Research Article

Downlink Traffic Scheduling with Contact Durations Awareness for Vehicular Infrastructures

Zhongjiang Yan and Bo Li

School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China

Correspondence should be addressed to Zhongjiang Yan; yangelc@msn.com

Received 11 July 2014; Revised 28 September 2014; Accepted 16 October 2014; Published 17 November 2014

Academic Editor: Joel Rodrigues

Copyright © 2014 Z. Yan and B. Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The vehicular infrastructures or roadside units (RSUs) in vehicular delay tolerant networks (VDTNs) can be used as the gateways of the distributed sensor networks. The different classes of service (CoS) support are desired when more than one type of the sensed data are collected by the RSUs. In this paper, the CoS support traffic scheduling problem for the RSU in VDTNs is considered. By exploring the contact information between the vehicles and the RSU, the CoS traffic scheduling problem is formulated as a maximum weighted triple matching problem, where the traffic scheduling strategy is a timeslot-vehicle-traffic matched pair. A flow network based method is proposed to optimally solve the maximum weighted triple matching problem. Both the offline version and the online version of the traffic scheduling algorithm are developed. Extensive simulations are conducted and the simulation results show the effectiveness and efficiency of the proposed flow network based algorithms.

1. Introduction

Traffic scheduling in vehicular delay tolerant networks (VDTNs) [1–3] has been attracting increasing research interests recently. One type of the VDTNs is installed in the sparse or less populated remote areas, which includes a limited number of unconnected vehicular infrastructures or roadside units (RSUs) and the passing by vehicles. The unconnected RSUs (without backbone network connections) are served as the gateways of the distributed sensor networks. For example, the distributed sensor networks are deployed to monitor the environment or the wildlife [1–6] in the sparse or less populated area. Since the direct communication connections are sparse in the less populated area which may not be covered by the cellular networks, a feasible low cost solution is proposed by using the passing by vehicles [7] carrying the sensed data from the unconnected RSU to the destination RSUs. In other words, after the sensed data is collected by the gateways or the RSUs, the unconnected RSU opportunistically sends the data to the vehicles. Then the vehicles store, carry, and forward the data to the destination RSUs when they arrive at the destination RSUs, that is, in a direct delivery routing scheme.

In this paper, we aim at solving the different classes of service (CoS) support downlink traffic scheduling problem, where the traffic is delivered by the unconnected RSU to the passing by vehicles. Generally speaking, there are two aspects of causes which may make the downlink traffic scheduling problem much more difficult than that of the conventional networks, for example, the cellular networks or the WLAN, which are the diversity of the traffics and the diversity of the vehicles.

1.1. Diversity of the Traffics. It means that the traffics have different transmission requests and need different classes of service (CoS) support [8]. Since the gateway or the RSU is much more expensive than the sensors, a limited number of RSUs will be deployed. And more than one type of the sensed data may be possibly collected by the RSU, waiting to be delivered to their respective destinations, for example, the wildlife tracking information [4], the agricultural information [5], and the environment information [6]. Additionally, these different types of data may be classified into different categories according to their importance, such as the TTL (time to live), destinations, or QoS (quality of service). Since the direct delivery routing scheme is assumed, thus the more
of the much more important data is delivered to the passing vehicles, the more gains will be obtained by the RSU. For example, in certain scenarios, the agriculture information is much more important than the environment information. More gains will be obtained by the RSU if it delivers more agriculture information to the vehicles. However, some vehicles may not drive to the destination RSU which collects the agriculture information, or the buffer capacity of the vehicles may be limited. Thus, another challenge arises, that is, the diversity of the vehicles.

1.2. Diversity of the Vehicles. It means that the vehicles are different from each other, in terms of their trajectories, buffer capacities, speeds, arrival times, and so on. Firstly, each vehicle drives in a different trajectory and may only pass a subset of the destination RSUs; thus only a subset of traffics can be carried by it. Secondly, each vehicle has a limited buffer capacity; thus the RSU can not send more data to avoid the buffer overflow. Thirdly, the velocities and the arrival times of the vehicles are different, which result in the contact durations of the vehicles and the RSU being different.

The short contact duration between the passing by vehicles and the RSU is one of the unique characteristics of the VDTNs [10]. Since the transmission range of the RSU is limited and the vehicles will not speed down to wait for the traffic downloading from the RSU, thus how to make a sufficient schedule during such a limited period becomes a significant problem. In addition, since the vehicles randomly arrive at the RSU, thus when there are more than one vehicle located in the transmission range, the problem with which order should the RSU deliver the data to the vehicles to maximize the total gains of the RSU is much more challenging.

To address these two challenges, in this paper we propose to explore the information of the contact durations between the vehicles and the RSU. The contact duration is defined as a time period, starting from the instant when the vehicle drives into the transmission area of the RSU and ending at the instant when the vehicle drives out of the transmission area of the RSU. During the contact duration, the RSU can use power control technology to transmit traffic to the passing by vehicles in a constant data rate [11–13]. Thus, the contact duration can be slotted and during each timeslot one unit of traffic (or one bundle) can be assumed to be transmitted from the RSU to the vehicle.

The main contributions of this paper are threefold. (1) By exploring the contact durations of each vehicle, the proposed different CoS support downlink traffic scheduling problem is formulated as a maximum weighted triple matching problem. A schedule strategy of the RSU is a timeslot-vehicle-traffic \((u-v-w)\) matched pair, which means that the RSU will transmit traffic \(w\) to vehicle \(v\) at the timeslot \(u\). (2) To compute the maximum weighted triple match of the triple matching graph, a flow network based algorithm is proposed. The online and the offline versions of the proposed algorithm are both developed. (3) Extensive simulations are conducted to evaluate the performance of the proposed algorithm. The simulation results are compared with the priority greedy based method, the FIFO based method, and the random based method, and the simulation results show the efficiency and the effectiveness of the proposed algorithm.

The rest of the paper is organized as follows. The availability analysis of the contact information is presented in Section 2. The related works are given in Section 3, and the differences between our work with the related works are also discussed. The system model is given in Section 4. A priority greedy based method is proposed in Section 5 firstly. Then, the flow network based traffic scheduling algorithm is developed. In Section 6, the online version of the proposed flow network based traffic scheduling algorithm is presented. Section 7 evaluates the performance of the proposed flow network based method through simulation, and Section 8 concludes this paper.

2. Contact Information Availability Analysis

Before exploring the information of the contact duration, we first analyse the availability of the contact information.

In [14], a layered architecture for the VDTNs is proposed where the bundle layer is placed below the network layer and above the data link layer. Particularly, the control plane functionalities and protocols are separated from the data plane. The former is responsible for connection setup for the data plane, by transmitting signalling control bundle out of band, while the latter is responsible for data bundle transmission, by aggregation and deaggregation the data bundles. These two are named as bundle signalling control (BSC) and bundle aggregation and deaggregation (BAD) in [14]. Normally speaking, a low powered, long-range, low bandwidth control plane link connection is used by the BSC to exchange control messages out of band, and a high powered, short-range and high bandwidth link is used by the BAD to exchange data bundles. In other words, the radius of the BSC, \(R_{BSC}\), is much larger than that of the BAD, \(R_{BAD}\), as shown in Figure 1.

When a vehicle is approaching the RSU (driving into the BSC area), the control plane works first to set up the connection. The vehicle registers itself to the RSU with the information in terms of the trajectory or destinations, geographical location, speed, power, and storage capacity. After that, the RSU will compute and update its traffic scheduling strategy when the vehicle drives into the BAD area. At the instant when the vehicle drives into the BAD area, an updated downlink traffic scheduling strategy will be invoked and the traffic will be delivered to this new arrived vehicle.

