Service-Oriented Dynamic Connection Management for Software-Defined Internet of Vehicles

Jiacheng Chen, Haibo Zhou, *Member, IEEE*, Ning Zhang, *Member, IEEE*, Wenchao Xu,
Quan Yu, *Senior Member, IEEE*, Lin Gui, *Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*— Internet of vehicles (IoV) is an emerging paradigm for accommodating the requirements of future intelligent transportation systems (ITSs) with the overwhelming trend of equipping vehicles with versatile sensors and communications modules, and facilitating drivers and passengers with a variety of innovative ITS applications. However, the implementation of IoV still faces many challenges, such as flexible and efficient connections, quality of service guarantee, and multiple concurrent support requests. To this end, in this paper we introduce the software-defined IoV (SD-IoV), which is able to tackle the above-mentioned issues by adopting the software-defined networking framework. We first present the architecture of SD-IoV and develop a centralized vehicular connection management approach. Then, we aim to allocate dedicated communications resources and underlying vehicular nodes to satisfy each service. We formulate the dynamic vehicular connection as an overlay vehicular network creation (OVNC) problem. A comprehensive utility function is also designed to serve as the optimization objective of OVNC. Finally, we solve the OVNC problem by developing a graph-based genetic algorithm and a heuristic algorithm, respectively. Extensive simulation results are provided to demonstrate the effectiveness of our proposed solution of dynamic vehicular connection management.**

***Index Terms*— Software defined Internet of vehicles (SD-IoV), intelligent transportation system (ITS), quality of service (QoS), vehicular connection management, overlay network creation.**

I. INTRODUCTION

INTELLIGENT transportation systems (ITS) are envisioned to greatly improve the transportation safety and efficiency by connecting everything on the road to cyberspace. Internet of vehicles (IoV) [1], [2], which is the instantiation of Internet of things (IoT) paradigm in vehicular networking scenario, is regarded as a promising way to realize the next generation ITS, especially with more and more vehicles embedded with various sensors and communications modules [3]. With the support of IoV, ITS applications can be enhanced by the advanced cloud computing and big data techniques.

The performance of communications among IoV entities can largely impact the quality of ITS services. However, the state-of-the-art vehicular communications protocol, i.e., dedicated short-range communications (DSRC) [4], can hardly satisfy the demands from future IoV-based ITS. The variety of ITS services requires flexible and efficient connections between vehicles and roadside units (RSU) as well as among vehicles, referred to as vehicle-to-roadside (V2R) and vehicle-to-vehicle (V2V), respectively. Furthermore, ITS services can have differentiated quality of service (QoS) requirements that need to be fulfilled in order to improve user satisfaction. Moreover, the capability of handling a large amount of concurrent requests is indispensable, due to the rapidly increasing number of connected vehicles. Recently, the emergence of software defined networking (SDN) [5] has triggered reconsideration of networking paradigm in both wired and wireless networks. Featured by the separation of control plane and data plane, SDN can not only improve network flexibility and efficiency but also provide a platform for advanced network management. Through the seamless integration of SDN and IoV, software defined IoV (SD-IoV) can fully inherit the merits of SDN and is regarded as a competitive solution for implementing future IoV-based ITS. In SD-IoV, controllers manage vehicular communications in a centralized manner by aggregating network state information and making appropriate decisions. With the realtime global knowledge, resource allocation can be dynamically optimized and service QoS can be guaranteed.

This paper focuses on the service-oriented dynamic vehicular connection management in SD-IoV and deals with the three aforementioned challenges in IoV-based ITS, namely flexible and efficient connections, service QoS guarantee, and multiple concurrent requests support. First, we introduce a reference scenario and present the architecture of SD-IoV. Based on the architecture, we develop a centralized control process for vehicular connection management in SD-IoV, which can realize flexible connections among network nodes (i.e., vehicles and RSUs). Then, we formulate the dynamic vehicular connection management as an overlay vehicular network creation (OVNC) problem. An overlay vehicular network comprises a subset of the underlaying communications nodes operating on an allocated channel. Dedicated overlay vehicular networks are dynamically created in order to satisfy

---

Manuscript received September 29, 2016; accepted May 6, 2017. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, in part by the National Natural Science Foundation of China under Grant 91638204 and Grant 61471236, and in part by Huawei Technologies Canada. The Associate Editor for this paper was K. Wu. *(Corresponding author: Haibo Zhou.)*

J. Chen, Q. Yu, and L. Gui are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: jiacheng1989@gmail.com; yuquan61@qq.com; guilin@sjtu.edu.cn).

H. Zhou, W. Xu, and X. Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: haibozhouuw@gmail.com; wenchaoxu.ru@gmail.com; xshen@bbcr.uwaterloo.ca).

N. Zhang is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 1A1, Canada (e-mail: nin.zhang@utoronto.ca).

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

Digital Object Identifier 10.1109/TITS.2017.2705978

the requests of users. The OVNC takes advantage of channel spatial reuse and generates multiple non-interfering overlay networks, such that multiple concurrent requests can be served simultaneously. The topology of an overlay network should follow a set of predefined rules so as to improve the data transmission efficiency and increase the chances to form the overlay network. We further design a novel utility function to evaluate an overlay network, which mainly takes into account resource utilization efficiency and QoS requirements. The objective of OVNC is to maximize the sum utilities of all the coexisting overlay networks. Finally, we design a graph-based genetic algorithm (GBGA) and a heuristic algorithm to solve the OVNC problem. Due to the complexity of the problem, a standard-type GA cannot be directed adopted. Therefore, we redesign the implementation of main procedures, including graph-based representation, initial population generation, crossover, mutation, and fitness evaluation. In summary, the main contributions of this paper are listed below:

- We design the system architecture of SD-IoV. We develop a centralized control process for vehicular connection management in SD-IoV, which can realize flexible connections among network nodes.
- We formulate the overlay vehicular network creation problem and design a novel utility function as its objective, such that resource utilization efficiency, service QoS, and multiple concurrent requests support can be guaranteed.
- We propose two algorithms to solve OVNC. One is the graph-based genetic algorithm that modifies the main procedures of a standard GA so as to fit in with OVNC, and the other is a heuristic algorithm with less computation load.

The remainder of this paper is organized as follows. Section II briefly summarizes the backgrounds and related works. In Section III, the SD-IoV system is described in details and the formulation of OVNC problem is given. GBGA and heuristic solutions of the problem are presented in Section IV, followed by the simulation results in Section V. Finally, Section VI draws the conclusion of the whole paper.

II. RELATED WORKS

A. Vehicular Communications Networks and IoV

Vehicular communications networks (VCN) [6] have attracted huge attentions from both academia and industries in recent years and have made significant progress in technology and market maturity. IoV [1], [2] is the extension of VCN in IoT [7] paradigm and is envisioned to pave the way for realizing next generation ITS. Similar to IoT, IoV is basically a three-level “client-connection-cloud” system [2]. Client system mainly consists of vehicles embedded with sensors and communications modules. Despite of acting as end users, such vehicles are also responsible for detecting and gathering realtime information in the surrounding area. Connection system implements communications functionalities. Typically, an infrastructure (e.g., RSU) is required as the gateway for data routing between client system and the Internet. Cloud system is the additional part of IoV compared with VCN, and is an ecosystem for all the ITS-related applications and services. Recently, research has been conducted on various aspects of IoV [8]–[10]. In [8], Wang *et al.* propose a novel decision-tree based vertical handoff method for IoV, which improves the performance of IoV when there exist multiple heterogeneous access networks. In [9], Guerrero-Ibáñez *et al.* discuss the potential challenges when integrating ITS with technologies including connected vehicles, cloud computing and IoT. Wang *et al.* [10] develop an analytical model for IoV based on IEEE 802.15.4 in nonbeacon-enabled mode, under the conditions of non-saturated traffic pattern and large-scale networks.

