Network-Coding-Assisted Data Dissemination via Cooperative Vehicle-to-Vehicle-/Infrastructure Communications

Kai Liu, Member, IEEE, Joseph Kee-Yin Ng, Senior Member, IEEE, Junhua Wang, Victor C. S. Lee, Member, IEEE, Weiwei Wu, Member, IEEE, and Sang Hyuk Son, Fellow, IEEE

Abstract—Vehicle-to-vehicle/vehicle-to-infrastructure (referred to as V2X) communications have potential to revolutionize current road transportation systems with respect to vehicle safety, transportation efficiency, and travel experience. This paper puts the first effort on applying network coding in cooperative V2X communication environments to improve bandwidth efficiency and enhance data service performance. Specifically, we investigate new arising challenges on network-coding-assisted data dissemination by considering both communication constraints and application requirements in vehicular networks. We present the system model and give an insight into the characteristics of cooperative data dissemination with network coding. On this basis, we formulate the problem and propose a network-coding-assisted scheduling algorithm to enable the hybrid of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications and exploit their joint effects on providing efficient data services. We design a cache strategy that allows vehicles to retrieve their unrequested data items. This strategy not only increases the opportunity of data sharing among vehicles but also gives higher probability of packet decoding, which in turn enhances the data service performance. We give an intensive analysis on the scheduling overhead, which shows the scalability of the algorithm. Finally, we build the simulation model and conduct a comprehensive performance evaluation to demonstrate the superiority of the proposed solution.

Index Terms—Data dissemination, V2X communication, scheduling algorithm, network coding.

I. INTRODUCTION

R ECENT advances in vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications (together abbreviated as V2X) have motivated the design and development of various emerging intelligent transportation applications, such as collision avoidance [1], roadway reservation [2], and intersection management [3], etc. The dedicated short-range communication (DSRC) [4] is an unprecedented wireless technology intended to support V2X communication in vehicular ad hoc networks (VANETs), where the roadside unit (RSU) is a stationary server installed along the road to provide information services via V2I communication, while the onboard unit (OBU) is mounted on vehicles to enable both V2I and V2V communications.

In the last decade, network coding has aroused significant interest in the wireless communication research and it is deemed as a promising solution to improving bandwidth efficiency and increasing network throughput ([5], [6]). W. Yeung [7] gives a review of network coding from the historical perspective and illustrates how the network coding, originated in information theory, has propagated to a great number of research fields. The analysis in [6] has shown the benefit of applying network coding on improving network throughput for unicast routing. Recently, some studies have exploited network coding to enhance data broadcast performance in mobile computing systems ([8], [9]). A few studies have incorporated network coding into VANETs ([10], [11]), but none of them considered unique characteristics in a typical V2X communication system for cooperative data dissemination.

This work is dedicated to exploring the benefit of network coding and implementing efficient cooperative data dissemination in VANETs. Compared with conventional mobile ad-hoc networks, there are a number of unique challenges when incorporating network coding into VANETs, especially considering the hybrid of V2I and V2V communications. First, to best exploit spatial reusability, vehicles are expected to be well cooperated when V2I and V2V data dissemination are taking place simultaneously. This makes it challenging to make encoding decisions, because both the RSU and vehicles can encode their stored data items to provide services at the same time via V2I and V2V communications, respectively.
Moreover, it is likely to have high complexity and overhead in scheduling due to a variety of requirements and constraints in VANETs, such as the stringent time limit on providing services, the dynamic data access pattern of vehicles, the half-duplex transmission of OBUs (i.e., vehicles cannot transmit and receive data packets at the same time) and the single radio communication of OBUs (i.e., a vehicle can only switch to either V2I or V2V communication mode at a time). Last but not least, it is non-trivial to design an efficient network coding assisted data dissemination solution when considering the interrelationship between the requested and cached data items among different vehicles.

To the best of our knowledge, this is the first work dedicated to addressing the aforementioned issues by incorporating network coding into cooperative data dissemination via the hybrid of V2I and V2V communications. The main contributions of this work are outlined as follows. First, we present the implementation of the cooperative data dissemination system in VANETs and analyze the system characteristics. Second, we propose a network coding assisted scheduling algorithm for data dissemination via V2X communication. Also, we design a cache strategy to further enhance the effect of network coding. In addition, we give an insight into the scheduling overhead and demonstrate the feasibility and scalability of the algorithm. Last, we build the simulation model and conduct a comprehensive performance evaluation, which shows the superiority of the algorithm.

The rest of this paper is organized as follows. Section II reviews the related work. Section III presents the cooperative data dissemination system. In Section IV, we propose a network coding assisted scheduling algorithm. In Section V, we build the simulation model and give performance evaluation. Finally, Section VI concludes this work and discusses future research directions.

II. RELATED WORK

Current studies on vehicular communications largely focused on enhancing communication quality and reliability at the MAC layer. F. Farnoud and S. Valaee [12] proposed a MAC protocol for delivering safety-critical messages in VANETs. It adopts positive orthogonal codes to implement topology-transparent broadcast and improve the broadcast reliability in highly dynamic vehicular environments. Y. Fallah et al. [13] investigated data dissemination in cooperative vehicle safety systems (CVSS). Two primary controllable parameters including the transmission rate and the transmission range are intensively analyzed, which guide the design of a robust range control scheme in VANETs. J. Zhang et al. [14] proposed a Vehicular Cooperative MAC (VC-MAC) protocol, which exploits the benefit of V2V communication to maximize the total throughput by leveraging V2V data sharing for serving vehicles which have left the RSU’s service region. Y. Bi et al. [15] proposed a Multi-Channel Token Ring MAC Protocol (MCTRP) for V2V communication. The asynchronous carrier sense multiple access with collision avoidance (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 providing non-safety multimedia applications. The above work form a solid foundation on improving wireless communication qualities in vehicular networks. Nevertheless, none of them focused on the data scheduling for providing information services by considering particular communication constraints and application requirements in VANETs.

Several studies considered the design of scheduling algorithms for data dissemination in VANETs. C. Chang et al. [16] aimed to minimize the handoff overhead for I2V data dissemination due to the incompleteness of file transmission in one RSU’s coverage. A scheduling algorithm called Maximum Freedom Last (MFL) is proposed, which incorporates several critical factors in scheduling, including remaining service dwell time, remaining data transmission time, queuing delay and maximum tolerable delay. J. Zhao et al. [17] proposed a data pouring and buffering paradigm, which can significantly improve data delivery ratio by making use of relay and broadcast stations to offload data dissemination workload at data centers. K. Liu et al. [18] investigated on real-time data services by considering both the time constraint of data dissemination and the freshness of data items, where a temporal data dissemination problem is formulated and a heuristic scheduling algorithm is proposed to enhance the system performance. The study in [19] considered the scalability, fairness and robustness of data dissemination in VANETs and a scheduling algorithm is proposed to best exploit the joint effects of V2I and V2V communications on data services. Unlike the above studies, this work investigates how to apply network coding into V2X-based cooperative data dissemination with the purpose of further improving the bandwidth efficiency and enhancing the data service performance.

