2016 IEEE 41st Conference on Local Computer Networks

Towards Real-Time and Temporal Information Services in Vehicular Networks via Multi-Objective Optimization

Penglin Dai*, Kai Liu*◇, Liang Feng*, Qingfeng Zhuge*, Victor Chung Sing Lee† and Sang Hyuk Son‡
*College of Computer Science, Chongqing University, Chongqing, 400044, China
Email: {penglindai, liukai0807,liangf,qfzhuge}@cqu.edu.cn
†Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong
Email: csvlee@cityu.edu.hk
‡ Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology, Daegu 711-873, Korea
Email: son@dgist.ac.kr

***Abstract*—Real-time and temporal information services are intrinsic characteristics in vehicular networks, where the timeliness of data dissemination and the maintenance of data quality interplay with each other and influence overall system performance. In this work, we present the system architecture where multiple road side units (RSUs) are cooperated to provide information services, and the vehicles can upload up-to-date information to RSUs via vehicle-to-infrastructure (V2I) communication. On this basis, we formulate the distributed temporal data management (DTDM) problem as a two-objective problem, which aims to enhance overall system performance on both the service quality and the service ratio simultaneously. Further, we propose a multi-objective evolutionary algorithm called MO-DTDM to obtain a set of pareto solutions and analyze how to fulfill given requirements on system performance with obtained pareto solutions. Finally, we build the simulation model and give a comprehensive performance evaluation, which demonstrates the superiority of the proposed optimization method.**

***Keywords*—vehicular networks; real-time data dissemination; temporal information service; multi-objective optimization**

I. INTRODUCTION

Vehicular networks are promising to enhance the transportation system in terms of safety, efficiency and sustainability by enabling vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications [1, 2]. In vehicular networks, there are a plenty of temporal information, such as real-time traffic information, location-based service information, etc., which requires to be timely updated [3]. Further, in many emerging real-time applications, such as road reservation [4], autonomous intersection control [5], etc., the data service has to be provided within certain time constraint. Therefore, it is necessary to maintain satisfactory quality of temporal information while supporting real-time services at the same time, which is non-trivial due to the limited wireless bandwidth and the highly dynamic vehicular environment.

In the literature, many research efforts have been deployed for data dissemination in vehicular networks. For instance, store-carry and forward [6–9] is a widely employed mechanism to disseminate data in vehicular networks, where data transmission from the source vehicle to the destination vehicle is a multi-hop process via a series of relay vehicles. However, due to the highly dynamic topology of vehicular networks, the store carry and forward strategy may cause serious long end-to-end delay [10, 11], and thus it is not suitable for real-time applications. A few studies have considered information services in vehicular ad hoc networks (VANETs). [12] considers the temporality of data item and schedule service request with dynamic snapshot consistency requirement. In [13], a RSU is responsible for broadcasting temporal information to passing vehicles, where the value of data is updated periodically via the backbone network. Nevertheless, the model does not consider the overhead of data update in terms of bandwidth consuming. In addition, [14] considers the system where a RSU is installed at the roadside to provide temporal information services. The information update at the RSU is completed by passing vehicles via V2I communication. [15] considers that the processes of data update and data broadcast are competing for the limited bandwidth resource, and a priority-based algorithm (PBS) is proposed to enhance the bandwidth efficiency. However, neither [14] nor [15] considers the cooperation of RSUs for providing information services.

With the rapid development of wireless communication technologies in VANETs, large-scale services are expected to be deployed where multiple RSUs are expected to incorporate with each other for providing efficient temporal and real-time information services [16, 17]. Taking this cue, in this work, we contribute to such services by developing a system in which multiple RSUs with local databases are connected together via the backbone network. Any data update in one local database will be synchronized in other local databases via the backbone network. Further, as the temporal data value

---

◇ Corresponding author

© 2016, Penglin Dai. Under license to IEEE.
DOI 10.1109/LCN.2016.117

needs to be updated timely to guarantee the usefulness of information, passing vehicles of a RSU are able to upload the up-to-date information collected along their travel trajectory. Meanwhile, RSUs are also capable of scheduling and disseminating information based on the requests submitted by vehicles. As we can see, this system architecture is scalable in supporting data services in a wide range of areas. However, the design of an efficient cooperative service scheme for such a system is challenging. First, the processes of data update and data broadcast compete for a limited wireless bandwidth, and thus the excessive resource consuming by one party will adversely impact on the performance of the other. Second, as different vehicles may update the different information to different RSUs, the efficient and effective cooperation on data update among RSUs is not a trivial task. Third, due to different service demands and traffic workloads in different places, it is expected to strike a balance among the RSUs in terms of both maintaining data quality and providing real-time services. Keep the above in mind, this paper makes the first attempt to explore the cooperation among the multiple RSUs as well as the passing vehicles to simultaneously improve the quality of temporal information and enhance service ratio of real-time applications.

The main contributions of this paper are outlined as follows. First, we present a distributed temporal information service system, in which multiple RSUs as well as passing vehicles are cooperated on data dissemination and data update. Second, we formulate the distributed temporal data management (DTDM) problem, which aims to enhance both the quality of temporal data and the service ratio of real-time requests simultaneously. Third, in order to optimize both the objectives, we further propose a multi-objective distributed temporal data management (MO-DTDM) algorithm to find a set of pareto-solutions. By adjusting the weight on data update and data broadcast, the proposed MO-DTDM can adaptively select the final solution to satisfy different given requirements on the trade-off between service quality and service ratio. Fourth, we build the simulation model and implement the proposed algorithm as well as two state-of-the-art solutions in literature for performance comparison. The comprehensive simulation results verify the effectiveness of the proposed algorithm under a wide range of scenarios.

The rest of the paper is organized as follows. Section II presents the system architecture. In Section III, we formulate the DTDM problem and in Section IV, we propose the MO-DTDM algorithm. In section V, we build the simulation model and present the performance evaluation. Section VI concludes the work and discusses future research directions.