The setting \(R_{BSC} > R_{BAD}\) allocates a time duration for the new arrived vehicles to register and to negotiate with the RSU. At the same time, when the vehicle’s information is obtained by the RSU a new traffic schedule strategy can be computed and broadcasted to the vehicles. For example, in [15] it is assumed that \(R_{BSC} = 90\) m and \(R_{BAD} = 30\) m, and the vehicles are assumed to be driving on the road with a constant speed, say, 50 km/h. Then the time duration in which the vehicle is located in the BSC area and out of the BAD area is 4.32 s. Assume the IEEE 802.11p [16] is used for the control plane, where the basic data rate is 3 Mbps, and the SIFS (short
3. Related Works

A recent survey on the VDTNs can be found in [2]. Reference [3] details the layered architecture of VDTN, including the bundle aggregation and deaggregation mechanisms, network protocols, scheduling and dropping policies, fragmentation mechanisms, and the created testbed for VDTNs performance evaluation, demonstration, and validation. In the VDTNs, three classical routing schemes are mainly employed [3], that is, direct delivery, epidemic [18], and spray and wait [19]. According to the employed routing schemes, the related works on traffic scheduling can be classified into two categories, that is, replicated copy based traffic scheduling and direct delivery based traffic scheduling. The replicated copy based traffic scheduling employs the epidemic or spray and wait routing schemes, and the direct delivery based traffic scheduling employs the direct delivery routing scheme.

3.1. Replicated Copy Based Traffic Scheduling

The epidemic and the spray and wait routing schemes always deliver the replicated copies of the bundles to the contact nodes when a connection opportunity appears, in order to increase their deliver probability and decrease their delivery delay. However the replicate approach may cause contention for network resources, such as link bandwidth and storage, which will reduce the performance of the network. Thus an efficient scheduling and dropping policy is necessary to optimize the performance of the network. The scheduling policies address the problem with which order the bundles should send when two nodes contact. The dropping policies are used to decide which bundles should be dropped when the buffer is full. The existing literatures can be classified into two categories according to their objectives, that is, to improve the message delivery delay and delivery ratio and to support the different classes of service (CoS).

3.1.1. To Improve the Message Delivery Delay and Delivery Ratio. The authors of [20] propose three scheduling policies, that is, FIFO (first in first out), random, and RL-DESC (remaining lifetime descending order), and three dropping policies, that is, head drop, random, and RL-ASC (remaining lifetime ascending order). Simulation results show that these policies based on the lifetime criteria decrease the average delivery delay and increase the delivery ratio. Reference [17] extends the works in [20], where the number of the replicated copies is considered to design the scheduling and dropping policies. In other words, when a contact opportunity appears or when the buffer is full, the bundles with less/more replicated copies are scheduled/dropped first. Reference [21] studies the impact of scheduling and dropping policies through a VDTNs testbed and concludes that the policies should consider the bundles’ lifetime as a criterion to improve the network performance.

3.1.2. To Support the Different Classes of Service (CoS). Reference [8] considers scheduling and dropping policies for traffic differentiation in VDTNs. An urban scenario is considered in reference [8], where the VDTNs support several asynchronous applications simultaneously. Each application generates different requirements traffic in terms of message delivery probability and message average delay, which motivates the traffic priority support of the scheduling and dropping policies. A method to specify the relative priority of messages is proposed, where the bulk messages have the lowest priority, and the normal messages are sent prior to the bulk ones, and the expedited messages have the higher priority. Three traffic differentiation scheduling policies are proposed, that

Figure 1: An example of the source RSU and the passing by vehicles in VDTN.
is, priority greedy, round robin, and time threshold. Two dropping policies are proposed where the messages with the lowest priority or the lowest remaining lifetime are dropped first. Reference [22] further extends the work in [8]. Two different buffer management schemes and corresponding dropping policies are proposed. The first method is to store the bundles in a sharing buffer, and when the buffer is full the lowest priority bundles are dropped. The second method is to store the bundles in separate buffers according to the bundle’s priority, and the bundles with the lowest lifetime are dropped when the buffer is full. A priority greedy scheduling policy is proposed with the sharing buffer management method. And a custom service time scheduling policy is proposed with the separated priority buffer management method. The performances are evaluated through simulations.

3.2.1 Bundle Relaying. In [23], the bundle delivery delay minimization problem is considered, where a Markov decision process framework is proposed, and the optimal traffic strategy is derived. Based on the optimal strategy, the source RSU should transfer the traffic to the vehicle with velocity larger than a threshold. Reference [24] also considers the delay minimization problem. The probability that the RSU should release its bundle to the passing by vehicles is predicted based on the vehicles’ information. Then based on the probability the RSU determines when to deliver its bundles such that the bundle delivery delay can be minimized.

To jointly minimize the transmitting energy consumption and the traffic delay penalties, an optimal traffic scheduling strategy is derived in [25]. The traffic scheduling problem is formulated as an optimal stopping problem. An optimal threshold based schedule strategy is derived, where when the penalty exceeds the derived threshold, the RSU is optimal to deliver its bundle to the passing by vehicles.

In [26], the mobile vehicle is considered as the constraint resource for the multiple source RSUs to communicate with their destination RSUs. Due to the contention for the limited network resource, that is, the vehicle, an optimal traffic strategy for the mobile vehicle is derived, whether or not to relay a specified RSUs’ traffic. Furthermore, the optimal strategies for the source RSUs are derived, when the source RSUs operate in a cooperative game mode or in a noncooperative game mode, respectively.

3.2.2 Requirements Satisfying. Reference [27] is one of the earliest works which considers the traffic scheduling problem, the objective of which is improving the service ratio of the vehicles. A $D \times S$ weight based traffic scheduling method is proposed first, where $D$ is the deadline of the request and $S$ is the size of the requested traffic. The basic idea is to schedule the small size data first and to schedule the earlier deadline data first. To further improve the service ratio, the common requested data with the minimum value of $(D \times S)/N$ is scheduled and broadcasted first, where $N$ is the number of the requested vehicles.

Reference [11] is another related work which considers how to minimize the RSU energy cost while meeting the vehicles request. By considering the exponential increase of the transmission power with the linear increase of the distance between the vehicle and the RSU, a heuristic strategy is proposed. The basic idea is to serve these vehicles first, which have the fastest speed and are nearest to the RSU. That scheme is motivated by the fact that if a fastest and nearest vehicle is served later more transmission power will be spent. Reference [12] further studies the energy efficient problem. Two cases of the energy efficient problem are discussed, that is, the packet-based case and the timeslot-based case. The packet-based schedule problem requests continuous downlink time to meet the vehicle’s request, which is proved to be an NP-complete problem. The timeslot-based problem is proposed to solve in a flow network scheme, where the optimal traffic schedule strategy is corresponding to a minimum cost flow of the flow network.

Reference [13] studies a jointly optimization problem, which aims to fairly serve the vehicles while minimizing the total energy cost. The proposed problem is a three-step optimization problem which is however proved to be optimally solved in a flow network method. The key is to assign the cost of the edges in the flow network with a square function.

3.3 Relationships with the Existing Literatures. From the viewpoint of routing schemes, our work belongs to the direct delivery one. From the viewpoint of the functionality of the RSU in VDTNs, our work belongs to the bundle relaying one. However, to the best knowledge of the authors, the traffic scheduling problem to support different classes of the service (CoS) has not been considered in this scenario, let alone the diversity of the vehicles. However, the CoS and the diversity of the vehicles are so realistic and practical which make the traffic scheduling problem significant and challenging. Therefore, we propose to study this practical traffic scheduling problem for the source RSU.

