Density Adaptive Urban Data Collection in Vehicular Sensor Networks

Zongjian He and Huijuan Zhang
School of Software Engineering
Tongji University
Shanghai, China
E-mail: {hezongjian, mszhj}@tongji.edu.cn

Abstract—Research on vehicular sensor networks (VSNs) has drawn increasing attentions recently. By collecting data from onboard sensors and gathering them with VSNs, vehicles can serve as data sources for many applications such as intelligent transportation system, smart cities and etc. Existing data collection solutions in VSNs focused on temporal or spatial influence but did not address the influence of vehicle density. In this paper, we present a novel data collection solution for collecting data within a specific geographical region in urban scenario. The objective of proposed algorithm is to satisfy the specific requirement of applications by collecting certain number or ratio of data. It is able to adjust the required collection life cycle adaptively according to vehicle density. The mobility nature of vehicles determines that the topology of VSN is dynamic. To adapt with mobility, the algorithm is designed to be decentralized and stateless. Therefore, vehicles implementing this algorithm do not require neighboring information or link state. The network transmission latency is analytically characterized and modeled. Evaluations show that the results of the algorithm match the mathematical model well and outperforms existing protocols in terms of efficiency and effectiveness.

Index Terms—data collection, VANET, traffic density, vehicular network.

I. INTRODUCTION

With the advancement of sensor and communication technology, vehicular sensor networks (VSN) are an emerging network paradigm. With the help of VSNs, many new applications become feasible. Besides safety related applications (e.g., collision avoidance), vehicles can also collect all kinds of data by utilizing their onboard sensors, such as vehicle’s velocity, location, fuel consumption and real-time emission. Particularly, when the data from all vehicles in a wide region are collected together and then analyzed, some important information, such as traffic condition [1], environmental pollution and noise level can be obtained. Such information can be used as a fundamental input for building intelligent transportation systems and smart cities.

Different approaches have been developed to transmit sensory data from vehicles to the cloud or base station (BS). Conventional approaches utilize the widely deployed cellular networks to transmit data [2]. However, this approach suffers from high communication cost, especially for stream media data, due to the expensive data plan offered by mobile operators. Therefore, using vehicular ad-hoc networks (VANET) to support data collection becomes attractive. Researchers have developed many multihop routing protocols for VANETS [3], which are very useful for one-way communication. However, data collection is usually a two-way communication: firstly application sends the data collection request to all vehicles, and then vehicles transmit the data back. In this scenario, one way communication is not always effective and there is a large space to optimize the overall communication performance. This motivates us to conduct the research of designing an effective two-way data collection protocol.

However, two-way data collection in VSNs is not an easy task. A number of challenging issues are raised; Firstly, the network topology of VSNs can be highly dynamic and VSNs can be disconnected from time to time. Therefore, many conventional data collection approaches which assume static neighbors such as spanning tree based data harvesting [4] can not be used in VSNs. Secondly, the density of vehicles changes with time. The data propagation speed of multi-hop communication is highly dependent on the vehicle density [5]. Therefore, vehicle density must be considered. Thirdly, choosing the appropriate life cycle of the data collection process is also difficult. The data collection process should be stopped after a period of time. Running too long or too short will both cause side effects to the collection process. If each single task runs for a long time, the network resources may be exhausted due to the synergistic effect of multiple instances. It is also meaningless to continue collecting data after the required amount of data have been collected. On the other hand, if the process stops too early, only limited number of data are collected and they may not satisfy the requirement and then affect the quality of data analysis. Moreover, the process must be stopped on both the BS and all the vehicles at the same time. It is also a tough problem due to the dynamic and distributed nature of the architecture.

In this work, we consider data collection scenario depicted in Fig. 1. A fixed base station (BS) is installed at the roadside. The BS can be regarded as a computer with strong computational capability and some data analysis...
applications run on it. Every period of time, the BS wish to collect data from a certain number of vehicles within a specific region called region of interest (ROI). The data collection process works like this: First, the BS initiates the data collection process by sending request to its nearby vehicles. The request is disseminated in a multi-hop way to nearby vehicles within ROI. Each vehicle inside the ROI will send the requested data back, also in a multi-hop manner, as soon as it receives the data collection request. It should be noted that in the process of flooding the request command sent by BS, each vehicle may need to diffuse the request to its neighbors multiple times, since there are always new neighbors due to the mobility of VSNs. After desired data has been collected, the data collection process terminates automatically.

