Traffic Condition Estimation Using Vehicular Crowdsensing Data

Lu Shao*, Cheng Wang*, Zhong Li†, Changjun Jiang*
* Department of Computer Science, Tongji University, Shanghai, China, 201804
† Department of Communication Engineering, Donghua University, Shanghai, China, 201620

***Abstract*—Urban traffic condition usually serves as a basic information for some intelligent urban applications, e.g., intelligent transportation system. But the acquisition of such information is often costly due to the dependency on equipments such as cameras and loop detectors. Crowdsensing can be utilized to gather vehicle-sensed data for traffic condition estimation. This way of data collection is economic. However, it has the problems of data uploading efficiency and data usage effectiveness. To deal with these problems, in this paper, we take into account the topology of the road net. We divide the road net into Road Sections and Junction Areas. Based on this division, we introduce a two-phased data collection and processing scheme named RTS (Road Topology based Scheme). It leverages the correlations among adjacent roads. In a junction area, data collected by vehicles is first processed and integrated by a sponsor vehicle. This sponsor vehicle will calculate the traffic condition locally. Both the selection of the sponsor and the calculation of the traffic condition utilize the road correlation. The sponsor then uploads the local data to a server. By employing the inherent relations among roads, the server processes data and estimates traffic condition for road sections unreached by vehicular data in a global vision. We conduct extensive experiments based on real vehicle trace data. The results indicate that, our design can commendably handle the problems of efficiency and effectiveness in the vehicular-crowdsensing-data based traffic condition evaluation.**

***Index Terms*—Crowdsensing, Vehicular networks, Traffic condition evaluation, Road topology.**

I. Introduction

Traffic condition information is very useful for works like transportation planning, road system design, traffic signal control, *etc.*. Mean velocity of a road for a given time interval is a perspective of the traffic condition information. However, the acquisition of the raw data (location, real time speed, *etc.*) for the velocity information is often costly from the financial aspect, *e.g.*, using cameras [1]–[5], dedicated sensors [6], loop detectors [7], [8]. Nowadays, vehicular crowdsensing is a popular data collection scheme, in which the data providers periodically store and incidentally provide their data rather than specially for data contributing [9]. This kind of data usually consists of position, real time velocity, and direction information of vehicles. With the development of vehicle electronic devices, the data can be easily obtained [10]. At the same time, it is a research trend to study the problem of traffic condition detection in vehicular networks (abbreviated as VN) [11]–[20].

In this paper, we integrate the paradigm of crowdsensing into the study of VN-based traffic condition detection. As above application scenario describes, usually, there is a central server to gather, store, and analyze the data collected from vehicles to detect traffic condition [21]–[23]. Any vehicles running on the roads can voluntarily offer their data about locations and corresponding speeds to join in the traffic condition detection. However, in this scenario, there are two main challenges as follows:

• Challenge 1: The *efficiency* of data transmission between vehicles and the server. It is noticeable that the data uploading process is multiple-to-one. It means that there will be a significant upstream bandwidth occupation. Furthermore, the VN architecture is not dedicated to vehicular data gathering only. Therefore, it is necessary to design a mechanism to save the bandwidth occupation, or in other words, to ensure the data uploading efficiency.

• Challenge 2: The *effectiveness* of the local and global traffic condition evaluation. We must ensure that for each road, the evaluation result can truly reflect the traffic condition of it. Furthermore, due to the incompleteness of the transmission equipment installation on all the vehicles for a city, as well as the nonuniform geological and temporal distribution of vehicles, it is hard to guarantee the coverage of VN-based crowdsensing data.

To deal with above challenges, we leverage the correlations among roads. According to the inter-relationships among roads, we first divide the road net into *Road Sections* and *Junction Areas* (as shown in Fig.2). Then, based on the road division, we propose a two-phased traffic data collection and processing scheme named **RTS** (Road Topology based Scheme). In the first phase, we design a *Sponsor-Follower* scheme to locally collect and integrate data. According to Sponsor-Follower scheme, vehicles in a same junction area choose a vehicle as a sponsor by a weighted-competing strategy. The sponsor is the vehicle who can collect data from a relatively largest range in shortest time and can transmit data to an RSU as quickly as possible. The sponsor will collect data from other vehicles and obtain the mean velocities of road sections in the junction area it belongs to. In these procedures, correlations among road sections are explicitly used. Then the sponsor transmits a packet containing the mean velocity to its nearest RSU which will finally transmit the packet to the central server. In the second phase, the server will handle the problem of geological and temporal coverage in a global vision. It recursively calculates the velocities of road sections, in which the calculation path follows the adjacent relations of road

978-1-4673-8590-9/15/$31.00 ©2015 IEEE

sections and the topology of road. By doing so, the road section correlations are used in depth again.

We conduct experiments based on the data set of $13,764$ taxis in Shenzhen, China, collected in April 2011, to verify the efficiency and effectiveness of our design. The results indicate that RTS can efficiently gather data from vehicles to server, with the wireless bandwidth occupation being drastically saved. Meanwhile, the local and global road condition is reflected effectively in our experiments.

The main contributions of this paper are listed as follows:

1) We design a two-phased vehicular data collection and process approach named RTS for urban traffic condition estimation. RTS effectively utilizes the topology of the road net.

2) In RTS, we propose a Sponsor-Follower scheme to choose the local data collector and uploader from vehicles in the same junction area. Upstream bandwidth occupation of data transmission is greatly reduced by the strategy of work division and cooperation.

3) We incorporate the inherent correlation between adjacent road sections into the study of local and global traffic condition detection. Our estimation method effectively reflects the local road condition and fulfills the coverage demand of global traffic condition estimation.

The rest of this paper is organized as follows. In Section II, we review the existing works. In Section III, we give a description of RTS from an overview perspective. In Section IV, we describe our approach of road net division. In Section V, we introduce our method for locally evaluating mean velocity for a road section. In Section VI, we give the detailed presentation of the Sponsor-Follower mechanism. In Section VII, we explain how to fulfill the coverage of global traffic condition. Then in Section VIII, we show our experiments and corresponding results. Finally, we conclude this paper and give our future works in Section IX.