Communications protocol is the fundamental building block of IoV. A variety of approaches have been proposed, such as cellular [11], Wi-Fi [12], DSRC [4], and opportunistic spectrum access over TV white space (TVWS) [13]. Among all these methods, DSRC is the only one that is specifically designed and standardized for vehicular networking scenario. However, it can barely satisfy the demands from future IoV-based ITS. From its implementation details [4], it can be known that DSRC-based system is not flexible in the following aspects: (i) WAVE service advertisements (WSA) are unaware of realtime requests of vehicles; (ii) vehicles can only passively select services advertised by WSAs; and (iii) services are bonded with the operating channels. Such inflexibility will lead to low channel utilization efficiency and will starve some of the requesting vehicles. What is more, the distributed contention-based MAC protocol (i.e., CSMA/CA) suffers from rate anomaly phenomenon and struggles in terms of QoS guarantee. Moreover, as the number of vehicles increases, the performance of CSMA/CA will degrade due to more contentions and collisions.

B. Software Defined Networking and SD-IoV

The concept of SDN has a long evolution history [5]. Nevertheless, it is only until the recent years that SDN has become a hot topic and has nearly swept all the fields of communications networks, owing to the seminal work by McKeown *et al.* [14], in which they propose the OpenFlow protocol that implements SDN. With the main idea of separating control plane and data plane, SDN gains potential benefits including flexibility, programmability, resource utilization efficiency, robustness, decoupling of hardware and services, etc. It is also a suitable platform for realizing advanced applications such as network virtualization and network function virtualization (NFV). Since SDN has proven its superiority in the wired networks [15], recently researchers also expect SDN can renovate the wireless networks [16], [17]. Compared with SDN-enabled 5G and Wi-Fi, study on software defined vehicular networks is still in its infancy. Chen *et al.* [18] surveys the recent works on this topic and points out the open challenges. Li *et al.* [19] aim to optimize the interactions between controllers and vehicles in software defined vehicular ad-hoc networks (VANET). In their model, vehicles can either use the more expensive cellular networks or the less reliable ad-hoc networks to send control messages. A rebating mechanism is designed and game theory is adopted to find the balance between cost and latency requirement. In [20], Wang *et al.* focus on reducing the number of rules installed in flow tables, due to the limited storage size of OpenFlow-enabled switches. In order to build compact rules for the scalability of IoV, multicast addresses are used for the wireless data plane and a destination-driven model is adopted in the wired counterpart. Truong *et al.* [21] propose an architecture for VANET that combines both SDN and fog computing. Besides the benefits of SDN, the system can also gain benefits from fog computing, which can provide delay-sensitive and location-aware services. Ku *et al.* [22] also design an SDN-based VANET architecture and introduce the potential benefits and services that it can provide. Routing protocol in SDN-based VANET is compared with traditional VANET routing protocols so as to demonstrate the feasibility. In [23], Salahuddin *et al.* propose RSU cloud for IoV. The SDN-enabled RSUs can dynamically reconfigure service hosts and data forwarding paths and implement service instantiation and migration so as to deal with node mobility. Based on reconfiguration cost analysis, an integer linear programming (ILP) problem is formulated to solve the RSU cloud resource management. Zheng *et al.* [24] study the software defined heterogeneous vehicular network from a high-level design perspective. They propose the multi-layer Cloud-RAN architecture and point out the corresponding challenges and solutions. In [25], Zheng *et al.* present a framework that integrates LTE, SDN and vehicular network. A dynamic scheduling scheme for virtualized radio resource within the framework is proposed. He *et al.* [26] utilize SDN for heterogeneous resources abstraction. Different network resources are properly scheduled in order to minimize communications cost. In [27], Liu *et al.* investigates cooperative data dissemination in software defined VANET. In their work, a scheduling period is divided into three phases. In the first phase, all the vehicles turn into V2V mode and find their neighbors. Then, all the vehicles turn into V2R mode and send information to the RSU. Finally, based on the decision of RSU, each vehicle stays in either V2R or V2V mode to complete data transmission. The scheduling problem is formulated as a maximum weighted independent set (MWIS) problem and solved by a greedy algorithm. Our work is different from the above literatures in various aspects. We study service-oriented vehicular connection management and deal with the challenges in the SD-IoV scenario. For multi-channel management, we develop a centralized control process that is able to utilize multiple channels simultaneously. For communications link model, we consider different data rates for different links and use protocol model to characterize interference. We consider IoV services with QoS requirements and the problem formulation is QoS-driven. The final vehicular connection management result is a set of overlay vehicular networks with general topologies such as multicast and multi-hop.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

Fig. 1 shows a reference scenario of SD-IoV. In the figure, a piece of road segment covered by one RSU is highlighted for elaboration. The RSU is connected to the SDN controller such that vehicles in the area can be centrally controlled. We adopt fog computing [28] paradigm, thus the SDN controller is located in the fog. Generally, fog can be regarded as localized lightweight cloud with lower latency, which is critical in the SD-IoV scenario. Besides, another trending concept that is closely related to SDN can also be implemented, namely network function virtualization (NFV) [29]. NFV further makes the network more flexible and programmable by decoupling network functions (NF) from dedicated hardware or entities. NFV can be realized based on SDN, so the NF server is also located in the fog. A fog can take control of multiple RSUs in a certain area so as to reduce implementation cost. Furthermore, such an implementation can enable cooperation among RSUs and facilitates certain NFs like mobility management. The RSU is also connected to the Internet cloud, in which various IoV-based ITS applications can be deployed, as well as other auxiliary functionalities such as data mining. By analogy, vehicles in SD-IoV are like the switches in SDN-enabled wired networks, which means they do not need to implement any network-level control functionalities. Instead, vehicles should be able to interpret the decisions made by SDN controller and take actions accordingly. The above SD-IoV can support all the transmission modes that may be required by different applications, including multicast, multi-hop V2V, V2R uplink, etc.

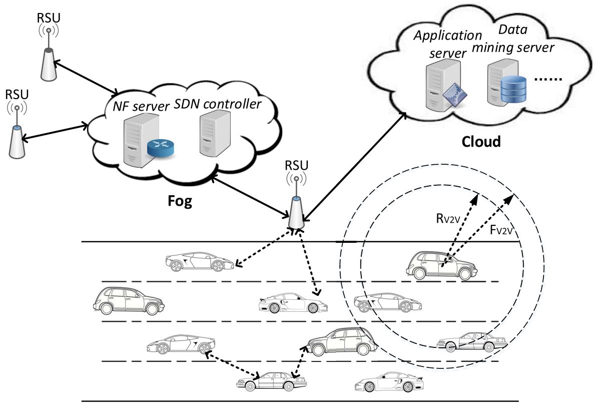


Fig. 1. A reference scenario of SD-IoV.

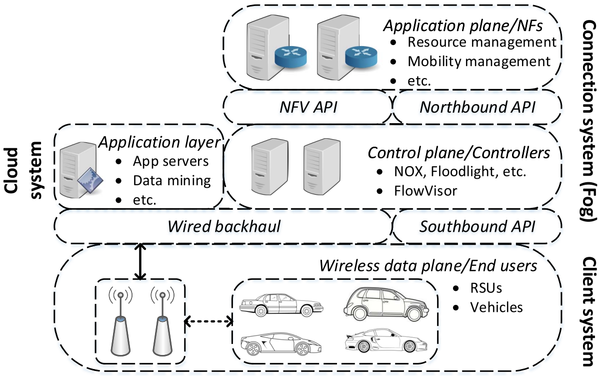


Fig. 2. Architecture of SD-IoV.

Fig. 2 illustrates the architecture of SD-IoV in the paradigm of both SDN and IoV. The wireless data plane is formed by end users and consists of RSUs and vehicles, which correspond to the client system in IoV. RSU also acts as the gateway for vehicles. The control plane is made up of controllers, which are servers running SDN controller programs like NOX and Floodlight as well as controller virtualization programs like FlowVisor. Southbound API is used for communications between control plane and data plane. On top of control plane lies the application plane, where NFs such as resource management and mobility management are implemented. Northbound API is adopted to interpret NFs into lower-level controller functions. NFs can further be decomposed into smaller modular functions through the NFV API so as to improve the reusability and robustness. Together with the wired backhaul, all these entities compose the connection system. In our architecture, most parts of the connection system are located in the fog. Finally, the cloud system comprises servers running various applications. Note that these applications are different from those in the application plane in that the former belong to the application layer while the latter belong to the network layer.