The contributions of this paper are listed as follows:

- We propose a novel two-way data collection algorithm for VSNs. To fit in with the mobility nature of vehicles, the algorithm is completely stateless. In the algorithm, vehicles neither store neighbor information locally nor detect neighbors change. In addition, the proposed algorithm can adaptively estimate its life cycle according to the current traffic flow density.
- The latency and efficiency of the proposed protocol is mathematically modeled and analyzed. In particular, the major factors driving system performance and latency are summarized. Based on this, some design issues are discussed. We focus on parameter preferences of the solution such as range of ROI or vehicle density.
- We conduct large-scaled evaluation using network simulators. We compare our solution with existing one. Evaluation results show that our solution fits the mathematical model well and outperforms existing protocols.

The rest of this paper is organized as follows. Previous studies and related work are summarized in Section II. Detailed explanation of the assumption and algorithm of proposed solution are presented in Section III. In Section IV, we model and analyze both the dissemination and aggregation phases of the solution. The vehicle density adaptive mechanism is also explained. In Section V, the algorithm is simulated and then evaluated. Finally, Section VI concludes this paper.

II. RELATED WORK

CGP [6] is a cluster based data gathering protocol for VSNs. It utilizes the geographical information to cluster vehicles on the road. Data within the same cluster is firstly collected to the cluster head and then transferred to the BS. CGP is applicable to single road scenario. If considering collecting data from a larger area with multiple roads, the road and lane based clustering may not applicable and the overhead to maintain the cluster will increase. Moreover, CGP works as a polling application and assumes that data from all pass-by vehicles should be collected. Under this circumstance, the data collection process only contains aggregation phase and the problem is simplified compared with ours.

Mobeyes [7] is a disruption tolerance and scalability vehicular sensing platform designed for urban data collection. It uses original opportunistic protocol that exploits intrinsic mobility of nodes to harvest summaries of sensed data. The Mobeyes’s on-demand data harvest mode deals with similar problem with us. It diffuses query in a k-hop manner and harvests data to a specific sink node. However, the authors claim that it is not practical to exploit on-demand strategy in Mobeyes due to several reasons and another proactive mode is preferred.

DB-VDG [8] is a delay-bounded vehicular data gathering protocol designed for gathering data from a certain geographic area while satisfying with a predefined delay bound. DB-VDG deals with the same problem as our solution and can be treated as the state of the art solution in VSN data collection. It uses opportunistic manner to transmit data, so the selection strategy that decides either to keep carrying or to forward to next hop is important. DB-VDG uses both distance-based and speed-based strategy for different situations.

Considering the dissemination and aggregation phases independently, more related works can be found in literature.

The analysis of one dimensional (1D) broadcast propagation speed in VANETs can be found in [9] and [10]. One dimensional model is the simplest scenario in vehicular network. However, it can not be directly applied to our scenario, which broadcasts in two dimensions. The node density of vehicular networks varies from time to time. Therefore, different strategies should be used under different situations. In sparse network case, information propagation speed in DTN is discussed in [11], both the upper bound and lower bound of the propagation delay are proven. The broadcasting delays of a geographical area is analyzed in [12]. However, in most literature, dense networks and sparse networks are discussed separately. In our solution, vehicle density can be estimated by the BS and the algorithm will be adjusted accordingly.
Definition

The vehicular network denoted as graph $G$. The transmission range of the wireless network is $d$. The radius of ROI is $r$. Density of vehicles is $\theta$. The ratio of vehicles not received broadcast at time $t$ is $P(t)$. The average collection speed at time $t$ is $E(t)$. The vehicular network denoted as graph $G$. The total number of vehicles inside ROI is $|V|$. The number of vehicles that have been collected to BS at time $t$ is $N(t)$. The number of vehicles that have not been collected to BS at time $t$ is $L(t)$. The total number of neighbors for vertex $v$ in graph $G$ is $N_c(v)$. The length of the sending duty cycle is $L_s$.

Due to the mobility of nodes in vehicular networks, it is impossible to maintain the complete network topology. Even for constructing a spanning tree like H-SPREAD [4], the cost of keeping the neighbors' information can be high. Instead, stateless geographical routing algorithm, which requires neither previous route computation prior to sending packets nor stateful routes to maintain in the network [13], is preferred. In our solution, beacon-less geographical routing is used.

III. THE DATA COLLECTION SOLUTION

A. System Model

The objective is to develop a density adaptive algorithm to collect data within a geographical region to meet the data collection requirements for applications. The data collection requirements can be defined as: collect data from certain number or a certain proportion of vehicles within a ROI.