A number of studies have proposed to apply network coding into mobile broadcast systems [8], [9], [20], but none of them is specifically designed for VANETs. A few work has incorporated network coding into VANETs. M. Li et al. [10] investigated popular content distribution via push-based broadcast in vehicular networks and proposed a cooperative content distribution protocol called CodeOn. The symbol level network coding (SLNC) is adopted to facilitate high speed content downloading in VANETs and counteract the packet loss issues. M. Firooz and S. Roy [11] 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. Distinguishing from the above efforts, this work focuses on designing a network coding assisted scheduling algorithm for data dissemination, which aims at maximizing the joint effect of V2I and V2V communications in providing data services.

III. COOPERATIVE DATA DISSEMINATION SYSTEM

The V2X-based cooperative data dissemination system is shown in Fig. 1. The RSU is installed along the road and provides services to passing vehicles via V2I communication. Meanwhile, vehicles are able to share their cached data items to their neighbors via V2V communication. In accordance with IEEE 1609.4 [21], we consider a multi-channel operation
environment, in which one control channel and two service channels are involved. The control channel is used for disseminating management information, service advertisements and control messages. Vehicles will be informed with available services via the control channel so that they may submit requests to the RSU for corresponding services. Based on the pending requests, the RSU schedules corresponding data items (or encoded packets) to broadcast via V2I and V2V communications. In this work, the bitwise exclusive-or ($\oplus$) coding operation is adopted due to its trivial implementation overhead. For example, given an encoded packet $p = d_1 \oplus d_2$, to decode an original data item (say $d_1$) from $p$, it requires the rest data items in $p$ (say $d_2$) by computing $d_1 = p \oplus d_2$.

The two service channels are used for disseminating data items (or encoded packets) via V2I and V2V communications, respectively [22]. We consider commonly adopted single radio OBUs, with which vehicles can tune in to only one of the channels at a time [23]. The time unit adopted in this model refers to a scheduling period, which consists of three phases. The start of each phase is controlled by the RSU and vehicles will be informed via the control channel.

- In the first phase, all the vehicles are set to the V2V mode and broadcast their heartbeat messages (i.e. the Basic Safety Message as defined in SAE J2735 [24]), so that each vehicle is able to identify a list of its neighboring vehicles (e.g., by measuring the signal-to-noise ratio through received heartbeat messages from other vehicles to recognize a set of vehicles, with which it can transmit and receive data items).

- In the second phase, all the vehicles switch to the V2I mode and report their updated information to the RSU, including the set of their current neighbors and the identifiers of the cached and requested data items. This information is piggybacked into the Probe Vehicle Message as defined in SAE J2735. Each request corresponds to one data item, and it is satisfied when the corresponding data item is retrieved by the vehicle. Outstanding requests are pending in the service queue at the RSU. The scheduling decisions are announced via the control channel (i.e. piggybacked into the WAVE service advertisement [25]) so that each vehicle within the RSU’s coverage can be informed.

- In the third phase, each vehicle switches to either V2I or V2V mode and starts data transmitting/receiving based on the received scheduling decision from the RSU. Note that multiple instances of data dissemination may take place simultaneously in this phase. Specifically, some vehicles may be instructed to retrieve data item from the RSU via V2I communication, while some others may be instructed to share their cached data items via V2V communication. This work considers only one-hop V2V data dissemination among vehicles driving along the same direction, and the vehicles are assumed to stay in the same neighborhood for a short period of time (e.g., during a scheduling period) [14].

Fig. 1 shows a toy example to better illustrate scheduling concerns in such a system. Each vehicle may have two sets of data items, namely, cached data items (in set $C$) and requested data items (in set $R$). The black dot on top of a vehicle indicates that this vehicle is tuning in to the V2I mode, while the white dot indicates the V2V mode. The double arrow edge represents the neighborhood relationship between two vehicles. Note that only vehicles tuning in to the V2V mode are considered when constructing the neighborhood relationship. The dotted circle represents the RSU’s service range.

At time $t$, a possible scheduling decision is shown in Fig. 1(a). Three vehicles are instructed to tune in to the V2I mode (i.e., $V_2$, $V_3$ and $V_7$). The encoded packet $a \oplus c$ is scheduled to broadcast from the RSU. Accordingly, $V_2$ and $V_3$ can decode this packet because they have cached $c$ and $a$, respectively. In conventional data broadcast systems, only those requests asking for the same data item can be served in one broadcast tick ([18], [26]). In contrast, with network coding, we note that even though $V_2$ and $V_3$ request for different data items (i.e., $a$ and $c$), both of them can be served by broadcasting $a \oplus c$ in one broadcast tick. This example indicates the potentiality on improving bandwidth efficiency using network coding. Besides, it is worth noting that, although $V_7$ does not request for $c$, it can overhear the broadcast packet and decode $c$ for free (as it has already cached $a$). To be discussed in Fig. 1(b), it is
preferable to let vehicles cache their unrequested data items so as to facilitate future scheduling.

At time $t$, four vehicles (i.e., $V_1$, $V_2$, $V_5$ and $V_6$) are instructed to tune in to the V2V mode. In particular, $V_4$ is scheduled to be a sender and broadcast $d$ to its neighbors (the single arrow edge represents the data transmission). In this case, $V_3$ is served by retrieving $d$ from $V_4$. On the other hand, although $V_5$ and $V_6$ do not request for $d$, they can overhear it and cache $d$ for the time being. Note that comparing with overhearing $d$ from $V_4$, one may expect to schedule $V_5$ as a sender at the same time, so that it can disseminate $b$ to serve $V_6$. Unfortunately, this is not a viable solution because $V_6$ is the neighbor of both $V_4$ and $V_5$. Due to the broadcast nature in wireless communication, simultaneously disseminating data items from the vehicles which are in the immediate or adjacent neighborhoods may lead to the transmission collision ([(14), (22)]). Therefore, $V_6$ cannot be served when $V_4$ and $V_5$ are transmitting data items at the same time.

After $\Delta t$, as shown in Fig. 1(b), suppose $V_1$, $V_2$ and $V_4$ have left the RSU’s service range and there are three new coming vehicles (i.e., $V_6$, $V_9$ and $V_10$). In this scheduling period, $V_2$ and $V_{10}$ are instructed to tune in to the V2I mode and retrieve $f$ from the RSU. Meanwhile, other vehicles switch to the V2V mode. Specifically, $V_7$ is designated as a sender to disseminate $c$, targeting at serving $V_9$. Recall that $V_7$ does not request for $c$ but just overheard it at time $t$. Therefore, caching unrequested data items may increase the chance for V2V data sharing. In addition, $V_5$ is designated as a sender to disseminate an encoded packet $b \oplus d$. In this case, since $V_5$ has cached $d$ and $V_9$ has cached $b$, both of them can decode $b \oplus d$ and retrieve their requested data items (i.e., $b$ and $d$ for $V_5$ and $V_9$, respectively). Recall that $V_9$ overheard $d$ from $V_4$ at time $t$, which helps for the current decoding. Therefore, caching unrequested data items may also give better chance to decode. Last, note that although $V_5$ and $V_7$ are neighbors, they cannot retrieve data items from each other while they are broadcasting at the same time due to the half-duplex transmission of OBU.