We suppose that RSU and vehicles are equipped with single radio and work in half-duplex mode. The centralized control process for vehicular connection management in SD-IoV works in time-division manner. Specifically, each time slot has two intervals. The first interval is used for overlay network creation. At the beginning, each vehicle sends its state information to the RSU on the predefined control channel. State information includes mobility-related information such as vehicle's location, velocity and direction, as well as service-related information such as request and service provision capability. Service provision capability indicates which services a vehicle can provide to others, and is determined by its equipped sensors and cached data. Afterwards, the RSU communicates with SDN controller, which creates overlay networks based on all the information collected and feedbacks the decision to RSU. Then, the RSU distributes the decision to vehicles. During the first interval, vehicles access the RSU in a polling manner, i.e., in a certain order. For example, the access order can be the same as the order that vehicles join the network. The second interval is used for data transmission. Nodes form the overlay networks based on the decision by switching to certain channels, setting addresses, etc. and start transmitting/receiving.

We use two simple examples to show how ITS applications are conducted in SD-IoV. In the first example, suppose a vehicle (denoted by $\mathcal{A}$) wants to overtake the vehicles (denoted by $\mathcal{B}$) in front of it. As $\mathcal{A}$ accelerates, it also sends a request to the RSU. The controller then creates a network in which $\mathcal{A}$ is sender and $\mathcal{B}$ are receivers. During the data transmission interval, $\mathcal{A}$ will multicast message to alert $\mathcal{B}$ so as to avoid collisions. In the second example, suppose the RSU needs to update the road condition information. The controller then checks all the vehicles' service provision capabilities and select an appropriate one to let it send the information to the RSU.

B. Problem Formulation

In this paper, we mainly focus on a piece of road segment covered by one RSU, as illustrated in Fig. 1. Since overlay vehicular network creation is conducted slot by slot, notations only refer to the studied slot in the following, if not specified. A road has N_R lanes, and the total number of vehicles on road is N_V. A node (denoted by $i, j, \ldots$) can refer to either a vehicle (denoted by v) or the RSU (denoted by r), hence the total number of nodes is $N_N = N_V + 1$. The indices of vehicles are from 1 to N_V and the index of RSU is N_N. The total number of channels is N_C, and the bandwidth of each channel is 10MHz. The length of a slot is denoted by δ_{slot}, and each interval δ_{intv} takes half of the slot length.

For IoV-based ITS applications, a service set D is predefined, and each element $d \in D$ represents a certain type of service. Each type of service is associated with a data size L_d, a priority P_d and a timeout limit T_d (in slots). We suppose that each node can only have no more than one request at the same time, and the request of a node i is denoted by $d_i \in D$ if it exists. If the request of a service d is not completed within T_d, then the request will be dropped. The service provision capability of vehicle v is represented by set $D_v \subseteq D$, meaning that v can provide the types of services given in D_v. Also, we have $d_v \notin D_v$, since a vehicle will not request data that it already caches. For the RSU, we suppose that its service provision capability is D, i.e., it can provide all types of services. However, it may also need to update its cached information by generating a request $d_r \in D$.

The length of studied road segment is denoted by R, which is also the coverage area of the RSU, given that it is located in the middle point of the road segment. Hence, the communications range of V2R is $R_{V2R} = R/2$. For V2V communications, the range is denoted by R_{V2V}, which is smaller than R_{V2R} for the sake of improving channel spatial utilization efficiency. In other words, the transmission power that vehicles use for V2R is larger than V2V, and RSU always uses the larger transmission power. On the other hand, V2V has interference range represented by F_{V2V}, which is a little larger than R_{V2V}. Illustrations of R_{V2V} and F_{V2V} are shown in Fig. 1. We suppose a dedicated channel used for V2R communications. The transmission rate between nodes i and j is denoted by ϕ_{ij}.

Considering the resource utilization efficiency and implementation feasibility in practice, we define the topology rules that allow three major types of topologies for the overlay vehicular networks. The first type is single-hop unicast ($sender - receiver$), including V2V, V2R uplink and downlink. The second type is single-hop multicast ($sender - receivers$), including V2V and V2R downlink. In this case, the transmission rate is equal to the minimum of all the links in the overlay network. The last type is V2V dual-hop unicast ($sender - helper - receiver$). We suppose that each hop takes half of the transmission time, and the transmission rate is equal to the lower of both hops. The intermediate helper node only forwards data, hence it does not need to have the same service provision capability as the sender. V2R dual-hop transmission is unnecessary due to the larger transmission power used in V2R communications. Also, V2V dual-hop multicast and V2V multi-hop (larger than two hops) are not considered, since they are not efficient and difficult to implement in practical vehicular networking scenarios.

1) QoS-Driven Utility Design: In this part, we aim to design a specific QoS-driven utility function to evaluate the overlay vehicular network. We take into account both service characteristics and resource utilization efficiency. First, we evaluate the potential gain of a requesting node by assuming that an overlay network is created for it. For node i with request d_i, we evaluate the gain through the service completion contribution (SCC) integrated with three terms and defined as:

$$SCC_i = P_{d_i}^{\alpha} \times \min\left\{\frac{L_{d_i}^{t} + r_i \delta_{intv} I_{Rx}(i)}{L_{d_i}}, 1\right\}^{\beta} \times \left(I_{\overline{Tout}}(d_i) \times \frac{t - t_{d_i} + 1}{T_{d_i}}\right)^{\gamma} \times I_{Req}(i) \tag{1}$$

where α, β, and γ are non-negative parameters that affect the weight of each term. The weight of a term can be eliminated by setting the corresponding parameter to 0, otherwise the value of each term is limited to $(0, 1]$ for the sake of consistency. Hence, the value of SCC belongs to $[0, 1]$. The first term reflects the influence of service's priority. Note that the priority value is normalized, namely $P_{d_i} \in (0, 1]$. In the second term, $L_{d_i}^{t}$ denotes the completed size of the service at the current slot t, r_i denotes the actual transmission rate (i.e., considering the effect of multicast or dual-hop), $I_{Rx}(i)$ is an indicator function that returns 1/0 if i belongs/does not belong to the set of all receiver nodes Rx, and the $min\{\}$ function ensures that the value of this term cannot exceed 1. The second term takes into account transmission rate on the one hand, and tends to finish a nearly completed request on the other hand. In the third term, $I_{\overline{Tout}}(d_i)$ returns 1/0 if d_i is/is not within the timeout limit, t_{d_i} denotes the time that the request d_i is generated, and $t - t_{d_i} + 1$ ensures the value of this term is not 0 if $t = t_{d_i}$. This term considers the urgency of a request so as to avoid timeout. At last, $I_{Req}(i)$ returns 1/0 if i has/does not have a request.

Then, we define the utility of an overlay vehicular network G_O based on the SCCs of all the nodes involved. U_{G_O} is given by:

$$\begin{aligned} U_{G_O} = {} & I_{\overline{Tout}}\left(\exists d_i, i \in Rx_{G_O}\right) \\ & \times \left(\sum\nolimits_{i \in Rx_{G_O}} SCC_i - NOP_{G_O} + 2\right) \\ & \times \left(1 - Norm\left(\max\left\{R_{Tx_{G_O},i}, i \in Rx_{G_O}\right\}\right)\right)^{\lambda I_{V2V}(G_O)} \end{aligned} \tag{2}$$

where Tx_{G_O} and Rx_{G_O} denotes the sender and receiver set of G_O respectively. U_{G_O} is a non-negative value and a zero value means that the overlay network is meaningless. The first term $I_{\overline{Tout}}(\exists d_i, i \in Rx_{G_O})$ returns 1/0 if there exist at least one/no requests that are within the timeout limit. In the second term, NOP_{G_O} refers to node occupation penalty (NOP) of G_O given by:

$$NOP_{G_O} = SCC_{Tx_{G_O}} + SCC_{Hp_{G_O}} \tag{3}$$

where Hp_{G_O} denotes the helper node in G_O. If Hp_{G_O} does not exist, then $SCC_{Hp_{G_O}} = 0$. Owing to the indicator functions in the definition of SCC, it is straightforward for NOP to reflect the costs for serving Rx_{G_O} with Tx_{G_O} and Hp_{G_O}. $-NOP_{G_O}+2$ in the second term ensures that the value of U_{G_O} is not negative, since the maximum of SCC is 1. In the third term, λ is a positive parameter that affects the weight of the term, $I_{V2V}(G_O)$ returns 1/0 if G_O is/is not a V2V type overlay network, $R_{Tx_{G_O},i}$ denotes the distance between the sender TX_{G_O} and receiver $i \in Rx_{G_O}$, and $Norm()$ is a normalization function that returns a value between 0 and 1. This term takes into account the channel spatial utilization efficiency in V2V scenarios by penalizing the overlay network that introduces a large interference area, which is represented by the maximum distance between sender and receivers.