4. System Model and Problem Formulation

In this paper, the vehicles are used as mobile relays of the RSUs and the direct delivery routing scheme is employed. In other words, when the vehicle receives bundles from the source RSU, it will store, carry, and forward these bundles to the destination RSUs when it arrives at the destinations; that is, no dropping policies are used by the vehicles. Each bundle can be sent by the RSU to one vehicle in one timeslot, and after that the bundle will be deleted from the buffer of the RSU. The buffers of the vehicles are shared by the variety of types of bundles. All types of bundles of the RSU are all delay tolerant and their lifetimes are considered to be long enough. In the following, we detail the system model used in this paper.
4.1. System Model. Suppose there is an RSU $S$ deployed along the highway road, which is used as a gateway of the underlay deployed distributed sensor networks. There are many types of the sensed data collected by the RSU, waiting for the passing by vehicles to be delivered to the destinations. According to the traffic differentiation mechanism, for example, proposed in [8, 22], the sensed data in the source RSU can be prioritized into $M$ types. Let $w_{m}(r_m, q_m)$, $1 \leq m \leq M$, and denote the $m$th type of sensed data, where $r_m$ is the gains obtained by the RSU when one bundle of $w_m$ is delivered by the RSU $S$ to a vehicle. And $q_m$ is the total quantity of bundles of $w_m$. The set of sensed data stored in the RSU $S$ can be written as

$$W = \{ w_{1}(r_1, q_1), \ldots, w_{m}(r_m, q_m), \ldots, w_{M}(r_M, q_M) \}. \tag{1}$$

Suppose there are totally $N$ vehicles located in the BAD area of the RSU $S$. These vehicles arrive at the RSU $S$ in a random process, where the interarrival times of the vehicles are independent and identically exponentially distributed [28]. Let $\{ \tau_1, \tau_2, \ldots, \tau_n, \ldots, \tau_N \}$ denote the interarrival times of the consecutive vehicles at the RSU $S$; then they are i.i.d exponential random variables with parameter $\mu$, where $\mu = 1/E[\tau_1]$ and $E[\cdot]$ is the expectation. Then, the arrival time of the $n$th vehicle is $t_n = \sum_{i=1}^{n-1} \tau_i$; that is, the instant when vehicle $v_n$ enters into the BAD area is $t_n$. Let $v_n = (t_n, D_n, s_n, p_n)$ denote the properties of the $n$th vehicle. $D_n$ is the delivery capability of vehicle $v_n$, a binary variable set with $|D_n| = M$, and the $k$th element of $D_n$ is 1 if the vehicle can take the $k$th type of data and is 0 otherwise. $s_n$ is the constant speed, and $p_n$ is the buffer capacity of $v_n$. Then the set of vehicles can be written as

$$V = \{ v_1(t_1, D_1, s_1, p_1), \ldots, v_n(t_n, D_n, s_n, p_n), \ldots, v_N(t_N, D_N, s_N, p_N) \}. \tag{2}$$

Suppose the RSU $S$ can use power control to transmit data bundles to the vehicles which are located in the BAD area. And, in each timeslot, only one data bundle will be transmitted from the RSU to the vehicles. Let $R_{\text{BSC}}$ and $R_{\text{BAD}}$ denote the BSC and BAD area transmission radius of the RSU $S$, respectively, where $R_{\text{BSC}} > R_{\text{BAD}}$. Then contact information between vehicle $v_n$ and the RSU $S$ can be denoted by $C_n = [t_n, t_{n} + (2 \times R_{\text{BAD}})/s_n]$, where recall that $t_n$ is the time instant when vehicle $v_n$ enters the BAD area and $(t_n + (2 \times R_{\text{BAD}})/s_n)$ is the time instant when vehicle $v_n$ drives out of the BAD area.

Let $\delta$ denote the length of a timeslot; then vehicle $v_n$’s contact information can be slotted as

$$C_n = \left\{ \left[ \frac{t_n}{\delta} \right], \left[ \frac{t_n}{\delta} + 1, \ldots, \left[ \frac{t_n + (2 \times R_{\text{BAD}})/s_n}{\delta} \right] \right] \right\}. \tag{3}$$

Figure 2 gives an example of sloting the contact durations of vehicles $v_1$, $v_2$, and $v_3$. Note that although these three vehicles enter the BAD area of the RSU at different time, part of the timeslots of them is overlapped; that is, $C_1 \cap C_2 \neq \emptyset$ and $C_2 \cap C_3 \neq \emptyset$. Let $u_k$ denote the $k$th timeslot. Let

$$U = \bigcup_{n=1}^{N} C_n = \{ u_1, u_2, \ldots, u_k, \ldots, u_K \} \tag{4}$$

de note the set of timeslot of the time duration $[t_1, t_1 + L]$, where

$$L = \max_{1 \leq n \leq N} \left( t_n + \frac{2 \times R_{\text{BAD}}}{s_n} \right) - t_1 \tag{5}$$

is the time length during which all of the $N$ vehicles are located in the BAD area.

Next we formally formulate the CoS support downlink traffic scheduling problem.

4.2. Problem Formulation. Based on the system model, a traffic scheduling strategy for the RSU $S$ can be represented as a timeslots-vehicles-traffic $(u_k, v_n, w_m)$ triple match. The physical meaning is that the RSU $S$ can send the traffic $w_m$ to vehicle $v_n$ at timeslot $u_k$. This motivates us to define a triple matching graph to formulate the proposed problem.

Define $G_{3m} = \{ V_{3m}, E_{3m} \}$ as a triple matching graph, as shown in Figure 3. $V_{3m} = U \cup V \cup W$ is the node set, composed of the timeslot set $U$, the vehicle set $V$, and the traffic set $W$. $E_{3m} = E_{\text{UV}} \cup E_{\text{VW}}$ is the edge set, where $E_{\text{UV}} = \{ e(u_k, v_n) \mid u_k \in U, v_n \in V \}$, and $E_{\text{VW}} = \{ e(v_n, w_m) \mid v_n \in V, w_m \in W \}$.

Recall that the objective of the proposed traffic scheduling problem is to maximize the total gains of the RSU $S$. The weight of the edge $h(\cdot, \cdot)$ is employed to translate the proposed problem to a maximum weighted triple matching problem. It concerns how to assign weight to the edges of $G_{3m}$ such that the sum of the weighted edge is maximized. If the weight $h(\cdot, \cdot) > 0$ then these two nodes of that edge are said to be matched with each other.
For each edge of $e(u_k, v_n) \in E_{UV}$, define $h(u_k, v_n)$ as the weight of $e(u_k, v_n)$, which is a boolean variable. In other words, for each edge of $e(u_k, v_n) \in E_{UV}$, the weight $h(u_k, v_n)$ can only be assigned to be 1 or 0. The physical meaning of $h(u_k, v_n)$ is whether or not to transmit bundle to vehicle $v_n$ at timeslot $u_k$. And $h(u_k, v_n) = 1$ means transmit, while $h(u_k, v_n) = 0$ means not to transmit. For each edge of $e(v_n, w_m) \in E_{3m}$, define $h(v_n, w_m)$ as the weight of $e(v_n, w_m)$, which is an integer variable; that is, $h(v_n, w_m) \geq 0$. The physical meaning of $h(v_n, w_m)$ is to transmit how many $w_m$ bundles to the vehicle $v_n$. Therefore, the proposed traffic scheduling problem can be formulated as a maximum weighted triple matching problem; that is,

$$\text{max} \quad \sum_{n=1}^{N} \sum_{m=1}^{M} h(v_n, w_m) r_m \quad (6)$$

subject to

$$\sum_{n=1}^{N} h(u_k, v_n) \leq 1, \quad \forall u_k \in U \quad (7)$$

$$\sum_{m=1}^{M} h(v_n, w_m) \leq p_n, \quad \forall v_n \in V \quad (8)$$

$$\sum_{n=1}^{N} h(v_n, w_m) \leq q_m, \quad \forall w_m \in W \quad (9)$$

$$\sum_{k=1}^{K} h(u_k, v_n) \geq \sum_{m=1}^{M} h(v_n, w_m), \quad \forall v_n \in V. \quad (10)$$

The objective of the maximum weighted triple matching problem is to maximize the total weighted gains of the RSU $S$. As indicated in the objective in (6), only the weight $h(v_n, w_m)$ of $e(v_n, w_m)$ contributes to the total gains of the RSU. The physical meanings of the four constraints are illustrated as follows. The first constraint in (7) means that any timeslot $u_k$ can be assigned to at most one vehicle. The second constraint in (8) means that the total number of bundles that each vehicle carries should be less than the capacity of the vehicle. The third constraint in (9) means that the total number of bundles that all of the vehicles carry should be less than the total amount of the traffic. The fourth constraint in (10) means that the total number of timeslots assigned to each vehicle, should be larger than the number of bundles that vehicle carries.