II. Related Works

In this section, we review some existing works in the following four aspects:

Architectures of VN-based traffic condition evaluation. There are a number of studies that applied vehicular networks into traffic condition evaluation. Some of them used distributed architectures, in which all the procedures of data processing were committed to VANET, *e.g.*, [13], [14], [16], [24]. The main problems of the distributed architecture are the data transmission delay and the lack of the global traffic condition knowledge. There are also some works that adopted centralized architecture into design, *e.g.*, [22] [15]. In these works, vehicles communicate with the central server respectively, so the problem of bandwidth occupation rises. Combining the centralized and decentralized ways, Miller *et al.* [12] proposed a hybrid architecture. In [12], data generated by vehicles is first gathered and aggregated by a Super Vehicle locally within a certain area to decrease the size of the data to be uploaded. But the strategy for choosing the Super Vehicle in [12] was somehow excessively complex.

Methods to save bandwidth. To save the bandwidth consumed by the data collection procedure, there are different methods proposed. From an overseeing perspective, Wang *et al.* examined the capacity of wireless communications in [25]–[27], aiming to design an effective data transmission protocol. Decreasing the packet loss probability can save the bandwidth from another perspective. To design a good routing strategy, there are works that incorporated the inner social property of moving objects into the data transmission of mobile networks. For example, Li *et al.* proposed several social-community based data forwarding strategies in [28]–[30]. The other method is to reduce the size of data from its source. For example, the compressive sensing method was examined in [10], [31], [32]. Skordylis *et al.* [11] used a probabilistic strategy to determine whether a sensor should be open to sense data. What's more, the local data process is another method. For instance, in TrafficView [21], the authors skillfully used the data semantic to aggregate vehicular data. Though ensuring the recoverability of the data is beneficial to reserve more information, we think it depends on the context of the specific application scenario. Calculating multiple records to a single result is more efficient to reduce data size.

Local vehicles organization. Several strategies were proposed to organize a local vehicular data collection and processing environment. Bauza *et al.* [16] proposed a front-to-back multi-hop strategy in which the congestion information is first produced by the vehicle in front of a series of vehicles on a same road. Then this information is iterated by the following vehicles until a vehicle on the back of the queue finds that there is no congestion. And in [12], the authors broke the road network into zones, in each zone there is a vehicle designated as a Super Vehicle that is responsible for the local data gathering and aggregation. However, to the best of our knowledge, none of the existing works strictly formalized the division of road net to match up the work of local cooperation of vehicles.

Coverage filling-up strategies. Alasmary *et al.* [33] used a branch and bound approximation algorithm to select the optimal number of sensors to guarantee the coverage in vehicular crowdsensing. But the method used in [33] is a predetermined optimization method and is costly in computation. Another methodology is trying to make full use of the deficient data to meet the coverage demand. A conventional way to do this is Matrix Completion (MC) [34], [35]. But MC is sensitive to data density and shows bad on sparse data set [34]. To overcome this, Du *et al.* [34] proposed a floating car control method to minimize the estimation error of MC. However, none of the above literatures took the road correlation into consideration. Differently, Pascale *et al.* [36] divided road net into blocks and considered the traffic flow between neighboring blocks. However, they didn't notice the bidirectional characteristic of roads.

III. Overview of RTS

Before introducing our design of RTS in detail, we first describe it from the perspective of giving an overview. As

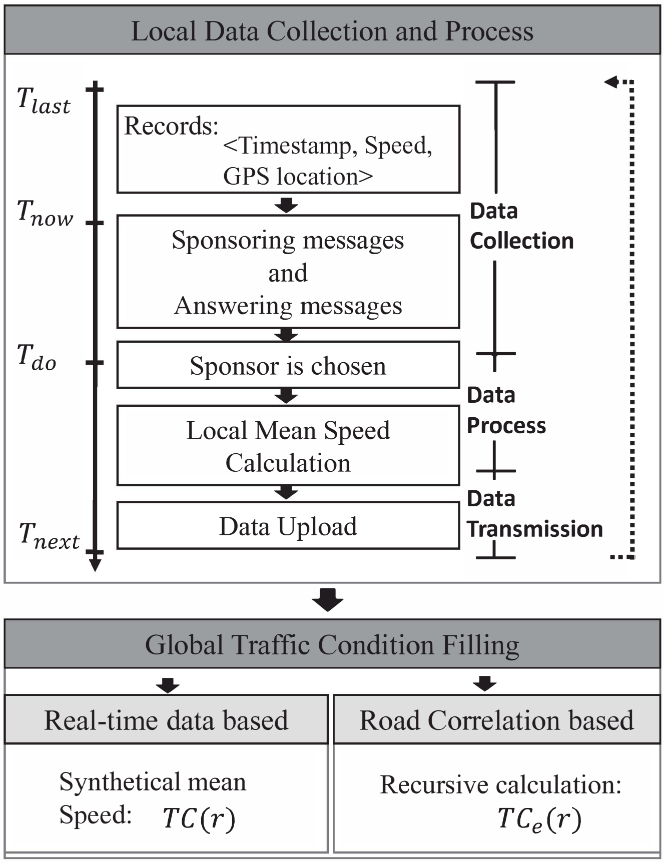

Local Data Collection and Process
T_{last}
T_{now}
T_{do}
T_{next}
Records: <Timestamp, Speed, GPS location>
Sponsoring messages and Answering messages
Sponsor is chosen
Local Mean Speed Calculation
Data Upload
Data Collection
Data Process
Data Transmission
Global Traffic Condition Filling
Real-time data based
Road Correlation based
Synthetical mean Speed: $TC(r)$
Recursive calculation: $TC_e(r)$

Fig. 1. The overview of RTS.

shown in Fig.1, RTS is designed as a two-phased scheme. The first phase is for local data collection and local traffic condition calculation. In this phase, vehicles will first collect data consisting of timestamp, speed and GPS location of them. Then the vehicles in a same junction area will participate in a procedure of competing the role of sponsor. The vehicle who successfully becomes a sponsor will be responsible for gathering data from other vehicles in this junction area. Then the sponsor will process this data to a mean speed value. This mean speed will reflect the traffic condition of this junction area in the last time interval. Next, the sponsor will upload the data to a server. The server is on the second phase of RTS. It will fill the vacancies of traffic condition values in a global vision. Depending on the availability of real-time data, the server will apply two different methods: (1) synthetical mean speed calculation based on real-time data, (2) recursive calculation based on road correlation.