2) Overlay Vehicular Network Creation Problem: In this part, we will formulate the OVNC problem. First, we present an overview of the problem. Generally, OVNC aims to create overlay vehicular networks based on a certain criteria, hence it is an optimization problem. The objective is to maximize the sum utilities of all the coexisting overlay networks. Also, constraints should be satisfied during the creation, which can be categorized into the following aspects: (i) the topology of an overlay network should not contradict with the radio characteristics (i.e., single-radio, half-duplex) and the predefined rules; (ii) in an overlay network, the service provision capability of the sender should cover the request of the receiver; and (iii) the coexisting overlay networks should not interfere with each other.

Then, we mathematically formulate the OVNC problem in a way that fits in with the GA framework, which is the main approach we will use to solve the problem. GA [30] is a global optimization method suitable for dealing with complex (e.g., NP-hard) combinatorial optimization problems. It belongs to the family of evolutionary algorithms (EA), which are different from traditional searching algorithms in that EAs are population-based. A *population* is a set of solutions drawn from the solution space. The population generated after each iteration is also called a *generation*. Hence, a new generation is returned for the next iteration instead of a single solution. Each solution is called a *phenotype*, and the *representation* of a solution in GA is called a *genotype*. Representation is a function that maps the solution space to search space comprising genotypes. A phenotype must have at least one corresponding genotype for a valid representation. *Genes* in a genotype code all the characteristics of a phenotype. The *reproduction* operations such as *crossover* and *mutation* create new solutions from the population. During GA, *selection* is used to filter better genotypes according to the *fitness*, which is an evaluation function defined on all genotypes. GA is a powerful optimization method and has many advantages. One of them that is worth mentioning is its parallelism. GA's major procedures such as initial population generation and fitness evaluation can be implemented simultaneously on different machines. Consequently, the computation time consumption of GA can be largely reduced.

In standard GAs, a solution is represented by a bit string. However, in our scenario, it is not efficient to code overlay vehicular networks in such a way. Therefore, we specifically design a GA tailored for our problem, i.e., the GBGA. The name itself indicates that the representation is graph-based.

We use an undirected connectivity graph G_C to describe the underlay vehicular network. Note that in the following, we adopt the same notation for the matrix representation of a graph. G_C is an N_N-by-N_N matrix and the value of its element G_C^{ij} is defined as:

$$G_C^{ij} = \begin{cases} 0, & i, j \in 1, \ldots, N_V, \quad R_{i,j} > R_{V2V} \\ 1, & otherwise \end{cases} \tag{4}$$

A link θ_{ij} is feasible only when $G_C^{ij} = 1$. We use an undirected interference graph G_I to indicate whether two links working on the same channel can coexist without interference. Hence, vertices in G_I are feasible links with directions (i.e., θ_{ij} and θ_{ji} are different links). The value of an element $G_I^{\theta_{ij},\theta_{kl}}$ in G_I is given by:

$$G_I^{\theta_{ij},\theta_{kl}} = \begin{cases} 0, & i, j, k, l \in 1, \ldots, N_V, \\ & R_{k,j} > F_{V2V}, \quad R_{i,l} > F_{V2V} \\ 1, & otherwise \end{cases}$$
$$i, j, k, l : G_C^{ij} = 1, G_C^{kl} = 1 \tag{5}$$

In our scenario, a phenotype refers to the set of all overlay vehicular networks. We use an undirected service graph G_S as the genotype to represent a phenotype. G_S is also an N_N-by-N_N matrix in which rows represent transmitting nodes and columns represent receiving nodes. $G_S^{ij} = 1$ indicates that the link θ_{ij} exists and $G_S^{ij} = 0$ otherwise. For G_S to be feasible, a set of constraints need to be satisfied. First of all, each link in G_S must be feasible, namely:

$$\forall i, j : G_S^{ij} = 1 \Rightarrow G_C^{ij} = 1 \tag{6}$$

The notation $a \Rightarrow b$ means that if a is true, then b must be true, otherwise G_S is not feasible. Then, considering the single-radio and half-duplex features of nodes as well as the topology rules defined above, a feasible G_S should also satisfy the following requirements:

Requirement 1:

$$\forall j : \sum_{i=1}^{N_N} G_S^{ij} \le 1 \tag{7}$$

which means a node cannot receive data from more than one transmitters simultaneously;

Requirement 2:

$$\forall j : \sum_{i=1}^{N_N} G_S^{ij} = 1 \Rightarrow \sum_{k=1}^{N_N} G_S^{jk} \le 1 \tag{8}$$

in which the inequality on the right side specifies single-hop unicast (when "less" holds) or dual-hop unicast (when "equal" holds) in that a receiver cannot be a sender simultaneously;

Requirement 3:

$$\forall i : \sum_{j=1}^{N_N} G_S^{ij} > 1 \Rightarrow \sum_{h=1}^{N_N} G_S^{hi} = 0 \tag{9}$$

which specifies single-hop multicast in that the sender cannot be a receiver simultaneously;

Requirement 4:

$$\forall i : \sum_{j=1}^{N_N} G_S^{ij} = 1, \quad \exists G_S^{hi} = 1 \Rightarrow \sum_{g=1}^{N_N} G_S^{gh} = 0, \quad \sum_{k=1}^{N_N} G_S^{hk} = 1 \tag{10}$$

which further specifies dual-hop unicast in that the sender can neither be a receiver nor multicast simultaneously;

Requirement 5:

$$\forall i, j : G_S^{ij} = 1 \Rightarrow G_S^{ji} = 0 \tag{11}$$

which means a transmitter and a receiver cannot exchange.

By checking each row of G_S, the corresponding phenotype can be interpreted and overlay vehicular networks can be known straightforwardly. Then, the overlay vehicular network creation problem can be formulated as below:

Objective: $\max_{G_S} \sum_{G_O \in G_S} U_{G_O}$

Subject to: G_S satisfies (7) - (11)

$$\forall G_O \in G_S : d_{Rx_{G_O}} \in D_{Tx_{G_O}} \tag{12}$$
$$\forall G_O, G'_O \in G_S, Ch_{G_O} = Ch_{G'_O} : G_I^{\theta_{ij},\theta_{kl}} = 0, \forall \theta_{ij} \in G_O, \theta_{kl} \in G'_O \tag{13}$$

where $d_{Rx_{G_O}}$ is the request of G_O (for multicast, the requests of all receivers are the same), and Ch_{G_O} is the channel of G_O. The objective is to maximize the sum utility of each overlay network. (12) means that the sender must have service provision capability for the request of G_O, and (13) ensures that different overlay networks working on the same channel do not interfere with each other.

By following the channel management scheme and topology rules of [27], we can get a simplified instantiation of the problem formulated above. Similar to the process in [27], based on the above constraints, a weighted graph can be created within polynomial time. In the graph, vertices are possible overlay networks, edges denote whether two vertices can coexist, and vertices' weights denote utilities. Then, the instantiation can be reduced to the NP-hard MWIS problem. Hence, the original problem with $N_C > 2$ and more complex topology rules is also NP-hard. Note that the original problem itself cannot be transformed into MWIS problem, hence the solution given in [27] cannot be adopted for our problem.