The above example illustrates potential benefits of applying network coding into V2X-based cooperative data dissemination. Meanwhile, it also reveals that it is non-trivial to design an efficient scheduling algorithm when considering both V2X communication constraints and data service requirements.

IV. PROPOSED SOLUTION

A. Preliminaries

The primary notations are summarized in Table I. The database $D = \{d_1, d_2, \ldots, d_D\}$ consists of $|D|$ data items. The set of vehicles is denoted by $V(t) = \{V_1, V_2, \ldots, V_{|V(t)|}\}$, where $|V(t)|$ is the total number of vehicles at time $t$. Depending on the communication mode of vehicles, $V(t)$ is divided into two sets: $V_I(t)$ and $V_V(t)$, which represent the set of vehicles in the V2I and V2V mode, respectively. Each vehicle stays in one of the modes at a time, namely, $V_I(t) \cap V_V(t) = \emptyset$ and $V_V(t) \cup V_I(t) = V(t)$.

Each $V_i(1 \leq i \leq |V(t)|)$ has a set of requests, which is denoted by $Q_i(t) = \{q_{i1}^V, q_{i2}^V, \ldots, q_{i|Q_i(t)|}^V\}$, where $|Q_i(t)|$ is the total number of requests submitted by $V_i$. Each $q_{ij}^V (1 \leq j \leq |Q_i(t)|)$ corresponds to one data item in the database. Once the data item is retrieved by $V_i$, $q_{ij}^V$ is satisfied. For each $V_i$, its requests have to be served before leaving the RSU’s service region.

Each $V_i(1 \leq i \leq |V(t)|)$ maintains a cache with size $|C|$ and the set of cached data items is denoted by $C_i(t)$. For each $V_i$ in the V2V mode, the set of its neighboring vehicles is denoted by $N_i(t)$, where $N_i(t) \subseteq V_i(t)$. The RSU maintains an entry in the service queue for each $V_i$, which is characterized by a 3-tuple: $<V_i, Q_i(t), N_i(t)>$. The values of $Q_i(t)$ and $N_i(t)$ are updated in every scheduling period.

A packet $p$ with $k$ encoded data items is denoted by $p = d_1^k \oplus d_2^k \oplus \ldots \oplus d_l^k$, where $d_j^k (1 \leq j \leq k)$ is the $j$th encoded data item and $d_l^k \in D$. Without loss of generality, we consider that an individual data item is a special encoded packet where $k = 1$.

At time $t$, the encoded packet broadcast from the RSU is denoted by $p_{rsu}(t)$. In V2V communication, we denote the current set of sender vehicles as $SV(t) = \{SV_1, SV_2, \ldots, SV_{|SV(t)|}\}$, where $|SV(t)|$ is the number of designated sender vehicles. The encoded packet disseminated by $SV_i(1 \leq i \leq |SV(t)|)$ is denoted by $p_{sv}(t)$. All the sender vehicles are in the V2V mode, namely, $SV(t) \subseteq V_V(t)$. Each sender vehicle can only disseminate one encoded packet at a time. Accordingly, the set of disseminated encoded packets in V2V communication is denoted by $p_{sv}(t) = \{p_{sv_1}, p_{sv_2}, \ldots, p_{sv_{|SV(t)|}}\}$. Note that all the encoded data items in $p_{sv}(t)$ have to be cached by $SV$, namely, $d_j^k \in C_{sv}(t)$, $\forall d_j^k \in p_{sv}(t)$.

In V2I communication, denote $SV_{rsu}(t)$ as the set of vehicles within the RSU’s coverage. When $p_{rsu}(t)$ is broadcast, the set of receiver vehicles is denoted by $RV_{rsu}(t)$. Each receiver vehicle $V_i$ (i.e., $V_i \in RV_{rsu}(t)$) has to satisfy the following conditions:

- $V_i$ is in the V2I mode. That is, $V_i \in V_I(t)$.
• $V_i$ is in the RSU’s coverage. That is, $V_i \in V_{\text{rsu}}(t)$.
• $V_i$ is able to decode $p_{\text{rsu}}(t)$. Specifically, suppose there are $k$ data items encoded in $p_{\text{rsu}}(t)$. Then, it requires that $V_i$ has to cache $k - 1$ out of the $k$ data items. That is, $p_{\text{rsu}}(t) - (C_{V_i}(t) \cap p_{\text{rsu}}(t)) = 1$.

Note that some vehicles in $RV_{\text{rsu}}(t)$ may not request any data item encoded in $p_{\text{rsu}}(t)$, but they can decode $p_{\text{rsu}}(t)$ to cache an unrequested data item. Therefore, among $RV_{\text{rsu}}(t)$, we denote the set of actually served vehicles as $RV_{p_{\text{rsu}}}(t)^*$, where $RV_{p_{\text{rsu}}}(t)^* \subseteq RV_{\text{rsu}}(t)$.

In V2V communication, due to the broadcast effect, simultaneous data dissemination from multiple sender vehicles may cause transmission collisions. Specifically, for any $V_i$ in the V2V mode, if $V_k$ is in the neighborhood of both $SV_i$ and $SV_j$ ($SV_i, SV_j \in SV(t)$), then the collision happens at $V_k$. That is, the set of vehicles $\{V_k|V_k \in V_i(t) \land V_k \in N_{SV_i}(t) \land V_k \in N_{SV_j}(t)\}$ will suffer transmission collision, and thus they cannot be the receiver at time $t$. In addition, a receiver vehicle cannot simultaneously be a sender due to the half-duplex transmission of OBU’s. Given the sender vehicle $SV_i$ with its encoded packet $p_{SV_i}(t)$, we denote $RV_{p_{SV_i}}(t)$ as the set of corresponding receiver vehicles. According to the above constraints, each receiver vehicle $V_j$ (i.e., $\forall V_j \in RV_{p_{SV_i}}(t)$) has to satisfy the following conditions.

• $V_j$ is in the V2V mode. That is, $V_j \in V_V(t)$.
• $V_j$ is in the neighborhood of $SV_i$. That is, $V_j \in N_{SV_i}(t)$.
• $V_j$ is not in the sender vehicle set. That is, $V_j \notin SV(t)$.
• $V_j$ is not the neighbor of any other sender vehicles excepting $SV_i$. That is, $V_j \notin N_{SV_k}(t) \land V_k \in \{SV(t) - SV_i\}$.
• $V_j$ is able to decode $p_{SV_i}(t)$. Specifically, suppose there are $k$ data items encoded in $p_{SV_i}(t)$. Then, it requires that $V_j$ has to cache $k - 1$ out of the $k$ data items. That is, $p_{SV_i}(t) - (C_{V_j}(t) \cap p_{SV_i}(t)) = 1$.