IV. Road Net Formalization

In the real road net, road segments can be separated by intersections or corners. For simplicity, we call an intersection and a corner on road net as **Junction** indiscriminately. What's more, it is noteworthy that traffic flow has its directional pattern, which means for a single road segment, traffic conditions of its two opposite directions are usually different. Thus, we further split one segment of a road into two individual **Road Sections** respectively. We then define the term **Junction Area** as the set of all the road sections connected by a same junction. According to the public transportation regulations, adjacent road sections have their inherent correlations for traffic flow to go through. Fig.2 illustrates the road sections, junction area and their relations in a local area.

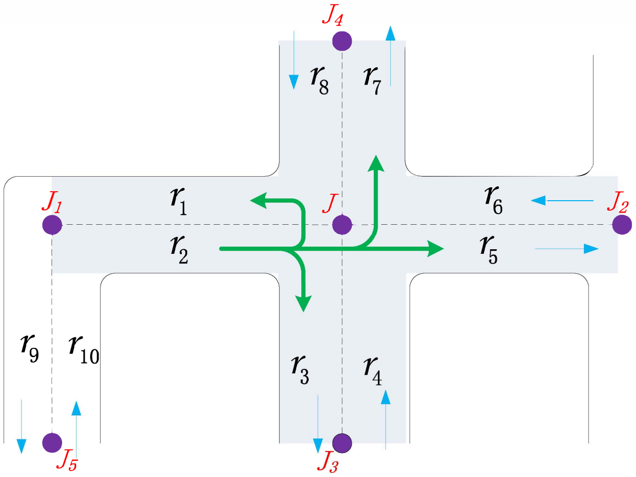


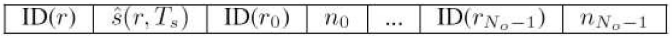
Fig. 2. The Road Sections and Junction Area. In this figure, the purple dots represent the junctions. By these junctions, this local road area is separated into 10 sections, *e.g.*, r_1 to r_{10}. The blue arrows show the driving directions, and the green arrows show the allowable traffic routes from r_2. The light-gray back-grounded area is the Junction Area defined around the junction J.

V. Local Mean Velocity Evaluation Strategy

In this section, we introduce our method for local traffic condition evaluating. It is also the pre-knowledge for our design of Sponsor-Follower mechanism in Section VI.

We take into consideration the correlation between adjacent road sections to design a local mean velocity evaluation strategy. Our main idea is that the traffic condition of a road section is influenced by its outward neighbors. If two road sections are adjacent, and vehicles can run from one of them to another complying with traffic regulation, then the former road section is an inward neighbor to the latter one, while the latter road section is an outward neighbor to the former one. For example, in Fig.2, road section r_2 has four outward neighbors: r_1, r_3, r_5, r_7. It indicates that the traffic flow of r_2 can be consumed by and only by the road sections r_1, r_3, r_5, r_7. Recall that vehicles can periodically check their locations and speeds and temporarily store this data. So if the vehicles on r_1, r_3, r_5, r_7 still have the data they generated and put into storage when they were running on r_2, we can combine the data collected from all the vehicles on r_1, r_3, r_5, r_7 and r_2 to calculate the mean velocity of r_2 for a period of past time.

Formally, assume that we now need to evaluate the traffic condition of a road section r for the past time duration T_s. r has a outward neighbor set $R_o = \{r_k | 0 \leqslant k < N_o\}$, where N_o is the number of outward neighbors of r. For each item r_k in R_o, it has a certain amount of vehicles now running on it but



ID(r)	$\hat{s}(r, T_s)$	ID(r_0)	n_0	...	ID(r_{N_o-1})	n_{N_o-1}

Fig. 3. The form of the local calculation result of road section r that to be uploaded to server via the sponsor.

went from r in the past T_s. Here, let n_k denote the number of vehicles that meet three criteria: (1) are now on r_k, (2) went from r, and (3) will offer their data.

We declare all the vehicles that are currently on r or went from r but currently on its outward neighbors as list V, and denote the number of its elements as N_{all}. We then define a sub list of V as V_Δ. A vehicle in V_Δ is a vehicle whose data contains at least two records, meeting (1) all of which belong to r or (2) at least one of them belongs to r and at least one of them belongs to its outward neighbor. Here, we only examine the data generated on r or its outward neighbors in the past T_s. We define t_v^0 as the earliest timestamp found in the records of vehicle v, and t_v^1 as the latest timestamp found in them. Also, we define P_v^0 as the corresponding location of t_v^0, and P_v^1 as the location corresponding to t_v^1. We denote the number of items in V_Δ as N_{V_Δ}. Further, we define $dist(P_1, P_2)$ as the distance between any points P_1 and P_2 along the road section(s) connecting them. So for a vehicle $v \in V_\Delta$, the mean speed from time t_v^0 to t_v^1 can be calculated as $\overline{s}_v = dist(P_v^1, P_v^0)/(t_v^1 - t_v^0)$.

Assuming that the location of the junction J that joints r and its outward neighbors is P_J, now we denote the distance that a vehicle v goes (went) on r as $D(v)$. If v is now on r, we have $D(v) = dist(P_v^0, P_v^1)$. And if v is now on a outward neighbor of r, we have $D(v) = dist(P_v^0, P_J)$.