IV. OVERLAY NETWORK CREATION ALGORITHMS

In this section, we will show details on how GBGA is implemented to solve the formulated problem. With the graph-based representation described above, redesign of GA's other procedures is presented. Also, a heuristic algorithm is presented as an alternative method.

A. Graph-Based Genetic Algorithm

GBGA's overall process and implementation of main procedures are shown in Alg. 1. It takes the state information reported by nodes as inputs and returns the overlay network creation decision. In the following, the design of each procedure/subroutine is discussed. Instead of focusing on programming-level details, we use general and descriptive languages to outline the logical steps in each pseudocode.

Algorithm 1 Graph-Based Genetic Algorithm (GBGA)

```
Input: state information of nodes
Output: overlay network creation decision
Initialize{
calculate intermediate variables: $G_C$, $G_I$, etc.
generate initial population
}
Main process{
while termination conditions not satisfied do
    crossover
    mutation
return the fittest genotype
}
Subroutine 1 — Random Genotype Generation{
Input: necessary variables
Output: genotype
while termination conditions not satisfied do
    randomly select an available node as sender
    randomly decide service and topology
    randomly select feasible nodes as receivers (and
    helper if needed)
    randomly allocate a feasible channel
    add overlay network to genotype
}
Subroutine 2 — Crossover{
Input: population
Output: updated population
select mating pool
while not enough offsprings generated do
    randomly select a pair of parents
    randomly perform overlay networks exchanging
    between parents
    add offsprings to population
return $N_P$ most fittest genotypes from population
}
Subroutine 3 — Mutation{
Input: population
Output: updated population
foreach genotype $G_S$ in population do
    randomly decide whether to mutate $G_S$ or not
    if mutate $G_S$ then
        randomly decide mutation type
        randomly mutate $G_S$ based on type
}
```

1) Initialization: In the initialization procedure, intermediate variables that will be used in the subsequent procedures are calculated, and the initial population is generated. The connectivity graph and interference graph are calculated based on the reported locations of nodes and certain path loss models, since there is no neighbor discovery function conducted in our system. Although the created overlay vehicular networks are supposed not to interfere with each other, the transmitters still work under the CSMA/CA mechanism so as to avoid collisions that may be introduced due to the possible inaccuracy of connectivity and interference graphs. Furthermore, the accuracy can be improved by historical data and learning-based methods, which are easy to implement in our SD-IoV architecture.

In GAs, initial population is generated randomly in most cases such that the whole search space can be covered. For GBGA, we also adopt random generation. Throughout GBGA, we maintain the size of population N_P in each iteration, which is equal to the size of the initial population. Due to the constraints (7)-(13), a simple random generation by randomly setting the bit elements of a genotype is likely to generate invalid genotypes and thus is not applicable. The design of GBGA aims to ensure the feasibility of each genotype no matter when and how it is generated. Therefore, we design a new random genotype generation procedure, and its general steps are shown in Subroutine 1. In each iteration, a sender is randomly selected from current available nodes, which are the nodes that have not been occupied by other overlay networks so far. A random service is chosen based on the service provision capability of the selected sender and the topology is randomly decided based on the neighbors of the sender. Then, receivers are randomly chosen from available nodes with the same request, and a helper is randomly chosen for the dual-hop case. Also, a random channel is allocated to the newly created overlay network, which should not interfere with existing overlay networks operating on the same channel. If there exist no available nodes or a certain number of iterations have been conducted, then the procedure ends. The procedure runs totally N_P times to generate the initial population.

2) Crossover: Crossover is one of the reproduction operations that generate new genotypes from the current population. A typical crossover exchanges part of the genes of a selected pair of genotypes (i.e., *parents*) and reproduces two new genotypes (i.e., *offsprings*). Such a process guarantees the effectiveness of GAs in that offsprings are expected to inherit superior genes from superior parents and thus the overall fitness of population can be improved generation by generation. For the same reason as the initial population generation, a specific crossover implementation is necessary for our problem. The major steps are shown in Subroutine 2. First, a subset of genotypes in population is selected to form the mating pool, from which parents are chosen. Generally, genotypes with higher fitness values are more likely to be selected, and such a tendency is referred to as *selection pressure*. Neither too high selection pressure nor too low is preferred. For GBGA, we adopt the *roulette wheel selection* [30]. Basically, the probability that a genotype is selected is proportional to its fitness. The selection procedure is conducted a certain number of times to form the mating pool. Then, an iteration procedure runs until a certain number of offsprings (usually slightly larger than N_P) are created. In each iteration, a pair of genotypes are randomly selected from the mating pool as parents. The parents reproduce a pair offsprings by randomly exchanging the overlay network with each other. Specifically, for each parent, one of its overlay networks is randomly chosen and removed from the genotype. Then, the overlay network is added to the other genotype. During the adding phase, if constraints are violated, the overlay networks that cause conflicts are temporarily removed, so as to ensure that the new overlay network can be added successfully. Afterwards, the temporarily removed overlay networks are re-added in a best-effort manner, for example, by changing the operating channel or receivers. In this way, offsprings can do their best to inherit from parents. The newly reproduced offsprings are added to the current population. Finally, N_P genotypes with the largest fitness values are selected to form the updated population. The the above crossover procedure performs a generational update method (i.e., $(\lambda + \mu)$-update in [30]).

3) Mutation: Mutation is another type of reproduction operations that generate new genotypes. Different from crossover, mutation operates on a single genotype and aims to introduce new random genes instead of inheriting from existing genotypes. Therefore, with mutation, the whole search space is more likely to be covered so as to avoid local optimum traps. *Mutation probability* defines the frequency of mutation, and *mutation type* defines how mutation is performed, such as fixed-length or variable-length. In terms of our graph-based representation, the length refers to the number of overlay networks in a genotype. In GBGA, mutation is performed after the crossover and thus it operates on the updated population after crossover. The general steps of our redesigned mutation procedure are shown in Subroutine 3. For each genotype in the current population, we first independently decide whether to mutate it or not based on the mutation probability. Then, in order to mutate a genotype, we start from deciding the mutation type randomly based on the corresponding probabilities. For fixed-length mutation, there are two steps: (i) an overlay network is randomly chosen and removed from the genotype, and (ii) a new random overlay network is generated and added to the genotype (same as the steps given in Subroutine 1). For variable-length mutation, only one of the above two steps is conducted for either overlay network removal or addition, respectively.

4) Termination: As an iterative method, GBGA needs certain conditions to terminate. We adopt two commonly used termination conditions that focus on different aspects. The first condition evaluates the result optimality. Specifically, the mean fitness of each generation is calculated, based on which the relative changes of mean fitness compared with the previous two generations are also calculated. If the changes are less than a certain termination threshold, then GBGA terminates. In other words, when the fitness values of generations become steady, we can infer that little improvement will be made further by more iterations, and the current population has reached a relatively high optimality. The second condition makes a compromise between time consumption and optimality. It is simply done by limiting the maximum number of iterations. After the iteration terminates, the genotype with the largest fitness value is returned as the final overlay vehicular network creation decision.

B. Heuristic Algorithm

Except for GBGA, we also develop a heuristic algorithm for the OVNC problem. The general steps are given in Alg. 2. At the beginning, the SCC of each node is calculated under the assumption that the transmitted data term in Eq. (1) is zero. Then, the SCCs are sorted in a descending order. In each iteration, an available potential receiver (i.e., node that has a request and has not been occupied) is selected according to the order. For the selected receiver, all nodes that can be used as its sender (i.e., potential senders) are found. Specifically, the sender should be an available node; have the capability to serve the receiver; and be at least within two-hop transmission range of sender. If there exist one-hop senders, then for each sender, the overlay network is temporarily created and its utility is calculated. The sender with the largest utility is selected finally. If there only exist dual-hop senders, we need to find all possible overlay networks with different senders and helpers and calculate the utilities. Similarly, the overlay network with largest utility is selected at last. A feasible channel is allocated for the overlay network created above. At the end, the service graph is updated with the overlay network. Note that the heuristic algorithm cannot create overlay networks with single-hop multicast topology.