II. SYSTEM MODEL

In this section, we introduce the system architecture of real-time and temporal information services in VANETs. As shown in Fig. 1, the system consists of multiple RSUs distributed along the roadside. Vehicles are assumed to be equipped with on-board units (OBU), which support both V2I and V2V communications. In addition, vehicles are capable of sensing and caching up-to-date information along their driving trajectories, such as real-time traffic information, location-based information, etc. The time stamp is recorded to reflect the status of temporal data. Each vehicle will send a notification message to the RSU once it enters into the RSU's communication range, which contains a list of caching data with the time stamp and a list of requesting data. Each RSU provides information to vehicles via an on-demand manner. The local databases of RSUs are connected via a high speed wired backbone network, and we assume that any data updated at one RSU will be synchronized in peer databases simultaneously. In addition, a centralized scheduler is used to coordinate RSUs for both data update and data broadcast.

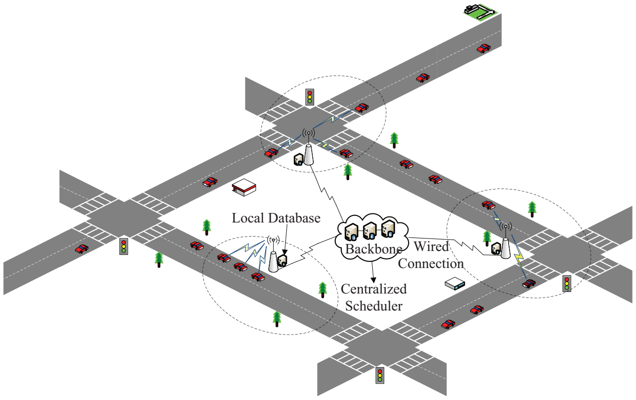


Figure 1. System architecture of real-time and temporal information services in VANETs

As the quality of temporal information will decrease over time, timely data update is required to keep the freshness and usefulness of the information stored in RSUs. According to the notification messages collected from passing vehicles, each RSU archives two lists, i.e., update list and request list, where the update list includes the available fresh data cached by vehicles and the request list includes the service requests sent by vehicles. The primary missions of RSUs are described as follows. First, the RSU is responsible for assigning upload tasks to passing vehicles and letting them to upload their cached data to refresh the low-quality temporal data in the database. Second, the RSU is responsible for serving the requests by broadcasting the requested data items to passing vehicles. The service has to be completed before the vehicle leaves the communication range of the RSU, which is denoted by the dotted ellipse in Fig. 1. Due to the limited wireless bandwidth, in order to guarantee both service quality and service ratio, RSUs have to strike a balance on bandwidth allocation for data update and data broadcast. Further, in the distributed system, a proper cooperation among RSUs is expected so as to better balancing the service workload. With the presented system, this paper is dedicated to investigating the scheduling for both data update and data broadcast via the coordination of multiple RSUs, so that the system is able to enhance the service ratio for real-time requests while maintaining satisfactory data quality for temporal information services.

In addition, we give one specific application to further clarify the concerned problem. For autonomous multiple-intersection control, a RSU is installed at the intersection for giving vehicles optimal speed advisory so that they'll pass the intersection safely and efficiently. The vehicles in RSU's coverage have to upload their temporal information, including GPS positions, velocities, accelerations, etc. Based on the collected data, the controller makes a global optimal solution. Then, the RSU broadcasts the instructions to vehicles via control messages so that they can cooperatively pass the intersection. Note that, either low quality of temporal data or non-real-time services may result in serious consequences this application.

III. PROBLEM FORMULATION

A. Notations

The set of RSUs in the system is denoted by $R = \left\{r_1, r_2, ..., r_{\|R\|}\right\}$ and the set of the temporal data items is denoted by $D = \left\{d_1, d_2, ..., d_{\|D\|}\right\}$. The communication range of each RSU is denoted by l. The set of vehicles in the communication range of $r_i \in R$ at time t is denoted by $V_i\left(t\right) = \left\{v_{i1}, v_{i2}, ..., v_{i\|V_i(t)\|}\right\}$. For each vehicle $v_{ik} \in V_i(t)$, the entering time and leaving time are denoted by E_{ik} and L_{ik}, respectively. The set of requested and cached data items of vehicle v_{ik} at time t are denoted by $H_{ik}\left(t\right)$ and $C_{ik}\left(t\right)$, respectively, where $H_{ik}\left(t\right) \subseteq D$, $C_{ik}\left(t\right) \subseteq D$ and $H_{ik}\left(t\right) \cap C_{ik}\left(t\right) = \emptyset$.

B. Quality of Temporal Data

In this section, we establish the data quality model to quantitatively evaluate the freshness of temporal information. Specifically, for d_j, we define the data quality function as $Q_{d_j}\left(t\right) = f(ts_{d_j}, l_{d_j}, t)$, where ts_{d_j} is the time stamp when d_j is generated and l_{d_j} is the valid period, which means that the value of d_j will be outdated at time $ts_{d_j} + l_{d_j}$. During $\left[ts_{d_j}, ts_{d_j} + l_{d_j}\right]$, the quality of d_j will degrade over time if it was not updated in due course. Clearly, the computation of current data quality depends on both ts_{d_j} and l_{d_j}. If the value of ts_{d_j} is closer to the current time t, it indicates that d_j is generated more recently. On the other hand, a larger value of l_{d_j} represents that the valid period of d_j is longer, which indicates that $Q_{d_j}\left(t\right)$ degrades more slowly over time. For the sake of better exhibition, $Q_{d_j}\left(t\right)$ is normalized in the range of $[0, 1]$, where 1 means the data has no loss of quality (when it is generated) and 0 means the data value is invalid. Accordingly, we have $0 \leq Q_{d_j}\left(t\right) \leq 1, \forall d_j \in D$. The time stamp and valid period is piggyback with the temporal data. Both the vehicles and the RSUs can obtain such information once they collect the data. $Q_{d_j}^{r_i}\left(t\right)$ and $Q_{d_j}^{v_{ik}}\left(t\right)$ represent the data quality of d_j maintained in the database of RSU r_i and cached by the vehicle v_{ik}, respectively. If v_{ik} uploads its cached d_j to the RSU r_i, $Q_{d_j}^{r_i}\left(t\right)$ would be updated to $Q_{d_j}^{v_{ik}}\left(t\right)$.