Then, we denote the set of all the speed values whose corresponding locations are on r as $S_r = \{s_l | 0 \leqslant l < N_S\}$, where N_S is the number of those values.

Now, we define a *local calculated mean speed* of r for the past T_s as

$$\hat{s}(r, T_s) = \frac{\sum_{l=0}^{N_S-1} s_l + N_{V_\Delta} \times \sum_{p=0}^{N_{V_\Delta}-1} \left(\overline{s}_{v_p} \times \frac{D(v_p)}{\sum_{m=0}^{N_{V_\Delta}-1} D(v_m)} \right)}{N_S + N_{V_\Delta}}, \tag{1}$$

where $s_l \in S_r$, $v_p, v_m \in V_\Delta$.

The form of the local calculation result of r that should be uploaded to the server by the sponsor is shown in Fig.3. The numbers $n_0, ..., n_{N_\bullet-1}$ will be used in the second phase of RTS. We call the list of values in this form of road section r as $data(r)$.

VI. The Sponsor-Follower Mechanism

The Sponsor-Follower mechanism deals with the role assignment for each vehicle involved in the procedure of local data gathering and processing.

Within a local area, we adopt both V2V (vehicle to vehicle) and V2R (vehicle to RSU) communication modes for data transmission. The vehicles and RSUs in this area actually form a distributed local wireless network. One of the fastest ways to realize the local data collection and process is to choose one vehicle to do it, with others just offering their data to the chosen one. This *one* is the sponsor in our mechanism. Due to the asynchronism of vehicle wireless communication environment, it is important to avoid collision in this sponsor choosing procedure.

An appropriate sponsor should be the one who can inform as many vehicles as possible in as short time as possible. In distributed wireless communication environment, shorter distance between devices usually means shorter communication time consumption. Considering the ability of a vehicle to locate itself, we hope to use it to decrease the time consumption in the phase of local data gather.

Here, our Sponsor-Follower mechanism is intended to settle the issue of collision avoidance and best-sponsor-choosing simultaneously. Before introducing this mechanism in detail, it is necessary to clarify the following two kinds of messages. The first is the *sponsoring message*, which is used to inform other vehicles that a procedure of data collection is sponsored by the generator of this message. The information contained in this message includes: timestamp, ID of the generator, ID of the junction area where the generator locates in. The second kind of message is the *response message*, which is used when a vehicle is willing to answer a sponsoring message from another vehicle. A response message contains the following information: timestamp, ID of the response vehicle, history timestamps, locations and corresponding speeds.

In general, the sponsor of a junction area $\mathcal{J}$ is responsible for collecting data from all other vehicles in $\mathcal{J}$. The followers in $\mathcal{J}$ voluntarily offer their data to the sponsor. The information included in a response message should be collected within the past time of T_s and should be collected in the area of $\mathcal{J}$.

As a common assumption, the RSUs are settled at the junctions of road. Meanwhile, transmitting data via a RSU is a fast way for data uploading. What's more, if a vehicle is near to the junction J, it will have a broader wireless coverage to communicate with other vehicles in $\mathcal{J}$. Here we define the location of junction J as the location of its central point and denote it as P_J. We consider the following two aspects when setting a priority to a vehicle who candidates the sponsor:

- Aspect 1. If two vehicles r_1 and r_2 are at same distance from P_J at the same time, but r_1 is coming closer to P_J while r_2 is leaving P_J, then r_1 will have higher priority than r_2 to be the sponsor. This is because the coming one has more time to communicate with the vehicles in $\mathcal{J}$ than the leaving one.

- Aspect 2. If r_1, r_2 are both coming toward P_J, and are very close to it while r_1 is closer than r_2. In this situation, we don't simply think that r_1 must have a higher priority. This is because due to the movements of vehicles, after a while, r_2 may be more suitable than r_1 to communicate with all other vehicles.

We denote the set of the road sections included in $\mathcal{J}$ as R_J, and denote the set of the vehicles now running in $\mathcal{J}$ as V_J. We now denote the priority of v at location P as $pri(P, v)$. Here,

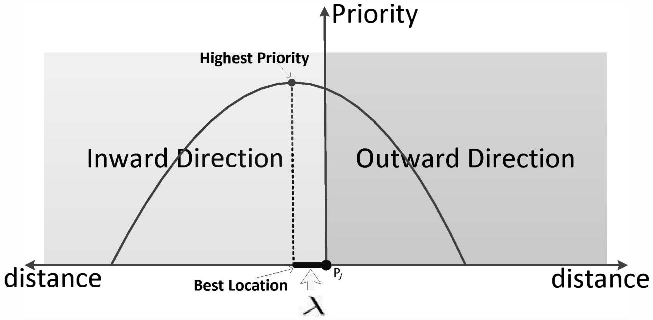


Fig. 4. The illustration of $pri(P, v)$

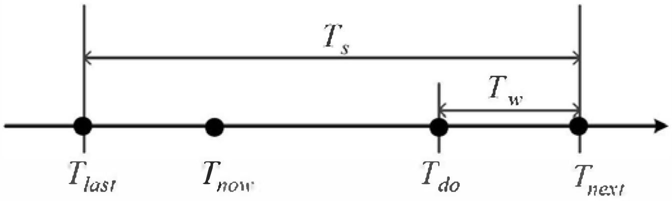


Fig. 5. The relationships among $T_{last}, T_{next}, T_{do}, T_{\omega}$ and T_{now}.

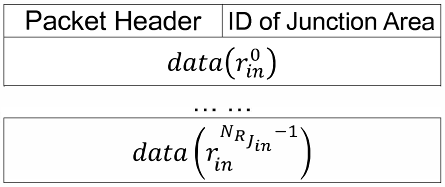


Fig. 6. The form of the packet to be uploaded by a sponsor.