Given the set of sender vehicles $SV(t)$ with the corresponding set of encoded packets $p_{SV_i}(t)$, we denote $RV_{p_{SV_i}}(t)$ as the set of receiver vehicles. It is the union of receiver vehicle sets for each encoded packet $p_{SV_i}(t)$ (i.e., $\forall p_{SV_i}(t) \in p_{SV}(t)$):

$$RV_{p_{SV_i}}(t) = \bigcup_{p_{SV_i}(t) \in p_{SV}(t)} RV_{p_{SV_i}}(t). \quad (1)$$

Similar with V2I scenarios, some vehicles in $RV_{p_{SV_i}}(t)$ may just overhear a packet and decode an unrequested data item via V2V communication. Therefore, among $RV_{p_{SV_i}}(t)$, we denote the set of actually served vehicles as $RV_{p_{SV_i}}(t)^*$, where $RV_{p_{SV_i}}(t)^* \subseteq RV_{p_{SV_i}}(t)$.

With the above analysis, the proposed scheduling algorithm aims to maximize the number of served vehicles by making the following decisions. First, it divides vehicles into V2I and V2V sets, namely, $V_I(t)$ and $V_V(t)$. Second, it selects a set of data items from $D$ and encodes them into a packet $p_{\text{rsu}}(t)$. Third, it designates a set of sender vehicles ($SV(t)$) together with the corresponding set of encoded packets ($p_{SV}(t)$). Considering $N$ scheduling periods, the objective is to maximize the total number of served vehicles via V2X communication. To sum up, the scheduling problem is to determine $V_I(t)$, $V_V(t)$, $p_{\text{rsu}}(t)$, $SV(t)$ and $p_{SV}(t)$ in each scheduling period so as to

$$\max \sum_{t=1}^{N} (|RV_{p_{\text{rsu}}}(t)^*| + |RV_{p_{SV_i}}(t)^*|). \quad (2)$$

B. Network Coding Assisted Scheduling

We propose a network coding assisted scheduling (NCAS) algorithm for cooperative data dissemination via V2X communication. Relevant concepts are introduced as follows to facilitate the algorithm design.

1) Tentative Schedule (TS): A tentative schedule (TS) refers to a service operation of packet dissemination (via either V2I or V2V communication) such that a request can be served. Note that a single data item is considered as a special case of an encoded packet. The TS is classified into two sets: $TS_{V2I}(t)$ and $TS_{V2V}(t)$, which represent V2I and V2V tentative schedules, respectively. Moreover, a TS $TS \in TS_{V2I}(t)$ can be considered as the service operation of $RIPv_r$, where $R$ represents the RSU. $\hat{p}$ represents the encoded packet broadcast by the RSU, and $V_r$ represents the receiver vehicle for $\hat{p}$. Note that $V_r$ has to be able to decode a data item from $\hat{p}$. On the other hand, a TS $TS \in TS_{V2V}(t)$ can be considered as the service operation of $V_s \hat{p}V_r$, where $V_s$ represents the sender vehicle. $\hat{p}$ represents the encoded packet broadcast by $V_s$, and $V_r$ represents the receiver vehicle for $\hat{p}$. Note that all the encoded data items in $\hat{p}$ have to be cached by $V_s$, and $V_r$ has to be able to decode a data item from $\hat{p}$. A valid TS must have potential to serve a request. For instance, as shown in Fig. 1(a), $R(a \oplus c)V_2$ is a TS ($TS \in TS_{V2I}(t)$) because $V_2$ is able to decode $a \oplus c$ and retrieve $a$. In contrast, $R(a \oplus c)V_5$ is not a TS because $V_5$ cannot decode $a \oplus c$, and there is no potential service by this operation.

2) Conflicting TSs: Due to communication constraints in VANETs, different TSs may be in conflict with each other when operating simultaneously. There are five rules to find conflicting TSs, which are described as follows.

• Rule 1: If the two TSs are both in $TS_{V2I}(t)$, but they have different encoded packets to broadcast, then they are in conflict with each other, because the RSU can only broadcast one encoded packet at a time.
• Rule 2: If the two TSs are both in $TS_{V2V}(t)$, but they designate the same sender vehicle to disseminate different encoded packets, then they are in conflict with each other, because one sender vehicle can only disseminate one encoded packet at a time.
• Rule 3: If the two TSs are both in $TS_{V2V}(t)$, where one TS designates a vehicle as the sender, while the other TS designates the same vehicle as the receiver, then they are in conflict with each other, because a vehicle cannot be both the sender and the receiver at the same time.
• Rule 4: If the two TSs are both in $TS_{V2V}(t)$, but a receiver is the neighbor of both the senders, then they are in conflict with each other, because the transmission collision will happen at the receiver.
• Rule 5: If one TS is in $TS_{V2I}(t)$ and the other TS is in $TS_{V2V}(t)$, but the receiver vehicle in V2I communication...
is a sender/receiver vehicle in V2V communication, then
they are in conflict with each other, because a vehicle
can be only in either V2V or V2I mode at a time.

The formal description of the above five rules are summa-
rized in Table II.

3) Scheduling Procedure: In each scheduling period, the
RSU needs to determine the broadcast packet (i.e., \( p_{rsu}(t) \)),
the sender vehicle set (i.e., \( SV(t) \)) and the corresponding set of
encoded packets to be disseminated via V2V communication
(i.e., \( \{p(t)\} \)). The algorithm is executed at the RSU, while
vehicles do not need to maintain any control information.

- **Step 1** Construct an undirected graph \( G \) based on
  the current requested and cached data items of each vehicle
  as well as the neighborhood relationship of vehicles: The
  algorithm first finds all the TSs, and then it creates a
  corresponding vertex \( v \) for each TS. For each pair of
  conflicting TSs, it finds the two corresponding vertices
  (e.g., \( v_i \) and \( v_j \)) and adds an edge between \( v_i \) and \( v_j \),
  so that the graph \( G \) is constructed.

- **Step 2** Select a subset of TSs which are not conflicting with
each other: According to the constructed \( G \), the set of
  non-conflicting TSs corresponds to an independent set of
  vertices in \( G \) (i.e. a set of vertices which share no edges).
  To maximize the performance, it is expected to find the
  maximum independent set (MIS) of \( G \). Due to the NP-
  hardness of the MIS problem [27], the algorithm adopts a
  greedy method proposed in [28] to approximately solve
  the MIS problem. The basic idea of the greedy method
  is recapitulated as follows. First, it computes the value of
  \( 1/((d(v) + 1) \) for each vertex \( v \) in \( G \), where \( d(v) \) repre-
sents the degree of \( v \). Second, it selects the vertex with
  the maximum value of \( 1/((d(v) + 1) \). Third, it updates \( G \)
  by removing the selected vertex and its adjacent vertices.
  Fourth, it repeats the above operations until there is no
  vertex remaining in \( G \).