Note that the degradation of data quality is determined by particular application scenarios. Therefore, the formulation of the data quality function varies with different applications. To emphasize the generality of our analysis, we adopt the general form of the data quality model when analyzing the problem, while a specific form will be considered in Section V when implementing the simulation model.

C. Distributed Temporal Data Management

In this section, we formulate the DTDM problem in detail. During the service interval of $[0, T]$, the operation sequence of RSU r_i is denoted by $O_i = \left\{\left(o_{i1}, s_{i1}\right), \left(o_{i2}, s_{i2}\right), ..., \left(o_{ij}, s_{ij}\right), ...\right\}$, $\forall r_i \in R$. The j^{th} operation of RSU r_i is represented by a two-tuple $\left(o_{ij}, s_{ij}\right)$, where o_{ij} denotes the processed data and s_{ij} indicates the processed operation. Specifically, $s_{ij} = 0$ indicates the data upload operation and $s_{ij} = 1$ indicates the data broadcast operation. Accordingly, the start time t_{ij} of the j^{th} operation $\left(o_{ij}, s_{ij}\right)$ is calculated by $t_{ij} = \sum\limits_{k=1}^{j-1}\left(\tau_1 + s_{ij}\left(\tau_2 - \tau_1\right)\right)$, where τ_1 and τ_2 represent the time slot for uploading one data item by a vehicle and broadcasting one data item by a RSU, respectively.

When RSU r_i broadcasts data o_{ij} at time t_{ij}, a vehicle v_{ik} will acquire o_{ij} if it satisfies the following two conditions: 1) v_{ik} can complete the retrieval of o_{ij} before it leaves the coverage of the RSU r_i, i.e., $t_{ij} + \tau_2 \leq L_{ik}$; 2) o_{ij} belongs to the set of requested data $H_{ik}\left(t_{ij}\right)$, i.e., $o_{ij} \in H_{ik}\left(t_{ij}\right)$. In order to evaluate the benefit of broadcasting o_{ij}, we define *Beneficial Vehicle Set* as follows.

Definition III.1. *Beneficial Vehicle Set: When RSU r_i broadcasts data d_j at time t, the beneficial vehicle set of d_j at time t is defined as the set of vehicles in the coverage of RSU r_i which can acquire d_j, expressed as follows:*

$$B_{d_j}^{r_i}\left(t\right) = \left\{v_{ik} \left| t + \tau_2 \leq L_{ik} \cap d_j \in H_{ik}\left(t\right), \forall v_{ik} \in V_i\left(t\right)\right.\right\} \tag{1}$$

When RSU r_i performs data update operation $\left(o_{ij}, 0\right)$ at time t_{ij}, it will assign the data upload task to a vehicle for enhancing data quality of o_{ij}. A vehicle v_{ik} can be chosen as a candidate if it satisfies the following two conditions: 1) v_{ik} can complete the data upload of o_{ij} before it leaves the coverage of RSU r_i, i.e., $t_{ij} + \tau_1 \leq L_{ik}$; 2) o_{ij} belongs to the set of cached data $C_{ik}\left(t_{ij}\right)$, i.e., $o_{ij} \in C_{ik}\left(t_{ij}\right)$. In order to evaluate the benefit of data update, we define the *Best Upload Quality* as follows:

Definition III.2. *Best Upload Quality: When RSU r_i performs data update operation for d_j at time t, the best upload quality of d_j at RSU r_i is defined as the highest data quality of d_j cached by the candidate vehicles in r_i, expressed as follows:*

$$Q_{d_j}^{u_i}\left(t\right) = \underset{\forall v_{ik} \in V_i(t)}{max}\left\{Q_{d_j}^{v_{ik}}\left(t\right) \left| t + \tau_2 \leq L_{ik} \cap d_j \in C_{ik}\left(t\right)\right.\right\} \tag{2}$$

Then, the vehicle assigned for uploading data d_j at RSU r_i is *determined as follows:*

$$v_i^* = arg \max_{\forall v_{ik} \in V_i(t)} \left\{ Q_{d_j}^{v_{ik}}(t) \left| t + \tau_2 \leq L_{ik} \cap d_j \in C_{ik}(t) \right. \right\} \tag{3}$$

If no vehicles cache d_j in coverage of r_i, $Q_{d_j}^{u_i}(t)$ equals 0.

After completing data update operation, the data quality of d_j in the database is updated to $Q_{d_j}^{u_i}(t_{ij} + \tau_1)$ (i.e. $Q_{d_j}^{r_i}(t_{ij} + \tau_1) \leftarrow Q_{d_j}^{u_i}(t_{ij} + \tau_1)$). The data quality of d_j in other local databases will be synchronized via the high speed backbone network accordingly.

Given the time interval $[0, T]$, in order to quantitatively measure the system performance, we define two metrics, namely, *service ratio* and *average quality of satisfied requests* as follows.

Definition III.3. *Service Ratio (SR):Given an operation sequence $x = \left(O_1, ..., O_{\|R\|}\right)$, SR is defined as the ratio of the number of satisfied requests to the total number of requests, which is computed by:*

$$f_1(x) = \frac{\sum_{i=1}^{\|R\|} \sum_{\forall (o_{ij}, s_{ij}) \in O_i} s_{ij} * \left\| B_{o_{ij}}^{r_i}(t_{ij}) \right\|}{\sum_{i=1}^{\|R\|} \sum_{\forall v_{ik} \in V_i(0)} \|H_{ik}(0)\|} \tag{4}$$

Definition III.4. *Average Quality of Satisfied Requests (AQSR): Given an operation sequence $x = \left(O_1, ..., O_{\|R\|}\right)$, AQSR is defined as the mean value of data quality of the satisfied requests, which is computed by:*

$$f_2(x) = \frac{\sum_{i=1}^{\|R\|} \sum_{\forall (o_{ij}, s_{ij}) \in O_i} s_{ij} * \left\| B_{o_{ij}}^{r_i}(t_{ij}) \right\| * Q_{d_j}^{r_i}(t_{ij})}{\sum_{i=1}^{\|R\|} \sum_{\forall (o_{ij}, s_{ij}) \in O_i} s_{ij} * \left\| B_{o_{ij}}^{r_i}(t_{ij}) \right\|} \tag{5}$$