$pri(P, v)$ should be a function satisfying the requirements of above two aspects. A proper form of $pri(P, v)$ could be:

$$pri(P, v) = \cos\left(\pi \times \frac{dist(P, P_J) \times Dire(v) - \lambda}{\sum_{i=0}^{N_J - 1} length_i}\right),$$

where $dist(P, P_J)$ is the distance between P and P_J, and $length_i$ is the length of the i^{th} road section of $\mathcal{J}$. The value of function $Dire(v)$ is 1 if v is running on an inward road of $\mathcal{J}$, and -1 if on an outward road of $\mathcal{J}$. Here λ is an adjustable offset value. We illustrate the meaning of $pri(P, v)$ in Fig.4.

Assume that the timestamp for now is T_{now}, the last timestamp of the data collection activity is T_{last}, and the next timestamp is T_{next}. We define T_{do} as the starting time at which the vehicles response to the sponsor for the next data collection activity in $\mathcal{J}$. Due to the time consumption of sponsoring message diffusion and response message reception, we define T_{ω} as the preparing time consumption before T_{next}. Fig.5 shows the relationship among $T_{last}, T_{next}, T_{do}, T_{\omega}$ and T_{now}.

Now we explain our strategy of sponsor choosing in the view of any vehicle v. v will know which vehicle will be the sponsor according to the following criteria and steps:

1) The vehicle v will first estimate its location at time T_{do}, denoted by P_{est_v}, according to its current location, speed, and moving direction. Then it calculates its priority at P_{est_v}, namely, $pri(P_{est_v}, v)$. Finally, it broadcasts its sponsoring message containing this priority value.

2) Vehicle v may receive multiple sponsoring messages from different other vehicles. Among all the priority values in the received messages and its own priority, it will find the biggest value and the corresponding ID of the generator of this priority.

3) When time comes to T_{do}, vehicle v will send a response message to the vehicle with the biggest priority at a random time between T_{do} and T_{next}. If the biggest priority belongs to v itself, it will not send response message.

4) Before v broadcasts its priority, if it receives a message containing a bigger priority than that of its own, it will give up broadcasting priority.

5) If v is a newcomer to $\mathcal{J}$, it will immediately calculate its value of priority. But if it has just participated in the last data collection activity of $\mathcal{J}$, it will randomly choose a time during T_{now} and T_{do} to broadcast. This design aims to avoid the collision when broadcasting sponsoring messages.

6) If v is a sponsor, it will open to receive data from other vehicles from time T_{do} to T_{next}.

It's reasonable that there may be more than one sponsors after this choosing procedure in the junction area $\mathcal{J}$. Due to the mobility of vehicles and the instability of wireless communication environment, the sponsoring message with biggest priority among all the vehicles may not be accepted by all the other vehicles.

After ceasing to receive data, the sponsor immediately begins to process the data it has received. Here, we further divide those road sections within $\mathcal{J}$ to two sets: $R_{J_{in}}$ and $R_{J_{\bullet ut}}$. The elements in $R_{J_{in}}$ are the inward road sections toward J, and $R_{J_{\bullet ut}}$ includes the outward road sections from J. The sponsor will only calculate the traffic condition values for the road sections included in $R_{J_{in}}$. This is because the traffic condition values of the road sections included in $R_{J_{\bullet ut}}$ will be calculated in other junction areas according to our design. To do so, the sponsor will first classify the collected data into different sets for the elements of $R_{J_{in}}$. For a road section $r \in R_{J_{in}}$, data from two kinds of vehicles will be included in its corresponding data set. The first kind of vehicles are those now (just now, actually) running on r, and the second kind of vehicles are those now running on a outward neighbor of r but went from r. Then, the sponsor will apply the local traffic condition evaluation method that we have introduced in section V to the classified data sets.

Then, the sponsor will upload the calculated results to the server via the RSU in $\mathcal{J}$. The form of the uploaded packet is shown in Fig.6, where $r_{in}^{l} \in R_o$, and $N_{R_{J_{in}}}$ denotes the number of elements of R_o.

VII. Global Traffic Condition Coverage Filling

We now introduce our method for global traffic condition vacancy completion.

We first introduce a slide window $T_{sw} = n \times T_s$, where n is a positive integer. We denote the begin time of this slide window as T_{begin}, and the end of it as T_{end}. We denote the i^{th} interval T_s in T_{sw} as T_s^i. For each T_s^i, there can be more than one piece of data uploaded to the server. We separate the data in a packet it received for a junction area into parts, corresponding to the road sections belonging to this junction area. For a road section r, we put the data received by the server to reflect the road condition for T_s^i in $D_r^i = \{d_j^i | 0 \leqslant j < N_r^i\}$, where N_r^i is the number of different pieces of data.

Then we have two different situations:

• If $N_r^i \geqslant 1$, the server should make full use of the received data to accurately illuminate the traffic condition of r. As described in Section V, in a piece of uploaded data, the number of vehicles contributing to constitute this data is also uploaded. For $d_j^i \in D_r^i$, we denote this numerical information of it by n_i^j. Correspondingly, we denote the local calculated mean speed in this data by s_i^j. So, in this situation, we calculate the *synthetical mean speed* of r for past T_s^i as:

$$\hat{s_r^i}(r, T_s^i) = \frac{\sum_{j=1}^{N_r^i}(n_i^j \times s_i^j)}{\sum_{j=1}^{N_r^i} n_i^j}.$$

We use $TC(r)$ to represent this way of traffic condition calculation.

• If $N_r^i = 0$, there is no uploaded data for the traffic condition of r for T_s^i. So we need to use the history correlation information to fill it up. In section VII, we said that in the uploaded data packet for road section r, there are fields representing the number of vehicles evacuated, to the outward neighboring road sections in the past T_s^i. We now reversely use this information to explain that for a road section r, in the past T_s^i, how its inward neighbors contributed to shape the traffic flow of r.

