Coding-Assisted Broadcast Scheduling via Memetic Computing in SDN-Based Vehicular Networks

Kai Liu, Member, IEEE, Liang Feng, Member, IEEE, Penglin Dai, Member, IEEE, Victor C. S. Lee, Member, IEEE, Sang Hyuk Son, Fellow, IEEE, and Jiannong Cao, Fellow, IEEE

Abstract—This paper embarks the first study on exploiting the synergy between vehicular caching and network coding for enhancing the bandwidth efficiency of data broadcasting in heterogeneous vehicular networks by presenting a service architecture that exercises the software defined networking concept. In particular, we consider the scenario where vehicles request a set of information and they could be served via heterogeneous wireless interfaces, such as roadside units and base stations (BSs). We formulate a novel problem of coding-assisted broadcast scheduling (CBS), aiming at maximizing the broadcast efficiency for the limited BS bandwidth by exploring the synergistic effect between vehicular caching and network coding. We prove the NP-hardness of the CBS problem by constructing a polynomial-time reduction from the simultaneous matrix completion problem. To efficiently solve the CBS problem, we employ memetic computing, which is a nature inspired computational paradigm for tackling complex problems. Specifically, we propose a memetic algorithm, which consists of a binary vector representation for encoding solutions, a fitness function for solution evaluation, a set of operators for offspring generation, a local search method for solution enhancement, and a repair operator for fixing infeasible solutions. Finally, we build the simulation model and give a comprehensive performance evaluation to demonstrate the superiority of the proposed solution.

Index Terms—SDN-based vehicular network, data broadcast, network coding, memetic algorithm.

Manuscript received January 12, 2017; revised June 9, 2017 and August 22, 2017; accepted August 24, 2017. Date of publication September 22, 2017; date of current version August 1, 2018. This work was supported in part by the National Science Foundation of China under Grant 61572088 and Grant 61603064, in part by the Chongqing Application Foundation and Research in Cutting-Edge Technologies under Grant cstc2017jcyjAX0026, in part by the Frontier Interdisciplinary Research Fund for Central Universities under Grant 106112017CDJQJ188828, in part by ICT Research and Development Program of MSIP/IITP, Resilient Cyber-Physical Systems Research under Grant 14-824-09-013, in part by GRL Program through NRF under Grant 2013K1A1A2A02078326, and in part by the DGIST Research and Development Program (CPS Global Center) funded by MSIP. The Associate Editor for this paper was L. Yang. (Corresponding authors: Kai Liu; Sang Hyuk Son).

K. Liu and L. Feng are with the Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing University, Chongqing 400040, China, and also with the College of Computer Science, Chongqing University, Chongqing 400040, China (e-mail: liukai0807@gmail.com; liangf@cqu.edu.cn).

P. Dai is with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China (e-mail: penglin@swjtu.edu.cn).

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

S. H. Son is with the Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology, Daegu 42988, South Korea (e-mail: son@dgist.ac.kr).

J. Cao is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: jiannong.cao@polyu.edu.hk).

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.2017.2748381

Vehicular networks have been envisioned as a promising paradigm for achieving breakthroughs in future intelligent transportation systems by improving road safety, enhancing driving experiences, and providing various value-added services. The dedicated short-range communication (DSRC) is being standardized as a de facto protocol in vehicular networks to support both infrastructure-to-vehicle (I2V) and vehicle-to-vehicle (V2V) communications [1]. Meanwhile, alternative wireless interfaces such as cellular networks, Wi-Fi and Bluetooth also coexist to form a heterogeneous vehicular network. Despite the advances in current wireless communication technologies, it is still challenging to provide efficient data services and optimize the bandwidth efficiency in vehicular networks, due to its intrinsic characteristics such as high mobility of vehicles, sparse distribution of roadside units (RSUs), diverse vehicle densities, dynamic traffic conditions and application requirements, etc.

A great number of studies have investigated RSU-based data services in vehicular networks [2]–[4]. However, with ever-increasing data demands by various applications, simply relying on I2V communications for data dissemination within RSUs’ coverage cannot support future large-scale systems. In addition, the data transmission with such an architecture may suffer from intermittent connections and unpredictable delay due to the short I2V communication range and the sparse deployment of RSUs. Many studies have incorporated V2V communications to assist I2V-based data dissemination and enhance system scalability [5]–[7]. Nevertheless, these studies cannot well exploit the heterogeneous wireless communication resources for data services. This work considers a heterogeneous vehicular network, where vehicles can retrieve part of their requested data items from RSUs via DSRC when they are passing by, and when they are out of RSUs’ coverage, the rest of services will be completed by the base station (BS) of the cellular network. Accordingly, vehicles may be jointly served via different interfaces. This work will focus on the data scheduling for the BS with the aims of maximizing its bandwidth efficiency and enhancing overall system performance.

Network coding has aroused significant interest in the community of wireless communication, and it has shown great potential on improving bandwidth efficiency in vehicular networks [8], [9]. In this work, the bitwise exclusive-or (⊕) coding operation is considered in data scheduling at BSs based
on the knowledge of vehicular cache information. On the other hand, the concept of software defined network (SDN) has shown great potential for facilitating data scheduling, improving resource utilization and enhancing service management in vehicular networks [10]–[13]. The core idea of SDN is the separation of the control plane and the data plane. The SDN controller, which has global network knowledge, formulates data broadcasting and forwarding rules for network nodes such as RSUs, BSs and vehicles via the control plane. As a result, the SDN controller can define the behavior of individual vehicles, RSUs and BSs by making scheduling decisions based on the global view. In this work, an SDN-based service architecture is presented to enable ‘logically centralized’ control on the global view. In this work, an SDN-based service architecture is presented to enable ‘logically centralized’ control in a heterogenous vehicular network, which forms the basis of incorporating network coding and vehicular caching in scheduling. Inspired from the superiority of memetic computing in solving complex problems [14], this work makes the first effort on proposing a memetic algorithm for efficient data scheduling at the SDN controller, with the objectives of maximizing bandwidth efficiency and enhancing system scalability. The main contributions of this work are outlined as follows.

- We present an SDN-based service architecture for heterogeneous vehicular communication environments, which enables the centralized scheduling at the controller with global view of vehicles, RSUs and BSs. The implementation, benefits as well as challenges of data services in such an architecture have been discussed.