A feasible solution of the maximum weighted triple matching problem is a feasible traffic schedule strategy. For example, for any triple matching pair $(u_k, v_n, w_m)$, if $h(u_k, v_n) > 0$ and $h(v_n, w_m) > 0$, then it means that the RSU $S$ can transfer traffic $w_m$ to $v_n$ at timeslot $u_k$. In Figure 3, a triple matching pair $(u_1, v_1, w_2)$ is shown to be denoted by the grey lines, and it means that the strategy for timeslot $u_1$ is to transmit one bundle of $w_2$ to vehicle $v_1$.

An optimal solution of the formulated maximum weighted triple matching problem is called a maximum weighted triple matching, which maximizes the total gains from the source RSU. However, it is not easy to derive from the formulated problem as shown in (6)–(10) directly. Thus, in the next section, we present a flow network based method to optimally solve the formulated problem.

5. Flow Network Based Traffic Scheduling

In this section, we firstly propose a priority greedy based method to solve the maximum weighted triple matching problem, which will be used in the simulation to compare the performance with our flow network based traffic scheduling algorithm. Next, we detail the flow network based method.

5.1. A Priority Greedy Based Method. To maximize the total gains of the RSU $S$, an intuitive method is to deliver the highest priority bundles as many as possible to the passing by vehicles, to obtain as many gains as possible. In other words, at each timeslot $u_k$, the RSU $S$ will greedily transmit the highest priority bundle in its buffer to a selected vehicle which can carry that type of bundle. This motivates us to design a priority greedy based method to solve the maximum weighted triple matching problem.

Suppose that at timeslot $u'$ the RSU $S$ can communicate with $V' = \{v_1, v_2, \ldots, v_n, \ldots, v_{N'}\}$, $1 \leq n \leq N'$. The largest benefit $S$ can obtain is to transmit one bundle of $w'$ to $v'$, where $v' \in V'$. Thus $S$ will greedily transmit one unit of $w'$ to $v'$ at slot $u'$. When there exists more than one pair of $(v', w')$, which deserves the same gain for $S$, then $S$ will randomly select one pair of them to perform the transmission. Finally, $S$ will execute above method from one timeslot to another, until there are no vehicles or there are no traffics. In other words, at each timeslot $u'$, $S$ will always select the maximum benefit pair $(v', w')$ and then transmit.

The priority greedy based method locally maximizes the benefit of $S$ at each transmission slot $u'$. However, it may not be globally optimal, that is, over the total transmission slots $U = \{u_1, u_2, \ldots, u_K\}$. An example, which compares the priority greedy based method to the optimal method, is presented in Section 5.4. Thus, the optimal traffic scheduling strategy for the maximum weighted triple matching problem is expected to be derived.
5.2. Flow Network Based Method. In this paper we propose to optimally solve the maximum weighted triple matching problem by using a flow network based method.

Flow network is a mathematical tool of graph theory [9], used to solve the problem of how to efficiently carry things from the source to the sink through routes between them. In specific, a flow over the routes in the flow network can be imagined as a water flow over the water pipes in a water network [10]. Let \( G_f = (V, E) \) denote the flow network graph, where \( V \) is the node set and \( E \) is the edge set. As the physical meaning of the flow implies, there are three attributes for each edge \( e(u, v) \in E \), which are the flow \( f(u, v) \), the capacity \( a(u, v) \), and the cost \( c(u, v) \). The amount of flow \( f(u, v) \) is the amount of things passing through \( e(u, v) \), and the capacity \( a(u, v) \) is the maximum flow that \( e(u, v) \) can bear. If there is one unit of flow passing through \( e(u, v) \), then it will cost the network \( c(u, v) \). Generally, the minimum cost maximum flow of the flow network \( G_f \) is devised, the physical meaning of which is how to carry the most/maximum things with the least/minimum cost. Classical flow network algorithms, such as the push-relabel algorithms [9], can be resorted to optimally compute the minimum cost maximum flow of the flow network graph \( G_f \).

Next we detail how to transform a maximum weighted triple matching problem to a minimum cost maximum flow problem. The flow network graph \( G_f \) is constructed based on the triple matching graph \( G_{3m} \), as shown in Figure 4. The pseudocode of flow network based traffic scheduling algorithm in given in Algorithm 1. In the following, when we say \( u_k, v_n, w_m \), we refer to any node in \( U \), \( V \), and \( W \), respectively. The details of Algorithm 1 are as follows.

5.2.1. Construction of \( G_f \). To construct the flow network graph \( G_f \) based on the triple matching graph \( G_{3m} \), a virtual source node \( s \) and a virtual sink node \( t \) are firstly added. Then, additional edges are added in order to enable the flows to be injected into and flowed out of the flow network graph \( G_f \). In other words, for each \( u_k \in U \) and for each \( w_m \in W \), an edge \( e(s, u_k) \) and an edge \( e(w_m, t) \) are added. Secondly, to formulate the finiteness of the traffic capacity of vehicle \( v_n \), that is, \( p_n \), we make a copy of \( v_n \), that is, \( v'_n \). An edge \( e(v_n, v'_n) \) is added, and the finiteness of \( p_n \) is assigned as the capacity of edge \( e(v_n, v'_n) \).

Note that the constraints in (7)–(10) have not all been indicated in \( G_f \), like the quantity of the traffics and the weight of the edges. In the flow network \( G_f \), these constraints can be assigned as the attributes of the edges.

5.2.2. Edge Attributes Assignment. In the flow network graph \( G_f \), the physical meaning of the flow of the edges is the number of the data bundles. For example, the flow \( f(v'_n, w_m) \) over the edge \( e(v'_n, w_m) \) means that the vehicle \( v_n \) takes \( f(v'_n, w_m) \) data bundles of \( w_m \). The capacity of the edges defines the upper bound of the flows over the edges. The cost of the edges defines the value of cost when the flow over the edge is 1. Recall that the objective of the proposed problem is to maximize the total gains of the RSU. Thus, in order to apply the conventional push-relabel algorithms [9], we assign the cost of the edges as the minus gains. The details are illustrated as follows.

5.2.3. Converting Flow to Traffic Scheduling Strategy. Finally, we present the details of how to convert a minimum flow to a traffic scheduling strategy.

Firstly, the timeslots allocated to each vehicle are determined with the information of the minimum cost maximum flow. For each edge \( e(u_k, v_n) \in E_{UV} \), if \( f(u_k, v_n) = 1 \) then the timeslot \( u_k \) is allocated to vehicle \( v_n \). Thus all of the timeslots which are allocated to vehicle \( v_n \) can be found by checking all of the edges \( e(u_k, v_n) \in E_{UV} \). Let \( U_n \) denote the set of timeslots allocated to vehicle \( v_n \).

Then the traffics to be delivered to each vehicle are determined. For each edge \( e(v'_n, w_m) \in E_{UV'} \), if \( f(v'_n, w_m) > 0 \) then \( f(v'_n, w_m) \) bundles of \( w_m \) will be delivered to vehicle \( v_n \). Thus all of the traffics which will be delivered to vehicle \( v_n \) can be found by checking all of the edges \( e(v'_n, w_m) \in E_{UV'} \). Let \( W_n \) denote the set of traffics to be delivered to vehicle \( v_n \).

Therefore, the traffic scheduling strategy can be represented as \( \mathbb{U} = \{(U_1, W_1), (U_2, W_2), \ldots, (U_N, W_N)\} \).

5.3. Discussion and Remarks. The equivalence of a minimum cost maximum flow with an optimal solution of the maximum weighted triple matching problem can be resulted from the properties of the flow. A flow in \( G_f \) is a real-valued function \( f : \mathbb{V}_f \times \mathbb{V}_f \rightarrow R \) that satisfies the following three properties [9].
Input: The set of vehicles $V$, the set of traffics $W$ and the set of timeslots $U$

Output: The traffic scheduling strategy $U$

(1) Construct the triple matching graph $G_{im} = \{V_{im}, E_{im}\}$, where $V_{im} = U \cup V \cup W$ and $E_{im} = E_{IV} \cup E_{IVW} \cup E_{IVW} = \{(u, v_\mu) | u_\mu \in U, v_\mu \in V, if~at~timeslot~u_\mu, v_\mu~can~communicate~with~S\}$, and $E_{IVW} = \{e(v_\mu, w_\nu) | v_\mu \in V, w_\nu \in W if v_\mu~can~carry~w_\nu\}$.