- **Step 3** Determine the cooperative data dissemination
  scheme by parsing each selected TS: Specifically, \( p_{rsu}(t) \)
  is set to the \( p \) in any selected \( RpV_r \). Note that \( p \) will be
  the same in all the selected TS from \( TSV_{21}(t) \) because of
  Rule 1 in avoiding conflicting TSs. The sender vehicle set
  in V2V (i.e., \( SV(t) \)) is set to the union of \( V_s \) from each
  selected \( V_sRpV_r \) in \( TSV_{2V}(t) \). Accordingly, the union of \( p \)
  from each selected \( V_sRpV_r \) forms the set of encoded packet
  \( p_{rsu}(t) \) to be disseminated via V2V communication.

4) **Caching Unrequested Data**: The above schedule only
considers to serve vehicles with outstanding requests. In-
deferred, after determining the receiver and the sender vehi-
cles in V2X communication (i.e. \( RV_{p_{rsu}}(t), SV(t) \)
and \( RV_{p_{rsu}}(t) \)), some other vehicles (i.e., \( V(t) - RV_{p_{rsu}}(t) \)
- \( SV(t) - RV_{p_{rsu}}(t) \)) may have chance to cache their unrequested
data items by overhearing a broadcast packet via either V2I
or V2V communication. We denote them as free vehicle set
\( FV(t) \) at time \( t \). Clearly, it is desirable to exploit the broadcast
effect and schedule free vehicles to retrieve their unrequested
data items. This will facilitate subsequent scheduling due to
the following reasons. First, the cached data item may be
requested by the vehicle later. In this case, it will be a local
hit of service, which not only saves the broadcast bandwidth,
but also significantly reduces the service time. Second, caching
unrequested data items may also increase the opportunity of
sharing data among neighboring vehicles, which makes it pos-
sible to better exploit spatial reusability for data services via
V2V communication. Third, considering the requirement of
coding, more data items in the cache will give the vehicle
higher opportunity to decode a packet, which in turn enhances
its service chance. With the above observations, we design the
following strategy for caching unrequested data items based on
the scheduling decisions made in Section IV-B3.

- If a vehicle \( V_f \in FV(t) \) is a neighbor of only one sender
  vehicle \( V_i \) and it is able to decode the packet sent by
  \( V_i \) (i.e., \( p_{V_i}(t) \)), then \( V_f \) will be set to V2V mode and cache
  the decoded data item from \( p_{V_i}(t) \). The condition for the
  free vehicle set in V2V communication \( FV_{V2V}(t) \) is
  represented as follows:

\[
FV_{V2V}(t) = \{V_f | V_f \in SV(t) \land \forall V_j \in SV(t) \land \forall V_j \neq V_f \land \forall p_{V_f}(t) \land (C_{V_f}(t) \cap p_{V_i}(t)) = 0 \}
\]

- For the rest free vehicles (i.e. \( FV(t) - FV_{V2V}(t) \)), if
  a \( V_f \) is able to decode the packet sent from the RSU
  (i.e. \( p_{rsu}(t) \)), then it is set to the V2I mode and cache
  the decoded data item from \( p_{rsu}(t) \). The condition for the
  free vehicle set in V2I communication \( FV_{V2I}(t) \) is
  represented as follows:

\[
FV_{V2I}(t) = \{V_f | V_f \in FV(t) - FV_{V2V}(t) \land (p_{rsu}(t) \land (C_{V_f}(t) \cap p_{rsu}(t))) = 1 \}
\]
broadcast packet in a scheduling period. In addition, we consider a limited cache size of vehicles and adopt first-in-first-out policy for cache replacement.

C. An Example

We give an example to illustrate the scheduling of NCAS based on the two scenarios shown in Fig. 1. In Step 1, NCAS constructs the graph $G$ by searching all the TSs and determining their conflicting relationships. The corresponding graphs at time $t$ and $t+\Delta t$ are shown in Fig. 2(a) and (b), respectively. Each node in $G$ corresponds to a TS. For example, the node $R(a \oplus c)V_3$ in Fig. 2(a) refers to the TS that RSU broadcasts the encoded packet $a \oplus c$ to serve $V_3$ at time $t$. Each edge in $G$ represents a pair of conflicting TSs. As there are five rules used for finding conflicting TSs, for clear exposition, Fig. 2 depicts the edges with five different formats and colors to distinguish different rules adopted.

In Step 2, NCAS selects a subset of non-conflicting TSs by finding an independent set of $G$. In Fig. 2(a), the three grey vertices form an independent set and they are selected to make scheduling decisions. Note that in Fig. 2, to clearly exhibit the independent set, only the edges connected to the selected vertices are depicted. For example, the edge between $V_4dV_1$ and $V_5bV_6$ represents that these two TSs are in conflict with each other due to rule 4, and hence they cannot be selected at the same time.

In Step 3, NCAS makes scheduling decisions based on the three selected TSs, namely, $R(a \oplus c)V_2$, $R(a \oplus c)V_5$ and $V_4dV_1$. Specifically, the following outputs are generated: $p_{rsu}(t) \leftarrow a \oplus c$, $RV_{p_{rsu}}(t) \leftarrow \{V_2, V_3\}$, $SV(t) \leftarrow \{V_4\}$, $p_{sv}(t) \leftarrow \{d\}$ and $RV_{p_{sv}}(t) \leftarrow \{V_1\}$.

Based on the above scheduling decisions, the free vehicle set will be set as $FV(t) \leftarrow \{V_5, V_6, V_7\}$. According to the policy presented in Section IV-B4, $V_5$ and $V_6$ are scheduled to cache $d$ via V2V communication, because they only have one neighboring sender vehicle (i.e., $V_4$). Finally, $V_7$ is scheduled as a free vehicle in V2I communication, because it does not have any neighboring sender vehicles and it is able to decode $a \oplus c$ from the RSU to retrieve $c$.

Similarly, Fig. 2(b) shows the constructed graph $G$ based on the scenario in Fig. 1(b). The five grey vertices are selected, including $RFV_{10}$, $RFV_3$, $V_7cV_6$, $V_5(b \oplus d)V_8$ and $V_6(b \oplus d)V_6$, which form an independent set. With these selected TSs, NCAS generates the following outputs: $p_{rsu}(t+\Delta t) \leftarrow f$, $RV_{p_{rsu}}(t+\Delta t) \leftarrow \{V_3, V_{10}\}$, $SV(t+\Delta t) \leftarrow \{V_5, V_7\}$, $p_{sv}(t+\Delta t) \leftarrow \{b \oplus d, c\}$ and $RV_{p_{sv}}(t+\Delta t) \leftarrow \{V_6, V_8, V_9\}$. With such a schedule, all the vehicles are participated into either V2I or V2V communication. Therefore, there is no free vehicle in this scheduling period.

D. Overhead Analysis

In this section, we analyze the algorithm complexity and discuss how to reduce the scheduling overhead while still maintaining decent performance.

