$$

Suppose there exist M conflicting edges and N multicast TTs, where $M = |E_c|$ and $N = |U_c|$. The constraints in (14) can be written in matrix form. We have a matrix $\boldsymbol{A}_{M*N}$, where for the tth conflicting edge (u_i, u_j), $\boldsymbol{A}_{t,i} = 1$, $\boldsymbol{A}_{t,j} = 1$ and $\boldsymbol{A}_{t,k} = 0$ if $k \neq i, j$. Therefore, (14) can be rewritten as

$$\begin{aligned} &\max \ \boldsymbol{W}_{N*1}^T \boldsymbol{X}_{N*1} \\ &s.t. \begin{cases} \boldsymbol{A}_{M*N} \boldsymbol{X}_{N*1} \leq \boldsymbol{1}_{M*1} \\ \boldsymbol{X}_{N*1} \in \{0, 1\}^{N*1}, \end{cases} \end{aligned} \tag{15}$$

where $\boldsymbol{W}_{N*1} = [w(u_0), w(u_1), \ldots, w(u_{N-1})]$ and $\boldsymbol{X}_{N*1} = [x(u_0), x(u_1), \ldots, x(u_{N-1})]$. Such an ILP is NP-hard. As the number of variables is tremendous large, direct solution may lead to a low efficiency, costing a heavy computation overhead. Therefore, we decompose this ILP into a master problem and several subproblems by applying Dantzig-Wolfe decomposition [45], [46]. By carefully orchestrating the problems, the computation complexity can be reduced.

Vehicles are constrained by the road topology, and therefore, the direct matched graph G_d is also shaped by the road topology. Each vehicle has a communication connection only with its neighbors, and thus, the multicast TTs have connection only in the local area for the conflict graph G_c, which can be shown in Fig. 6. This segmentation is equivalent to partitioning the vertex set U_c of the conflict graph G_c into P parts, $Q_1, \ldots, Q_P$. For every $p \in \{1, 2, \ldots, P\}$, we define subgraph $G_p = G_c[U_p]$ with edge set $E_p = E_c[G_p]$.

Edges of G_c whose ends lie in different sets of partition constitute set $\hat{E} = E_c \setminus \bigcup_{p=1}^{P} E_p$, which induces the subgraph $G[\hat{E}]$ with vertex set $\hat{E}$. Therefore, the matrix $\boldsymbol{A}_{M*N}$ has a block-diagonal structure and (15) can be rewritten as

$$\max \sum_{p=1}^{P} \boldsymbol{W}_p^T \boldsymbol{X}_p \tag{16}$$

$$s.t. \begin{cases} \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{B}_2 & \cdots & \boldsymbol{B}_P \\ \boldsymbol{D}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{D}_P \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \\ \vdots \\ \boldsymbol{X}_P \end{bmatrix} \leq \boldsymbol{1}_{M*1} \\ \boldsymbol{X}_{N*1} \in \{0,1\}^{N*1}, \end{cases}$$

where $\boldsymbol{B_p}$ is the matrix of coefficients in inequalities associated with edges $(u_i, u_j) \in \hat{E}$; $\boldsymbol{D}_p$ is the matrix of coefficients in inequalities associated with edges $(u_i, u_j) \in E_p$, $1 \leq p \leq P$; $\boldsymbol{X}_p$ is the pth vector of decision variables associated with vertices $u_i \in U_p$; and $\boldsymbol{W}_p$ is the pth vector of weights associated with vertices $u_i \in U_p$.

Considering the inequalities associated with edges $(u_i, u_j) \in \hat{E}$, we get following ILP:

$$\begin{aligned} &\max \sum_{p=1}^{P} \boldsymbol{W}_p^T \boldsymbol{X}_p \\ &s.t. \begin{cases} \sum_{p=1}^{P} \boldsymbol{B}_p \boldsymbol{X}_p \leq \boldsymbol{1} \\ \boldsymbol{X}_{N*1} \in \{0,1\}^{N*1}. \end{cases} \end{aligned} \tag{17}$$

Applying Dantzig-Wolfe decomposition to the linear relaxation of (17), we obtain the Restricted Master Problem (RMP), a restricted version of the master problem that includes a subset of the columns associated with the extreme points of subproblem polytopes

$$\begin{aligned} &\max \sum_{p=1}^{P} \sum_{j \in J_p} \lambda_{jp} (\boldsymbol{W}_p^T \boldsymbol{X}^{jp}) \\ &s.t. \begin{cases} \sum_{p=1}^{P} \sum_{j \in J_p} \lambda_{jp} (\boldsymbol{B}_p \boldsymbol{X}^{jp}) \leq 1 & (18.1) \\ \sum_{j \in J_p} \lambda_{jp} = 1, \ \forall p \in \{1, \ldots, P\} & (18.2) \\ \lambda_{jp} \geq 0, \ \forall p \in \{1, \ldots, P\}, j \in J_p, \end{cases} \end{aligned} \tag{18}$$

where J_p is the set of integer extreme points of $\{x(u_i) + x(u_j) \leq 1, \forall (u_i, u_j) \in E_p, x(u_i) \in \{0,1\}\}$ associated with generated columns in RMP; $\boldsymbol{X}^{jp}$ is the $|U_p|$ vector that defines extreme point $j \in J_p$; λ_{jp} is the RMP decision variable that corresponds to extreme point $j \in J_p$.

The subproblem $p \in \{1, 2, \ldots, P\}$ has the form

$$\begin{aligned} &Z_p^* = \max (\boldsymbol{W}_p - \boldsymbol{A}_p^T \boldsymbol{\alpha}) \boldsymbol{X}_p \\ &s.t. \begin{cases} \boldsymbol{A}_{M*N} \boldsymbol{X}_{N*1} \leq \boldsymbol{1}_{M*1} \\ \boldsymbol{X}_{N*1} \in \{0,1\}^{N*1}, \end{cases} \end{aligned} \tag{19}$$

where $\boldsymbol{\alpha}$ is the $\hat{E}$-vector of dual variables associated with the rows of constraints (18.1). In this model, x^{jp} is an improving column if $Z_p^* - \beta_p \geq 0$, where β_p is the dual variable associated with the pth convexity constraint (18.2). The role of RMP is to co-ordinate subproblem solutions (through the values of dual variables) to prescribe a solution to the scheduling algorithm on graph G_c. For more details on Dantzig-Wolfe decomposition of ILP, please refer to [45], [46]. Depending on the subgraph structure, the subproblem may also be NP-hard; however, subgraphs are smaller than the original graph G_c, and they will be less challenging, on average, for optimizing algorithms to solve.

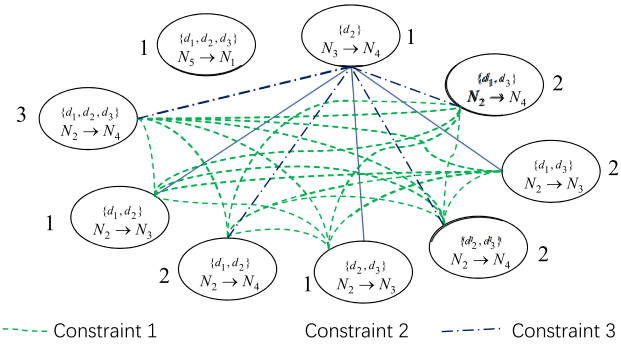


Fig. 7. Example of conflict graph based on unicast TTs.