(2) Construct the flow network graph $G_f = \{V_f, E_f\}$. The node set $V_f = U \cup V \cup V' \cup W \cup \{s, t\}$, where $V' = V$ and $s$ is the virtual source node and $t$ is the virtual sink node of the flow network graph. The edge set $E_f = E_{IV} \cup E_{IVW} \cup E_{IVW} \cup \{e(s, u_\mu) | u_\mu \in U \} \cup \{e(w_\nu, t) | w_\nu \in W\}$, where $E_{IVW} = \{e(u_\mu, v'_{\mu}) | v_\mu \in V\}$, and $E_{IVW} = \{e(u_\mu, w_\nu) | v_\mu \in V', w_\nu \in W if v_\mu~can~carry~w_\nu\}$.

(3) For each edge in $E_f$, assign the flow, the capacity and the cost.

(4) for each each edge in $E_f$ do

(5) set the initial flow as 0.

(6) for each $e(s, u_\mu) \in \{e(s, u_\mu) | u_\mu \in U\}$ do

(7) set $a(s, u_\mu) = 1$, and $c(s, u_\mu) = 0$.

(8) for each $e(u_\mu, v_\mu) \in E_{IVW}$ do

(9) set $a(u_\mu, v_\mu) = 1$, and $c(u_\mu, v_\mu) = 0$.

(10) for each $e(v'_\mu, u_\mu) \in E_{IVW}$ do

(11) set $a(v'_\mu, u_\mu) = p_\mu$, and $c(v'_\mu, u_\mu) = 0$, where $p_\mu$ is the buffer capacity of vehicle $u_\mu$.

(12) for each $e(v'_\mu, w_\nu) \in E_{IVW}$ do

(13) set $a(v'_\mu, w_\nu) = c_\nu$, and $c(v'_\mu, w_\nu) = 0$.

(14) for each $e(w_\nu, t) \in \{e(w_\nu, t) | w_\nu \in W\}$ do

(15) set $a(w_\nu, t) = q_\nu$, and $c(w_\nu, t) = -r_\nu$, where $q_\nu$ is the amount of traffic $w_\nu$, and $r_\nu$ is the gain obtained by the RSU $S$ when one bundle of $w_\nu$ is delivered to a vehicle.

(16) Apply the Push-relabel Algorithms in [9] on the flow network graph $G_f$ to compute the minimum cost maximum flow.

(17) Convert the minimum cost maximum flow to the traffic scheduling strategy $U$.

Algorithm 1: Flow network based traffic scheduling.

(i) Capacity constraint: for all $u, v \in V_f$, we require $f(u, v) \leq c(u, v)$.

(ii) Skew symmetry: for all $u, v \in V_f$, we require $f(u, v) = -f(v, u)$.

(iii) Flow conservation: for all $u \in V_f - \{s, t\}$, we require $\sum_{v \in V_f} f(u, v) = 0$.

Therefore, with the capacity constraint property of the flow, the constraints in (7)-(9) are perfectly mapped to the flow network. With the properties of the skew symmetry and the flow conservation, the constraint (10) is perfectly mapped to the flow network, where the total number of timeslots assigned to vehicle $v_\mu$ will always equal the number of bundles to be delivered to vehicle $v_\mu$.

Since the computed minimum cost maximum flow maximizes the total number of timeslots in the flows and minimizes the total cost simultaneously, thus the total gains of the RSU can be maximized (recall that the minus gain is mapped as the cost of the flow). Therefore, from Algorithm 1, it is clear that the minimum cost maximum flow of the constructed flow network graph is exactly the optimal solution of the maximum weighted triple matching problem. Therefore the maximum triple matching problem is transformed as a minimum cost maximum flow problem.

We remark that the optimal traffic scheduling strategy for the maximum weighted triple matching problem may not be only one. Since there may exist more than one minimum cost maximum flow of $G_f$, the optimal traffic scheduling strategy may be more than one. It can also be learned from another way. In the maximum weighted triple matching problem, since the first matching is only concerned about how much transmission slots are assigned to each vehicle and the second matching is only concerned about how much traffic are assigned to each vehicle, not the form of these matchings, thus even for the same amount matching there may exist several types.

5.4. An Example for Comparison. Figure 5 shows a comparison example between the priority greedy based method and the flow network based method.

Firstly, Figure 5(a) presents the original triple matching problem. In the original triple matching graph, there are 3 vehicles, 10 timeslots, and 3 kinds of traffic. The vehicles’ delivery capability for each type of data are given as $D_1 = \{1, 1, 0\}$, $D_2 = \{0, 1, 1\}$, and $D_3 = \{1, 0, 1\}$. The buffer capacity of the vehicles is given as $p_1 = p_2 = p_3 = 10$. The slotted contact duration of $v_1$ is $C_1 = \{u_1, \ldots, u_6\}$, and the slotted contact duration of $v_2$ is $C_2 = \{u_2, \ldots, u_6\}$, and the slotted contact duration of $v_3$ is $C_3 = \{u_6, \ldots, u_{10}\}$. The total timeslot set is given as

$$U = \{u_1, \ldots, u_{10}\}.$$ \hspace{1cm} (11)

The traffic set is given as

$$W = \{w_1 (5, 5), w_2 (4, 9), w_3 (4, 10)\}.$$ \hspace{1cm} (12)

Note that there exists overlaps among the slotted contact durations. That is, from timeslots $u_3$ to $u_6$ both $v_1$ and $v_2$
The transferred traffic is given as \( p_1 = p_2 = p_3 = 10 \). The slotted contact durations are \( C_1 = \{u_1, \ldots, u_k\}, C_2 = \{u_1, \ldots, u_k\}, \) and \( C_3 = \{u_6, \ldots, u_{10}\} \). The traffic set is given as \( W = \{w_1(5,5), w_2(4,9), w_3(4,10)\} \). In the priority greedy based method, the total gains obtained by the RSU \( S \) are 82, and the total bundle of the transmitted traffic is 10. In the flow network based method, the total gains obtained by the RSU \( S \) are 86, and the total unit of the transferred traffic is 10.

Firstly, the complexity of the priority greedy based method is analysed. The core step of this method is to find the maximum valued vehicles-traffic (\( v_m-w_n \)) pair in each timeslot. However, in the worst case there are at most \( NM \) pairs, and the computational complexity of finding the largest one is given as \( \Theta(NM) \). Recall that there are \( K \) timeslots in total; thus in the worst case the complexity of the priority greedy based method is \( \Theta(KMN) \).

Next, the complexity of the flow network based method is analysed. The computational complexity of the flow network base method depends on the computational complexity of the push-relabel algorithm. The computation complexity of push-relabel algorithm is \( \Theta(|V|^2|E|) \), where \(|V|\) is the total number of the nodes and \(|E|\) is the total number of the edges of the flow network graph. In our constructed flow network graph \( G_f \), there are \(|V| = K + 2N + M + 2 \) nodes and there are \(|E| = K + KN + N + NM + M \) edges in the worst case. Thus the complexity of the proposed flow network based method is \( \Theta(\max\{K^3, M^3, N^3\}) \).

Normally, the number of vehicles \( N \) and the types of traffic \( M \) are finite numbers, and thus the computation complexity depends on \( K \), that is, the total number of the timeslots (which is inverse to the length of the timeslot \( \delta \)). In this case, the computation complexity of the priority greedy based method is \( \Theta(K) \) while the proposed flow network based method is \( \Theta(K^3) \). Recall that \( K \) is the total number of slots in period \( [t_1, t_1 + L] \); thus the complexity analysis implies that to reduce the computation complexity of the problem the total number of timeslots should be limited.

### 6. Online Traffic Scheduling

In both Sections 4 and 5, the complete information of the \( N \) vehicles (including the information of the contact durations) and the information of the traffic \( W \) are known by the RSU. This can be referred to as the offline version of the proposed traffic scheduling problem. However, in the real environment the vehicles arrive randomly one by one, which is unknown and cannot be predicted by the RSU. Therefore, it is necessary to design the online traffic scheduling strategy for the RSU.
In this section, the online version of the traffic scheduling algorithm is given. The basic idea of the online version of the traffic scheduling algorithm is also applying Algorithm 1 with the following modifications.