From the system point of view, it desires both high service ratio and high service quality. With the above definitions, we formulate DTDM as a two-objective optimization problem as follows:

$$\begin{aligned} & maximize\ F(x) = (f_1(x), f_2(x)) \\ & subject\ to \sum_{j=1}^{\|O_i\|} (\tau_1 + s_{ij}(\tau_2 - \tau_1)) \leq T, i = 1, 2, ..., \|R\| \\ & \qquad o_{ij} \in D, s_{ij} \in \{0, 1\}, \\ & \qquad i = 1, 2, ..., \|R\|, j = 1, 2, ..., \|O_i\| \end{aligned} \tag{6}$$

Due to the limited wireless bandwidth and the dynamic vehicular environment, the two objectives are in conflict with each other. In the following section, we propose a multi-objective technique to solve the DTDM problem.

IV. ALGORITHM DESIGN

To tackle the multi-objective optimization problem, the population based evolutionary algorithm has been widely verified as an effective paradigm. In this section, we propose an evolutionary algorithm, namely Multi-Objective Distributed Temporal Data Management (MO-DTDM) to consider the optimization of both the service quality of temporal information and the service ratio of real-time requests simultaneously. First, we present the basic idea of the MO-DTDM algorithm. Then, we explain the detail mechanism of each component of the proposed MO-DTDM algorithm.

A. Multi-Objective Decomposition

First, we decompose the multi-objective optimization in Eq. 6 into N scalar optimization problems, where N is the population size of solutions for optimization. Let $w^1, ..., w^N$ be a set of evenly spread weight vectors, where $w^k = \left(w_1^k, w_2^k\right)$ and $w_1^k + w_2^k = 1$. Each weight vector w^k corresponds to a scalar optimization problem. The multi-objective problem in Eq. 6 can be decomposed into N scalar optimization subproblems by using Weighted Sum approach [18, 19]. Then, for each weight vector w^k, the k^{th} corresponding subproblem is expressed as follows:

$$g\left(x \left| w^k, z\right.\right) = \sum_{i=1}^{2} w_i^k f_i(x), subject\ to\ x \in \Omega \tag{7}$$

where Ω is the decision space. For each weight vector w^k, the best solution of the k^{th} subproblem is maintained for optimization purpose. Therefore, the population is composed of the best solution found so far for each subproblem.

B. Chromosome Encoding

In the proposed algorithm, a scalable representation is applied to represent a scheduling solution x including the scheduling decision of each RSU during the time interval $[0, T]$, as shown in Fig. 2. One rectangle represents an operation sequence of one RSU and the arrow represents the operation order. For example, O_2 represents the scheduling solution for the RSU r_2 and the two-tuple $(d_6, 0)$ indicates an update operation of d_6. Accordingly, the operations of r_2 in this example are to update d_6 and d_3 in sequence, and then broadcast d_3, and finally update d_7. Note that, the total time consumed for performing these operations cannot exceed T.

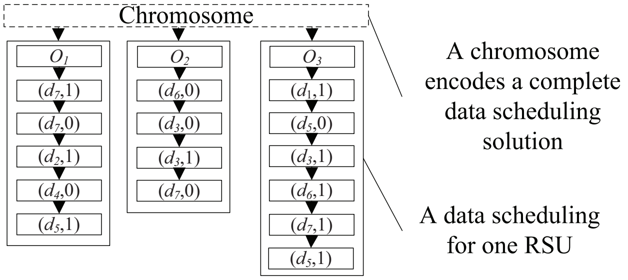


Figure 2. Encoding of the chromosome

C. Procedure of the MO-DTDM algorithm

In this section, the procedure of the proposed MO-DTDM is presented in detail. As outlined in Alg. IV.1, in Step 1, we generate the neighborhood set of each weight vector w^k and initialize the population. In Step 2, we update the population by evolutionary reproduction and selection. In Step 3, the stopping criterion is checked. If the stopping criterion is not satisfied, it goes back to Step 2. Otherwise, the set of non-dominated set is the output and the best solution of this set is chosen according to the designed metrics. In the following, each component of the algorithm will be explained in detail.

Algorithm IV.1 MO-DTDM algorithm

```
Step 1 Initialization:
  Generate neighborhood set $NB(k)$ of each weight vector $w_k$
  Initialize the population $x^1, x^2, ..., x^N$ and set $EP = \emptyset$
Step 2 Update:
  for $k = 1, 2, ..., N$ do
    Randomly select two indexes $m, n$ from $NB(k)$
    Generate a new solution $y$ from $x^n$ and $x^m$ by genetic operators.
    Apply an improvement method on $y$ to get new solution $y^{'}$
    Update neighborhood solutions of $x^k$ according to $y^{'}$
    Update $EP$ according to $y^{'}$
  end for
Step 3 Output:
  if Stopping criteria is satisfied then
    Output $EP$
  else
    Go to Step 2
  end if
  Choose the best solution from $EP$ by the given weight vector
```

1) Initialization: First, we initialize the non-dominate set EP as an empty set, which is used for storing the best solutions found so far. Let x^n and x^m be the two solutions, and $F\left(x^n\right)$ dominates $F\left(x^m\right)$ if and only if $f_i\left(x^n\right) \geq f_i\left(x^m\right), i = 1, 2$ and $f_i\left(x^n\right) > f_i\left(x^m\right)$ for at least one index $i \in \{1, 2\}$. The non-dominate set EP consists of the solutions which cannot be dominated by any other solutions in the population.

Second, we generate the neighborhood set of w^k by considering Euclidean distances between weight vectors. For any two weight vectors $w^n, w^m, \forall n, m \leq N$, the Euclidean distance $\|w^n - w^m\| = \sqrt{\sum_{l=1}^{2}\left(w_l^m - w_l^n\right)}$ is computed. Then, for each weight vector w^k, we can choose the M closest weight vectors as the neighborhood set $NB\left(k\right) = \left(k_1, k_2, ..., k_M\right)$, where $w^{k_1}, w^{k_2}, ..., w^{k_M}$ are the M closest weight vectors to w^k. The neighborhood of the k^{th} subproblem consists of all the subproblems with the weight vectors from $NB\left(k\right)$. Only the current solutions to its neighborhood subproblems are exploited for optimizing a subproblem.