In our research, we assume that all the vehicles have onboard computers installed. The onboard computer contains positioning system like GPS to obtain current position information. It can also establish V2V and V2I network connections using standard wireless network protocols like IEEE 802.11p. The transmission range for both BS and vehicles is denoted as $r$. We further assume that the shape of ROI is a circle with radius $d$ and the BS is located at the center of the circle. The algorithm is designed for urban scenario, thus the ROI is a large area such as an urban area with thousands of vehicles. In this circumstance, the number of vehicles move in or out of the ROI during the data collection process is small and can be neglected. To simplify the analysis, vehicles in the ROI are treated to be uniformly distributed with density $\theta$. So, in our solution we will not consider vehicles enter or leave the ROI.

Notations and symbols used in this paper are summarized in Table I.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r$</td>
<td>The transmission range of the wireless network</td>
</tr>
<tr>
<td>$d$</td>
<td>The radius of ROI, $d &gt; 2r$</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Density of vehicles.</td>
</tr>
<tr>
<td>$R(t)$</td>
<td>Number of vehicles received the request at time $t$.</td>
</tr>
<tr>
<td>$L(t)$</td>
<td>Number of vehicles not received the request at time $t$.</td>
</tr>
<tr>
<td>$N(t)$</td>
<td>Number of vehicles that have been collected to BS at time $t$.</td>
</tr>
<tr>
<td>$E(t)$</td>
<td>The average collection speed at time $t$, $E(t) = \frac{N(t)}{t}$.</td>
</tr>
<tr>
<td>$G$</td>
<td>The vehicular network denoted as graph.</td>
</tr>
<tr>
<td>$</td>
<td>V</td>
</tr>
<tr>
<td>$P(t)$</td>
<td>The ratio of vehicles that have not received broadcast at time $t$.</td>
</tr>
<tr>
<td>$N_c(v)$</td>
<td>Number of neighbors for vertex $v$ in graph $G$.</td>
</tr>
<tr>
<td>$L_s$</td>
<td>Length of the sending duty cycle.</td>
</tr>
</tbody>
</table>

B. The data dissemination phase

The entire data collection process begins with a request sent from BS. The request is then disseminated to the vehicles. We define the procedure of sending the data collection request from the BS to vehicles as dissemination phase of the algorithm. Ideally, all vehicles inside the ROI will receive the broadcast. However, implementing a guaranteed solution to diffuse the request to all vehicles requires additional efforts. For example, it should be aware of which vehicles have or have not received the request, as well as their locations. Considering vehicles are mobile and the network topology changes rapidly, the overhead of doing this is considerably large. Therefore, instead of the guaranteed scheme, we prefer to use the best effort scheme, which means the request is disseminated without any guarantee that it will be received by all vehicles, and the senders will not be acknowledged if the requests are received or not.

In VSNs, there will always be new neighbors that have not received the request entering it transmission range. If every vehicle broadcasts only once, it is possible that certain vehicles will never receive the broadcasted request. To overcome this issue, in our solution, all vehicles that have received the request are requested to re-broadcast periodically to their neighbors. The broadcast will not stop during the entire life cycle of the algorithm.

An intuitive question to the broadcast approach is that whether it will cause broadcast storm problem [14] and exhaust the network resource. Moreover, compared with time critical applications such as collision avoidance, the priority of data collection is not that high. Since all applications on vehicles share the same V2V network channels and bandwidth, data collection tasks should be scheduled to leave the major bandwidth to time critical tasks and only occupy the channel to send and receive data in limited time. To deal with these problems, two mechanisms are developed. The first one is called Re-broadcast Filtering and the second one is Duty Cycled Execution.

Re-broadcast Filtering tries to reduce the redundant re-broadcasts by filtering the received broadcast. Only on receiving broadcast for the first time, the vehicles have the obligation to re-broadcast. Later on, duplicated request will be disregarded.

Duty Cycled Execution tries to reduce the contention and collision by restricting the transmission frequency. Both the BS and the vehicles are only allowed to broadcast once in a duty cycle. The period length of the duty cycle, namely transmission time slot is denoted as $t_s$. The length of time slot is usually much longer than the broadcast period (e.g. 3 seconds). In this way, the bandwidth occupation is strictly controlled. If $t_s$ approximates to 0, it is reduced to conventional flooding.