The GMWIS algorithm would generate a suboptimal scheduling with low complexity. This may lead to wasteful computation power. This decomposition algorithm should improve the scheduling based on the solution obtained by GMWIS. Here we give the basic idea of this approach. First, we can obtain a feasible solution $\boldsymbol{X}^0 = (x_0^0, x_1^0, \ldots, x_{N-1}^0)$ by GMWIS. The corresponding objective function at $\boldsymbol{X}^0$ is $w(\boldsymbol{X}^0) = \sum_{i=0}^{|T_c|} w(u_i) x_i^0$. $\boldsymbol{X}^0$ can be regarded as the initial feasible basis both for the master problem and subproblems. The decomposition method iterates back and forth between the master problem and the subproblems, exchanging information to solve the overall problem to optimality. Each iteration would get a better solution and can be stopped at any time. Therefore, we repeat the iteration until an optimal solution is found out or expiration of execution time. As a consequence, the computation power allocated to this service can be fully utilized.

6 Performance Evaluation

6.1 Theoretical Analysis

Our proposed algorithm CCDEC is similar to the approach in [32]. However, CCDEC has a big difference to the one in [32]. [32] assumes that the channel capacity is the same and creates a unicast Tentative Scheduling (TS, Section 5.2 in [32]. TS and unicast TT refer to the tentative scheduling in [32]) for each possible transmission. For example, as for N_2 shown in Fig. 4, there exist 12 TSs according to [32], which are $N_2 \xrightarrow{\{d_1\}} N_3$, $N_2 \xrightarrow{\{d_1\}} N_4$, $N_2 \xrightarrow{\{d_2\}} N_4$, $N_2 \xrightarrow{\{d_3\}} N_3$, $N_2 \xrightarrow{\{d_3\}} N_4$, $N_2 \xrightarrow{\{d_1, d_2\}} N_3$, $N_2 \xrightarrow{\{d_1, d_2\}} N_4$, $N_2 \xrightarrow{\{d_1, d_3\}} N_3$, $N_2 \xrightarrow{\{d_1, d_3\}} N_4$, $N_2 \xrightarrow{\{d_2, d_3\}} N_3$, $N_2 \xrightarrow{\{d_2, d_3\}} N_4$, and $N_2 \xrightarrow{\{d_1, d_2, d_3\}} N_4$. As for N_2 shown in Fig. 4, there exist 7 TSs based on our directed matched graph shown in Fig. 4, and the corresponding conflict graph is shown in Fig. 7. The larger size of the conflict graph, the more complex to solve the MWIS problem. Furthermore, by applying the Preference Order to refine the remaining 7 TSs generated by [32], only 2 TSs ($N_2 \xrightarrow{\{d_1, d_2, d_3\}} N_4$, $N_2 \xrightarrow{\{d_1, d_3\}} N_3$) are left. However, this

refinement would degrade the performance. As for our proposed approach, there exist 3 multicast TTs for N_2 shown in Fig. 4, and the corresponding conflict graph is shown in Fig. 5. The preference order on our proposed multicast TT would not degrade the performance and can reduce the size of the constructed conflict graph.

By applying GMWIS algorithm to the conflict graph generated by [32], which is shown in Fig. 7, the independent set is $X^*(GMWIS) = \{N_5 \xrightarrow{\{d_1,d_2,d_3\}} N_1, N_2 \xrightarrow{\{d_1,d_2,d_3\}} N_4\}$. Then, the TSR can be derived, with N_5 and N_2 broadcasting $\{d_1, d_2, d_3\}$, where the total received valid data items are 4. We refer to GMWIS algorithm applied to the conflict graph generated by our proposed algorithm (multicast TTs) as GMWIS+. As shown in Fig. 5, the $X^*(GMWIS+) = \{N_5 \xrightarrow{\{d_1,d_2,d_3\}} \{N_1\},\ N_2 \xrightarrow{\{d_1,d_3\}} \{N_3, N_4\}\}$, where the total received valid data items are 5. Therefore, the proposed multicast TTs can not only reduce the scheduling complexity but also improve the scheduling efficiency when applying the greedy algorithm, which is theoretically proved by Observation 1 and Proposition 3.

According to the unicast TS presented in [32], each node has at most $\Delta_0 2^{L_0}$ TSs in our scenario (channel capacity of different links is not the same), where Δ_0 is the maximum number of neighbors for a node. Therefore, the number of vertices in G_c is at most $\Delta_0(m+n)2^{L_0}$. In the worst case, a TT has to check all other TSs to find conflicting TSs, giving the complexity of $O((\Delta_0(m+n)2^{L_0})(\Delta_0(m+n)2^{L_0}-1)) \approx O(\Delta_0^2 2^{2L_0}|m+n|^2)$. Our proposed algorithm achieves a complexity of $O(2^{2\Delta_0}|m+n|^2)$. Obviously, a larger L_0 would output a higher sharing efficiency. Meanwhile, for less interference and better channel link quality [47], this data sharing should be proceeded in a line-of-sight link. Therefore, Δ_0 tends to be very small. In a nutshell, a larger L_0 and smaller Δ_0 would lead a higher data sharing efficiency. Under this scenario, we derive Observation 1, which demonstrates low computation efficiency of our proposed algorithm. Meanwhile, our proposed algorithm would never degrade the performance, which is indicated by Proposition 3.

Observation 1. *The multicast TTs will greatly reduce the size of the constructed conflict graph when we resort to higher efficiency. This means that our proposed algorithm has a lower complexity when we resort to higher efficiency (larger L_0).*

Proposition 3. *The multicast TTs will not degrade the performance of CCDEC.*

Proof. Applying the approach in [32] to generate all the unicast TSs, the scheduling algorithm generates the maximum weighted independent set X^*. We could get the optimal scheduling decisions TSR. For sender N_i, the transmitted data items set is $T^i_{k^*}$, and the intended receivers are a. The $T^i_{k^*}$ must equal to $\arg\max_{T^i_k}\{\sum_{N_j \in a}|V_{i,j}(T^i_k)|\}$, otherwise, this X^* is not optimal, which is contradictory to the fact that X^* is maximum. N_i transmitting $T^i_{k^*}$ to receivers a is the multicast TT $N_i \xrightarrow{T^i_{k^*}} a$. Therefore, the multicast TTs include all the unicast TTs that belong to X^*, which means that the multicast TTs will not degrade the performance of the scheduling algorithm. □

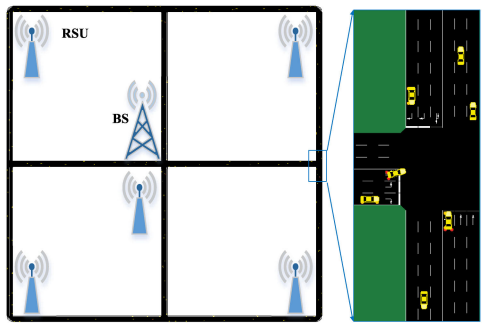


Fig. 8. Simulation map in SUMO.

As compared with the approach in [32], our proposed algorithm can reduce the complexity, without lowering down the performance, which is achieved by a series of schemes. First, transforming the undirected neighbor graph into a directed matched graph can eliminate a lot of impractical transmissions. Second, we refine the vertices for the constructed conflict graph by presenting a novel multicast TT, resulting that the size of the conflict graph is not related to the total data blocks, greatly reducing the size of the conflict graph. Third, we have come up with a novel approach (Preference Order) to remove the set of vertices in the conflict graph that cannot lead to a better solution. These schemes can reduce the size of the constructed conflict graph and do not degrade the performance.