At the time instant $t_1 = 0$ (or the timeslot $u_1 = 0$), when the first vehicle $v_1$ enters the \( R_{\text{BAD}} \) transmission range of the RSU, a traffic scheduling strategy is computed with Algorithm 1 for vehicle $v_1$. The node set is defined as $V' = \{ v_1 \}$, and the timeslot set is defined as $U' = \{ u_1, u_2, \ldots, u_K \}$, where $K' = \lfloor 2R_{\text{BAD}}/s \rfloor$. And the traffic set of the RSU is that the traffic currently stored in the RSU is denoted by $W'$. The other details are the same as shown in Algorithm 1.

Suppose that, at the time instant $t_n$ (or the timeslot $u_k$), a new arriving vehicle $v_n$, $1 \leq n \leq N$ enters the BAD area of the RSU, which has finished the registering and negotiating processes with the RSU before $t_n$. At timeslot $u_k$, suppose there also exist $N', 0 \leq N' \leq N - 1$ vehicles in the BAD area of the RSU, which are being served by the RSU. No matter whether the existing traffic scheduling strategy is finished or not at timeslot $u_k$, a new traffic scheduling strategy will be computed with Algorithm 1. The node set is defined as $V' = \{ v_{n-N'}, v_{N'-N'+1}, \ldots, v_n \}$. Let $u_k+K' = K \leq K-1$ be the timeslot when the last vehicle in $V'$ leaves the BAD area of the RSU; note that the last vehicle may not be $v_n$ due to different velocities of the vehicles. Then the timeslot set can be defined as $U' = \{ u_k, u_{k+1}, \ldots, u_{k+K'} \}$. The other details are done as the same as shown in Algorithm 1. At timeslot $u_k$, the RSU updates itself to the new traffic scheduling strategy. The online version will continue running until there are no arriving vehicles.

7. Performance Evaluation

In this paper, we aim at developing the CoS support traffic scheduling algorithms. Particularly, the direct delivery routing scheme is employed, and the vehicle registering and negotiating processes are omitted. Therefore, a customized C++ simulator is developed to evaluate the performance of the proposed flow network based algorithm.

Note that as forementioned in the related work, this paper is the first work to address the proposed traffic scheduling problem in the open literature. And thus, in our simulator, the flow network based method, the priority greedy based method, the FIFO (first-in first-out) based scheduling method, the random-max based scheduling method, and the bi-random based scheduling method are implemented. They are labelled as “FlowNet,” “PritGred,” “FifoMax,” “RndMax,” and “BiRnd” in short, respectively. The basic ideas of the FIFO based method (FifoMax), the random-max based method (RndMax), and the bi-random based method (BiRnd) are illustrated as follows.

(i) \textit{FifoMax}. The basic idea is that the RSU delivers the bundles to the first arriving vehicle in a priority first ordering, until the buffer of that vehicle is full or the bundles in the RSU are empty.

(ii) \textit{RndMax}. The basic idea is that at each timeslot the RSU randomly selects a vehicle (if any). Then the bundle is delivered in a priority first ordering.

(iii) \textit{BiRnd}. The basic idea is that at each timeslot the RSU randomly selects a vehicle (if any). Then a bundle (if any) which that vehicle can carry is randomly selected. Finally, the selected bundle is delivered to the selected vehicle.

Note that these three methods are usually used as benchmarks in the existing literatures, and in this paper we compare the simulation results of the proposed algorithms with them.

Next, we initially analyse the relevance relationship between the simulation parameters, based on which we set the parameters in our simulation. We analyse the relevance relationships between the parameters from the aspects of conditions of overlapped contact durations and conditions of RSU buffer overflow avoidance. Particularly, we analyse the conditions of RSU’s buffer overflow avoidance from two aspects, that is, the viewpoint of the timeslot and the viewpoint of the vehicle.

7.1. Relevance Relationship Analysis between Parameters

7.1.1. Conditions of Overlapped Contact Durations, That Is, \( \mu>s/2R_{\text{BAD}} \). Recall that $1/\mu$ is the expectation of the interarrival time between two consecutive vehicles, $s$ is the constant speed of the vehicles, and $R_{\text{BAD}}$ is the transmission radius of the BAD area of the RSU.

Note that when the vehicles’ arrival density $\mu$ is very low where there may exist only one vehicle located in the BAD transmission area of the RSU, then the optimal traffic scheduling strategy is to deliver the bundles in a priority first ordering. Therefore, in this case, the FlowNet, PritGred, FifoMax, and RndMax methods all can achieve the optimal performance. However, when the vehicles’ arrival density is not very low, there may exist overlaps among the contact durations of the vehicles. In this case, how will our proposed algorithms perform needs to be evaluated.

Then, in the following we derive the conditions when the contact duration may be overlapped. Note that the interarrival time between two consecutive vehicles is i.i.d. exponential random variables, and in average the interarrival time is $1/\mu$, and the length of the contact duration is $2 \times R_{\text{BAD}}/s$. Then if the length of the contact duration is less than the interarrival time, the contact durations of the vehicles will be overlapped in average. Therefore, if $1/\mu < (2 \times R_{\text{BAD}})/s$, that is, $\mu > s/2R_{\text{BAD}}$, the contact durations between two consecutive vehicles will be overlapped in average. This means that the RSU has more than one selection of vehicles to transmit bundles. Therefore, in the following simulation settings, we will set

$$
\mu > \frac{s}{2R_{\text{BAD}}}. \quad (13)
$$

7.1.2. Conditions of RSU’s Buffer Overflow Avoidance, That Is, \( \delta>\sum_{m=1}^{M} \lambda_m \) and \( p_0>(2R_{\text{BAD}}/s) \times \sum_{m=1}^{M} \lambda_m \). Recall that $\delta$ is the length of a timeslot and $p_0$ is the buffer capacity of vehicle $v_n$. Let the $mth$ bundles arrival process be a Poisson process with parameter of $\lambda_m$. Then the density of all of the $M$ types
of bundles arrival process is $\sum_{m=1}^{M} \lambda_m$. The interarrival time of any two consecutive bundles is $1/\sum_{m=1}^{M} \lambda_m$.

From the viewpoint of the timeslot, since in one timeslot only one bundle can be delivered from the RSU to the vehicles, therefore when the interarrival time between the bundles is less than $\delta$, then the buffer of the RSU will be overflowed. In other words, if $1/\sum_{m=1}^{M} \lambda_m < \delta$, then the buffer of the RSU will be overflowed. Therefore, to avoid the RSU’s buffer overflow we will set

$$\delta > \frac{1}{\sum_{m=1}^{M} \lambda_m}. \quad (14)$$

From the viewpoint of the vehicle, if the number of the arrived bundles during the contact duration $\nu_n$ is larger than $p_n$, then the bundles will be stored in the buffer of the RSU which also will make the buffer of the RSU overflowed. The number of bundles arriving during the contact duration of $\nu_n$ is $(2R_{BAD}/s) \times \sum_{m=1}^{M} \lambda_m$ in average, where recall $2R_{BAD}/s$ is the length of the contact duration and $\sum_{m=1}^{M} \lambda_m$ is the density of the bundles. Therefore, to avoid the RSU’s buffer overflow we will set

$$p_n > \frac{2R_{BAD}}{s} \times \sum_{m=1}^{M} \lambda_m. \quad (15)$$

Therefore, with some mathematical manipulations on (13) and (15), we have

$$\frac{\sum_{m=1}^{M} \lambda_m}{p_n} < \frac{s}{2R_{BAD}} < \mu. \quad (16)$$

By combining (16) with (14), then the relevance relationships between the simulation parameters can be represented as

$$\frac{1}{\delta} < \sum_{m=1}^{M} \lambda_m < \frac{sp_n}{2R_{BAD}} < \mu p_n. \quad (17)$$

Figure 6 shows the relevance relationship of the simulation parameters given in (17). Since $R_{BAD}$ and $\delta$ are constant which are determined by the maximum transmission power of the RSU, the length of the bundle packet, and the transmission rate, we investigate the performance of the proposed algorithm mainly by changing the values of $\sum_{m=1}^{M} \lambda_m$, $s$, and $\mu$. From Figure 6, it can be seen that, to evaluate the performance of the proposed algorithm, three intervals should be investigated, that is, changing $\sum_{m=1}^{M} \lambda_m$ in $(1/\delta, sp_n/2R_{BAD})$, changing $s$ in $(\sum_{m=1}^{M} \lambda_m, \mu p_n)$, and changing $\mu$ in $(sp_n/2R_{BAD}, \infty)$.