Due to the incorporation of network coding, the operation of searching valid TSs dominates the scheduling overhead. The upper bound of searching TSs is analyzed as follows. Given $|D|$ data items in the database, $|V|$ vehicles and the cache size of $|C|$ for each vehicle, the number of encoding options at the RSU is bounded by $C_i^{|D|} + C_{i+1}^{|D|} + \cdots + C_{|C|}^{|D|} = 2^{|D|}$, where $C_i^{|D|}$ $(1 \leq i \leq |D|)$ represents the number of options of selecting $i$ data items from the database for encoding (i.e., the coding length is $i$). Nevertheless, according to the decoding rule, the vehicle cannot decode a packet with length greater than its cache size. In other words, given a packet with length $L$ (i.e. it encodes $L$ data items), there is no chance to decode the packet if $L-1 > |C|$, because it requires $L-1$ cached data items for decoding. With this observation, the actual upper bound of encoding options at the RSU is $C_1^{|D|} + C_2^{|D|} + \cdots + C_{|C|}^{|D|}$. Since we have $|C| \ll |D|$ in practice, the number of encoding options at the RSU will not be the hurdle of algorithm scalability. In addition, for each encoding option, it requires to traverse the cache of every vehicle to determine whether there is a valid TS. Therefore, denote Overhead$_{TS|V2I}$ as the upper bound of searching TSs in V2I, it is:

$$\text{Overhead}_{TS|V2I} \leq \left(C_1^{|D|} + C_2^{|D|} + \cdots + C_{|C|}^{|D|}\right) \cdot |V| \cdot |C|.$$  

(5)

In V2V communication, since each vehicle can at most cache $|C|$ data items, the number of encoding options for a vehicle is bounded by $C_1^{|C|} + C_2^{|C|} + \cdots + C_{|C|}^{|C|} = 2^{|C|}$, where $C_j^{|C|}$ $(1 \leq j \leq |C|)$ represents the number of options of selecting $j$ data
items for encoding. Therefore, denote Overhead\(_{TS_{V2V}}\) as the upper bound of searching TS\(_{V2V}\), it is:

\[
\text{Overhead}_{\text{TS}_{V2V}} \leq 2|C| \cdot |V| \cdot |C|. \tag{6}
\]

To sum up, the upper bound for searching TSs in V2X (denoted by Overhead\(_{\text{TS}_{V2V}}\)) is the sum of Overhead\(_{\text{TS}_{V2V}}\) and Overhead\(_{\text{TS}_{V2V'}}\), which is:

\[
\text{Overhead}_{\text{TS}_{V2V}} \leq |V| \cdot |C| \cdot \left(2|C| + C^3_D + C^2_D + \cdots + C^{|C|}_D\right). \tag{7}
\]

Note that in this system model, the number of vehicles \(|V|\) is constrained by the RSU’s coverage. Even in urban areas with high traffic density (e.g. 100 vehicles/km), due to the limited RSU’s coverage (e.g. with the communication radius of 600 m), \(|V|\) is still in the order of 10^2. So, based on Eq. (7), the overall scheduling overhead is reasonable.

To further enhance the system scalability, we observe from Eq. (5) and Eq. (6) that the overhead of searching TSs is dominated by the maximum allowed encoding length of a packet. In view of this, we propose to set a threshold for the encoding length, and demonstrate that it can significantly reduce the scheduling overhead while still maintaining decent algorithm performance. To start with, we derive the following theorem, from which we can observe that the chance would be slim to decode a packet with long encoding length.

Theorem 1: Given an encoded packet \(E = e_1 \oplus e_2 \ldots \oplus e_L\) with length \(L\), where \(e_i \in D\) and \(1 \leq i \leq L\), the probability of decoding \(E\) (denoted by \(P(E)\)) satisfies:

\[
P(E) \leq 1 - \prod_{k=1}^{L} \left(1 - \prod_{m=1}^{L-1} \left(1 - (1 - p_{\text{access}}(e^k_m))^{|V|}\right)\right)
\]

where \(a^k_m \in E\) and \(p_{\text{access}}(a^k_m)\) is the data access probability of \(a^k_m\).

Proof: Considering an ideal case, as long as \(e_i\) is requested by any vehicle, it will be scheduled to broadcast and all the other vehicles will cache \(e_i\) at the same time. Given \(|V|\) vehicles, the probability that none of them requests \(e_i\) is \((1 - p_{\text{access}}(e_i))^{|V|}\). Accordingly, the probability of caching \(e_i\) (denoted by \(p_{\text{cache}}(e_i)\)) is bounded by \(1 - (1 - p_{\text{access}}(e_i))^{|V|}\):

\[
p_{\text{cache}}(e_i) \leq 1 - (1 - p_{\text{access}}(e_i))^{|V|}. \tag{8}
\]

To decode \(E\), \(L - 1\) out of the \(L\) encoded data items have to be cached. Denote \(a^1_k, a^2_k, \ldots, a^{L-1}_k\) as the \(k\)th combination of choosing \(L - 1\) data items from the set \(\{e_1, e_2, \ldots, e_L\}\). The probability of caching \(a^1_k, a^2_k, \ldots, a^{L-1}_k\) is \(\prod_{m=1}^{L-1} p_{\text{cache}}(a^k_m)\), and hence the probability that the set of data items corresponding to the \(k\)th combination are not in the cache is computed by \(1 - \prod_{m=1}^{L-1} p_{\text{cache}}(a^k_m)\). Since there are \(C^L_{L-1} = L\) combinations, the probability that none of the combination of data items in the cache is computed by \(\prod_{k=1}^{L} \left(1 - \prod_{m=1}^{L-1} p_{\text{cache}}(a^k_m)\right)\).

As \(E\) can be decoded if any combination exists, the probability of decoding \(E\) is computed by:

\[
P(E) = 1 - \prod_{k=1}^{L} \left(1 - \prod_{m=1}^{L-1} p_{\text{cache}}(a^k_m)\right). \tag{9}
\]

Based on Eq. (8) and Eq. (9), the theorem is proved.

Suppose \(|D| = 100\), \(|V| = 30\), \(L = 3\) and the data access pattern follows the Uniform distribution, namely \(p_{\text{access}}(d_i) = 1/|D|, \forall d_i \in D\). Based on Eq. (8), we get \(p_{\text{cache}}(d_i) \leq 0.26, \forall d_i \in D\). Therefore, for any packet \(E\) with length \(L = 3\), according to Theorem 1, we have \(P(E) \leq 0.189\). In fact, this upper bound is derived based on a strong precondition. That is, as long as a data item is requested by any vehicle, it will be cached by all the other vehicles. In practice, the probability \(p_{\text{cache}}(e_i)\) could be much lower than the upper bound derived in Eq. (8). To sum up, we observe from this example that the chance could be very slim to decode a packet with long coding length.

With the above analysis, it is reasonable to set a threshold of coding length \(L_{\text{max}}\) such that the algorithm only considers the packet with length \(L \leq L_{\text{max}}\) when searching TSs. The above theorem guarantees that a properly selected \(L_{\text{max}}\) can significantly reduce the scheduling overhead while maintaining the algorithm performance.