C. The data aggregation phase

After vehicles have received the data collection request for the first time, they begin to forward the request to other vehicles. At the same time, they have to respond to the request by sending the data from their onboard sensors (e.g. temperature, velocity etc.) back to the BS. The aggregation phase of the algorithm is defined as the
procedure of sending collected data back to the BS from vehicles. The model of aggregation is different from dissemination. Dissemination is a typical one to all broadcast communication, which contains only one data source (BS) and multiple destinations. If we treat each vehicle’s data transfer individually, the aggregation is a one to one unicast. That is, from one specific vehicle to the BS.

The objective of the aggregation algorithm is to collect data to the BS as soon as possible while not adding much overhead to the network communication. Therefore, instead of designing a separate algorithm that does not related to the dissemination algorithm to deal with aggregation, we prefer to utilize the existing dissemination broadcast to transfer collected data by appending the data at the end of each broadcast package. We have two major reasons to implement like this. Firstly, vehicles start to send data back as soon as they receive the broadcasted request. So there is no explicit boundary between the dissemination and aggregation phases in our algorithm. Meanwhile, they have to relay the broadcast in the rest of process life cycle periodically. It is possible to appending data to the end of broadcast packages. Secondly, compared with designing a new algorithm for aggregation, reuse the existing broadcast package by appending extra data to it can reduce the overall overhead of network. Some information in the package like source ID and locations can be reused.

However, keeping appending data to the end of the package without any constraint will cause the package size increasing explosively. Thus, data must be selectively added. In our solution, we use vehicles geographical and velocity information to determine whether the data should be carried or dropped by a vehicle. All carried data on a vehicle will be sent out along with the next broadcast. The idea is to choose the vehicle in all neighbors whose future location is closest to the BS to carry the data. Since vehicles are mobile and their locations change from time to time, only considering current location is not enough. Vehicles travel in different directions and the velocity must also be considered. The complete mechanism works like this: after one hop neighbors receive the broadcasted data, they will calculate whether they are more suitable to be the next hop, if so, they will keep the data, otherwise the data will be dropped. The calculation is done by $r = |P_{src} - \vec{V}_{src}t_s| - |P_{cur} - \vec{V}_{cur}t_s|$, where $P_{src}$ and $\vec{V}_{src}$ is the position and velocity of source vehicle, while $P_{cur}$ and $\vec{V}_{cur}$ is the position and velocity of current vehicle. $t_s$ is the length of transmission time slot. The coordinate of BS is (0, 0). If the result of equation $r$ is greater than 0, which means current vehicle will be closer to the BS after a duty cycle, the data will be kept. Otherwise, it will be dropped.

To ensure the statelessness of the algorithm, if the receiver node decides to carry the data, it should not send an response or ACK to inform the source node. Instead, it will broadcast the carried data in next duty cycle. After the source node receives the broadcast, it will check the appended data and found that the data has found a better next hop and then delete it from its carrying list. In this way, the data duplication can not be avoided completely since the original node may not receive the broadcast due to mobility. But the length of package can be reduced effectively.

The entire algorithm is shown as Algorithm 1. It is executed on vehicles when they receive packages each time.

**Algorithm 1** vehicle package receiving

**Input:** The received package: pkg

**Input:** The set of harvested data in the received package: pkg.data

**Input:** The set of harvested data carried locally in vehicle sCarry

1: if pkg.data $\neq \emptyset$ then
2: \hspace{1cm} r $\leftarrow$ $|P_{src} - \vec{V}_{src}t_s| - |P_{cur} - \vec{V}_{cur}t_s|$
3: \hspace{1cm} if $r > 0$ then
4: \hspace{2cm} sCarry $\leftarrow$ sCarry $\cup$ pkg.data
5: \hspace{1cm} else
6: \hspace{2cm} sCarry $\leftarrow$ sCarry $\setminus$ pkg.data
7: \hspace{1cm} end if
8: \hspace{1cm} end if

The package structure used in the algorithm is shown as Fig. 2. The detailed explanation of each field are:

- **SrcVID :** Unified vehicle ID of the source node, a.k.a sender.
- **ReqID :** Data collection request ID, generated by BS at the beginning of each data collection process. Used to distinguish different data collection request.
- **Position :** The position information of the source node.
- **Velocity :** The velocity information of the source node. Including speed and orientation.
- **TTL :** Time to live, used to control the life cycle of the algorithm.
- **Timeslot :** The length of time slot. Predefined by the BS.
be formulated as follows: the relationship between the two factors. The problem can received the broadcast. We are interested in finding out the time spent and the number of vehicles that have

Suppose node $S$ currently carries data from $\{X, Z\}$. It broadcasts its data to one hop neighbors. After receiving data from $S$, the one hop neighbors will merge the data from $S$ with its own. Then, re-broadcast it. Suppose node $D$ is closest to the BS. Fig. 3(b) shows the case when $D$ rebroadcast. After every neighbor of $D$ receives the data, they will delete their own copy because they know a node closer to BS has already successfully carried the data.