Algorithm 2 Heuristic Algorithm for OVNC

```
Input: necessary variables
Output: overlay network creation decision
calculate SCC of each node
sort SCCs in descending order
foreach available potential receiver do
    find all potential senders
    select the best sender (and helper if needed)
    randomly allocate a feasible channel
    update the service graph
```

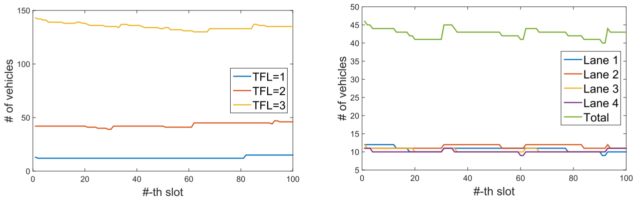
V. SIMULATION RESULTS

In this section, we will evaluate the proposed GBGA itself as well as compare it with the heuristic algorithm and the cooperative data dissemination (CDD) algorithm proposed in [27]. First, we will introduce the simulation setup. We do not modify the values of DSRC-related parameters, which can be found in [4]. Other parameters' values are given in Tab. I. We define the traffic flow level (TFL) parameter to characterize the macroscopic traffic model. TFL can take integer values from 1 to 3, and the corresponding vehicle density (number of vehicles per mile per lane) belongs to the range $[0, 12]$, $[12, 30]$, and $[30, 67]$, respectively [31]. The vehicle speed is also based on TFL, which belongs to the range $[100, 120]$, $[80, 100]$, and $[60, 80]$, respectively. Vehicle speed is randomly generated, but the distance between neighbor vehicles on the same lane cannot be less than the minimum inter-vehicle distance. We suppose that the traffic flow is relatively stable during the simulation period, namely TFL does not change. In order to achieve this, the time interval between two vehicles that consecutively drive into the road is dynamically determined according to the current vehicle density and mean speed. Fig. 3 shows the evaluation of the macroscopic traffic model. We can find that the total number

TABLE I

PARAMETER VALUES IN SIMULATION

Parameter	Value	Parameter	Value
N_R	4 (2 for each direction)	N_C	7
δ_{slot}	100ms	R	1km
R_{V2V}	200m	F_{V2V}	250m
ϕ_{ij}	{12,9,6,4.5,3}Mbps	$\|D\|$	100
L_d	[0.5,2]MB	$\|D_v\|$	3



(a)

(b)

Fig. 3. Macroscopic traffic model: (a) Total vehicles on the road at each time slot (with different TFLs); (b) Total vehicles on each lane at each time slot (TFL=2).

of vehicles on the road/on each lane is almost steady during the simulation period. We suppose that the popularity of ITS services follows Zipf's law, and the probability that a service is requested is calculated the same way as [27]. The timeout limit of a service is proportional to its size, with a little randomness. The priority of each service is also random and takes a value from {0.2,0.4,0.6,0.8,1}. A vehicle without an ongoing request will randomly generate one in the next few slots. When a vehicle finishes a request, its cache space updates in a first-in-first-out manner, which means its service provision capability is also updated. All the weight tuning parameters appearing in the utility definition (i.e., α, β, γ, λ) are set to 1.

A. GBGA Evaluation

In this subsection, we will evaluate the performance of GBGA itself under different settings. As an iterative stochastic algorithm, GBGA's performance can be characterized from two aspects, namely convergence speed and stability (optimality). Convergence speed describes how fast the algorithm terminates. Stability is reflected by the fluctuation of GBGA's final outcomes from different runs. A small fluctuation can also indicate that the whole search space is almost fully covered, which means the final outcome is near-optimal. Figs. 4 and 5 show the evaluation of GBGA in terms of convergence speed and stability under different TFLs. In Fig. 4, we show the outcome (i.e., the maximum fitness) after each iteration during a single run of GBGA. For normalization purpose, we use the relative difference between the outcome after each iteration and the outcome after the run in Fig. 4. It can be found that GBGA can quickly converge to the near-optimal final outcome. Also, the convergence speed in high traffic density case is slower due to the larger search space. In order to verify stability, we run GBGA for 10 times under the same environment (i.e., the same connectivity graph, vehicles' request, etc.). The mean of the outcomes from all the runs is calculated and we show the relative difference between the outcome from each run and the mean in Fig. 5. It can be found that the fluctuation is small (for $TFL = 1$ case, GBGA has the same outcome from all the runs), thus stability and optimality are validated.

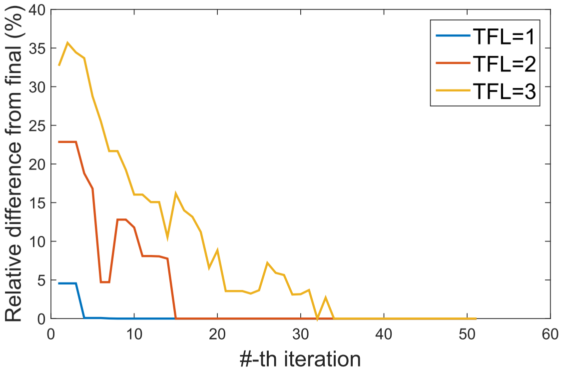


Fig. 4. Convergence speed evaluation of GBGA. The figure shows the relative difference between the outcome after each iteration and the outcome after the run, with different TFLs.

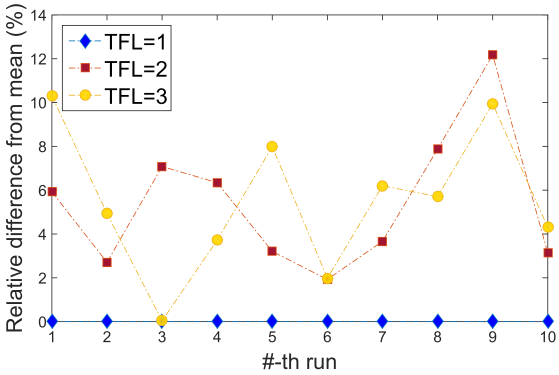
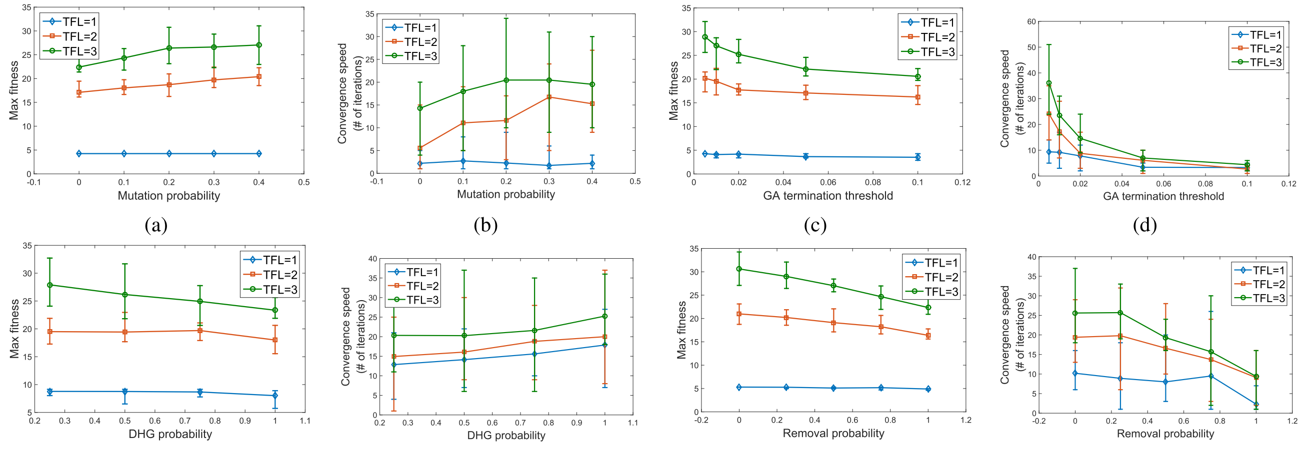


Fig. 5. Stability and optimality evaluation of GBGA. The figure shows the relative difference between the outcome from each run and the mean of outcomes from all the runs, with different TFLs.