- We formulate a novel problem of Coding-assisted Broadcast Scheduling (CBS), which targets at incorporating vehicular caching and network coding into the scheduler at the control plane, and maximizing the bandwidth efficiency of the BS on data services.

- We prove that CBS is NP-hard by constructing a polynomial-time reduction from the simultaneous matrix completion problem [15].

- This is the first known study on proposing a particular memetic algorithm (MA) for solving the data dissemination problem in vehicular networks. Specifically, based on the characteristics of the formulated CBS problem, the proposed MA consists of a binary vector representation for encoding solutions, a fitness function for solution evaluation, a set of reproduction operators (i.e., parents selection, crossover and mutation) for solution evolution, as well as a repair operator for fixing infeasible solutions.

- We build the simulation model and evaluate the performance of MA under a variety of application scenarios. The comprehensive simulation results demonstrate the effectiveness and the scalability of the proposed solution.

The rest of this paper is organized as follows. Section II reviews the related work. Section III presents the SDN-based service architecture. Section IV formulates the coding-assisted broadcast scheduling problem and proves that it is NP-hard. In Section V, we propose a memetic algorithm. In Section VI, we build the simulation model and evaluate the algorithm performance. Finally, Section VII concludes this work and discusses future research directions.

II. RELATED WORK

Great efforts have been devoted to data scheduling in vehicular networks. Zhang et al. [16] proposed a VC-MAC (Vehicular Cooperative Media Access Control) protocol, which jointly exploits the cooperative of I2V and V2V communications to enhance spatial reusability. It considers the scenario where vehicles request files of common interest when passing through the RSU, and the VC-MAC enables vehicles to cooperatively share their cached information via V2V communication when they are out of the RSU’s coverage, so as to improve the total throughput. Zeng et al. [17] incorporated channel prediction into the centralized cooperative data dissemination scheduling strategy for vehicular communications. Specifically, a recursive least squares algorithm is proposed to realize large-scale channel prediction with low computational complexity, and on this basis, a scheduling strategy is proposed for cooperative data dissemination. Bi et al. [18] proposed a Multi-Channel Token Ring MAC Protocol (MCTRP) for V2V communication. The CSMA/CA mechanism is applied for delivering emergency messages with low delay. In addition, a token-based data exchange protocol is designed to improve bandwidth efficiency for non-safety applications. Cheng et al. [19] investigated the feasibility of V2V and V2I integration by operating V2V links as underlay device-to-device (D2D) links. A suite of methods are proposed for low-complexity yet high-performance D2D operations, including an interference coordination approach, a resource-allocation scheme, and a transmission scheduling framework.

From the application point of view, a number of studies have incorporated network coding into data dissemination in vehicular networks. Li et al. [8] investigated popular content distribution via push-based broadcast and proposed a cooperative content distribution protocol called CodeOn. The symbol level network coding technique is adopted to facilitate high speed content downloading and counteract the packet loss problem. Firooz and Roy [20] proposed a collaborative data download scheme using network coding. They considered a two-phase data dissemination scenario where V2I and V2V communications are cooperated to complete the service. Liu et al. [8] considered both communication constraints and application requirements in vehicular networks. A network coding-assisted scheduling algorithm is proposed to best exploit the joint effect of V2V and V2I communications and provide efficient data services. Zhu et al. [21] proposed a multiple-vehicle protocol for collaborative data downloading using network coding. A multiple-vehicle collaborative download model is presented, and a probability mass function of the downloading completion time is derived to quantify the performance gains obtained from network coding.

A few work has considered the SDN concept to further facilitate the data service in vehicular networks. He et al. [10] proposed an SDN-based architecture to enable rapid network innovation for vehicular communications, in which vehicles and roadside units are abstracted as SDN switches. In addition, network resources such as wireless bandwidth can be allocated and assigned by the logically centralized control plane, which provides a more agile configuration capability. Liu et al. [13] presented an SDN architecture for GeoBroadcast
in vehicular networks. The SDN controller helps the source vehicles find the path towards the destination vehicles with knowing the topological and geographical information. Liu et al. [22] investigated cooperative data dissemination in a hybrid I2V and V2V communication environment, and a cooperative data scheduling problem is formulated. The proposed model and solution, which are based on the centralized scheduler at the RSU, represent the first known vehicular network implementation of SDN concept. Cheng et al. [23] proposed a 5GenCIV framework which integrates the design of 5G and intelligent vehicles to enable affordable and reliable autonomous driving. Specifically, the SDN feature in the 5G network is exploited to reduce the network latency for specific self-driving operations with the centralized control plane.

Distinguishing from previous efforts, this work aims at best exploiting the benefit of network coding and vehicular caching in heterogeneous vehicular communication environments by presenting an SDN-based data service architecture and proposing a dedicated scheduling solution at the controller, so as to optimize the utilization of network resources and enhance overall system performance.

III. SYSTEM ARCHITECTURE

Fig. 1 shows a typical system architecture of SDN-based heterogeneous vehicular networks [10], where RSUs, BSs, vehicles and other wireless devices are abstract as switches in conventional SDN, representing the data plane. On the other hand, given specific services, the scheduling decisions including the bandwidth allocation, routing protocol, data scheduling, etc., are exercised by the control plane. Based on the received control messages, switches such as RSUs and BSs will operate accordingly.

In the concerned service scenario, vehicles ask for the service with common interest such as parking slots, road conditions, gas stations, etc., which are jointly provided by heterogenous wireless communication interfaces such RSUs and BSs. RSUs are installed sparsely along the roads. Vehicles in the RSU’s coverage (denoted by the dotted circle in Fig.1) can retrieve information via DSRC. Nevertheless, due to the limited communication range and intermittent connections of RSUs, services may not be able to be completed in the RSU’s coverage. For those partially served vehicles by RSUs, they are able to continue retrieving data items via other communication interfaces with larger coverage, such as cellular networks. Although the capacity of modern cellular networks (e.g., 4G and 5G) have improved significantly, due to the ever increasing data service demand by various mobile applications, it is still critical to make best use of the scarce bandwidth resource of BSs allocated to vehicular information services. Therefore, it is desirable to maximize the bandwidth efficiency of BSs for data services in heterogeneous vehicular communication environments. To this end, we present an SDN-based service architecture to enable ‘logically centralized’ control and facilitate the data scheduling in such a system. Detailed service procedures are presented as follows.