6.2 Simulation Setup

We validate our proposed approach using simulations based on the discrete-event network simulator NS3 and the simulation of urban mobility SUMO. NS3 is adopted to simulate the network (DSRC, cellular communication and SDN), while SUMO serves as the simulator for vehicle mobility. These two simulators are connected by the trace file [48], which records the trajectory of vehicles in SUMO. Then, this file is feed to the Mobility Module in NS3, which can be used to control the movement of vehicles in NS3. For evaluation purposes, we simulate a heterogeneous vehicular network scenario as shown in Fig. 8, in which a BS is deployed at the center, and 5 RSUs are located at the corners and the center, respectively. The BS is connected with a MEC server through a wired link, where there exist a large amount of computing and storage resources. The LLC can be maintained and stored in the MEC server and every vehicle is interested in the content. Each vehicle is equipped with a cellular interface and an 802.11p interface. These two communication modes cooperate to enable higher efficiency.

The simulation map is shown in Fig. 8, and each road segment has its own speed limits. There are three kinds of vehicles, distinguished by maximum acceleration and maximum deceleration, as shown in Table 2. Vehicles are randomly placed on the roads and each vehicle has a random destination ("Random Trips" model). The route of the source to the destination is randomly generated (all the vehicles will not stop until it leaves the area). The movement of the vehicles is affected by vehicle attributes (e.g., acceleration, deceleration), mobility model, and vehicular traffic features (e.g., vehicle density in each road lane, velocity limits, traffic lights,

TABLE 2
Vehicle Type

Vehicle type	Acceleration (m/s^2)	Deceleration (m/s^2)	Possibility
I	8	8	0.2
II	4	4	0.6
III	1	1	0.2

etc.). We record the trajectories of vehicles into a trace file ("Tools/TraceExporter"), which consists of coordination points of all vehicles at the period of 0.5 s.

NS3 is organized into separate modules that are built as a separate software library. NS3 models are abstract representations of real-world objects, protocols, devices, etc., and several modules (libraries) can be linked to conduct a simulation. In our simulations, there exist three kinds of nodes, which are BS, RSU, and vehicles, as shown in Fig. 9. These nodes are implemented as NS3 Nodes in the simulation. Each NS3 Node can be regarded as a physical computer (a container), where multiple Modules can be linked together. The BS is regarded as the SDN controller node (OpenFlow Module). Each RSU is equipped with an 802.11p interface and interconnected with the BS through a wired connection. As for each vehicle, there exist two communication interfaces. We treat the control information transmitted by CDPI as 5G uRLLC messages and assume all the control information can successfully transmit to the controller and deliver to vehicles. Therefore, we create a virtual interface for each vehicle and the communication channel between vehicle and RSU is achieved via logic communication. The rationality of such consideration is that 5G can fully fulfill the requirements of updating control information for the CCDEC problem, since 5G uRLLC can support sub-millisecond latency with error rates that are lower than 1 packet loss in 10^5 packets [49]. As for the DSRC, we implement a new medium access control mechanism based on the WAVE Module, since each vehicle does not contend the channel and directly transmits the content based on the TSR in our scenario. Meanwhile, each vehicle serves as an OpenFlow switch (OpenFlow Module), and communicates with the controller through CDPI interface (cellular link). The wireless communication channel adopts the YansWifiChannel [50], since it has the following advantages: 1) coupling between various models is very high; 2) object-oriented; 3) widely use of C++ standard library and 4) coupling between C++ and OTcl is very low.

As for the DSRC communication, the communication quality is highly affected by severe shadowing in VANETs [41]. Therefore, we assume that the communication links exist only between vehicles with a LOS, or equivalently between "neighbors" [43]. We adopt the Rician model for small-scale fading with the propagation loss factor $n = 4$ [43]. The channel capacity between any two nodes $N_i, N_j \in \mathcal{N}$ at the scheduling period, is then given by

$$C_{i,j} = \begin{cases} W \log_2\left(1 + \eta |h|^2 d_{i,j}^{-4}\right), & \text{LOS exists} \\ 0, & \text{otherwise,} \end{cases} \tag{20}$$

where $d_{i,j}$ is the distance between node i and node j, η is the SINR at transmitters, W is the channel bandwidth, and h is

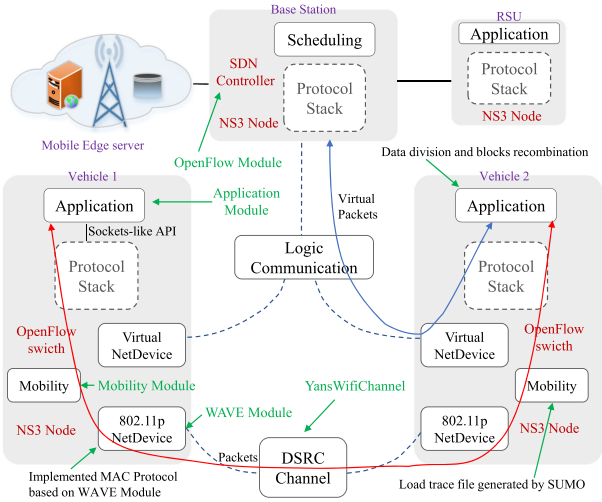


Fig. 9. Simulation of software-defined cooperative data dissemination in edge computing assisted 5G-VANET in NS3.

the Rician channel gain, which is given by

$$h = \sqrt{\frac{\kappa}{\kappa + 1}} e^{j\theta} + \sqrt{\frac{1}{\kappa + 1}} \omega, \tag{21}$$

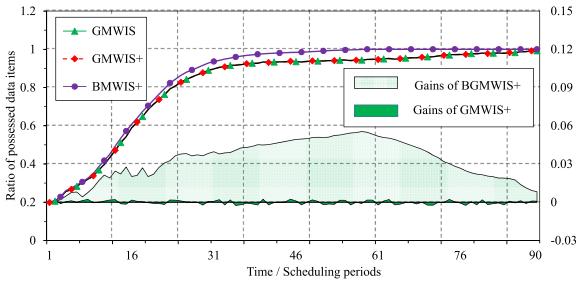
where θ is a random variable uniformly distributed in $[0,\ 2\pi]$, ω is a complex Gaussian random variable with unit variance and zero mean, and κ is the ratio of the energy in the LOS path to the energy in the scattered paths. Thus, the maximum size of data transmitted between node N_i and node N_j in each time slot, is given by $C_{i,j}T_0$. Other simulation parameters are shown in Table 3.

OBUs are initialized with $L_0\rho_0$ number of data items randomly, where ρ_0 is the ratio of initialized items. Each RSU has all L_0 data items all the time, as it acts as the source of LLC contents. These data items are managed by a specially designed application (Application Module). This designed application will encapsulate the data items into a packet, and then transmit the packet through the socket-like API in NS3.