Let $R_{in_r} = \{r_k | 0 \leqslant k < N_{in_r}\}$ denote the inward neighbors of r, where N_{in_r} is the number of the elements in R_{in_r}. For $\forall r_k \in R_{in_r}$, we denote the number of reported vehicles went from r_k to r as in_k.

We now define the *Inward Contribution Factor* of an inward neighbor r_k to r for the past T_{sw} as,

$$ICF(r_k, r) = \frac{\sum_{i=0}^{n-2}\left(in_k / \sum_{j=0}^{N_{in_r}-1} in_j\right)}{n-1}.$$

Here we take the former $n-1$ T_ss in T_{sw} into consideration to mine the historical relationship between r_j and r.

Then in this situation, the *synthetical mean speed* of r can be calculated as,

$$\hat{s_r^i} = \sum_{k=0}^{N_{in_r}-1} \hat{s_{r_k}^i} \times ICF(r_k, r),$$

where $\hat{s_{r_k}^i}$ denotes the synthetical mean speed of r_k for T_s^i. We name this way of calculation as $TC_e(r)$.

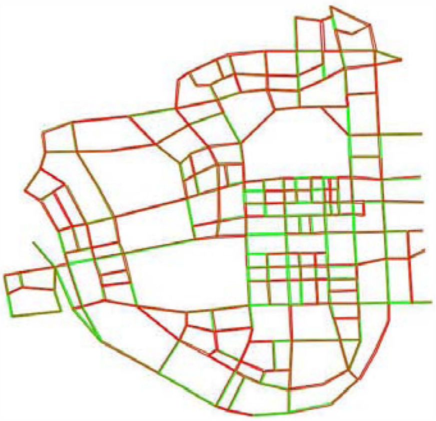


Fig. 7. The graphical representation of traffic condition evaluation results. Traffic condition is better if the color is greener, and worse if redder.

But if for an inward neighbor of v, *e.g.*, u, the server also has no data about it for T_s^i, we will apply $TC_e(u)$ to get its evaluation value first. If the same situation happens to u in this way, we will apply the method recursively. If the evaluation value of r is successfully calculated, the next step of our algorithm is to calculate the evaluation value of its neighbors one by one if they have not been calculated yet.

VIII. Experiments And Results

To evaluate the performance of our approach, we perform simulations based on real vehicular traces. The trace data was collected in the Futian district of the city Shenzhen, China from April 18th 00:00:00, 2011 to April 26th 00:00:00, 2011. There are 13764 taxis that participated in this work of data collection.

A. Graphical Representation of Traffic Evaluation

We use the traffic methods presented in Section V and VII to simulate the traffic condition evaluation. We take 716 main road sections in Futian district as experimental object. As an example, Fig.7 represents the evaluation results for the time duration of 12:50:30-12:51:00 in the day of April 18th, 2011. A greener color in Fig.7 of a road section indicates a better traffic condition. Through this graphical presentation, we can intuitively see the difference of traffic condition between different road sections.

B. Performance of Bandwidth Saving

An important goal of our design for RTS is to save the upstream bandwidth consumption. We noticed that if vehicle data is not integrated locally and every vehicle chooses to upload its data directly to the server, the bandwidth to be occupied would be much bigger. Based on this observation, we simulate the counting of the size of uploading data packets (in the unit of byte) during the experiment time. For conciseness, we denote the total size of the data packets to be uploaded by every vehicles respectively in a specific T_s as $B_{T_s}^{res}$, and denote the corresponding size of packets to be uploaded by

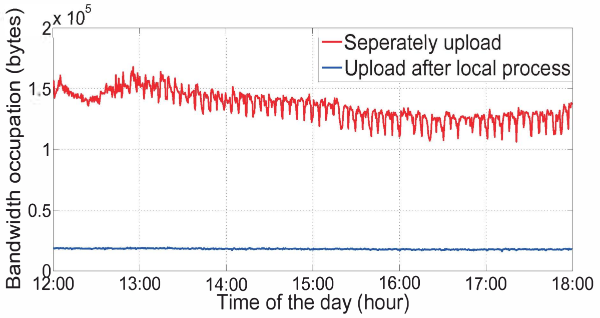


Fig. 8. The bandwidth occupation savings when $T_s = 30s$

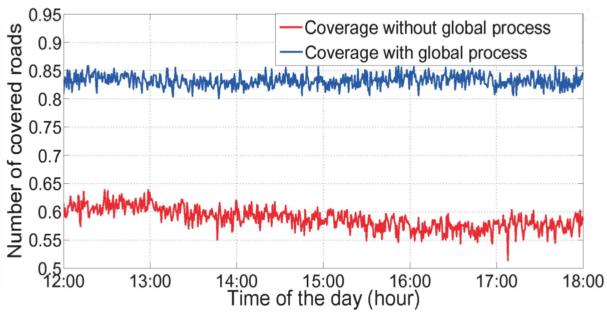


Fig. 10. Road coverage ratio when $T_s = 30s$

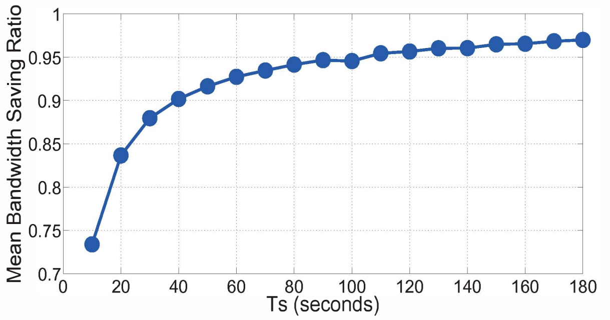


Fig. 9. The bandwidth occupation saving ratio when T_s varies.