Vehicles periodically update their status to the controller via the cellular network interface, including positions, cached contents and kinematic information, etc. Vehicles in the RSU’s coverage will be instructed to turn into the DSRC interface and communicate with the RSU, so that the RSU is aware of the set of vehicles in its coverage as well as their service status. RSUs can either follow the control instructions from the control plane, or alternatively, they may exercise certain intelligence and make part of scheduling decisions individually. At early stage of exploring such a data dissemination problem in SDN-based vehicular networks, this work does not consider tight cooperation between RSUs and BSs on information services. In other words, any existing scheduling algorithms proposed for RSUs such as [3] and [24] can be incorporated into the presented system architecture. In any case, vehicles will be able to cache part of their requested data items via RSUs’ services. Accordingly, when vehicles are not in the coverage of RSUs, they will be instructed by the controller to turn into the cellular network interface to complete the service. Finally, with the global information of vehicle service status, the control plane makes scheduling decisions and informs the BS via the control message, so that the BS will provide data services accordingly via the cellular network interface.

With the above SDN-based service architecture, the system aims at maximizing overall system performance by implementing an efficient scheduling policy at the control plane. In particular, the bitwise exclusive-or (⊕) coding operation is adopted at the BS for data broadcast due to its trivial implementation overhead. For example, given an encoded packet $p = d_1 \oplus d_2$, to decode a data item (say $d_1$) from $p$, it requires the remaining data items in $p$ (say $d_2$) by computing $d_1 = p \oplus d_2$. Meanwhile, different from traditional wireless sensor nodes which have very limited cache size, vehicles can support large storage, and the cache size will not be a hurdle of system performance. In view of this, we consider that vehicles are allowed to cache those encoded packets even if these packets cannot be decoded out immediately.

Fig. 1 shows an example to illustrate the benefit of caching encoded packets, and meanwhile, it reveals the challenges in designing an efficient scheduling policy. Considering there are four vehicles $V_1$, $V_2$, $V_3$ and $V_4$, which are out of the RSUs’ coverage. The set of data items is $d_1$, $d_2$, $d_3$ and $d_4$. The current cached packets are $d_4$ and $d_1 \oplus d_3$ for $V_1$; $d_2$ and $d_1 \oplus d_3$ for $V_2$; $d_4$ and $d_2 \oplus d_3 \oplus d_4$ for $V_3$; and $d_1$ and
d₂ for V₄. To maximize the bandwidth efficiency of the BS, it is expected to complete the service to all the vehicles with the minimum number of broadcast transmissions. Consider the duration for broadcasting a data item (or an encoded packet) as a time unit. In this example, it is observed that at least two time units is required to complete the service by broadcasting a time unit. In this example, it is observed that at least two time durations for broadcasting a data item (or an encoded packet) as well as the best solutions is to broadcast sequence to serve all the vehicles, which requires 3 time slots.

The primary notations in this paper are summarized in Table I.

With the above knowledge, the coding-assisted broadcast scheduling (CBS) problem is described as follows. Given the set of vehicles \( V = \{V_1, V_2, \ldots, V_{|V|}\} \), the set of data items \( D = \{d_1, d_2, \ldots, d_{|D|}\} \), and the set of cached packets of each vehicle \( C(V_m) \) \((V_m \in V)\), the control plane targets at maximizing the bandwidth efficiency of the BS by determining the set of encoded packets \( P \), which can complete the service to all the vehicles with the minimum number of time slots.

**B. NP-Hardness**

We prove the NP-hardness of CBS by constructing a polynomial-time reduction from the simultaneous matrix completion problem, which is a well-known NP-hard problem [15]. The general concept of the simultaneous matrix completion problem is described as follows. Given a set of matrices, each matrix contains a mixture of numbers and variables. Each particular variable can only appear once per matrix but may appear in several matrices. The objective is to find values for these variables such that all resulting matrices simultaneously have full rank. It has been shown in [15] that the simultaneous matrix completion problem over \( GF(2) \) is NP-complete. To have a clear exposition, before giving the formal proof, we illustrate a sketch of the idea with an example.

Suppose \( D = \{d_1, d_2, d_3, d_4\} \), and there are two vehicles \( V_1 \) and \( V_2 \) with cache sets \( C(V_1) = \{d_4, d_4 \oplus d_3 + d_1 \oplus d_3 + d_4\} \) and \( C(V_2) = \{d_2, d_1 \oplus d_1\} \). For \( V_i \), the number of cached packets is denoted by \( |C(V_i)| \), and each cached packet is represented by a \( |D| \)-dimensional coefficient vector. Then, we can construct a \( |C(V_i)| \times |D| \) matrix for \( V_i \), representing its cached contents. For example, the constructed matrix for \( V_1 \) is:

\[
\begin{bmatrix}
0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 \\
1 & 0 & 1 & 1
\end{bmatrix}
\]

In this example, we observe that the packet \( p^1_1 \) (i.e. \( d_1 \oplus d_3 \oplus d_4 \)) in \( C(V_1) \) can be derived from the other two packets \( p^1_1 \) and \( p^2_2 \) in \( C(V_1) \) by computing \( p^1_1 \oplus p^2_2 \). It indicates that