7.2. Performance Metrics. The most interesting performance metrics are the number of total obtained gains and the number of total transmitted bundles.

Recall that the objective of the proposed problem is to maximize the total obtained gains of the RSU, as shown in (6). It can be seen that it depends on $r_m$ and $h(v_n, w_m)$. In other words, it not only depends on the gains of the $m$th type of bundles, but also depends on the total number of the $m$th bundles carried by the vehicles. The former one, $\{r_1, \ldots, r_M\}$, usually is assigned as the input of the traffic scheduling algorithms, for example, the input of the proposed algorithm including $W$ as shown in Algorithm 1. The latter one is determined by the traffic scheduling algorithms and will be evaluated through extensive simulations.

Therefore, the metric of the number of total obtained gains can be defined as

$$\sum_{n=1}^{N} \sum_{m=1}^{M} r_m h(v_n, w_m). \quad (18)$$

And the metric of the number of total transmitted bundles can be defined as

$$\sum_{n=1}^{N} \sum_{m=1}^{M} h(v_n, w_m). \quad (19)$$

7.3. Computer Simulations. In the following, we will evaluate the impacts of the arrival density of the $m$th type bundles $\lambda_m$, the density of vehicles $\mu$, and the speed of the vehicles $s$ on the number of total obtained gains and the number of total transmitted bundles.

The simulation parameters are set as follows. The length of a timeslot $\delta$ is assumed to be constant and $\delta = 0.05$ s. The radius of the BAD area of the RSU $R_{BAD}$ is assumed to be constant and $R_{BAD} = 100$ m. The speed of the vehicles $s$ is assumed constant and uniformly distributed in $[0.8 s, 1.2 s]$, and the average value is set as $s = 30$ m/s. The buffer capacity of each vehicle $p_n$ is assumed to be constant and uniformly distributed in $[0.8 p_n, 1.2 p_n]$, and the average value is set as $p_n = 200$. The vehicles arrive at the RSU in a Poisson process with density $\mu$, and the classical value of $\mu$ is set as $\mu = 0.175$. All of the $M$ types of traffic are assumed arriving at the RSU in a Poisson process with the same density $\lambda_m$, and the classical value of $\lambda_m$ is set as $\lambda_m = 8$. Assume $M = 5$ and each type of the traffic can be carried by any vehicle with a probability of 0.5; that is, for each vehicle $D_n = \{0.5, 0.5, 0.5, 0.5, 0.5\}$. The gains of each type of traffic are assumed to be constant and are set as $r_1 = 20$, $r_2 = 18$, $r_3 = 16$, $r_4 = 14$, $r_5 = 12$.

We note that the values of $D_n$ and $r_m$ may directly impact the value of the total number of the gains obtained by the RSU, that is, $\sum_{n=1}^{N} \sum_{m=1}^{M} r_m h(v_n, w_m)$. However, it does not affect the performance comparisons among the traffic scheduling algorithms. Therefore, we set both $D_n$ and $r_m$ as constants.

Based on the relevant relationship analysis of the simulation parameters, we evaluate the behaviour of the traffic scheduling algorithms in terms of by changing $s$ from 20
to 40 with a step of 2, by changing \( \mu \) from 0.125 to 0.225 with a step of 0.01, and by changing \( \lambda_m \) from 4 to 12 with a step of 1. When one parameter is changed, the others are set as the classical values. And, for each setting of parameters, 30 network scenarios are generated and the average value is taken as the simulation result. The confidence interval of the simulation results is given with confidence level of 95%. The simulation results are validated through experiments.

7.3.1. Impacts of \( \lambda_m \). Firstly, the impacts of \( \lambda_m \) are evaluated and given in Figures 7 and 8. From Figure 7, it can be seen that with the increase of \( \lambda_m \) the total number of the transmitted bundles firstly increases and then almost remains the same when \( \lambda_m > 8 \). This is mainly due to that when \( \lambda_m > 8 \) almost all of the timeslots are exploited by each traffic scheduling algorithm. Recall that \( \delta = 0.05 \) and \( D_n = [0.5 \ 0.5 \ 0.5 \ 0.5 \ 0.5] \); thus when \( \lambda_m = 8 \), we have \( 1/\delta = 0.5 \sum_{n=1}^{5} \lambda_m = 20 \), and when \( \lambda_m > 8 \), we have \( 1/\delta < 0.5 \sum_{n=1}^{5} \lambda_m \), which is the condition of RSU buffer overflow avoidance, as shown in (14). Among all of the traffic scheduling algorithms, FlowNet outperforms the others. The key behind this is that when computing the traffic scheduling strategies FlowNet takes all of the information of the vehicles and the traffics into account. However, the other ones, like the PritGred, FifoMax, and RndMax, do not consider all of the information, only explore the traffic information at a different level. Among all of the traffic scheduling algorithms, RndMax is the worst one.

However, as shown in Figure 8 BiRnd becomes the worst one instead of RndMax. This is because RndMax always delivers the bundles to randomly selected vehicles in a priority first ordering, and thus the obtained gains are larger. Among all of the traffic scheduling algorithms, FlowNet outperforms the others, and PritGred is the second best one.

FlowNet outperforms BiRnd from at least 9.8% to at most 20.9% and, in average 18.4%, and outperforms PritGred from at least 2.3% to at most 3.1% and in average 2.5%.

7.3.2. Impacts of \( \mu \). Secondly, the impacts of \( \mu \) on the performance metrics are evaluated and given in Figures 9 and 10. From Figure 9, it can be seen that with the increase of \( \mu \) the total number of transmitted bundles increases. This is mainly because with the increase of \( \mu \), more vehicles arrive at the RSU. Therefore, the number of the contact durations
increases and thus the total number of transmitted bundles increases. Among all of the traffic scheduling algorithms, FlowNet outperforms the others as expected and RndMax is the worst one. The other three algorithms are almost the same.

However, from Figure 10 it can be seen that BiRnd is the worst one instead of RndMax. This is mainly due to the fact that RndMax can support traffic differentiation and thus obtains more gains since it always delivers the bundles to randomly selected vehicles in a priority first ordering. Among all of the traffic scheduling algorithms, FlowNet outperforms the others as expected. And the second best one is shown as the PritGred. This is mainly because at each timeslot PritGred always delivers the most gained bundles. In other words, PritGred has the second best CoS support capability among the other four (except for FlowNet). FlowNet outperforms BiRnd by almost 19.5% in average and outperforms PritGred by almost 2.5% in average.

7.3.3. Impacts of $s$. Finally, the impacts of $s$ on the performance metrics are evaluated and given in Figures 11 and 12. From Figure 11, it can be seen that, with the increase of $s$, the total number of transmitted bundles decreases. This is because with the increase of $s$ the length of the contact durations decreases. Thus, less bundles can be taken by the vehicles. Among all of the traffic scheduling algorithms, FlowNet outperforms the others and RndMax is also the worst one. The other three algorithms are almost the same.

Similar to the behaviour of changing $\mu$, BiRnd obtains the least gains instead of RndMax as shown in Figure 12. Among all of the traffic scheduling algorithms, FlowNet outperforms the others and PritGred is the second best one. FlowNet outperforms BiRnd by almost 19.4% in average and outperforms PritGred by almost 2.6% in average.