Third, we generate the initial population by considering the benefit of operations. For each operation $\left(o_{ij}, s_{ij}\right)$, we can calculate the benefit of the operation in two aspects: broadcast benefit and update benefit. First, the broadcast benefit of an operation $\left(o_{ij}, s_{ij}\right)$ is computed as follows:

$$\Phi_1\left(\left(o_{ij}, s_{ij}\right), t\right) = Q_{o_{ij}}^{r_i}\left(ts_{o_{ij}}, l_{o_{ij}}, t\right) \cdot B_{o_{ij}}^{r_i}\left(t\right) \cdot s_{ij} \tag{8}$$

For a broadcast operation $\left(o_{ij}, s_{ij}\right)$, it equals the data quality of o_{ij} in the database of r_i multiplied by the total number of vehicles in r_i at time t which request o_{ij}. For an update operation $\left(o_{ij}, s_{ij}\right)$, it equals zero since it has no effect on data broadcast. Similarly, the update benefit of an operation $\left(o_{ij}, s_{ij}\right)$ is computed as follows:

$$\begin{aligned}\Phi_2\left(\left(o_{ij}, s_{ij}\right), t\right) = &\left(Q_{o_{ij}}^{u_i}\left(ts_{o_{ij}}, l_{o_{ij}}, t\right) - Q_{o_{ij}}^{r_i}\left(ts_{o_{ij}}, l_{o_{ij}}, t\right)\right)\\ &\cdot\left(\sum_{i=1}^{\|R\|} B_{o_{ij}}^{r_i}\left(t\right) + \varepsilon\right) \cdot \left(1 - s_{ij}\right)\end{aligned} \tag{9}$$

where ε is a constant and $0 < \varepsilon \ll 1$. For an update operation $\left(o_{ij}, s_{ij}\right)$, it equals the augmented data quality of o_{ij} multiplied by the total number of vehicles requesting o_{ij} under all RSUs at time t. For a broadcast operation, it equals zero. Then, we can evaluate the benefit of an operation $\left(o_{ij}, s_{ij}\right)$ under a weight vector w^k as follows:

$$\Phi\left(\left(o_{ij}, s_{ij}\right), t \left| w^k \right.\right) = \max_{1 \leq l \leq 2}\left(w_l^k \cdot \Phi_l\left(\left(o_{ij}, s_{ij}\right), t\right)\right) \tag{10}$$

For each weight vector w^k, the benefit of each operation is computed according to Eq. 10 and all the candidate operations of one RSU r_i, $(i = 1, 2, ..., \|R\|)$ are sorted in descending order. Then, we generate the scheduling solution of each weight vector w^k as follows: for each RSU r_i, the candidate operations are selected iteratively based on the predefined order until the total time consumed does not exceed T. Accordingly, we acquire the initial population, whose population size is N.

2) Update: For each index k, $k = 1, 2, ..., N$, we do the following procedures:

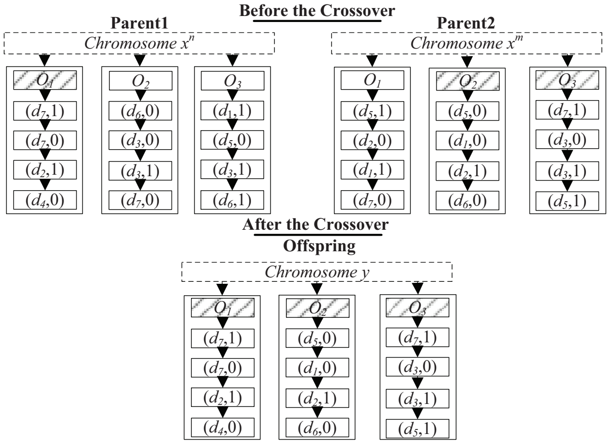
1. Reproduction: we randomly select two indexes n, m from $NB(k)$ and generate a new solution y from x^n and x^m by the designed genetic operators. The genetic operators are the uniform crossover operator and the standard mutation operator. For two solutions x^n and x^m, we apply uniform crossover operator to generate one child solution y. By using uniform crossover operator, each O_i of y is randomly selected from x^n and x^m. For example, in Fig. 3(a), O_1 of x^n and O_2 and O_3 of x^m are chosen to generate the solution y. Then, a standard mutation operator is applied to y. It mutates each O_i of the solution y independently with mutation probability ρ_1. If the O_i is selected to be mutated, then one operation of O_i is randomly selected to be mutated. In Fig. 3(b), the operation $(d_1, 0)$ of O_2 is selected and mutated to $(d_2, 0)$.

2. Improvement: In this step, an improvement method is proposed on the solution y to generate a new solution $y^{'}$. The basic idea of this improvement method is to exploit the cooperation among multiple RSUs to enhance the service quality. The improvement rules are described as follows. Rule 1, if both update and broadcast operations of one data belong to one O_i, then the order of update operation is placed before the broadcast operation to improve the broadcast data quality. Rule 2, if there exists an update operation $(d_i, 0)$ in an O_i, then the order of broadcast operations $(d_i, 1)$ of any other O_j $(O_j \neq O_i)$ are arranged after the order of the update operation $(d_i, 0)$ of O_i. For example, in Fig. 4, both $(d_7, 0)$ and $(d_7, 1)$ belong to O_1, and then $(d_7, 0)$ is placed before $(d_7, 1)$

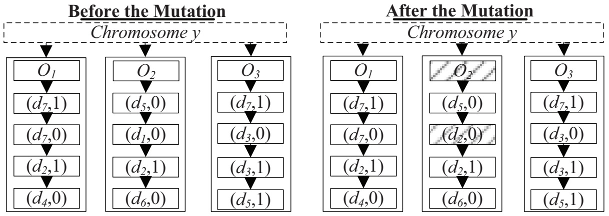
according to Rule 1. In addition, $(d_7, 1)$ of O_3 is placed after the order of $(d_7, 0)$ of O_1 according to Rule 2.