### TABLE I

<table>
<thead>
<tr>
<th><strong>Notation</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>( V )</td>
<td>The set of vehicles, ( V = {V_1, V_2, \ldots, V_{</td>
</tr>
<tr>
<td>( D )</td>
<td>The set of requested data items, ( D = {d_1, d_2, \ldots, d_{</td>
</tr>
<tr>
<td>( p^i_j )</td>
<td>The ( i )-th encoded packet of vehicle ( V_m ), where ( V_m \in V )</td>
</tr>
<tr>
<td>( a(p^i_j) )</td>
<td>Coefficient vector of ( p^i_j ), ( a(p^i_j) = [a(p^i_j)_1, a(p^i_j)<em>2, \ldots, a(p^i_j)</em>{</td>
</tr>
<tr>
<td>( C(V_m) )</td>
<td>The set of cached packets by vehicle ( V_m ), ( C(V_m) = {p^m_1, p^m_2, \ldots, p^m_{</td>
</tr>
<tr>
<td>( A(V_i) )</td>
<td>A (</td>
</tr>
<tr>
<td>( T )</td>
<td>The maximum scheduling period</td>
</tr>
<tr>
<td>( N )</td>
<td>The total number of ( D )-dimension coefficient vectors initialized for solution search</td>
</tr>
<tr>
<td>( \chi_m )</td>
<td>A feasible solution, represented by a binary vector with ( N )-dimension which satisfies ( 0 &lt; \sum_{k} \chi_m[k] \leq T )</td>
</tr>
<tr>
<td>( M )</td>
<td>The number of feasible solutions in the population</td>
</tr>
<tr>
<td>( \alpha_0 )</td>
<td>Initialized population, ( \alpha_0 = {x_1, x_2, \ldots, x_M} )</td>
</tr>
<tr>
<td>( r_j )</td>
<td>The rank of the matrix ( A(V_j) ) constructed for vehicle ( V_j )</td>
</tr>
<tr>
<td>( f )</td>
<td>The fitness function, ( f = {\sum_{j=1}^{</td>
</tr>
</tbody>
</table>
given $p_1^*$ and $p_2^*$, $p_3^*$ actually has no further contribution to decode out any new data item for $V_1$. Therefore, it is expected to find the set of cached packets which are independent with each other. To this end, we further transform the matrix to its reduced row echelon form by Gaussian elimination. In this example, the reduced row echelon form of the constructed matrix is 

$$
\begin{bmatrix}
1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

The transformed matrix for $V_1$ is denoted by $A(V_1)$, which is a $[C'(V_1)] \times |D|$ matrix, represented by $[a'_(p_1^*) \ a'_(p_2^*) \ \ldots \ a'_(p_{[C'(V_1)]]})^T$, where $|C'(V_1)|$ is the number of independent vectors and $[C'(V_1)] \leq |C(V_1)|$. In addition, since the rows of $A(V_1)$ are independent, we have $|C'(V_1)| \leq |D|$. In particular, if $|C'(V_1)|=|D|$, $A(V_1)$ is a full rank matrix (it is an identity matrix in the reduced row echelon form). Then, all the data items can be decoded out from $C(V_1)$. Otherwise (e.g. $|C'(V_1)| < |D|$), at least $|D| - |C'(V_1)|$ packets are required to serve $V_1$ completely.

By obtaining $A(V_1) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and $A(V_2) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$, we may observe that there are at least two packets (e.g. $p_1^*$ and $p_2^*$) are required to serve both $V_1$ and $V_2$. Suppose we could find such two packets, by pendaing the two corresponding coefficient vectors $a(p_1^*)$ and $a(p_2^*)$ to both $A(V_1)$ and $A(V_2)$, the two matrices $[a'_(p_1^*) \ a'_(p_2^*) \ a'_(p_{[C'(V_1)]})^T] \text{and} \ [a'_(p_1^*) \ a'_(p_2^*) \ a'_(p_{[C'(V_1)]})^T]$ should be full rank. In this example, we can check that by setting $a(p_1^*) = [0 \ 0 \ 1 \ 0]$ and $a(p_2^*) = [0 \ 1 \ 0 \ 1]$, the two matrices will become full rank. Therefore, both $V_1$ and $V_2$ will be completely served.

**Theorem 1: The Coding-assisted Broadcast Scheduling Problem is NP-hard.**

**Proof:** Given $V$, $D$ and $C(V_i)$ ($V_i \in V$), we construct a set of coefficient matrices $A = \{A(V_1), A(V_2), \ldots, A(V_{|V|})\}$, where $A(V_i)$ is represented by $[a'_(p_1^*) \ a'_(p_2^*) \ \ldots \ a'_(p_{[C'(V_i)]})^T]$, which is the reduced row echelon form of the matrix $[a(p_1^*) \ a(p_2^*) \ \ldots \ a(p_{[C'(V_i)]})^T]$. Accordingly, there are $|C'(V_i)|$ independent packets cached by $V_i$. On the other hand, as a vehicle can decode out at most one data item by receiving a packet, it requires at least another $|D| - |C'(V_i)|$ packets to serve $V_i$ (i.e. to decode out $|D|$ data items). Further, considering the service to all the vehicles, denote $D = \min (|[C'(V_i)]|) \forall V_i \in V$, then it requires at least $|D| - L$ packets to complete the service. Therefore, the CBS problem is to schedule the $|D| - L$ packets to serve all the vehicles and maximize the bandwidth efficiency.

Denote the set of scheduled packets as $P^* = \{p_1^*, p_2^*, \ldots, p_{|D|-L}^*\}$, and the coefficient vector of $p_j^*$ is denoted by $a(p_j^*)$ ($p_j^* \in P^*$). Then, by pendaing the set of coefficient vectors $a(p_j^*)$ $(\forall p_j^* \in P^*)$ to each $A(V_i) \in A$, we obtain a new set of matrices $A^* = \{A^*(V_1), A^*(V_2), \ldots, A^*(V_{|V|})\}$, where $A^*(V_i) = [a'_(p_1^*) \ a'_(p_2^*) \ \ldots \ a'_(p_{[C'(V_i)]})^T] \forall V_i \in V$.

Since $V_i$ can be served if and only if $A^*(V_i)$ can be transformed to an identity matrix by Gaussian elimination (i.e. $A^*(V_i)$ is full rank), the CBS problem is equivalent to the determination of the set of vectors $a(p_1^*), a(p_2^*), \ldots, a(p_{|D|-L}^*)$ over $GF(2)$, so that all the matrices in $A^*$ are in full rank simultaneously. The above proves that the CBS problem is NP-hard.

The above analysis shows that to find the optimal solution, namely, $P^* = \{p_1^*, p_2^*, \ldots, p_{|D|-L}^*\}$, it is equivalent to determining the set of coefficient vectors $a(p_1^*), a(p_2^*), \ldots, a(p_{|D|-L}^*)$, so that the set of matrices $A^* = \{a'_(p_1^*) \ a'_(p_2^*) \ \ldots \ a'_(p_{|D|-L}^*)^T\}$ ($\forall V_i \in V$) are in full rank simultaneously. As $a(p)$ is a $D$-dimension vector over $GF(2)$ (i.e., $a(p) = \{a(p_1), a(p_2), \ldots, a(p_{|D|})\}$, and $a(p) \in \{0, 1\}$), there are in total $2^{|D|}$ candidate coefficient vectors. Then, there are $C_{|D|-L}$ combinations for selecting $|D| - L$ coefficient vectors out of the whole set. For the $i$th combination, denote the corresponding set of packets as $P^i = \{p_1^i, p_2^i, \ldots, p_{|D|-L}^i\}$. Denote $P = \{P^1, P^2, \ldots, P^{|D|-L}\}$. Then, the optimal solution $P^*$ is represented by:

$$
P^* = \{P^k | P^k \in P \text{ and } C_{fullrank}\}
$$

where $C_{fullrank}$ stands for the condition that the corresponding set of coefficient vectors of $P^k$ (i.e., $a(p_1^k), a(p_2^k), \ldots, a(p_{|D|-L}^k)$) results in the simultaneous matrix completion for the set of matrices $A^*(V_i) = [a'_(p_1^k) \ a'_(p_2^k) \ \ldots \ a'_(p_{[C'(V_i)]})^T] \forall V_i \in V$.