8. Conclusion

In the sparse or unpopulated area, one type of VDTNs including the vehicular infrastructures or roadside units (RSUs) and mobile vehicles is deployed. The unconnected RSUs of the VDTNs are used as the gateways of the distributed sensor networks, and the mobile vehicles are used as the mobile relays of the data. The sensed data are collected by the RSU and delivered to the passing by vehicles. Then the vehicles store, carry, and forward them to the destinations. In general, more than one type of the sensed data are collected by
the RSU, and different classes of service (CoS) support are desired.

In this paper, we consider the CoS traffic scheduling problem for the RSU, with the objective of maximizing the total obtained gains. The availability of the contact information between the vehicles and the RSU is analysed. By exploring the contact durations, the proposed problem is formulated as a maximum weighted triple matching problem and is optimally solved with a flow network based method. Both the offline version and the online version of the proposed algorithms are developed. Extensive simulations are conducted to evaluate the performance of the proposed algorithms. Simulation results show the efficiency and effectiveness of the proposed algorithms.

In the future, the analytical results of the proposed traffic scheduling problem can be derived to deeply understand the behaviour of the CoS traffic scheduling strategy. And the fairness issue for the different CoS traffic scheduling will be considered.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundations of China (Grant nos. 61201157 and 61271279), the National 863 plans project (Grant no. 2014AAA01A707), the National Science and Technology Major Project (Grant no. 2015ZX03002006), and the Fundamental Research Foundation of NPU (Grant no. JCY20130131).

References

[1] A. McMahon and S. Farrell, "Delay-and disruption-tolerant networking," IEEE Internet Computing, vol. 13, no. 6, pp. 82–87, 2009.
[2] P. R. Pereira, A. Casaca, J. J. P. C. Rodrigues, V. N. G. J. Soares, J. Triay, and C. Cervelló-Pastor, "From delay-tolerant networks to vehicular-delay-tolerant networks," IEEE Communications Surveys and Tutorials, vol. 14, no. 4, pp. 1166–1182, 2012.
[3] J. N. G. Isento, J. J. P. C. Rodrigues, J. A. F. F. Dias, M. C. G. Paula, and A. Vinel, "Vehicular delay-tolerant networks? A novel solution for vehicular communications," IEEE Intelligent Transportation Systems Magazine, vol. 5, no. 4, pp. 10–19, 2013.
[4] P. Jiang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design, tradeoffs and early experiences with ZebraNet," in Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 96–107, October 2002.
[5] H. Ochiai, H. Ishizuka, Y. Kawakami, and H. Esaki, "A DTN-based sensor data gathering for agricultural applications," IEEE Sensors Journal, vol. 11, no. 11, pp. 2861–2868, 2011.
[6] L. Selavo, A. Wood, Q. Cao et al., "LUSTER: wireless sensor network for environmental research," in Proceedings of the 5th ACM International Conference on Embedded Networked Sensor Systems (SenSys ’07), pp. 103–116, November 2007.
[7] O. Truellols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, “Cooperative download in vehicular environments,” IEEE Transactions on Mobile Computing, vol. 11, no. 4, pp. 663–678, 2012.
[8] V. N. G. J. Soares, F. Farahmand, and J. J. P. C. Rodrigues, “Scheduling and drop policies for traffic differentiation on vehicular delay-tolerant networks,” in Proceedings of the 17th International Conference on Software, Telecommunications and Computer Networks (SoftCOM ’09), pp. 353–357, IEEE, September 2009.
[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, MIT Press, 2nd edition, 2001.
[10] Z. Yan, H. Jiang, Z. Shen, Y. Chang, and L. Huang, “k-connectivity analysis of one-dimensional linear VANETs,” IEEE Transactions on Vehicular Technology, vol. 61, no. 1, pp. 426–433, 2012.
[11] A. A. Hammad, G. H. Badawy, T. D. Todd, A. A. Sayegh, and D. Zhao, “Traffic scheduling for energy sustainable vehicular infrastructure,” in Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM ’10), December 2010.
[12] A. A. Hammad, T. D. Todd, G. Karakostas, and D. Zhao, “Downlink traffic scheduling in green vehicular roadside infrastructure,” IEEE Transactions on Vehicular Technology, vol. 62, no. 3, pp. 1289–1302, 2013.
[13] Z. Yan, B. Li, X. Zuo, and T. Gao, “Fair downlink traffic scheduling for energy sustainable vehicular roadside infrastructure,” in Proceedings of the International Conference on Connected Vehicles & Expo (ICCVE ’14), pp. 1–6, Vienna, Austria, November 2014.
[14] V. N. G. J. Soares, F. Farahmand, and J. J. P. C. Rodrigues, “A layered architecture for vehicular delay-tolerant networks,” in Proceedings of the IEEE Symposium on Computers and Communications (ISCC ’09), pp. 122–127, Sousse, Tunisia, July 2009.
[15] V. N. G. J. Soares, J. J. P. C. Rodrigues, F. Farahmand, and M. Denko, “Exploiting node localization for performance improvement of vehicular delay-tolerant networks,” in Proceedings of the IEEE International Conference on Communications (ICC ’10), pp. 1–5, IEEE, May 2010.
[16] IEEE 802.11p-2010, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments,” IEEE Standards Association, 2010.
[17] V. N. G. J. Soares, F. Farahmand, and J. J. P. C. Rodrigues, “Performance analysis of scheduling and dropping policies in vehicular delay-tolerant networks,” International Journal on Advances in Internet Technology, vol. 3, no. 1-2, pp. 137–145, 2010.
[18] A. Vahdat and D. Becker, “Epidemic routing for partially connected ad hoc networks,” Tech. Rep., Duke University, 2000.
[19] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and wait: an efficient routing scheme for intermittently connected mobile networks,” in Proceedings of the ACM Workshops: Conference on Computer Communications (SIGCOMM ’05), pp. 252–259, Philadelphia, Pa, USA, August 2005.
[20] V. N. G. J. Soares, J. J. P. C. Rodrigues, P. S. Ferreira, and A. Nogueira, “Improvement of messages delivery time on vehicular delay-tolerant networks,” in Proceedings of the 8th International Conference on Parallel Processing (ICPP ’09) Workshops—The 2nd International Workshop on Next Generation of Wireless and Mobile Networks (NGWMN ’09), Vienna, Austria, September 2009.
[21] J. A. Dias, J. N. Isento, V. N. G. J. Soares, and J. J. P. C. Rodrigues, “Impact of scheduling and dropping policies on the performance of vehicular delay-tolerant networks,” in Proceedings of
[22] V. N. G. J. Soares, F. Farahmand, and J. J. P. C. Rodrigues, “Traffic differentiation support in vehicular delay-tolerant networks,” *Telecommunication Systems*, vol. 48, no. 1-2, pp. 151–162, 2011.

[23] V. Ramaiyan, E. Altman, and A. Kumar, “Delay optimal scheduling in a two-hop vehicular relay network,” *Mobile Networks and Applications*, vol. 15, no. 1, pp. 97–111, 2010.

[24] M. J. Khabbaz, W. F. Fawaz, and C. M. Assi, “Probabilistic bundle relaying schemes in two-hop vehicular delay tolerant networks,” *IEEE Communications Letters*, vol. 15, no. 3, pp. 281–283, 2011.

[25] Z. Yan, Z. Zhang, H. Jiang, Z. Shen, and Y. Chang, “Optimal traffic scheduling in vehicular delay tolerant networks,” *IEEE Communications Letters*, vol. 16, no. 1, pp. 50–53, 2012.

[26] D. Niyato, P. Wang, and J. C. M. Teo, “Performance analysis of the vehicular delay tolerant network,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’09)*, pp. 1–5, Budapest, Hungary, April 2009.

[27] Y. Zhang, J. Zhao, and G. Cao, “On scheduling vehicle-roadside data access,” in *Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks (VANET ’07)*, pp. 9–18, September 2007.

[28] N. Wisitponghan, F. Bai, P. Muralige, V. Sadekar, and O. Tonguz, “Routing in sparse vehicular ad hoc wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1538–1556, 2007.

[29] G. Chartrand and P. Zhang, *Introduction to Graph Theory*, McGraw-Hill, New York, NY, USA, 2005.