3. Update of neighborhood solutions: For each index $l \in B(k)$, if $g\left(x^l \left| w^k, z\right.\right) \leq g\left(y' \left| w^k, z\right.\right)$, then set $x^l = y'$.

4. Update of EP: In this step, all the solutions whose $F(x)$ dominated by $F\left(y'\right)$ are removed from EP and add y' to EP if no solution y in EP whose $F(y)$ dominates $F\left(y'\right)$.



(a) Uniform crossover operator



(b) Mutation operator

Figure 3. An example of genetic operators

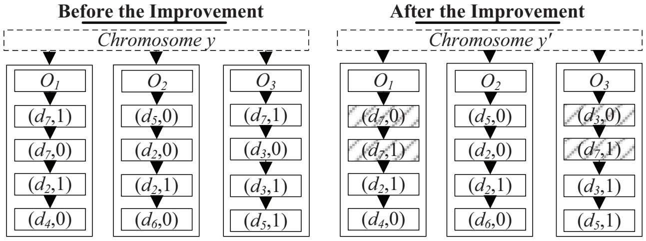


Figure 4. An example of improvement method

3) Output: The stopping criterion is described as follows: the iteration is terminated and output the EP if it satisfies that the iteration number reaches the maximum value, which is an experience value determined by particular problem. Once the non-dominate set EP is obtained, the scheduler chooses a solution from EP to execute. In order to adaptively fulfill the given requirement on service ratio and service quality, a weight vector is given to select the final solution. In particular, given $w' = \left(w'_1, w'_2\right)$ and $w'_1 + w'_2 = 1, w'_i \geq 0, i = 1, 2$, we choose the solution with the highest priority in EP, which is defined as follows:

$$P(x) = \sum_{i=1}^{2} f_i(x) w'_i \tag{11}$$

Then, the final solution x^* is chosen with the maximum value of $P(x^*) = \max\limits_{\forall x \in EP} \{P(x)\}$. In practice, the weight vector can be determined by certain evaluation method based on statistical information of particular application scenarios.

V. PERFORMANCE EVALUATION

A. Simulation Model

The simulation model is built based on the system architecture illustrated in Section II and it is implemented by SUMO [20] and Python. SUMO is used to simulate the traffic scenario and generate the mobility traces of vehicles. A 2km × 2km road network with 9 junctions is simulated and four RSUs are evenly distributed along the roadside to provide information services. In addition, the arrival of vehicles follows the Poison process with parameter λ. Python is used to simulate the vehicular communication environment and implement the proposed algorithm. For each vehicle, the number of cached data items by a vehicle is randomly generated in the range of cn and the number of requested data items by a vehicle is randomly generated in the range of rn. The total number of data items maintained in the database is 200. The time slots for broadcasting and uploading one data item are set to $1s$ and $2s$, respectively. This is reasonable because according to Dedicated Short Range Communication (DSRC) [21], the data rate of DSRC using BPSK modulation is about 3 Mbps and it consumes 1s to complete the transmission of one data item with 3Mb. The weight vector in MO-DTDM is set to $(0.5, 0.5)$ and the mutation probability is set to $\rho_1 = 0.05$. The default parameters are summarized in Table I. Unless stated otherwise, the simulation is conducted under the default setting.

To quantitatively evaluate the data quality, a commonly-used linear function [22] is adopted, which is a typical setting for evaluating quality of temporal information in vehicular networks. The formulation is expressed as follows:

$$Q_{d_j}(t) = \begin{cases} 1 - \frac{s_{d_j}}{l_{d_j}}, & s_{d_j} \leq t \leq s_{d_j} + l_{d_j} \\ 0, & t > s_{d_j} + l_{d_j} \end{cases} \tag{12}$$

For performance comparison, we have implemented two alternative solutions: a) PBS [15] algorithm, which designs a priority to evaluate the benefit of both broadcast operation and update operation and selects an operation with the optimal benefit at each time slot. b) Alternative-EDF, which alternatively chooses update operation or broadcast operation. The order of update (or broadcast) operation is sorted by the deadline. We consider the two objectives for evaluating system performance, namely, service ratio (SR) and average quality of satisfied requests ($AQSR$), which are defined in Eq. 4 and Eq. 5, respectively.

Table I
DEFAULT SETTING

Param.	Default	Description
$\|D\|$	200	Size of database
λ	$\frac{2}{3}veh/s$	The arrival rate of vehicles
L	1000m	The diameter of coverage of a RSU
cn	[8,10]	The range of cached number of data items
rn	[9, 11]	The range of requested number of data items
vp	[400,500]	The range of valid period of data items
τ_1	$2s$	The time slot for uploading one data item
τ_2	$1s$	The time slot for broadcasting one data item

B. Experimental Results

1) Effect of the weight vector: Fig. 5 shows SR and $AQSR$ of MO-DTDM under different weight vectors. We test nine evenly distributed weight vectors $w^k = \left(w_1^k, w_2^k\right), k = 1, 2, ..., 9$, to show the effect of weight vectors on the performance of SR and $AQSR$. As Fig. 5 shows, w_1^k ranges from 0.1 to 0.9, which indicates that the allocated bandwidth for broadcast operation changes from low to high. When w_1 keeps increasing, the SR grows dramatically at the beginning, and then it slows down. When w_1 increases, the broadcast operations with high broadcast benefit $\Phi_1\left(\left(o_{ij}, s_{ij}\right), t\right)$ are firstly chosen, which contribute to high growth on SR. Finally, when w_1 keeps increasing, the broadcast operations with low broadcast benefit $\Phi_1\left(\left(o_{ij}, s_{ij}\right), t\right)$ are chosen, which slows down the growth of SR. Conversely, when w_2^k increases gradually, the $AQSR$ grows dramatically at the beginning and then it slows down. It is because that the update operations with high update benefit $\Phi_2\left(\left(o_{ij}, s_{ij}\right), t\right)$ are chosen first, which contribute to fast growth on $AQSR$. When w_2^k keeps increasing, the update operations with low update benefit $\Phi_2\left(\left(o_{ij}, s_{ij}\right), t\right)$ are chosen, which result in slower growth on $AQSR$. As Fig. 5 shows, MO-DTDM is able to provide an adaptive solution to satisfy different requirements on the performance of SR and $AQSR$ by adjusting the given weight vector. Further, as Fig. 5 shows, excessively allocating bandwidth for one metric results in severe degradation of performance on another. Therefore, a reasonable selection of weight vectors are expected depending on the specific application requirements.