**V. PROPOSED MEMETIC ALGORITHM**

With the fast growth of learning and optimization techniques [25], it is promising to have dedicated intelligent algorithms for solving complex data dissemination problems in vehicular networks. Memetic computing is a new paradigm proposed in the literature, which has been successfully applied to solve many real-world problems such as permutation flow shop scheduling, quadratic assignment problem, feature selection, etc. [14]. In this section, we propose a memetic algorithm (MA) to solve the CBS problem. The general idea of the proposed MA is outlined as follows.

Firstly, MA generates a population of solutions. Each solution represents a set of selected packets in a scheduling period. Subsequently, based on the designed fitness function, parent selection is performed to identify the solutions to undergo crossover and mutation for offspring generation. Further, a local search process kicks in to refine the solution. A repair operator is designed to fix infeasible solutions in due course. Afterwards, the fitter solutions among both parent solutions and the generated offspring solutions will be survived for the next generation through the population replacement process. The above procedures will repeat until the predefined stopping criterion is satisfied. Detailed steps of MA is presented as follows and the pseudo-code is shown in Alg. 1.

**A. Initialization**

Denote a solution as $\chi$, and it is represented by a binary vector with $N$-dimension, where $N$ is the total number of the
Algorithm 1 Memetic Algorithm (MA)

Input: The set of cached packet $C(V_m), m = 1, 2, \ldots, |V|$, the population size $M$, and the maximum scheduling period $T$

Output: The solution with the highest fitness value $\Lambda_0 = \{x_1, x_2, \ldots, x_M\}$

// Initialization: initialize the population
1: for $m = 1$ to $M$ do
2: for $i = 1$ to $N$ do
3: Randomly determine $x_m[i]$ as 0 or 1
4: if $\sum_{k=1}^{i} x_m[k] \geq T$ then
5: break
6: end if
7: end for
8: end for
9: Compute fitness value $f_{x_m}$ of each $x_m \in \Lambda_0$ according to Eq. 2
10: while the stopping criterion is not satisfied do
11: for $m = 1$ to $M/2$ do
// Offspring generation: generate the new population $\Lambda'$
12: Randomly select $k$ individuals from $\Lambda$
13: Sort the $k$ individuals in descending order according to their fitness values $f$
14: Select the top-2 individuals $x_l$ and $x_n$ as parent solutions
15: for $i = 1$ to $N$ do
16: Randomly assign $x_m'[i] \leftarrow x_l[i]$ or $x_m'[i] \leftarrow x_n[i]$
17: $x_{m+1}'[i] \leftarrow x_l[i] + x_n[i] - x_m'[i]$
18: end for
19: for each child $x_j' \in \{x_m', x_{m+1}'\}$ do
20: for $i = 1$ to $N$ do
21: if $\text{rand} < \rho$ then
22: $x_j'[i] \leftarrow 1 - x_j'[i]$
23: end if
24: end for
25: if $\sum_{i=1}^{N} x_j'[r] > T$ then
26: Apply the repair operator to $x_j'$
27: end if
28: end for
// Local Search:
29: for each child $x_j' \in \{x_m', x_{m+1}'\}$ do
30: for $i = 1$ to $N$ do
31: $\text{tmp} \leftarrow x_j'$
32: $\text{tmp}[i] \leftarrow 1 - x_j'[i]$
33: if $\sum_{i=1}^{N} x_j'[r] > T$ then
34: Apply the repair operator to $x_j'$
35: end if
36: Compute fitness value $f_{\text{tmp}}$ of $\text{tmp}$
37: if $f_{\text{tmp}} > f_{x_j'}$ then
38: $x_j' \leftarrow \text{tmp}$
39: end if
40: end for
41: end for
// Population replacement:
42: Sort the individuals in $\Lambda \cup \Lambda'$ according to $f$
43: Choose the top-$M$ individuals as the new population
44: end while
45: Output the best individual in the population $\Lambda$

randomly generated coefficient vectors with dimension $|D|$. Specifically, we set $\chi[i] = 1$ if the corresponding coefficient vector $a_i$ $(1 \leq i \leq N)$ is selected. Otherwise, $\chi[i] = 0$. Given the maximum scheduling period $T$, as most $T$ packets can be scheduled at a time. Accordingly, a feasible solution should satisfy $0 < \sum_{i=1}^{N} \chi_m[i] \leq T$. Fig. 2 shows the representation of a feasible solution.

B. Fitness Function
Recall that the objective of the CBS problem is to maximize the bandwidth efficiency by finding the minimum set of packets to complete the service. With this in mind, we design the following function to evaluate the fitness of a solution. Given a solution $\chi_m$ $(1 \leq m \leq M)$, if $\chi_m[i] = 1$ $(1 \leq i \leq N)$, then we attach the vector $a_i$ to the coefficient matrix of each vehicle. The total number of attached vectors is denoted by $k_m$, where $k_m = \sum_{i=1}^{N} \chi_m[i]$. For vehicle $V_j$, denote its originally and newly obtained matrices as $A(V_j)$ and $A(V_j)'$, respectively, and their corresponding ranks are denoted by $r_j$ and $r_j'$. Then, the fitness function $f$ of a solution is defined as the average matrix rank improvement contributed by each packet in this solution, which is computed by:

$$f = \frac{\sum_{j=1}^{N} (r_j' - r_j)/|V|}{k_m}$$

As defined, if the two solutions contain the same number of coefficient vectors (i.e., the same number of packets is scheduled), then the one improving more of the average matrix rank is preferred, because more data items can be retrieved by vehicles with the same bandwidth consumption. On the other hand, if the two solutions improve the same amount of average matrix rank (e.g., the same number of data items can be retrieved by vehicles), then the one containing less number of coefficient vectors is preferred, because it can achieve the same portion of services with less bandwidth consumption.