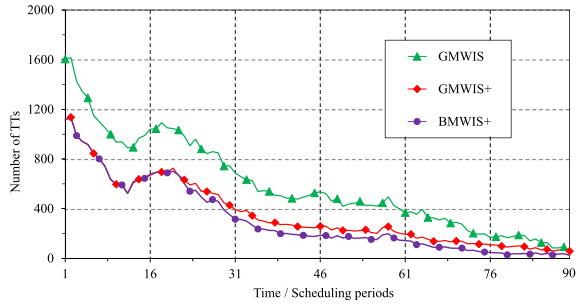
To give an evaluation to the performance of this software-defined cooperative data dissemination algorithm, we have conducted extensive simulations as compared with the one in [32]. Besides, we have implemented two other algorithms. The first one is a random approach, in which every node N_i broadcasts a set of items randomly (Random-Random), once the channel is detected to be unoccupied. The other one is a greedy non-cooperative approach

TABLE 3
Simulation Parameters

Parameter	Value
Simulation map	Generated by SUMO
V2V communication range	50 m
V2I communication range	100 m
Scheduling duration T_0	0.5 s
The size of data file	100 Mb
Bandwidth	10 Mhz
Initialization ratio ρ_0	0.2
The number of RSUs n	5
The number of OBUs m	$45 - 400$



(a) The ratio of possessed data items



(b) The number of TTs in each scheduling period

Fig. 10. Comparison of the GMWIS, GMWIS+, and BMWIS+ in terms of complexity and efficiency under same channel capacity, $L_0 = 15$, $\rho = 0.2$.

(RandomGreedy), in which each node N_i broadcasts the data items. RandomGreedy ensures that the maximum number of valid data items can be received as long as the channel is detected to be unoccupied. The transmitted data items T_k^{i*} for node N_i for RandomGreedy algorithm are

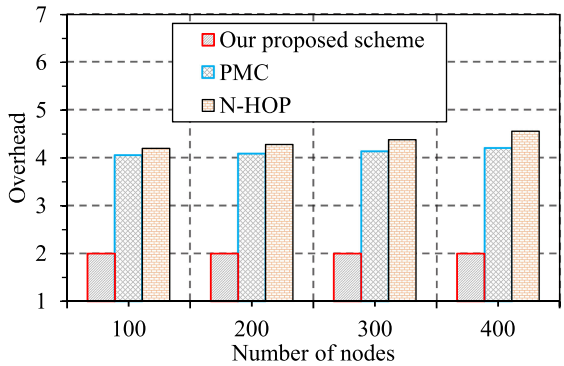
$$T_k^{i*} = \arg\max_{T_k^i \in \mathcal{T}(N_i)} \sum_{N_j \in \Omega(N_i, G_d)} |V_{i,j}(T_k^i)|. \quad (22)$$

6.3 Simulation Results

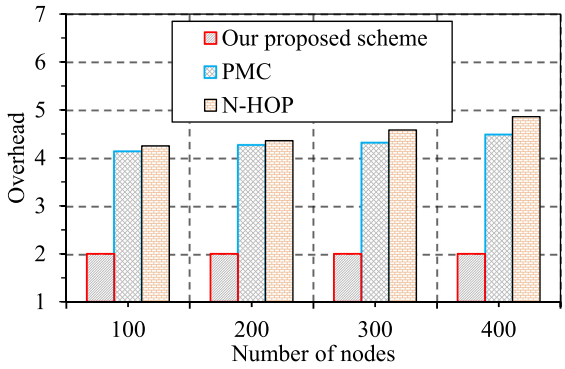
In this subsection, we will evaluate our proposed algorithm via NS3 and SUMO. First, we conduct simulations to show the overhead of our proposed scheme as compared with the clustering schemes. Then, we show the gains of different channel capacity over identical channel capacity, since the latter is widely assumed in various works and the former is more practical. Finally, we evaluate the performance of the balanced greedy algorithm and ILP decomposition algorithm.

6.3.1 Overhead

We first conduct simulations to show the overhead of the proposed scheme as compared with two well-known clustering schemes, PMC [51] and N-HOP [52]. We first analyze the overhead. There exists two kinds of overhead by our proposed scheme: 1) contextual information transmission to the MEC via cellular communication (each node generates a packet in each scheduling period); 2) scheduling results transmission to each vehicle via cellular communication (each node generates a packet in each scheduling period). Consequently, our proposed algorithm generates 2 packets overhead per node in each scheduling period. However, as for the typical clustering algorithm, there exists three kinds of overhead: 1) contextual information transmission to the cluster head via multi-hop communication; 2) scheduling results transmission to each vehicle via multi-hop communication; 3) clustering overhead: vehicles have to communicate with each other to select a cluster head and maintain the cluster. Therefore, the overhead of the typical clustering algorithm is greater than that of our proposed algorithm. Furthermore, we have conduct extensive simulations to demonstrate it. We adopt the number of transmissions per node in each scheduling period to reveal the overhead, which is shown in Fig. 11. Figs. 11a and 11b show the results when the maximum hops from the cluster head to a cluster member are 3 and 4, respectively. PMC produces more than twice as many overheads as our proposed algorithm. The clustering algorithms have to depend on the real-time status of neighbors, i.e., each node must be aware of the position and direction of the neighbors. Therefore, each node broadcasts beacon messages periodically and extra event-triggered packets have to be generated with the mobility of vehicles. N-HOP has the heaviest overhead, since it only considers the mobility of a vehicle when selecting the cluster head. Besides, these clustering algorithms would generate inter-cluster interference, which would deteriorate the performance.



(a) 3 hops



(b) 4 hops

Fig. 11. Overhead of the proposed scheme as compared with two clustering schemes. The overhead is indicated by the number of transmissions per node in each scheduling period.

6.3.2 The Gains of Different Channel Capacity

We compare our proposed algorithm with the approach in [32], where identical channel capacity is assumed. [32] constructs the conflict graph by iterating all the unicast TTs and formulates the scheduling as an MWIS problem. This problem is solved by a greedy MWIS algorithm, which is termed

as GMWIS. Compared with the work in [32], our work is different in both the scenario and the algorithm, and meanwhile, our proposed algorithm can improve the performance with lower complexity. The differences are summarized as follows: 1) We consider the software-defined cooperative data sharing in 5G-VANET, which enables the cooperation of cellular communication and DSRC, which is of higher performance and scalability. 2) The proposed algorithm can reduce the complexity, without degrading the performance, which is achieved by a series of schemes. First, transforming the undirected neighbor graph into the directed matched graph can eliminate a lot of impractical transmissions. Second, we refine the unicast TTs into multicast TTs, which reduces the size of the conflict graph, thus reducing the complexity without lowering down the performance. Third, we have come up with a novel approach to remove the set of multicast TTs that cannot lead to a better solution. 3) We have proposed two algorithms to improve the solution of the MWIS problem, carefully considering the characteristics in 5G-VANET. The first one is a balanced greedy algorithm, which can distribute the data items in a more balanced manner, benefiting the data dissemination in the following time slots. The second one is the ILP based decomposition algorithm, which can make full use of the computing power. Once this software-defined cooperative data sharing is deployed as a service, a fixed amount of computing power should be allocated to this service. This proposed algorithm can improve the solution with execution goes on, thus making full use of the computing resources. We refer to the greedy MWIS in [32] as GMWIS. The greedy MWIS based on our proposed refined multicast TTs is referred to as GMWIS+. The balanced greedy algorithm is termed as BMWIS+.

The first simulation is conducted to compare the proposed algorithm with the one in [32], where the channel capacity for each link is the same, following the assumptions in [32]. The popular content is divided into 15 data items, and each vehicle caches 20 percent of the data items randomly ($L_0 = 15$, $\rho_0 = 0.2$). Fig. 10a shows the ratio of possessed items in each scheduling period. The area diagram shows the gains of GMWIS+ and BMWIS+, as compared with GMWIS. The BMWIS+ has better performance, as the popular content can be delivered to each vehicle at the fastest speed. The reason is that in each scheduling period, BMWIS+ always transmits the set of data items that is most unbalanced. Therefore, with balanced item distribution, each item request can be served by their neighbors at a higher possibility. GMWIS and GMWIS+ have nearly the same performance and different complexity, which can be shown in Fig. 10b. The GMWIS has the maximum number of TTs while GMWIS+ and BMWIS+ have a very small number of TTs. The reason behind this is that we merge the unicast TTs into multicast TTs, and we have removed the TTs that cannot lead to a better solution. These two mechanisms can greatly reduce the size of the conflict graph, thus reducing the computation overhead in solving the MWIS problem. Meanwhile, these mechanisms do not lower down the performance. Therefore, our proposed algorithm achieves a significant performance boost both in terms of complexity and efficiency, compared with [32].