V. PERFORMANCE EVALUATION

A. Setup

The simulation model is built based on the system architecture described in Section III, and it is implemented by CSIM19 [29]. Specifically, CSIM19 is a library of routines, which provides interfaces to simulate the RSU and vehicles by creating pseudo processes. The scheduling algorithm is implemented in the pseudo process of the RSU, which will be invoked in every scheduling period. When a vehicle drives into the RSU’s coverage, a new pseudo process will be created to simulate its both physical (e.g. velocity, coordinate, etc.) and cyber (cached/requested information, communication interface, etc.) features. The coordination and the communication among different pseudo processes are handled by specific CSIM interfaces.

Traffic features are generated based on the Greenshield’s model [30], which is widely adopted in simulating macroscopic traffic scenarios [31]. It is a model of uninterrupted traffic flow that predicts and explains the trends observed in real traffic flows. It assumes that under uninterrupted flow conditions, the speed and density of vehicles are linearly related. Specifically, the relationship between the velocity (\(v\)) and the density (\(k\)) is represented by:

\[
v = V_f - V_f/K_j \cdot k \tag{10}
\]

where \(V_f\) is the free flow speed, which is the maximum speed when the traffic density is zero. \(K_j\) is the jam density, which is the density at which there is a traffic jam and the flow speed equals zero. Three lanes are simulated, and the free flow speeds of each lane are set to \(V_1 = 100\) km/h, \(V_2 = 80\) km/h and \(V_3 = 60\) km/h, respectively. The same jam density \(K_j\) is set for each lane, which is 100 vehicles/km. Consider that all the vehicles drive in the same direction and the arrival of vehicles in each lane follows the Poisson process. A wide range of vehicle arrival rates is simulated to evaluate the scheduling performance under different traffic workloads. Relevant statistics including...
vehicle velocities and vehicle densities are collected under different traffic workloads, which are summarized in Table III.

Vehicular communication characteristics are simulated based on DSRC. In particular, the communication radius of the RSU is set to 600 m, and the V2V communication range is set to 300 m. We do not specify absolute values of the data size and the data transmission rate, but setting the scheduling period to 1 s. This is reasonable because it has been shown that DSRC is able to support the data transmission rate of $6 \sim 27$ Mbps [32]. Including the overhead of exchanging control messages in each scheduling period, 1 s is sufficient to disseminate a data item (an encoded packet) with normal sizes (e.g., in the order of KBytes) [33]. The database contains 100 data items and the default cache size of vehicles is 20 (data items). Each vehicle randomly generates 1 to 7 requests when passing through the RSU’s coverage. The data access pattern follows the Zipf distribution [34] with the parameter $\theta = 0.6$, and the access probability of a data item $d_i$ is computed by:

$$\left(\frac{1}{|D|}\right)^\theta \sum_{j=1}^{|D|} \left(\frac{1}{j}\right)^\theta$$

where $|D|$ is the size of the database.

We implement three representative solutions for performance comparison. One is FCFS (First Come First Served) [35], which broadcasts data items according to the arrival order of requests. The second algorithm is MRF (Most Requested First) [36], which broadcasts the data item with the maximum number of pending requests. Note that since there is no V2X coordination mechanisms which can be directly adopted for FCFS and MRF, they are scheduling via pure V2I communication. The third algorithm is SFR [19], which considers both V2I and V2V communications for cooperative data dissemination in VANETs. To the best of our knowledge, SFR is the most relevant yet competitive solution for comparison. For NCAS, we set the threshold of coding length to 2, which strikes the best balance between minimizing the scheduling overhead and maximizing the service performance in the default setting. We adopt the following two metrics to quantitatively evaluate the performance of different algorithms.

- **Service ratio.** It is the ratio of the number of served requests $n_s$ to the total number of submitted requests ($n$) by all passing vehicles, which is computed by:

$$\text{ServiceRatio} = \frac{n_s}{n}.$$  \hspace{1cm} (12)

It is the most important metric to evaluate algorithm performance in real-time systems.

- **Average service delay.** For a served request $i$, the service delay is the duration from the time when the request is submitted to the time when the data item is retrieved, which is denoted by $\tau_i$. The average service delay is computed by:

$$\text{AverageServiceDelay} = \frac{\sum_{i=1}^{n_s} \tau_i}{n_s}.$$  \hspace{1cm} (13)

A shorter average service delay indicates better system responsiveness on serving requests.

### B. Simulation Results

Fig. 3 shows the service ratio of algorithms under different traffic scenarios. The ID of each traffic scenario ($x$-axis) corresponds to the index number in Table III, where the traffic workload increases from traffic scenario 1 to 5. As noted, NCAS and SFR perform much better than MRF and FCFS in all scenarios. This is because MRF and FCFS can only utilize the V2I service channel in data services while they cannot exploit the benefit of V2V communication. For SFR, although it considers the coordination between V2I and V2V communications, its performance is limited by conventional broadcast constraint. That is, in one broadcast tick, it can only serve those requests asking for the same data item. In contrast, NCAS breaks such a limit by applying network coding and it further improves the bandwidth efficiency. Therefore, NCAS achieves the best performance on maximizing the service ratio. When the traffic workload is getting heavier, there are more requests submitted

### TABLE III

<table>
<thead>
<tr>
<th>Traffic Scenario</th>
<th>Mean Arrival Rate (vehicles/h)</th>
<th>Mean Velocity (km/h)</th>
<th>Mean Density (vehicles/km)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lane1</td>
<td>Lane2</td>
<td>Lane3</td>
</tr>
<tr>
<td>1</td>
<td>1198</td>
<td>1000</td>
<td>882</td>
</tr>
<tr>
<td>2</td>
<td>1012</td>
<td>1292</td>
<td>1123</td>
</tr>
<tr>
<td>3</td>
<td>1990</td>
<td>1735</td>
<td>1242</td>
</tr>
<tr>
<td>4</td>
<td>2322</td>
<td>2138</td>
<td>1479</td>
</tr>
<tr>
<td>5</td>
<td>2728</td>
<td>2401</td>
<td>1677</td>
</tr>
</tbody>
</table>

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.
by vehicles, causing higher service workload. Meanwhile, the vehicle density is also getting higher, causing longer dwell time of vehicles in the RSU's service region, which implies a looser service deadline. Therefore, as shown in Fig. 3, the service ratio of algorithms is getting higher in heavier traffic workload scenarios because the loose service deadline dominates the scheduling performance.

Fig. 4 shows the average service delay of algorithms under different traffic scenarios. As MRF gives higher priority to data items with more pending requests, it helps to exploit the broadcast effect, which makes MRF outperform FCFS in reducing service delay. This is consistent with observations in previous work [26]. Meanwhile, NCAS and SFR achieve much shorter service delay than MRF. Although the improvement of NCAS compared with SFR is not so significant, it is actually a non-trivial achievement. This is because NCAS serves more requests than all the other algorithms as shown in Fig. 3, and this makes it more challenging for NCAS to achieve shorter average service delay at the same time.