IV. ANALYSIS OF THE MODEL

In this section, we will model the proposed algorithm mathematically.

A. Mathematical model of the algorithm

During the dissemination phase, the major tradeoff is the time spent and the number of vehicles that have received the broadcast. We are interested in finding out the relationship between the two factors. The problem can be formulated as follows:

Given:
1) A VSN denoted as Graph: $G = (V, E)$.
2) Region of Interest with radius $d$
3) Vehicles with transmission range $r$ and density $\theta$
4) $|V| = R(t) + L(t) = \theta \pi d^2$

Subject To:
$\forall v \in V, N_G(v) = \theta \pi r^2 - 1 > 0$

Find: $R(t)$

Input 4) shows the total vehicle numbers. Constraint 1) shows the assumption that vehicles have at least one neighbor. Considering the urban scenario, the vehicle density is usually high and it is reasonable to make such assumption. This assumption is only for modeling purpose. The algorithm itself still works without this assumption. Since the algorithm is stateless, we can not utilize the connection state, velocity or location information when building up the model. The problem can only be modeled using probability theory.

Globally, the ratio of vehicles that have not received the broadcast at time $t$ can be represented as:

$$P(t) = \frac{L(t)}{|V|} = \frac{L(t)}{\theta \pi d^2} \quad (1)$$

Because all vehicles follow uniform distribution, the ratio of neighbors that have not received the broadcast for a single vehicle should be identical with the global value. And then, the mean number of neighbors that will receive the request after the broadcast for a single vehicle can be calculated by:

$$P(t) \times N_G(v) = (\theta \pi r^2 - 1) \frac{L(t)}{\theta \pi d^2} \quad (2)$$

The increasing rate of vehicles at time $t$ can be expressed as a differential equation. Combine Formula 2, the given condition and the constraints, we can get the simultaneous differential equations for the problem:

$$\begin{align*}
\frac{dR(t)}{dt} &= (\theta \pi r^2 - 1) \frac{L(t)}{\theta \pi d^2} R(t) \\
L(t) + R(t) &= \theta \pi d^2 \\
R(0) &= 1
\end{align*} \quad (3)$$

Here $R(0) = 1$ means at the beginning of data collection process, only the BS can be treated as if it has "received" the broadcast. The solution for $R(t)$ can be get using separation of variables. The result is:

$$R(t) = \frac{\theta \pi d^2}{1 + (\theta \pi d^2 - 1)e^{-(\theta \pi r^2 - 1)r}} \quad (4)$$

Equation 4 is the function we want to find. In this equation, $R(t)$ is determined by four variables, $\theta$, $d$, $r$ and $t$. The value $t$ here can be considered as number of duty cycle.

In aggregation phase, data is transmitted back to the BS using our modified version of geographical routing. In previous dissemination process modeling, we do not use any geographical information such as distance to the BS. However, in order to figure out the delay in aggregation phase, geographical information must be used. Again, the problem will be analyzed using probability model.

As vehicles are mobile with randomness, it is impossible to know the accurate location of each vehicle when it receives the broadcast. To model the problem, we will use average distance of all vehicles when receive the request. The average distance $h$ to the ROI can be represented as the mathematical expectation of $h$ or $E[h]$. If $h$ or $E[h]$ can be found, the delay can be calculated by distance divided by speed then.

Since vehicles are uniformly distributed in the ROI. We can assume the coordinate of vertex $v$ consists of two random variables denoted as $(x, y)$, then the joint density function of $x$ and $y$ can be given by:

$$f(x, y) = \begin{cases} 
\frac{1}{\pi d^2} & x^2 + y^2 \leq d^2 \\
0 & x^2 + y^2 > d^2 
\end{cases} \quad (5)$$

The probability of the distance to ROI is less than or
equal to $a$, where $0 \leq a \leq d$, can be calculated as:

$$F_h(a) = P\{ \sqrt{x^2 + y^2} \leq a \} = \int_0^a \int_0^{\sqrt{a^2 - y^2}} f(x, y) dy \, dx = \frac{\pi a}{2d^2} \int_0^a \sqrt{a^2 - y^2} \, dy \, dx$$