The second simulation is conducted to verify that different channel capacity has a big influence on scheduling

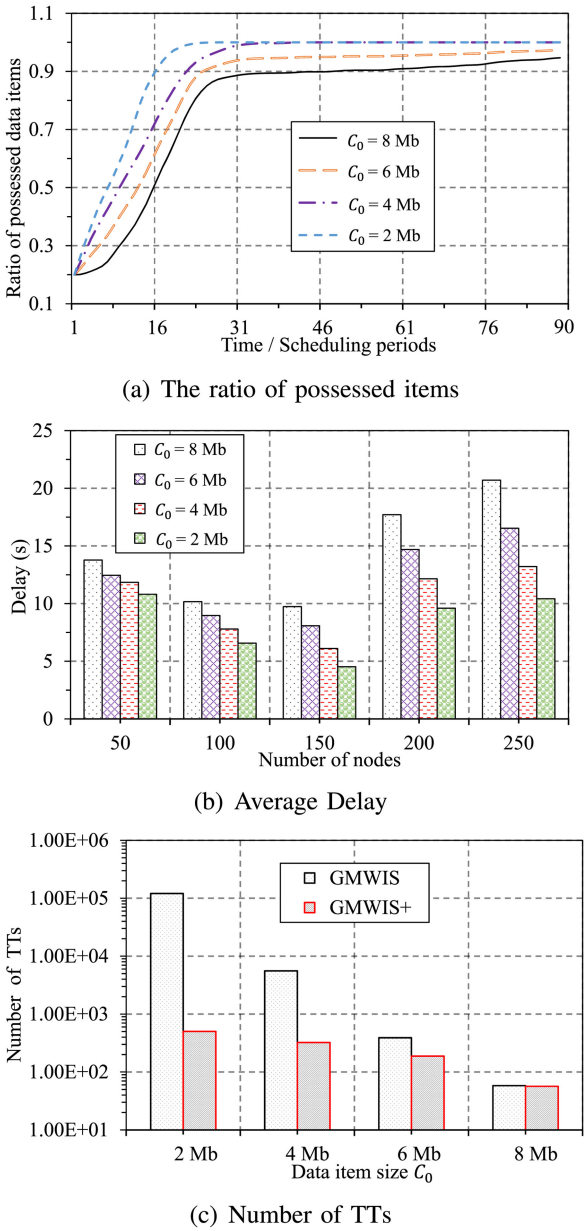


Fig. 12. Performance of scheduling under different C_0 when the channel capacity is different. A smaller C_0 leads to a higher scheduling performance.

performance. [32] assumed the same channel capacity for each link, which is impractical and of low efficiency. The channel capacity of different communication links is different. Therefore, we conduct the scheduling algorithm under a practical channel capacity condition. Fig. 12a shows the ratio of possessed data items under different size of data item C_0. Smaller C_0 leads to higher performance, as the diverse channel capacity can be fully utilized, thus improving the efficiency of cooperative data sharing. A larger C_0 would waste a part of links. For example, when certain channel capacity is 5 Mb, if $C_0 = 6$ Mb, this channel cannot disseminate any data item. Under this scenario, $C_0 = 2.5$ Mb is better than $C_0 = 4$ Mb, as two items can be delivered simultaneously if $C_0 = 2.5$ Mb. Therefore, a smaller C_0 leads to a faster dissemination speed of the popular content, which can be shown in Fig. 12b. $C_0 = 2$ Mb has the fastest speed to deliver all the data items to each vehicle, while $C_0 = 8$ Mb has the lowest. Therefore, a

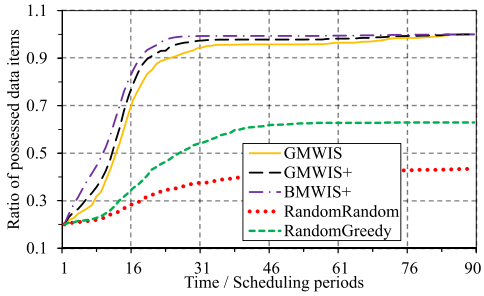


Fig. 13. The ratio of possessed data items in each scheduling period for different scheduling algorithms.

smaller C_0 means better cooperative sharing efficiency. On the other hand, a smaller C_0 would generate more TTs according to the approach in [32], and more TTs would increase the complexity of the scheduling. A smaller C_0 means a higher L_0, which would lead to a larger number of unicast TTs. As described in Section 4.3, the number of unicast TTs is in exponential order with L_0, as the complexity is $O(\Delta_0^2 2^{2L_0}|m+n|^2)$. Our proposed algorithm generates at most $2^{\Delta_0}-1$ multicast TTs for each vehicle, achieving the complexity of $O(2^{2\Delta_0}(m+n)^2)$. Meanwhile, we could further reduce the number of TTs by the preference order to eliminate the set of TTs that would not lead to a better solution. Therefore, our proposed algorithm makes sure that the number of TTs is not influenced by C_0, at the same time, without affecting the scheduling performance, as shown in Fig. 12c, where GMWIS has a larger number of TTs than GMWIS+. Therefore, with C_0 becomes smaller, our proposed algorithm achieves a higher performance and will not increase the scheduling complexity, which is of great significance to cooperative data dissemination in 5G-VANET.

6.3.3 Evaluation of BMWIS+ and GMWIS+

The third simulation is conducted to evaluate the performance of the proposed algorithm BMWIS+ and GMWIS+. We compare our algorithm with GMWIS, RandomRandom, and RandomGreedy. Fig. 13 shows the ratio of possessed data items in each scheduling period. We can easily conclude that BMWIS+ has the best performance and GMWIS+ has a better performance than GMWIS. GMWIS+ adopts a series of schemes to reduce the number of TTs, i.e., eliminating the set of impractical transmissions by transforming the G_u to G_d, merging unicast TTs to multicast TTs, and removing the set of TTs that could not lead to a better solution by the preference order. These schemes can reduce the number of TTs, thus reducing the complexity of the scheduling. What's more, these schemes achieve a better performance, as the content is delivered to each vehicle at the fastest speed. The reason is that the multicast TTs are more robust over the greedy algorithm. For the same directed matched graph shown in Fig. 4, the unicast TT (the algorithm in [32]) generates the conflict graph, which is shown in Fig. 7. The greedy algorithm applied to this conflict graph, which is GMWIS, generates the independent set $\{N_5 \xrightarrow{\{d_1,d_2,d_3\}} N_1, N_2 \xrightarrow{\{d_1,d_2,d_3\}} N_4\}$. Then, the TSR could be obtained. The sender nodes are N_2 and N_5, and the transmitted data items for N_2 and N_5 are $\{d_1,d_2,d_3\}$. Under this scenario, 4 data items will be successfully received.