Fig. 5 shows the service ratio of NCAS under different cache sizes. As designed, NCAS tries to enhance the cache utilization by finding free vehicles in each scheduling period and assigning them into either V2I or V2V service channel, so that these free vehicles can retrieve their unrequested data items without affecting the service to other vehicles. This strategy not only makes local cache hit possible, but also helps subsequent scheduling to better exploit the advantage of V2V data sharing and network coding. Therefore, it is expected that the performance of NCAS is getting better with larger cache sizes. This demonstrates that NCAS is able to enhance the system performance by best combing the cooperative V2X communication and the caching strategy.

As analyzed in Section IV-D, it may have high scheduling overhead if the coding length of packets is long. Therefore, it is expected to set an appropriate threshold of the coding length to make NCAS scalable and practical. Fig. 6 shows the service ratio of NCAS under different thresholds of coding length. As noted, NCAS has almost the same performance when the maximum allowed coding length increases from 2 to 3. We observe from this result that even though it allows to code three data items, there is little chance to schedule packets with such a length to further improve bandwidth efficiency, because the decoding opportunity is slim. This observation is consistent with the theoretical analysis in Section IV-D and it verifies that NCAS is able to schedule with low overhead while maintaining decent performance.

VI. CONCLUSION AND FUTURE WORK

This is the first study on applying network coding for cooperative data dissemination in V2X communication. Specifically, we present the implementation of the cooperative data dissemination model and analyze the system characteristics by considering both communication constraints and application requirements in VANETs. On this basis, we propose a network coding assisted scheduling algorithm called NCAS. It transforms the scheduling problem by constructing a graph containing all the possible service operations (i.e., tentative schedule) and their relationships (i.e., whether they are in conflict with each other), and then it makes the scheduling decision.
based on finding the maximum independent set of the graph. In addition, we design a strategy to enable vehicles to cache their unrequested data items, which not only facilitates future V2V data sharing, but also increases the chance of packet decoding. Further, we analyze the scheduling overhead and demonstrate that NCAS can be more scalable by setting a proper threshold for the encoding length, while still maintaining satisfactory performance. Finally, we built the simulation model and give a comprehensive performance evaluation. The simulation results under a wide range of traffic workloads conclusively demonstrate the superiority of NCAS.

In future work, we will further investigate how to enable vehicles to cooperatively relay and share information via multi-hop V2V communication. In addition, it is desirable to incorporate the impacts from MAC and PHY layers into consideration. For instance, the signal quality may vary with environments and transmission distances, resulting in packet loss and heterogeneous data transmission rates. More robust solutions are expected to be developed in realistic wireless communication environments.

REFERENCES


Joseph Kee-Yin Ng (M’91–SM’00) received the B.Sc., M.Sc., and Ph.D. degrees from University of Illinois at Urbana-Champaign, all in Computer Science, in 1986, 1988, and 1993, respectively. Since 1993, he has been with the Department of Computer Science, Hong Kong Baptist University (HKBU), where he is currently a Professor. He is the Programme Coordinator of the Computer Science degree program and introduced health information technology and health informatics into the undergraduate and graduate programs in HKBU. Dr. Ng served as a Steering Chair, a Program Chair, and a General Chair for numerous international conferences and as an Associate Editor and a member of the editorial board of several international journals. He is the author of 135 technical papers published in journals and presented in conferences and the holder of two patents. He is also a Director of the Hong Kong Internet Registration Corporation Limited. His current research interests include real-time and embedded systems, multimedia communications, and ubiquitous/pervasive computing. He has been a member of the IEEE Computer Society since 1991. He had also served as the Region 10 Coordinator for the Chapter Activities Board of the IEEE Computer Society, and was the Coordinator of the IEEE Computer Society Distinguished Visitors Program (Asia/Pacific). He has served as an Executive Committee Member, General Secretary, Vice Chair, and Chair for the IEEE, Hong Kong Section, Computer Chapter. He is a member of the IEEE Communication Society, the Association for Computing Machinery, Hong Kong Computer Society, and the Founding Member and Exco-Member & Treasurer for the Internet Society (ISOC)-Hong Kong Chapter. He received numerous awards and certificates of appreciation from IEEE, IEEE Region 10, IEEE Computer Society, and from IEEE Hong Kong Section, for his leadership and services to the ICT Industry.

Wewei Wu (M’13) received the B.Sc. degree from South China University of Technology in 2006 and the Ph.D. degree from City University of Hong Kong and University of Science and Technology of China, in 2011. In 2012, he went to Mathematical Division, Nanyang Technological University, Singapore, for postdoctoral research in 2012. He is currently an Associate Professor with Southeast University, China. His research interests include wireless communications, optimizations and algorithm analysis, scheduling, game theory, and network economics.

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 Association for Computing Machinery and IEEE Computer Society. He has been the Chair of the IEEE, Hong Kong Section, Computer Chapter from 2006 to 2007.

Junhua Wang received the B.S. degree in computer science from Chongqing University, in 2014. She is currently working toward the Ph.D. degree with the College of Computer Science, Chongqing University. Her research interests include mobile computing and vehicular ad-hoc networks.

Wehei Wu (M’13) received the B.Sc. degree from South China University of Technology in 2006 and the Ph.D. degree from City University of Hong Kong and University of Science and Technology of China, in 2011. In 2012, he went to Mathematical Division, Nanyang Technological University, Singapore, for postdoctoral research in 2012. He is currently an Associate Professor with Southeast University, China. His research interests include wireless communications, optimizations and algorithm analysis, scheduling, game theory, and network economics.

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 KAIST, and the Ph.D. degree in computer science from University of Maryland, College Park in 1976, 1978, and 1986, respectively. He serves as the Dean of the Graduate School and the Director of CPS Global Center at DGIST. He has been a Professor of the Department of Computer Science, University of Virginia, and the WCU Chair Professor with Sogang University. He has been a Visiting Professor at KAIST, City University of Hong Kong, Ecole Centrale de Lille, France, and Linkoping University and University of Skovde, Sweden. He is a member of both the Korean Academy of Science & Technology and the National Academy of Engineering of Korea. He has served as the Chair of the IEEE Technical Committee on Real-Time Systems from 2007 to 2008, He is currently serving as an Associate Editor for the ACM Transactions on Cyber Physical Systems and has served on the editorial board of the IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and Real-Time Systems Journal. He is a founding member of ACM/IEEE CPS Week, and serving as a member of the steering committee for RTCSA, Cyber Physical Systems Week, and SEUS. He received the Outstanding Contribution Award form Cyber Physical Systems Week in 2012. His research interests include cyber—physical systems, real-time and embedded systems, database and data services, and wireless sensor networks. He has written or co-authored over 300 papers and edited/authored four books in these areas. His research has been funded by the Korean Government, National Research Foundation, USA National Science Foundation, DARPA, Office of Naval Research, Department of Energy, National Security Agency, and IBM.