We also conduct evaluations under the same environment (averaged over 10 to 15 runs) with different parameter settings under different TFLs, as shown in Fig. 6. Specifically, totally four settings are experimented, in which GBGA's optimality (represented by maximum fitness) and convergence speed (represented by the number of iterations used to find maximum fitness) are used as the metrics. Fig. 6a and Fig. 6b show the evaluation results for different mutation probabilities. It can be found that larger mutation probability can lead to a raise in maximum fitness, due to more complete search. On the other hand, when the mutation probability is smaller, the algorithm converges to a less optimal final outcome. In both cases, the convergence speed is a little faster. In Fig. 6c and Fig. 6d, GBGA is evaluated with different termination threshold. The results show that maximum fitness decreases while convergence speed increases as the termination threshold becomes larger. Although being a stochastic algorithm, GBGA can still have certain level of *local determinism* [30], which means the choices in some procedures can be biased based on experience or rules instead of being determined by absolute randomness. For GBGA, we evaluate the following two types



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 6. GBGA evaluation with different settings under different TFLs: (a&b) GBGA evaluation with different mutation probabilities; (c&d) GBGA evaluation with different termination thresholds; (e&f) GBGA evaluation with different dual-hop overlay network probabilities; (g&h) GBGA evaluation with different removal probabilities in mutation.

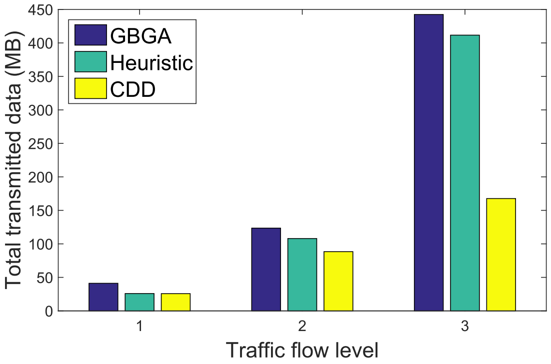


Fig. 7. Total transmitted data with different TFLs.

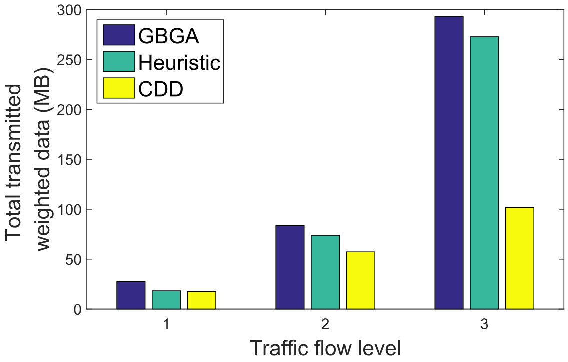


Fig. 8. Total transmitted weighted data with different TFLs.

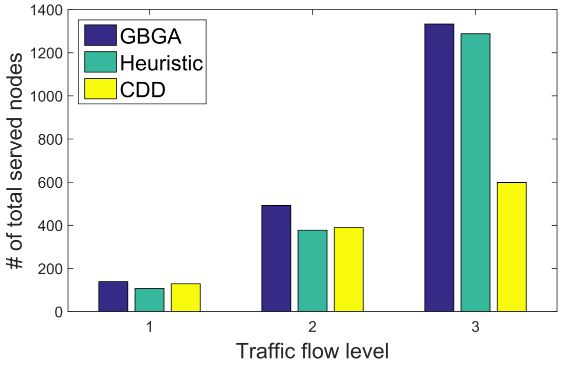


Fig. 9. Total number of served nodes with different TFLs.

of local determinism. Fig. 6e and Fig. 6f show the evaluation results when the dual-hop topology generation probability in the initial population generation procedure is changed. It can be found that the maximum fitness decreases when the dual-hop probability increases since dual-hop transmission is less efficient. On the other hand, the algorithm takes more time to converge when dual-hop probability is larger, due to the badly generated initial population. We also evaluate how mutation affects GBGA. In Fig. 6g and Fig. 6h, we evaluate the overlay network removal probability in a variable-length mutation, which has a probability of 0.5 itself. We can see that the maximum fitness drops when it is more likely to remove an overlay network, although it is easier for the algorithm to converge.

B. Algorithm Comparisons

In this subsection, we will evaluate GBGA, heuristic algorithm and CDD in the context of IoV from different aspects. The main differences between CDD and our solutions are described in Section II. The results under different TFLs are given in Figs. 7-10. Specifically, Fig. 7 shows the total transmitted data during the simulated period. It can be seen that GBGA is better than heuristic algorithm, and both of them are better than CDD. When the traffic density is low, the gain is not significant since there are less chances to create overlay networks and thus the resources are relatively sufficient. On the contrary, for high traffic density scenario, GBGA and heuristic algorithm can perform much better than CDD. Since we consider requests with different priorities in our scenario, we also compare the total transmitted weighted data in Fig. 8, and results similar to Fig. 7 can be found.

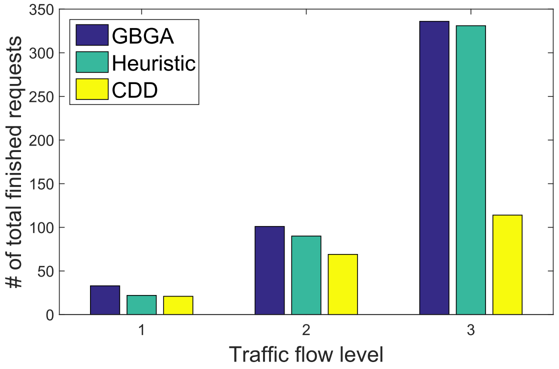


Fig. 10. Total number of finished requests with different TFLs.

In Fig. 9, the results on the total numbers of served nodes during the simulated period are shown, which reflect the concurrent services support capability. Although both CDD and our methods can serve nearly all the requests in low traffic density cases, CDD cannot deal with the large number of requesting nodes as traffic density grows. Finally, in Fig. 10, we show the results on the total number of finished requests during the simulated period. As can be seen in the figure, both GBGA and heuristic algorithm can finish a large amount of the requests in high traffic density case, whereas CDD does not have the same capability.

VI. CONCLUSION

In this paper, we have studied the service-oriented dynamic vehicular connection management in SD-IoV that is able to improve the efficiency and flexibility of resource utilization, enhance the QoS guarantee, and support multiple concurrent requests. The proposed architecture of SD-IoV exemplifies how the cutting-edge technologies including SDN, NFV, and IoV can be integrated towards realizing the next generation ITS. The vehicular connection management results, i.e., overlay vehicular networks, serve as a novel resource utilization pattern in the SD-IoV scenario and can take advantage of the new features of SD-IoV. The developed GBGA and heuristic algorithm can generate near-optimal results under different traffic conditions. For future works, we will further improve the SD-IoV system by designing adaptive and learning algorithms based on historical resource allocation results under different conditions.

REFERENCES

[1] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, Aug. 2014.

[2] N. Liu. (Aug. 2016). *Internet of Vehicles: Your Next Connection*. [Online]. Available: http://www1.huawei.com/enapp/28/hw-110836.htm

[3] Gartner. (Aug. 2016). *Gartner Says by 2020, a Quarter Billion Connected Vehicles Will Enable New in-Vehicle Services and Automated Driving Capabilities*. [Online]. Available: http://www.gartner.com/newsroom/id/2970017

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

[5] D. F. Macedo, D. Guedes, L. F. M. Vieira, M. A. M. Vieira, and M. Nogueira, "Programmable networks—From software-defined radio to software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 1102–1125, 2nd Quart., 2015.

[6] G. Karagiannis *et al.*, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 584–616, 4th Quart., 2011.