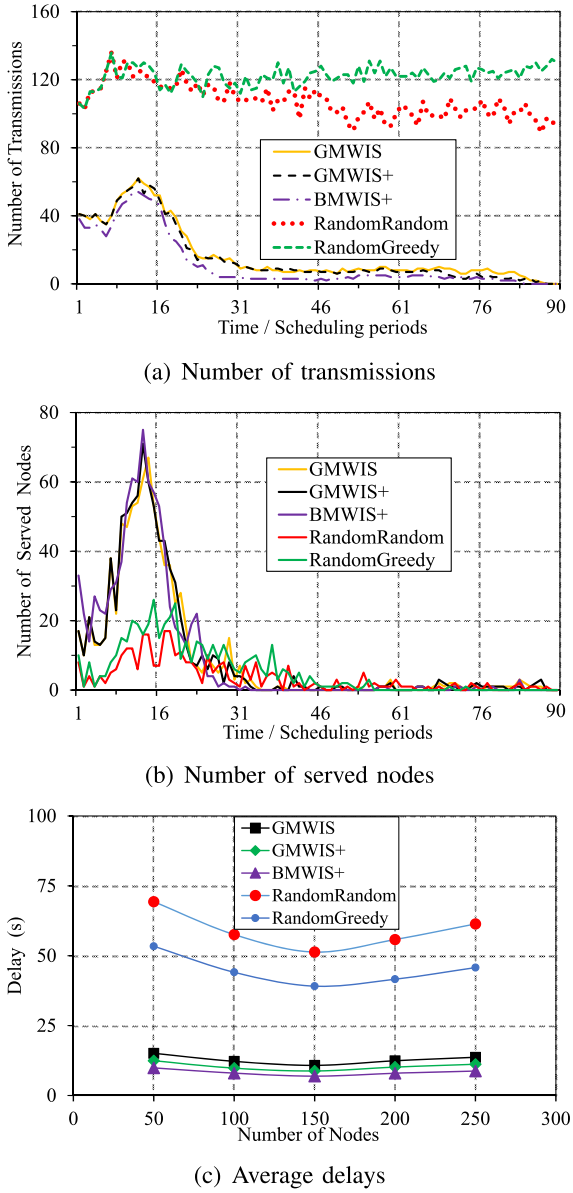
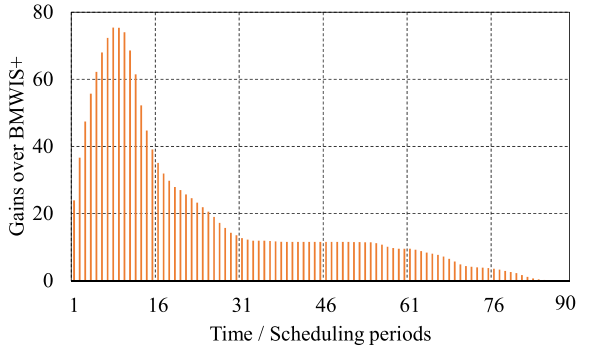
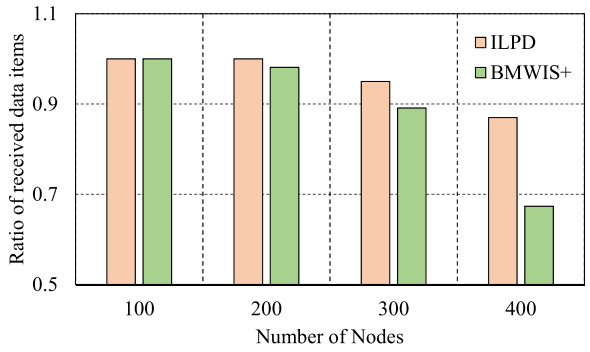


Fig. 14. Performance of the five scheduling algorithms.

The multicast TTs generate the conflict graph, as shown in Fig. 5. The greedy algorithm applied to this conflict graph, which is GMWIS+, generates the independent set $\{N_5 \xrightarrow{\{d_1,d_2,d_3\}} \{N_1\}, N_2 \xrightarrow{\{d_1,d_3\}} \{N_3,N_4\}\}$. Then, the TSR could be obtained. The sender nodes are N_2 and N_5, and the transmitted data items for N_2 and N_5 are $\{d_1,d_2,d_3\}$ and $\{d_1,d_3\}$, respectively. Under this scenario, 5 data items will be successfully received. Therefore, these schemes could not only reduce the algorithm complexity, but also improve the performance when the greedy algorithm is applied to solve this MWIS problem. GMWIS makes a separate decision on each scheduling period. Therefore, BMWIS+ considers the dissemination influence of different time slots, scheduling the set of data items that have the highest balanced rate. The BMWIS+ can make the distribution of items more balanced. Therefore, BMWIS+ achieves a better performance than GMWIS+, which is shown in Fig. 13. Fig. 14a shows the number of transmission in each scheduling period. The BMWIS+ has the least number of transmissions, and RandomRandom



(a) Gains over BMWIS+



(b) Ratio of received data items

Fig. 15. Performance of ILPD as compared with BMWIS+.

and RandomGreedy have a higher number of transmissions. At the same time, RandomRandom and RandomGreedy can serve a smaller number of nodes, compared with BMWIS+, GMWIS+, and GMWIS, which can be shown in Fig. 14b. Fig. 14c shows the average delay, where RandomRandom and RandomGreedy have a higher delay. BMWIS+ achieves the minimum delay, which demonstrates that our proposed algorithms can reduce the latency of cooperative data sharing of LLC content.

6.3.4 *Evaluation of the ILP Based Decomposition Algorithm*

The last simulation is conducted to evaluate the performance of the ILP based decomposition algorithm, which is referred to as ILPD. ILPD can improve the solution of BMWIS+ with execution goes on. Based on the solution of BMWIS+, ILPD can explore a better one by adopting the branch-and-bound framework, meanwhile, the conflict graph is divided into multiple subgraphs by considering the sparsity of the coefficient matrix, which reduces the search space, leading to higher efficiency. Therefore, we evaluate the performance of ILPD as compared with BMWIS+. Fig. 15a shows the gains of ILPD over BMWIS+, which is defined as the total possessed data items of ILPD minus that of BMWIS+. ILPD can make more data items received successfully, as in the previous time slots, the gains reach the highest. With the sharing goes on, the gains will diminish since the requested data items are fixed and the gains will always reach zero. ILPD can transmit more contents as compared with BMWIS+, which is shown in Fig. 15b. With the nodes $m + n = 200$, BMWIS+ cannot ensure that all the vehicles receive their requested data items, while ILPD can successfully meet the data requests, transmitting all the requested data items. With the number of nodes increases, the advantages of ILPD over BMWIS+ are more obvious, as more requested data items can be successfully transmitted as compared with BMWIS+.

7 Conclusion

In this work, we have proposed a software-defined cooperative data sharing architecture and designed a cooperative data sharing scheduling algorithm in 5G-VANET. The proposed architecture has decoupled contextual information sensing from the data transmission. To enable efficient cooperation between 5G and VANETs, and cooperation between communication and computing resources, we have proposed a graph theory based algorithm to schedule the cooperative data dissemination. We have formulated the cooperative data sharing algorithm as an MWIS problem based on the constructed conflict graph, which achieves significant performance boost both in terms of complexity and efficiency, compared with the state-of-the-art methods. Considering the cooperative data sharing characteristics in 5G-VANET, we have proposed a balanced greedy algorithm and an ILP based decomposition algorithm. Simulation results have demonstrated the superiority and efficiency of the proposed algorithms.