C. Offspring Generation
Parent selection, crossover and mutation are the reproduction operators conducted for generating offspring solutions. In particular, the tournament selection [26] is employed to identify the two parents in the population for reproduction. The basic idea of tournament selection is to select the best two
solutions $\chi_l$ and $\chi_n$ in terms of the fitness value $f$ from $k$ individuals in $A$. The method is shown in lines 12 ~ 14 in Alg.1. In addition, the uniform crossover and bit mutation [27] are adopted. Specifically, each bit of the offspring is randomly selected from either $\chi_l$ or $\chi_n$, and hence each offspring has approximately half of the genes from each parent. Further, to increase the diversity of the search process, the bit mutation operator is applied to all dimensions in each offspring solution with a predefined mutation probability $p$. The procedures of crossover and mutation operators are shown in lines 15 ~ 29 in Alg.1.

D. Local Search

The local search is implemented by a sequential flipping operations conducted on each solution. In particular, given a solution $\chi_m$, the local search will be executed from the first to the last dimension with value flipping, i.e., 0 to 1 or 1 to 0. Given the fitness function $f$, only if the flipping with fitness improvement would be accepted and stored in the solution. For example, suppose $\chi_m[i]$ is the newly obtained solution by flipping the $i^{th}$ bit of $\chi_m$. Then, we have $\chi_m[i] \leftarrow 1 - \chi_m[i]$ and $\chi_m[j] \leftarrow \chi_m[j], \forall j \neq i$. Afterwards, we compute and compare the fitness values of $\chi_m$ and $\chi_m'$, denoted by $f_{\chi_m}$ and $f_{\chi_m'}$, respectively. If $f_{\chi_m'} > f_{\chi_m}$, then $\chi_m$ is replaced by $\chi_m'$ in this round of local search. The procedures are shown in lines 30 ~ 43 in Alg.1.

E. Repair Operator

Note that the operations of crossover, mutation and local search may induce infeasible solutions where $\sum_{i=1}^{N} \chi_m[i] > T$. To address this issue, we design the following repair operator. First, it finds the set of ‘1’ bit in the solution $\chi_m$, which is $S = \{|i|\chi_m[i] = 1\}$. Second, it randomly selects $|S| - T$ elements from $S$ and sets them to 0. Clearly, we have $\sum_{i=1}^{N} \chi_m[i] = T$ by applying this repair operator.

F. Population Replacement

To determine the solutions for survival in the next generation and keep the population size unchanged, as shown in lines 44 ~ 45 in Alg.1, MA compares the fitness value of solutions in the fusion of parent and offspring. Then, the top-$M$ solutions are kept in the population for further evolution.

G. Termination

The stopping criterion in this study is defined as the condition that the predefined maximum number of generations ($G_{\text{max}}$) or the global optimum is reached, which is commonly used for memetic algorithms in the literature [28], [29]. Note that it is also straightforward to adopt other criterions for the proposed MA, such as maximal number of fitness evaluations, etc.

Based on the above described procedures, the computational complexity of MA is analyzed as follows. In the initialization phase, to generate a feasible solution $\chi$, it traverses at most $N$ bits. So, given the population size of $M$, the overhead of initialization is $O(M \cdot N)$. In the offspring generation phase, for parent generation, it computes the fitness value of $k$ random individuals. Base on the definition in Eq. 2, to compute $f$, it requires to check the coefficient matrix of $|V|$ vehicles, and the matrix size is bounded by $|D| \cdot |D|$, where $|D|$ is the size of database. For the crossover operation, it will traverse the two children solutions once with the overhead of 2$N$. For the mutation operation, it will traverse the two $N$-dimension children solutions. For each traverse, at most $N$ times of repair operation will be applied. As illustrated, the overhead of applying the repair operator is $N$. Accordingly, the overhead of offspring generation is $O(k \cdot |V| \cdot |D|^2 + 2N + 2N^2)$, which is $O(|V| \cdot |D|^2 + N^2)$. In the local search phase, it will traverse the two $N$-dimension children solutions. When examining each element, the overhead of applying repair operator is $N$, and the overhead of computing the fitness value is $|V| \cdot |D|^2$. Accordingly, the overhead of local search is $O(2N \cdot (N + |V| \cdot |D|^2))$, which is $O(N \cdot |V| \cdot |D|^2 + N^2)$. According to the stopping criterion, the bound of iteration is $G_{\text{max}}$.

Overall, the computational complexity of the proposed MA is $O(M \cdot N) + O(G_{\text{max}} \cdot (|V| \cdot |D|^2 + N^2)) + O(G_{\text{max}} \cdot (N \cdot |V| \cdot |D|^2 + N^2))$, which is $O(M \cdot N + G_{\text{max}} \cdot (N \cdot |V| \cdot |D|^2 + N^2))$. In practice, $M$, $N$ and $G_{\text{max}}$ are constant parameters defined by the algorithm, and their values do not depend on the system scale. For example, $M$, $N$ and $G_{\text{max}}$ are set to 100, 30 and 100, respectively in the simulation, which give the algorithm near optimal performance. Therefore, the computational complexity is reduced to $O(|V| \cdot |D|^2)$. Furthermore, since $D$ is the set of data items requested by all the vehicles, it is reasonably to assume a constant and small size of $|D|$ given particular applications. With above analysis, we may conclude that the complexity of MA can be approximate to a linear function of the number of vehicles in the service region. Therefore, the computational overhead of MA will not be the hurdle of system scalability.