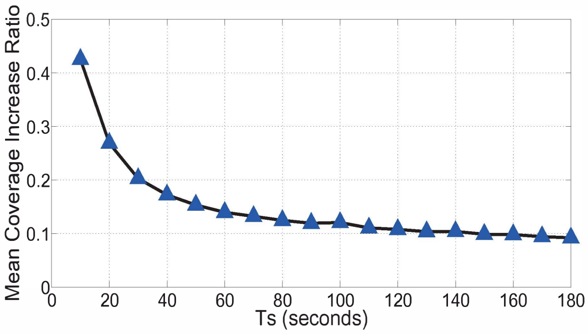


Fig. 11. The coverage gain when T_s varies.

sponsors after local integration as $B_{T_s}^{loc}$. Fig.8 shows the results of bandwidth consumption when $T_s = 30s$ during 12:00:00-18:00:00 in April 18th, 2011. The results show that our design of RTS can significantly save the upstream bandwidth via local data aggregation.

To observe the variation of bandwidth saving when T_s varies, we define the bandwidth saving ratio for experimental duration time $\mathcal{T}$ as

$$\mathcal{B}_{\mathcal{T}} = \frac{1}{N_{T_s}} \left(\sum_1^{N_{T_s}} \frac{B_{T_s}^{res} - B_{T_s}^{loc}}{B_{T_s}^{res}} \right),$$

where N_{T_s} is the number of T_s within $\mathcal{T}$.

Then we change the value of T_s from 10 seconds to 180 seconds with 10 seconds as step length. Fig.9 shows our bandwidth saving results for different T_s when $\mathcal{T}$ starts from April 18th 00:00:00, 2011 and ends at April 20th 00:00:00, 2011. We can see that, when T_s increases, the ratio of the saved bandwidth also increases, but it tends to be stable.

C. Performance of Coverage Gain

The global process in the second phase of RTS aims to increase the spatiotemporal coverage of traffic condition evaluation. We denote the number of roads covered in local process phase for T_s as $C_{T_s}^{loc}$, and denote the number of roads covered after global process as $C_{T_s}^{glo}$. As pre-mentioned, we have 716 road sections in total, now we denote this number as N_{roads}. Fig.10 shows the ratios (the proportions of $C_{T_s}^{loc}$ and $C_{T_s}^{glo}$ over N_{roads} respectively) of covered roads when $T_s = 30s$, which indicates that our method can significantly increase the ratio of coverage.

We also examine the change of coverage for different T_s. We define the coverage gain for experimental time $\mathcal{T}$ as

$$\mathcal{C}_{\mathcal{T}} = \frac{1}{N_{T_s}} \left(\sum_1^{N_{T_s}} \frac{C_{T_s}^{glo} - N_{roads}}{C_{T_s}} \right),$$

where N_{T_s} is the number of T_s within $\mathcal{T}$. Fig.11 shows the coverage results for different T_s which changes from 10 to 180 with step 10 ($\mathcal{T}$ is the same as that of section VIII-B). So we can see that when T_s increases, the coverage gain decreases.

IX. Conclusion and Future Work

In this work, we introduce a two-phased data collection and process approach named RTS using vehicular crowdsensing. RTS uses the topology of road net to divide it into road sections and junction areas. With this division, RTS exploits the power of local cooperation among vehicles to calculate traffic conditions for road sections and to save the upstream bandwidth. It also provides a global traffic evaluation method that utilizes the correlation between adjacent roads, which significantly increase the spatiotemporal coverage for road net. The efficiency of data collection and the effectiveness of data usage are jointly improved. In future, we need to do studies about extending this approach to broader application scenarios not just the traffic condition evaluation.

Acknowledgments

The research of authors is partially supported by "Shuguang Program" from Shanghai Education Development Foundation under grant No. 14SG20, Shanghai Rising-Star Program under grant No. 14QA1403700, the National Natural Science Foundation of China (NSFC) under grant No. 61571331 and 61202383, the Program for New Century Excellent Talents in University (NCET) under grant No. NCET-12-0414, the Natural Science Foundation of Shanghai under grant No. 12ZR1451200, the Integrated Project for Major Research Plan of the National Natural Science Foundation of China under grant No. 91218301, the Shanghai Key Basic Research Project under grant No. JC1405800, the National Basic Research Program of China (973 Program) under grant No. 2014CB340404, the Research Fund for the Doctoral Program of Higher Education of China (RFDP) under grant No. 20120072120075, and the Tongji Young Talents Training Program under grant No. 2013KJ048.

References

[1] K. Kiratiratanapruk and S. Siddhichai, "Practical Application for Vision-based Traffic Monitoring System," in *Proc. IEEE International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology 2009*.

[2] T. Semertzidis, K. Dimitropoulos, A. Koutsia, and N. Grammalidis, "Video Sensor Network for Real-time Traffic Monitoring and Surveillance," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 103–112, 2010.

[3] S. Ahmed, S. Bahaa, and E. Mohamad, "Intelligent Cross Road Traffic Management System," in *Proc. IEEE ICCTD 2010*.

[4] B. Tian, Q. Yao, Y. Gu, K. Wang, and Y. Li, "Video Processing Techniques for Traffic Flow Monitoring: A Survey," in *Proc. IEEE ITS 2011*.

[5] K. Robert, "Video-based Traffic Monitoring at Day and Night Vehicle Features Detection and Tracking," in *Proc. IEEE ITS 2009*.

[6] K. Mandal, A. Sen, A. Chakraborty, S. Roy, S. Batabyal, and S. Bandyopadhyay, "Road Traffic Congestion Monitoring and Measurement Using Active RFID and GSM Technology," in *Proc. IEEE ITSC 2011*.

[7] C. Bachmann, M. J. Roorda, B. Abdulhai, and B. Moshiri, "Fusing a Bluetooth Traffic Monitoring System with Loop Detector Data for Improved Freeway Traffic Speed Estimation," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 17, no. 2, pp. 152–164, 2013.

[8] S. Jeng and L. Chu, "A High-Definition Traffic Performance Monitoring System with the Inductive Loop Detector Signature Technology," in *Proc. IEEE ITSC 2014*.