In our future work, we will extend application scenarios with multiple channel access. Following IEEE 1609.4, there exist 6 service channels. Cooperative data sharing over multiple channels could significantly increase throughput. Also, other applications and services (e.g., computation offloading, service migration) will be considered. Furthermore, the impacts of data dissemination at medium access control layers and physical layers are expected to be examined to validate the model in realistic vehicular communication environments.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China under Grants 61876023, 91638204, and 61902035, BUPT Excellent Ph.D. Students Foundation under Grant No. CX2018210, the Natural Science Foundation of Beijing under Grant No. 4181002, and Natural Sciences and Engineering Research Council (NSERC), Canada.

References

[1] H. Zhou, W. Xu, Y. Bi, J. Chen, Q. Yu, and X. Shen, "Toward 5G spectrum sharing for immersive-experience-driven vehicular communications," *IEEE Wireless Commun.*, vol. 24, no. 6, pp. 30–37, Dec. 2017.

[2] N. Cheng, N. Lu, N. Zhang, X. Shen, and J. W. Mark, "Vehicular WiFi offloading: Challenges and solutions," *Veh. Commun.*, vol. 1, no. 1, pp. 13–21, 2014.

[3] W. Xu *et al.*, "Internet of vehicles in big data era," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 19–35, Jan. 2018.

[4] K. Lin, J. Luo, L. Hu, M. S. Hossain, and A. Ghoneim, "Localization based on social big data analysis in the vehicular networks," *IEEE Trans. Ind. Inform.*, vol. 13, no. 4, pp. 1932–1940, Aug. 2017.

[5] N. Cheng, N. Zhang, N. Lu, X. Shen, J. W. Mark, and F. Liu, "Opportunistic spectrum access for CR-VANETs: A game-theoretic approach," *IEEE Trans. Veh. Technol.*, vol. 63, no. 1, pp. 237–251, Jan. 2014.

[6] G. Luo *et al.*, "Cooperative vehicular content distribution in edge computing assisted 5G-VANET," *China Commun.*, vol. 15, no. 7, pp. 1–17, Jul. 2018.

[7] J. Li *et al.*, "An end-to-end load balancer based on deep learning for vehicular network traffic control," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 953–966, Feb. 2019.

[8] G. Einziger, C. F. Chiasserini, and F. Malandrino, "Scheduling advertisement delivery in vehicular networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 12, pp. 2882–2897, Dec. 2018.

[9] N. Cheng *et al.*, "Big data driven vehicular networks," *IEEE Netw.*, vol. 32, no. 6, pp. 160–167, Nov./Dec. 2018.

[10] D. Naboulsi and M. Fiore, "Characterizing the instantaneous connectivity of large-scale urban vehicular networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1272–1286, May 2017.

[11] F. Lyu *et al.*, "SS-MAC: A novel time slot-sharing MAC for safety messages broadcasting in VANETs," *IEEE Trans. Veh. Technol.*, vol. 67, no. 4, pp. 3586–3597, Apr. 2018.

[12] J. He, Y. Ni, L. Cai, J. Pan, and C. Chen, "Optimal dropbox deployment algorithm for data dissemination in vehicular networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 3, pp. 632–645, Mar. 2018.

[13] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," ETSI White Paper, vol. 11, no. 11, pp. 1–16, 2015.

[14] H. Zhou, N. Cheng, Q. Yu, X. ShermanShen, D. Shan, and F. Bai, "Toward multi-radio vehicular data piping for dynamic DSRC/TVWS spectrum sharing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2575–2588, Oct. 2016.

[15] J. Wang, C. Jiang, Z. Han, Y. Ren, and L. Hanzo, "Network association strategies for an energy harvesting aided super-WiFi network relying on measured solar activity," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3785–3797, Dec. 2016.

[16] H. Zhou, N. Cheng, Q. Yu, X. Shen, D. Shan, and F. Bai, "Toward multi-radio vehicular data piping for dynamic DSRC/TVWS spectrum sharing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2575–2588, Oct. 2016.

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

[18] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[19] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. Shen, "Toward effcient content delivery for automated driving services: An edge computing solution," *IEEE Netw.*, vol. 32, no. 1, pp. 80–86, Jan./Feb. 2018.

[20] M. Chen, Y. Qian, Y. Hao, Y. Li, and J. Song, "Data-driven computing and caching in 5G networks: Architecture and delay analysis," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 70–75, Feb. 2018.

[21] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A scalable and quick-response software defined vehicular network assisted by mobile edge computing," *IEEE Commun. Magazine*, vol. 55, no. 7, pp. 94–100, Jul. 2017.

[22] F. Rebecchi, M. D. De Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, "Data offloading techniques in cellular networks: A survey," *IEEE Commun. Surveys Tut.*, vol. 17, no. 2, pp. 580–603, Apr.–Jun. 2015.

[23] M. Xing, J. He, and L. Cai, "Utility maximization for multimedia data dissemination in large-scale VANETs," *IEEE Trans. Mobile Comput.*, vol. 16, no. 4, pp. 1188–1198, Apr. 2017.

[24] J. Liu, X. Wang, G. Yue, and S. Shen, "Data sharing in VANETs based on evolutionary fuzzy game," *Future Gener. Comput. Syst.*, vol. 81, pp. 141–155, 2018.

[25] Q. Yuan, J. Li, Z. Liu, and F. Yang, "Space and time constrained data offloading in vehicular networks," in *Proc. IEEE Int. Conf. High Perform. Comput. Commun.*, Dec. 2016, pp. 398–405.

[26] Y. Gu, L. X. Cai, M. Pan, L. Song, and Z. Han, "Exploiting the stable fixture matching game for content sharing in D2D-based LTE-V2X communications," in *Proc. IEEE Global Commun. Conf.*, 2016, pp. 1–6.

[27] S. Ucar, S. C. Ergen, and O. Ozkasap, "Multihop-cluster-based IEEE 802.11 p and LTE hybrid architecture for VANET safety message dissemination," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2621–2636, Apr. 2016.

[28] G. El Mouna Zhioua, N. Tabbane, H. Labiod, and S. Tabbane, "A fuzzy multi-metric QoS-balancing gateway selection algorithm in a clustered VANET to LTE advanced hybrid cellular network," *IEEE Trans. Veh. Technol.*, vol. 64, no. 2, pp. 804–817, Feb. 2015.

[29] G. Luo, S. Jia, Z. Liu, K. Zhu, and L. Zhang, "sdnMAC: A software-defined networking based MAC protocol in VANETs," in *Proc. IEEE/ACM 24th Int. Symp. Qual. Service*, 2016, pp. 1–2.

[30] J. Chen *et al.*, "Service-oriented dynamic connection management for software-defined internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2826–2837, Oct. 2017.

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

[32] K. Liu, J. K. Y. Ng, V. C. S. Lee, S. H. Son, and I. Stojmenovic, "Cooperative data scheduling in hybrid vehicular ad hoc networks: VANET as a software defined network," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1759–1773, Jun. 2016.

[33] G. S. Aujla, R. Chaudhary, N. Kumar, J. J. P. C. Rodrigues, and A. Vinel, "Data offloading in 5G-enabled software-defined vehicular networks: A stackelberg-game-based approach," *IEEE Commun. Magazine*, vol. 55, no. 8, pp. 100–108, Aug. 2017.

[34] G. Luo, J. Li, L. Zhang, Q. Yuan, Z. Liu, and F. Yang, "sdnMAC: A software-defined network inspired MAC protocol for cooperative safety in VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 2011–2024, Jun. 2018.