VI. PERFORMANCE EVALUATION

A. Setup

The simulation model is built based on the system architecture presented in Section III for performance evaluation. Specifically, SUMO [30] is adopted to simulate vehicle mobility and generate vehicle traces. The map is extracted from a 2.2km $\times$ 2.2km area of the University Town in Chongqing City, in which 4 RSUs and 1 BS are simulated. Further, a control module based on C programming is implemented for enabling logically centralized control, including trace analysis, information collection, service coordination and management, scheduling and service interfaces for RSUs and BSs, etc. The MA is implemented by MATLAB, which runs the algorithm based on the parameters from the control module, and then outputs scheduling decisions to the controller. To give a clear view of the developed simulation model, Figure 3 illustrates the key functions and relationship of the three modules.
Due to the dedicated service architecture as presented in Section III, previous data scheduling algorithms designed for vehicular networks are not suitable to be adopted into the concerned system environment. Therefore, for comparison purpose, we implement two classic data broadcast algorithms in mobile computing systems. One is the Most Requested First (MRF) [31], which broadcasts the data item with the maximum number of pending requests. The other is Round-Robin [32], which broadcasts all the data items periodically. For MA, by default, the population size $M$ is set to 100; the dimension $N$ of the solution vector is set to 30; the maximum number of generations $G_{\text{max}}$ is set to 100; the mutation probability $\rho$ is set to 0.01; and the maximum scheduling period is set to 5 time units. For the system, a random linear coding strategy is implemented in each RSU, and the default transmission rate is 0.4 packet per time unit. The number of requested data items is set to 50, and 120 vehicles are simulated. Unless stated otherwise, all the experiments are conducted with the default setting.

In order to quantitatively evaluate the performance of algorithms, we design and analyze the following metrics.

- **Number of Broadcast Packet (NBP):** It is defined as the number of broadcast packets to serve all the requests. According to the analysis in IV-B, the lower bound of NBP can be computed by $|D| - L$, where $|D|$ is the set of requested data items and $L = \min_{V_j \in V} (|C'(V_j)|)$.

- **Average Service Delay (ASD):** It is designed for evaluating the algorithm performance on satisfying delay tolerant services. Specifically, denote the service delay for $V_j$ as $l_j$. That is, it takes $l_j$ time slots to achieve a full rank of the coefficient matrix for $V_j$. Then, ASD is defined as the summation of each vehicle’s service delay over the number of vehicles, which is computed by:

$$ASD = \frac{\sum_{j=1}^{|V|} l_j}{|V|}$$  \hspace{1cm} (3)

A lower value of ASD indicates a better performance of the algorithm on satisfying delay tolerant services.

- **Broadcast Productivity (BP):** It is designed for evaluating the algorithm performance on enhancing the bandwidth efficiency. Specifically, denote the ranks of the originally (at time $t_j^{\text{start}}$) and newly obtained (at time $t_j^{\text{end}}$) matrices of $V_j$ as $r_j$ and $r_j'$, respectively, and the service duration (i.e. $t_j^{\text{end}} - t_j^{\text{start}}$) is denoted by $t_j$. Then, BP is defined as the summation of each vehicle’s improved rank per time unit over the number of vehicles, which is computed by:

$$BP = \frac{\sum_{j=1}^{|V|} (r_j' - r_j)}{t_j}$$  \hspace{1cm} (4)

A higher value of BP indicates a better performance of the algorithm on enhancing the efficiency of broadcast bandwidth. The upper bound of BP is 1 because at most the average rank improves by one in a time unit when all the vehicles receive an outstanding data.

- **Average Service Ratio (ASR):** It is designed for evaluating the algorithm performance on scheduling real-time services. Specifically, given a slack time $\tau$, denote the rank of the coefficient matrix of $V_j$ as $r_j'$ after $\tau$. Then, the service ratio for $V_j$ is computed by $r_j' / |D|$. Accordingly, ASR of the system is defined as the summation of each vehicle’s service ratio over the number of vehicles, which is computed by:

$$ASR = \frac{\sum_{j=1}^{|V|} r_j' / |D|}{|V|}$$  \hspace{1cm} (5)

A higher value of ASR indicates a better performance of the algorithm on scheduling real-time data services.

### B. Simulation Results

1) **Effect of System Workload on Delay-Tolerant Services:**

The first set of experiments evaluates the algorithm performance on scheduling delay-tolerant services under different system workloads (Figs. 4~6). The more data items requested by vehicles represents a higher system workload. Specifically, Fig.4 compares the NBP of algorithms. As analyzed, the lower bound of the NBP is computed by $|D| - \min_{V_j \in V} (|C'(V_j)|)$, where $|C'(V_j)|$ is obtained based on the cached packet of $V_j$ in the simulation. We observe from Fig.4 that MA can achieve the
optimal performance in terms of minimizing NBP. In contrast, either MRF or Round-Robin almost reaches the upper bound of the NBP in all cases, which is the number of requested data items. This is because neither MRF nor Round-Robin adopted network coding by considering the cached packets of vehicles, which indicates that a data item has to be broadcast even if only one of the vehicles is waiting for it. Due to the diversity of outstanding requests by different vehicles, most likely, the algorithm has to schedule all the data items for broadcasting to complete the service.

Fig. 5 compares the ASD of algorithms under different system workloads. As shown, the ASD of all the algorithms increases with the increasing of the system workload. Nevertheless, MA always achieves the lowest ASD in all cases. To better comprehend and verify such a superiority of MA, we further examine the result shown in Fig. 6, where the BP of algorithms under different system workloads are compared. As analyzed, the higher value of BP demonstrates the better capability of the algorithm on improving the bandwidth efficiency and the upper bound is 1. Evidently, MA achieves near optimal BP in all scenarios, which is much higher than other algorithms. Also, note that MRF performs better than Round-Robin, because MRF considers data productivity in scheduling by broadcasting the data item with the most pending requests. This observation is consistent with findings in previous work [8].

2) Effect of Data Transmission Rates of RSUs on Delay-Tolerant Services: This set of experiments evaluates the algorithm performance on scheduling delay-tolerant services under different data transmission rates of RSUs (Figs. 7~9).

As stated above, for fair comparison, the same random linear coding strategy is adopted at RSUs for all the algorithms. Nevertheless, the different data transmission rate of RSUs will result in different number of cached packets of vehicles, which may influence algorithm performance at BSs. Fig. 7 compares the NBP of algorithms under different data transmission rates of RSUs. Clearly, when the data transmission rate of RSUs is higher, vehicles are able to cache more packets when they are passing by. According to the collected statistics, the average number of cached packets of each vehicle increases from 9 to 25 with the increasing of the data transmission rate. Intuitively, the BS should be able to broadcast fewer number of packets to complete the service when more packets are cached by vehicles. Nevertheless, we note that the NBP of MRF and Round-Robin almost unchanged when the data rate of RSUs is getting higher. This is because MRF and Round-Robin can barely take the advantage of those cached packets as most of them are in the encoded form. In contrast, MA can always achieve the optimal result by well exploiting the benefit of vehicular caching and network coding.