2) Effect of service workload: Fig. 6 shows the SR of the three algorithms under different service workloads. In x axis, a higher value of request number per vehicle indicates a heavier service workload. As shown, with the increasing of service workload, the SR of all the algorithms decreases dramatically, but MO-DTDM manages to achieve higher SR than other two algorithms in all cases. Fig. 7 shows the $AQSR$ of all the algorithms under different service workloads. The trend of $AQSR$ of both three algorithms increases gradually. This is because increasing service workload brings more benefit of broadcast effect and some vehicles receive requested data in advance and hence, it enhances the received data quality. In particular, the $AQSR$ of MO-DTDM and A-EDF increases faster than PBS. This is because MO-DTDM chooses the data with high value of broadcast benefit $\Phi_1\left(\left(o_{ij}, s_{ij}\right), t\right)$, which benefits more vehicles with higher data quality. Further, for A-EDF, increasing service workload makes service deadline more urgent, which enhances the broadcast data quality.

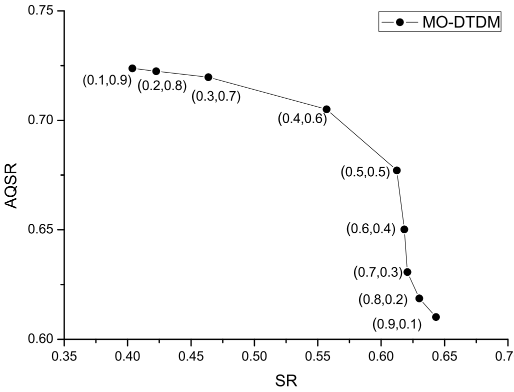


Figure 5. SR and $AQSR$ of MO-DTDM algorithm under different weight vectors

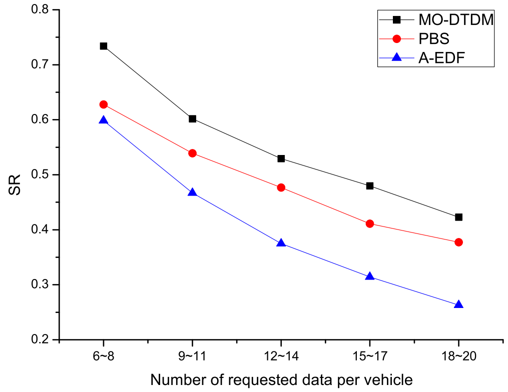


Figure 6. SR under different service workloads

3) Effect of data valid period: Fig. 8 shows the SR of the three algorithms under different data valid periods. The shorter data valid period indicates that the data quality degrades faster. When the data valid period increases, the SR of both MO-DTDM and PBS increases since they allocate more bandwidth for broadcast operations. The SR of A-EDF maintains the same since A-EDF always allocates half bandwidth for broadcast operation. Further, the MO-DTDM achieves better SR than other two algorithms in all cases. Fig. 9 shows the $AQSR$ of the three algorithms under different data valid periods. When the data valid period increases, the $AQSR$ of all three algorithms increases since the data quality degrades slower and MO-DTDM achieves better $AQSR$ than PBS and A-EDF in all cases. Also, we note that the increasing rate of $AQSR$ of MO-DTDM slows down but A-EDF increases dramatically. This is because MO-DTDM and PBS allocate

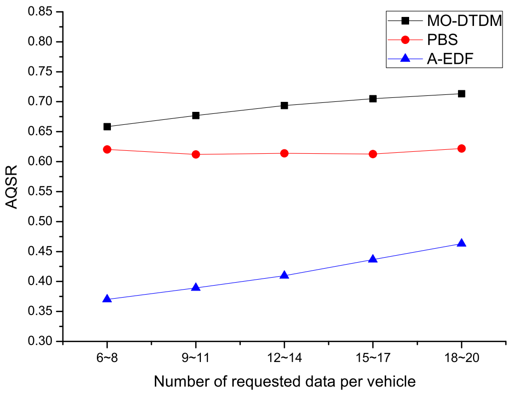


Figure 7. $AQSR$ under different service workloads

less bandwidth for update operation when data valid period increases while A-EDF always allocates the same bandwidth for update operation. However, the A-EDF always chooses the upload data with the closest service deadline, which degrades the upload quality of data. This explains why the SR of A-EDF maintains at a stable level and it is much lower than MO-DTDM and PBS when data valid period increases. With the above analysis, we may safely conclude that MO-DTDM can achieve better performance than existing solutions in terms of enhancing both service ratio and service quality in a wide range of application scenarios.

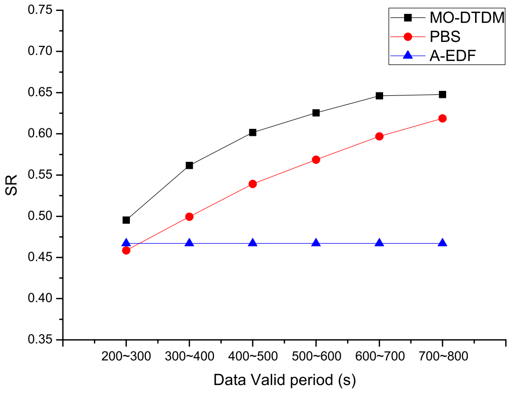


Figure 8. SR under different data valid periods

VI. CONCLUSION

In this paper, we present the system where multiple RSUs are cooperated to provide temporal and real-time services to passing vehicles. Passing vehicles can sense up-to-date temporal information and upload fresh data cooperatively to RSUs. We have investigated the DTDM problem and formulated it as a two-objective problem, which aims to improve both the service quality and the service ratio simultaneously. Further, we have proposed a scheduling algorithm called MO-DTDM, which can explore the cooperation of multiple RSUs to enhance both $AQSR$ and SR by providing a set of pareto solutions. By adjusting the given weight vectors, MO-DTDM adaptively selects the solution from the non-dominate set to satisfy the given requirement by striking the best balance on the service quality and the service ratio. Lastly, we have built a simulation model and provided a comprehensive performance evaluation. The simulation results demonstrated that MO-DTDM significantly outperformed the existing solutions in a wide range of scenarios.