[9] R. K. Ganti, F. Ye, and H. Lei, "Mobile Crowdsensing: Current State and Future Challenges," *IEEE Communications Magazine*, vol. 3, no. 11, pp. 32–39, November 2011.

[10] C. Liu, C. Chigan, and C. Gao, "Compressive sensing based data collection in vanets," in *Proc. IEEE WCNC 2013*.

[11] A. Skordylis and N. Trigoni, "Efficient Data Propagation in Traffic-Monitoring Vehicular Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 680–694, 2011.

[12] J. Miller, "Vehicle-to-Vehicle-to-Infrastructure (V2V2I) Intelligent Transportation System Architecture," in *Proc. IEEE Intelligent Vehicles Symposium 2008*.

[13] S. Dornbush and A. Joshi, "StreetSmart Traffic: Discovering and Disseminating Automobile Congestion Using VANETs," in *Proc. IEEE Vehicular Technology Conference 2007*.

[14] S. Vaqar and O. Basir, "Traffic Pattern Detection in A Partially Deployed Vehicular Ad Hoc Network of Vehicles," *IEEE Wireless Communications*, vol. 16, no. 6, pp. 40–46, 2009.

[15] W. Shi and Y. Liu, "Real-time Urban Traffic Monitoring with Global Positioning System-equipped Vehicles," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 113–120, 2010.

[16] R. Bauza, J. Gozalvez and J.S. Soriano, "Road Traffic Congestion Detection Through Cooperative Vehicle-to-Vehicle Communications," in *Proc. IEEE Local Computer Networks 2010*.

[17] J. Aslam, S. Lim, and X. Pan, "City-Scale Traffic Estimation from a Roving Sensor Network," in *Proc. ACM Sensys 2012*.

[18] X. Chen, X. Wu, X.-Y. Li, Y. He, and Y. Liu, "Privacy-Preserving High-Quality Map Generation with Participatory Sensing," in *Proc. IEEE INFOCOM 2014*.

[19] X. Jiang and D. Du, "Bus-Vanet: A Bus Vehicular Network Integrated with Traffic Infrastructure," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 2, pp. 45–57, 2015.

[20] D. Wu, Y. Zhang, J. Luo, and R. Li, "Efficient Data Dissemination by Crowdsensing in Vehicular Networks," in *Proc. IEEE 22nd International Symposium of Quality of Service 2014*.

[21] T. Nadeem, S. Dashtinezhad, C. Liao and L. Iftobe, "TrafficView: Traffic Data Dissemination Using Car-to-Car Communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 3, pp. 6–19, 2004.

[22] Q. Kong, Q. Zhao, C. Wei, and Y. Liu, "Efficient Traffic State Estimation for Large-scale Urban Road Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 398–407, 2013.

[23] X. Shi, J. Xing, Z. Jun, Z. Lei, and W. Li, "Application of Dynamic Traffic Flow Map by Using Real Time GPS Data Equipped Vehicles," in *Proc. International Conference on ITS Telecommunications 2006*.

[24] B.-J. Chang, B.-J. Huang, and Y.-H. Liang, "Wireless Sensor Network-Based Adaptive Vehicle Navigation in Multihop-Relay WiMAX Networks," in *Proc. IEEE AINA 2008*.

[25] C. Wang, L. Shao, Z. Li, L. Yang, X.-Y. Li, and C.-J. Jiang, "Capacity Scaling of Wireless Social Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 7, pp. 1839–1850, 2015.

[26] C. Wang, C. Jiang, X.-Y. Li, S. Tang, and P. Yang, "General Capacity Scaling of Wireless Networks," in *Proc. IEEE INFOCOM 2011*.

[27] C. Wang, C. Jiang, X.-Y. Li, S. Tang, Y. He, X. Mao, and Y. Liu, "Scaling Laws of Multicast Capacity for Power-Constrained Wireless Networks under Gaussian Channel Model," *IEEE Transactions on Computers*, vol. 61, no. 5, pp. 713–725, 2012.

[28] Z. Li, C. Wang, S.-Q. Yang, C.-J. Jiang, and I. Stojmenovic, "Space-Crossing: Community-Based Data Forwarding in Mobile Social Networks Under the Hybrid Communication Architecture," *IEEE Transactions on Wireless Communications*, vol. 14, no. 9, pp. 4720–4727, 2015.

[29] Z. Li, C. Wang, S.-Q. Yang, C.-J. Jiang, and X.-Y. Li, "LASS: Local-Activity and Social-Similarity Based Data Forwarding in Mobile Social Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 174–184, 2015.

[30] Z. Li, C. Wang, S.-Q. Yang, C.-J. Jiang, and I. Stojmenovic, "Improving Data Forwarding in Mobile Social Networks with Infrastructure Support: A Space-Crossing Community Approach," in *Proc. IEEE INFOCOM 2014*.

[31] H. Wang, Y. Zhu, and Q. Zhang, "Compressive Sensing Based Monitoring With Vehicular Networks," in *Proc. IEEE INFOCOM 2013*.

[32] Z. Li, Y. Zhu, H. Zhu, and M. Li, "Compressive Sensing Approach to Urban Traffic Sensing," in *Proc. IEEE ICDCS 2011*.

[33] W. Alasmary, H. Sadeghi, and S. Valaee, "Crowdsensing in Vehicular Sensor Networks with Limited Channel Capacity ," in *Proc. IEEE ICC 2013*.

[34] R. Du and C. Chen, "Effective Urban Traffic Monitoring by Vehicular Sensor Networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 273–286, 2014.

[35] R. Du, C. Chen, B. Yang, and X. Guan, "VANETs Based Traffic Estimation: A Matrix Completion Approach," in *Proc. IEEE GLOBECOM 2013*.

[36] A. Pascale, M. Nicoli, and U. Spagnolini, "Cooperative Bayesian Estimation of Vehicular Traffic in Large-Scale Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2074–2088, 2014.