[7] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[8] S. Wang, C. Fan, C.-H. Hsu, Q. Sun, and F. Yang, "A vertical handoff method via self-selection decision tree for Internet of vehicles," *IEEE Syst. J.*, vol. 10, no. 3, pp. 1183–1192, Sep. 2016.

[9] J. A. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, "Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and Internet of Things technologies," *IEEE Wireless Commun.*, vol. 22, no. 6, pp. 122–128, Dec. 2015.

[10] C.-H. Wang, C.-T. Chou, P. Lin, and M. Guizani, "Performance evaluation of IEEE 802.15.4 nonbeacon-enabled mode for Internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3150–3159, Dec. 2015.

[11] AT&T. (Aug. 2016). *Connected Car*. [Online]. Available: http://about.att.com/sites/internet-of-things/connected_car

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

[13] H. Zhou *et al.*, "Whitefi infostation: Engineering vehicular media streaming with geolocation database," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2260–2274, Aug. 2016.

[14] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Apr. 2008.

[15] S. Jain *et al.*, "B4: Experience with a globally-deployed software defined wan," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, Oct. 2013.

[16] I. T. Haque and N. Abu-Ghazaleh, "Wireless software defined networking: A survey and taxonomy," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2713–2737, 4th Quart., 2016.

[17] H. Huang, P. Li, S. Guo, and W. Zhuang, "Software-defined wireless mesh networks: Architecture and traffic orchestration," *IEEE Netw.*, vol. 29, no. 4, pp. 24–30, Jul./Aug. 2015.

[18] J. Chen, H. Zhou, N. Zhang, P. Yang, L. Gui, and X. Shen, "Software defined Internet of vehicles: Architecture, challenges and solutions," *J. Commun. Inf. Netw.*, vol. 1, no. 1, pp. 14–26, Jun. 2016.

[19] H. Li, M. Dong, and K. Ota, "Control plane optimization in software-defined vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7895–7904, Oct. 2016.

[20] X. Wang, C. Wang, J. Zhang, M. Zhou, and C. Jiang, "Improved rule installation for real-time query service in software-defined Internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 2, pp. 225–235, Feb. 2017.

[21] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular adhoc network with fog computing," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 1202–1207.

[22] I. Ku, Y. Lu, M. Gerla, R. L. Gomes, F. Ongaro, and E. Cerqueira, "Towards software-defined VANET: Architecture and services," in *Proc. 13th Annu. Medit. Ad Hoc Netw. Workshop (MED-HOC-NET)*, Jun. 2014, pp. 103–110.

[23] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Software-defined networking for RSU clouds in support of the Internet of vehicles," *IEEE Internet Things J.*, vol. 2, no. 2, pp. 133–144, Apr. 2015.

[24] K. Zheng, L. Hou, H. Meng, Q. Zheng, N. Lu, and L. Lei, "Soft-defined heterogeneous vehicular network: Architecture and challenges," *IEEE Netw.*, vol. 30, no. 4, pp. 72–80, Jul./Aug. 2016.

[25] 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.

[26] Z. He, D. Zhang, and J. Liang, "Cost-efficient sensory data transmission in heterogeneous software-defined vehicular networks," *IEEE Sensors J.*, vol. 16, no. 20, pp. 7342–7354, Oct. 2016.

[27] 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.

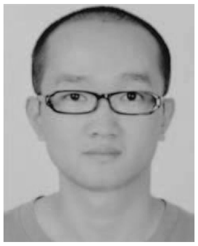
[28] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, Aug. 2016.

[29] T. Wood, K. K. Ramakrishnan, J. Hwang, G. Liu, and W. Zhang, "Toward a software-based network: Integrating software defined networking and network function virtualization," *IEEE Netw.*, vol. 29, no. 3, pp. 36–41, May/Jun. 2015.

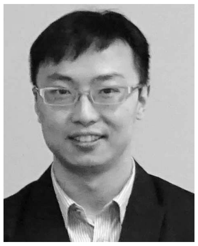
[30] S. Sivanandam and S. Deepa, *Introduction to Genetic Algorithms*. Springer, 2007.

[31] H. Lieu, N. Gartner, C. Messer, and A. Rathi, "Traffic flow theory," *Public Roads*, vol. 62, pp. 45–47, 1999.

Jiacheng Chen received the B.S. degree from the School of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, China, in 2013. He is currently working toward the Ph.D. degree with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China. He was a Visiting Ph.D. Student with the BBCR Lab, University of Waterloo, Canada, from 2015 to 2016. His research interests include resource management in wireless networks, vehicular networks, and white space networks.



Haibo Zhou received the Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University, Shanghai, China, in 2014. He is a Post-Doctoral Fellow with the Broadband Communications Research Group, University of Waterloo. His research interests include resource management and performance analysis in cognitive radio networks and vehicular networks.



Ning Zhang (S'12–M'16) received the B.Sc. degree from Beijing Jiaotong University in 2007; the M.Sc. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2010; and the Ph.D. degree from University of Waterloo in 2015. From 2015 to 2016, he was a Post-Doctoral Research Fellow with the BBCR Lab, University of Waterloo. He is currently a Post-Doctoral Research Fellow with the University of Toronto. His research interests include next generation wireless networks, software-defined networking, green communication, and physical layer security.

Wenchao Xu received the B.E. and M.E. degrees from Zhejiang University, Hangzhou, China, in 2008 and 2011, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. In 2011, he joined Alcatel Lucent Shanghai Bell Co., Ltd., where he was a Software Engineer for telecom virtualization. His interests include wireless communications with emphasis on resource allocation, network modeling, and vehicular ad hoc network.

Quan Yu (SM'16) received the B.S. degree in information physics from Nanjing University in 1986, the M.S. degree in radio wave propagation from Xidian University in 1988, and the Ph.D. degree in fiber optics from University of Limoges in 1992. He was invited to serve as the Honorary Dean of the School of Telecommunications Engineering, Xidian University, in 2012. Since 1992, he has been a Senior Engineer with the Faculty of the Institute of China Electronic System Engineering Corporation, where he is currently a Research Fellow. He also is an Adjunct Professor and a Ph.D. Supervisor with Xidian University and Shanghai Jiao Tong University. His main areas of research interest are the architecture of wireless networks, optimization of protocols, and cognitive radio. He is a Fellow of the Chinese Academy of Engineering.

Lin Gui (M'08) received the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2002. Since 2002, she has been with the Institute of Wireless Communication Technology, Shanghai Jiao Tong University, Shanghai, China, where she is currently a Professor. Her research interests include HDTV and wireless communications.

Xuemin (Sherman) Shen (M'97–SM'02–F'09) received the B.Sc. degree from Dalian Maritime University, China, in 1982, and the M.Sc. and Ph.D. degrees from Rutgers University, NJ, USA, in 1987 and 1990, respectively, all in electrical engineering. He is currently a Professor and the University Research Chair with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is also the Associate Chair for Graduate Studies. His research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He is an elected member of the IEEE ComSoc Board of Governor and the Chair of the Distinguished Lecturers Selection Committee. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. He received the Excellent Graduate Supervision Award in 2006, the Outstanding Performance Award in 2004, 2007, 2010, and 2014 from the University of Waterloo, the Premiers Research Excellence Award in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. He served as the Technical Program Committee Chair/Co-Chair of the IEEE Globecom16, the IEEE Infocom14, the IEEE VTC10 Fall, and the IEEE Globecom07; the Symposia Chair of the IEEE ICC10; the Tutorial Chair of the IEEE VTC11 Spring and the IEEE ICC08; the General Co-Chair of ACM Mobihoc15, Chinacom07, and QShine06; and the Chair of the IEEE Communications Society Technical Committee on Wireless Communications and P2P Communications and Networking. He also serves/served as the Editor-in-Chief of IEEE NETWORK, *Peer-to-Peer Networking and Application*, and *IET Communications*; a Founding Area Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS; an Associate Editor of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *Computer Networks*, and *ACM/Wireless Networks*; and the Guest Editor of IEEE JSAC, IEEE WIRELESS COMMUNICATIONS, *IEEE Communications Magazine*, and *ACM Mobile Networks and Applications*.