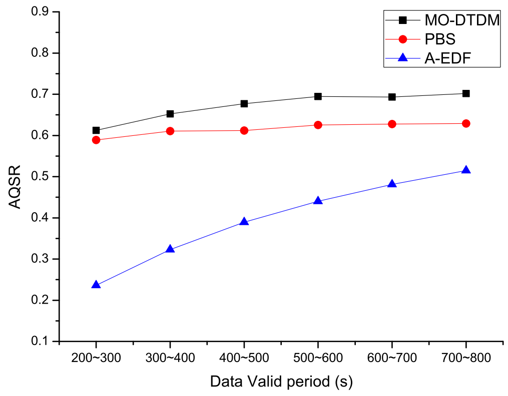


Figure 9. $AQSR$ under different data valid periods

For future work, we would like to incorporate the V2V communication for data services to further improve channel utilization and enhance system scalability. In addition, we would also like to investigate the communication issues resided in lower layers of VANETs, such as packet loss and interference, to establish a more realistic system model.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61572088); Fundamental Research Funds for the Central Universities (Grant No.106112015CDJZR185518); ICT R&D program of MSIP/IITP (14-824-09-013, Resilient Cyber-Physical Systems Research) and GRL Program (2013K1A1A2A02078326) through NRF.

REFERENCES

[1] J. Santa, A. F. Gómez-Skarmeta, and M. Sánchez-Artigas, “Architecture and evaluation of a unified v2v and v2i communication system based on cellular networks,” *Computer Communications*, vol. 31, no. 12, pp. 2850 – 2861, 2008.

[2] K. Liu, J. K. Y. Ng, V. C. S. Lee, S. H. Son, and I. Stojmenovic, “Cooperative data scheduling in hybrid

vehicular ad hoc networks: Vanet as a software defined network," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1759–1773, June 2016.

[3] S. H. Bouk, G. Kim, S. H. Ahmed, and D. Kim, "Hybrid adaptive beaconing in vehicular ad hoc networks: A survey," *Int. J. Distrib. Sen. Netw.*, vol. 2015, pp. 4:4–4:4, Jan. 2015.

[4] P. Su and B. B. Park, "Auction-based highway reservation system an agent-based simulation study," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 211 – 226, 2015.

[5] H. Fu, A. J. Pel, and S. P. Hoogendoorn, "Optimization of evacuation traffic management with intersection control constraints," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 376–386, Feb 2015.

[6] J. Zhao and G. Cao, "Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, May 2008.

[7] R. Jiang, Y. Zhu, X. Wang, and L. M. Ni, "Tmc: Exploiting trajectories for multicast in sparse vehicular networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 262–271, Jan 2015.

[8] J. Wu, S. Yang, and F. Dai, "Logarithmic store-carry-forward routing in mobile ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 6, pp. 735–748, June 2007.

[9] V. N. Soares, J. J. Rodrigues, and F. Farahmand, "Geospray: A geographic routing protocol for vehicular delay-tolerant networks," *Information Fusion*, vol. 15, pp. 102 – 113, 2014, special Issue: Resource Constrained Networks.

[10] M. Zhu, J. Cao, D. Pang, Z. He, and M. Xu, "Sdn-based routing for efficient message propagation in vanet," in *Wireless Algorithms, Systems, and Applications*. Springer, 2015, pp. 788–797.

[11] A. Abdrabou and W. Zhuang, "Probabilistic delay control and road side unit placement for vehicular ad hoc networks with disrupted connectivity," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 129–139, January 2011.

[12] K. Liu, V. C. S. Lee, J. K. Y. Ng, S. H. Son, and E. H.-M. Sha, "Scheduling temporal data with dynamic snapshot consistency requirement in vehicular cyber-physical systems," *ACM Trans. Embed. Comput. Syst.*, vol. 13, no. 5s, pp. 163:1–163:21, Oct. 2014. [Online]. Available: http://doi.acm.org/10.1145/2629546

[13] K. Liu, V. C. S. Lee, J. K. Y. Ng, J. Chen, and S. H. Son, "Temporal data dissemination in vehicular cyber physical systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2419–2431, Dec 2014.

[14] Y. Zhang, J. Zhao, and G. Cao, "Service scheduling of vehicle-roadside data access," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 83–96, 2009.

[15] P. Dai, K. Liu, E. Sha, Q. Zhuge, V. Lee, and S. H. Son, "Vehicle assisted data update for temporal information service in vehicular networks," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sept 2015, pp. 2545–2550.

[16] W. Saad, Z. Han, A. Hjorungnes, D. Niyato, and E. Hossain, "Coalition formation games for distributed cooperation among roadside units in vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 48–60, January 2011.

[17] G. N. Ali, E. Chan, and W. Li, "Supporting real-time multiple data items query in multi-rsu vehicular ad hoc networks (vanets)," *Journal of Systems and Software*, vol. 86, no. 8, pp. 2127 – 2142, 2013.

[18] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, Dec 2007.

[19] K. Miettinen, *Nonlinear multiobjective optimization*. Springer Science & Business Media, 2012, vol. 12.

[20] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo–simulation of urban mobility," in *The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain*, 2011.

[21] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the united states," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, July 2011.

[22] C. Anagnostopoulos and S. Hadjiefthymiades, "Delay-tolerant delivery of quality information in ad hoc networks," *Journal of Parallel and Distributed Computing*, vol. 71, no. 7, pp. 974 – 987, 2011.