Fig. 8 compares the ASD of algorithms under different data transmission rates of RSUs. First, the performance of all the algorithms are getting better when the data transmission rate of RSUs is getting higher, which makes sense because the overall system workload is relieving. Second, we note that MA constantly outperforms other algorithms in all the scenarios, especially when the service rate of RSUs is getting higher. It further demonstrates the superiority of MA on exploiting the vehicular cache content in making coding decisions. Further, we analyze the BP of algorithms under different data transmission rates of RSUs.
transmission rates of RSUs in Fig. 9. As observed, MA always manages to achieve the upper bound of BP in different scenarios, while the performance of MRF and Round-Robin is decreasing significantly. This is because when there are more packets received from RSUs, it is more likely that a broadcast data item by MRF and Round-Robin can only serve a few number of vehicles, which results in the decreasing of BP.

3) Effect of Slack Time on Real-Time Services: The following experiments evaluate algorithm performance on scheduling real-time services (Figs. 10∼12). Specifically, Fig.10 evaluates the algorithm performance on scheduling real-time services with different slack time. A shorter slack time gives a more stringent time constraint on data services, beyond which the service will be failed. As expected, all the algorithms have higher ASR when the slack time is getting longer. Also, MA constantly outperforms other algorithms in all ranges.

4) Effect of System Workload on Real-Time Services: Fig. 11 evaluates the algorithm performance on scheduling real-time services under different system workloads. As shown, the ASR of all the algorithms decreases with an increasing number of requested data items, but MA always achieves the highest ASR in all the scenarios, which demonstrates the scalability of the solution.

5) Effect of Data Transmission Rate of RSUs on Real-Time Services: Fig. 12 evaluates the algorithm performance on scheduling real-time services under different data transmission rates of RSUs. As shown, with higher data transmission rates of RSUs, vehicles are able to cache more packets when they are passing by. Accordingly, it is expected that all the algorithms will achieve a higher ASR. Also, we note that MA achieves the best performance across the whole range.

VII. CONCLUSION AND FUTURE WORK

In this work, we present an SDN-based data service architecture in heterogeneous vehicular communication environments, where RSUs, BSs and vehicles are abstract as the data plane, while the scheduling decisions are exercised by the logically centralized control plane. On this basis, we give a formal description of the coding-assisted broadcast scheduling problem, CBS, which aims at maximizing the bandwidth efficiency by incorporating vehicular caching and network coding in making scheduling decisions. We prove that CBS is NP-hard by constructing a polynomial-time reduction from the simultaneous matrix completion problem. We propose a memetic algorithm MA to solve the CBS problem, which consists of a binary vector representation for encoding solutions, a fitness function for solution evaluation, a set of reproduction operators (i.e. parent selection, crossover and mutation) for offspring generation, a local search method for solution enhancement and a repair operator for fixing infeasible solutions. Finally, we build the simulation model and give a comprehensive performance evaluation. The simulation results conclusively demonstrate the effectiveness of the proposed solution under a variety of circumstances.

It would be a meaningful extension of the current work by further considering different data service requirements of different applications, and investigating the tight cooperation between RSUs and BSs in data scheduling under the SDN-based service architecture. In addition, we will further look into the design of efficient routing protocols in such an SDN-based service architecture, where the control plane...
can direct multi-hop communications among the data plane, including BSs, RSUs and vehicles. Finally, the communication impacts at MAC and physical layers are expected to be examined so as to validate the system model in realistic vehicular communication environments.

**References**


Penglin Dai (S’15–M’17) received the B.S. degree in mathematics and applied mathematics and the Ph.D. degree in computer science from Chongqing University, Chongqing, China, in 2012 and 2017, respectively. He is currently an Assistant Professor with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China. His research interests include intelligent transportation systems and vehicular cyber-physical systems.

Victor C. S. Lee (M’92) received the Ph.D. degree in computer science from the City University of Hong Kong in 1997. He is currently an Assistant Professor with the Department of Computer Science, City University of Hong Kong. His research interests include data dissemination in vehicular networks, real-time databases, and performance evaluation. He is a member of the ACM and the IEEE Computer Society. He has been the Chairman of the IEEE, Hong Kong Section, Computer Chapter in 2006–2007.

Sang Hyuk Son (M’85–SM’98–F’13) received the B.S. degree in electronics engineering from Seoul National University, the M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), and the Ph.D. degree in computer science from the University of Maryland, College Park, MD, USA. He has been a Professor with the Department of Computer Science, University of Virginia, and WCU Chair Professor with Sogang University. He has been a Visiting Professor at KAIST, the City University of Hong Kong, Ecole Centrale de Lille in France, Linkoping University, and the University of Skövde, Sweden. He is currently the President of the Daegu Gyeongbuk Institute of Science and Technology. His research has been funded by the Korean Government, National Research Foundation, National Science Foundation, DARPA, Office of Naval Research, Department of Energy, National Security Agency, and IBM. He has authored or co-authored over 340 papers and edited/authored four books in his research areas. His research interests include cyber physical systems, real-time and embedded systems, database and data services, and wireless sensor networks.

Prof. Son is a member of both the Korean Academy of Science & Technology and the National Academy of Engineering of Korea. He is a Founding Member of the ACM/IEEE CPS Week, and serving as a member of the steering committee for the IEEE RTCSA and Cyber Physical Systems Week. He received the Outstanding Contribution Award from the Cyber Physical Systems Week in 2012. He has served on the Editorial Board of the ACM Transactions on Cyber Physical Systems, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and Real-Time Systems Journal.

Jiannong Cao (F’15) received the B.Sc. degree in computer science from Nanjing University, China, in 1982, and the M.Sc. and Ph.D. degrees in computer science from Washington State University, Pullman, WA, USA, in 1986 and 1990, respectively. He is currently a Chair Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. He is also the Director of the Internet and Mobile Computing Laboratory in the Department and the Director of the University’s Research Facility in Big Data Analytics. He has co-authored five books in mobile computing and wireless sensor networks, co-edited nine books, and authored or co-authored over 500 papers in major international journals and conference proceedings. His research interests include parallel and distributed computing, wireless sensing and networks, pervasive and mobile computing, and big data and cloud computing. He served as the Chair of the Technical Committee on Distributed Computing of the IEEE Computer Society 2012–2014.