[35] R. Yu, J. Ding, X. Huang, M. T. Zhou, S. Gjessing, and Y. Zhang, "Optimal resource sharing in 5G-enabled vehicular networks: A matrix game approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7844–7856, Oct. 2016.

[36] Q. Zheng, K. Zheng, H. Zhang, and V. C. M. Leung, "Delay-optimal virtualized radio resource scheduling in software-defined vehicular networks via stochastic learning," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7857–7867, Oct. 2016.

[37] X. Duan, Y. Liu, and X. Wang, "SDN enabled 5G-VANET: Adaptive vehicle clustering and beamformed transmission for aggregated traffic," *IEEE Commun. Magazine*, vol. 55, no. 7, pp. 120–127, Jul. 2017.

[38] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Introduction to ultra reliable and low latency communications in 5G," *Comput. Res. Repository (CoRR) abs/ 1704.05565*, 2017. [Online]. Available: https://128.84.21.199/abs/1704.05565v1

[39] Y. L. Morgan, "Notes on DSRC & WAVE standards suite: Its architecture, design, and characteristics," *IEEE Commun. Surveys Tut.*, vol. 12, no. 4, pp. 504–518, Oct.–Dec. 2010.

[40] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the united states," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.

[41] M. Boban, W. Viriyasitavat, and O. K. Tonguz, "Modeling vehicle-to-vehicle line of sight channels and its impact on application-layer performance," in *Proc. ACM Int. Workshop Veh. Inter-Netw. Syst. Appl.*, 2013, pp. 91–94.

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

[43] T. Wang, L. Song, Z. Han, and B. Jiao, "Dynamic popular content distribution in vehicular networks using coalition formation games," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 538–547, Sep. 2013.

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

[45] A. Schrijver, *Theory of Linear and Integer Programming*. Hoboken, NJ, USA: Wiley, 1998.

[46] D. Warrier, W. E. Wilhelm, J. S. Warren, and I. V. Hicks, "A branch–and–price approach for the maximum weight independent set problem," *Netw.*, vol. 46, no. 4, pp. 198–209, 2005.

[47] X. Cheng, Q. Yao, M. Wen, C. X. Wang, L. Y. Song, and B. L. Jiao, "Wideband channel modeling and intercarrier interference cancellation for vehicle-to-vehicle communication systems," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 434–448, Sep. 2013.

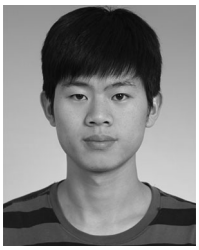
[48] C. Celes, F. A. Silva, A. Boukerche, R. M. D. C. Andrade, and A. A. F. Loureiro, "Improving VANET simulation with calibrated vehicular mobility traces," *IEEE Trans. Mobile Comput.*, vol. 16, no. 12, pp. 3376–3389, Dec. 2017.

[49] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 124–130, Jun. 2018.

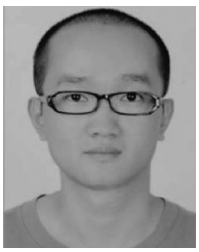
[50] M. Lacage and T. R. Henderson, "Yet another network simulator," in *Proc. Workshop NS-2: The IP Netw. Simulator*, 2006, Art. no. 12.

[51] D. Zhang, H. Ge, T. Zhang, Y.-Y. Cui, X. Liu, and G. Mao, "New multi-hop clustering algorithm for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1517–1530, Apr. 2019.

[52] Z. Zhang, A. Boukerche, and R. Pazzi, "A novel multi-hop clustering scheme for vehicular ad-hoc networks," in *Proc. 9th ACM Int. Symp. Mobility Manage. Wireless Access*, 2011, pp. 19–26.



Guiyang Luo received the BE degree in electrical and communications from Beijing Jiaotong University (BJTU), Beijing, China, in 2015. He is currently working toward the PhD degree at the Beijing University of Posts and Telecommunications. His current research interests includes software-defined networks, medium access control, and intelligent transportation system.



Haibo Zhou received the PhD degree in information and communication engineering from Shanghai Jiao Tong University, China, in 2014. From 2014 to 2017, he worked as a postdoctoral fellow with the Broadband Communications Research Group, ECE Department, University of Waterloo, Canada. Currently, he is an associate professor with the School of Electronic Science and Engineering, Nanjing University. His research interests include resource management and protocol design in cognitive radio networks and vehicular networks.

Nan Cheng (S'12-M'16) received the PhD degree from the Department of Electrical and Computer Engineering, University of Waterloo. He is currently working as a postdoctoral fellow with the Department of Electrical and Computer Engineering, University of Toronto, and with Department of Electrical and Computer Engineering, University of Waterloo. His research interests include performance analysis and opportunistic communications for vehicular networks, unmanned aerial vehicles, and cellular traffic offloading. He is a member of the IEEE.

Quan Yuan received the BS degree in computer science and technology from the Beijing University of Posts and Telecommunications, Beijing, China, in 2011. He is currently working as a postdoctoral fellow at the Beijing University of Posts and Telecommunications. His current research interests include crowdsensing, connected vehicle, mobile Internet, and intelligent transportation system.

Jinglin Li received the PhD degree in computer science and technology from the Beijing University of Posts and Telecommunications, in 2004. He is currently an associate professor of computer science and technology, and director of the Switching and Intelligent Control Research Center (SICRC) at the State Key Laboratory of Networking and Switching Technology, China. His research interests include mobile Internet, Internet of Things, Internet of vehicles, convergence network, and service technologies.

Fangchun Yang received the PhD degree in communications and electronic systems from the Beijing University of Posts and Telecommunication, in 1990. He is currently professor with the Beijing University of Posts and Telecommunication, China. He has published six books and more than 80 papers. His current research interests include network intelligence, service computing, communications software, soft-switching technology, and network security. He is a fellow of the IET.

Xuemin (Sherman) Shen (M'97-SM'02-F'09) received the BSc degree from Dalian Maritime University, China, in 1982 and the MSc and PhD degrees from Rutgers University, New Jersey, all in electrical engineering, in 1987 and 1990, respectively. He is a university professor, Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests include resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He served as the technical program committee chair/co-chair for IEEE Globecom16, Infocom14, IEEE VTC10 Fall, and Globecom07, the symposia chair for IEEE ICC10, the tutorial chair for IEEE VTC11 Spring and IEEE ICC08, the general co-chair for ACM Mobihoc15, Chinacom07 and QShine06, and the chair for IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He also serves/served as the editor-in-chief for the *IEEE Internet of Things Journal*, *IEEE Network*, *Peer-to-Peer Networking and Application*, and *IET Communications*; a founding area editor for the *IEEE Transactions on Wireless Communications*; an associate editor for the *IEEE Transactions on Vehicular Technology*, *Computer Networks*, and *ACM/Wireless Networks*, etc. He received the Excellent Graduate Supervision Award, in 2006, and the Outstanding Performance Award, in 2004, 2007, 2010, and 2014 from the University of Waterloo, the Premiers Research Excellence Award (PREA), in 2003 from the Province of Ontario, Canada, the Distinguished Performance Award, in 2002 and 2007 from the Faculty of Engineering, University of Waterloo, and the Joseph LoCicero Award from the IEEE Communications Society. He is a registered professional engineer of Ontario, Canada, an IEEE fellow, an Engineering Institute of Canada fellow, a Canadian Academy of Engineering fellow, a Royal Society of Canada fellow, and a Distinguished lecturer of IEEE Vehicular Technology Society and Communications Society.