(6)

From Equation 6, we can get the probability density function (PDF) of all vehicles:

$$f_h(a) = \frac{dF_h(a)}{da} = \frac{2a}{d^2}, \quad 0 \leq a \leq d$$

(7)

Hence, using PDF, the mathematical expectation $E[h]$ can be calculated by:

$$E[h] = \int_0^d 2a \frac{a}{d^2} \, da = \frac{2}{d^2} \int_0^d a^2 \, da = \frac{2d}{3}$$

(8)

From Equation 8, we know that in probability point of view, all vehicles can be treated as at $\frac{2d}{3}$ far from the ROI when they receive the data collection request. We can calculate the time cost to transmit data from $\frac{2d}{3}$ to the BS. In our algorithm, there may exists duplicated data in the network. We can just consider the optimal condition, that is the BS receives the first copy of the data.

Considering one hop transmission range, the optimal solution is $r$, which means data can be transmitted as far as the transmission range in each hop. This situation is applicable when vehicle density is high. In contrast to sparse network, we assume that each vehicle has at least one neighbor, the mean transmission range is the same as Equation 8, that is $\frac{2d}{3}$. Therefore, the transmission time of the aggregation phase $t_d$ is bounded by:

$$\frac{2d}{3r} \leq t_d \leq \frac{d}{r}$$

(9)

Combine formula 4 and 9 together. The number of nodes aggregated at time $t$ can be calculated as $N(t) = R(t - t_d)$. That is:

$$N(t) = \frac{\theta \pi d^2}{1 + (\theta \pi d^2 - 1)e^{-(\theta \pi d^2 - 1)(t - \frac{t_d}{r})}}$$

(10)

$$N(t) = \frac{\theta \pi d^2}{1 + (\theta \pi d^2 - 1)e^{-(\theta \pi d^2 - 1)(t - \frac{r}{r})}}$$

(11)

Formula 10 is suitable for dense traffic while Formula 11 is suitable for sparse traffic. Considering the urban scenario, the traffic density is high for most of the time, we’ll use Formula 10 to evaluate the result in the following paragraphs.

B. Discussion of the model

In this section, we’ll discuss the characteristics of the model and how can the algorithm be adjusted to fit in with the specific requirement of application.

**Lemma 1:** $\lim_{t \to \infty} N(t) = \theta \pi d^2 = |V|

The proof is straight forward since $\lim_{t \to -\infty} e^t = 0$

Lemma 1 shows that given infinite time, data from all vehicles within ROI will finally be collected. However, in reality, it is impossible to wait infinitely to collect information from all vehicles. Luckily, for most of the applications, it is not necessary to wait for such a long time to collect all data in a region. What we need to do is just collecting the required amount of data.

**Lemma 2:** $\exists$ s.t. $Max(\lim_{t \to 0} \frac{\Delta N(t)}{\Delta t})$ exists.

**Proof:** Since $N(t) = R(t - t_d)$, we can discuss $R(t)$ instead to simplify the proof. To prove Lemma 2, we only need to prove $\frac{dR(t)}{dt}$ has a maximum value. The first derivative of formula 4 is:

$$\frac{dR(t)}{dt} = \frac{\pi d^2(\pi r^2 - 1)(\pi d^2 - 1)}{e^{(\pi r^2 - 1)t}(\pi d^2 - 1)(\pi r^2 - 1) + 1}$$

(12)

If the maximum value exists, it $t$ can be calculated by finding the stationary point of $\frac{dR(t)}{dt}$, that is $\frac{d^2R(t)}{dt^2} = 0$.

And the solution is:

$$t = \frac{\ln(\theta \pi d^2 - 1)}{\theta \pi r^2 - 1}$$

(13)

And it is easy to prove that $\frac{d^3R(t)}{dt^3}$ where $t$ equals to the value in Formula 13 is negative, which means it is the maximum value of $\frac{dR(t)}{dt}$.

In terms of $N(t)$, $\frac{dN(t)}{dt}$ reaches maximum value when

$$t = \frac{\ln(\theta \pi d^2 - 1) + d}{\theta \pi r^2 - 1}$$

(14)

The analytical output of $R(t)$ and $\frac{dR(t)}{dt}$ can be depicted as Fig. 4. Lemma 2 shows that data collection speed is not always the same. At the beginning of the collection process, the collection speed is relatively lower but increasing rapidly. However, the speed will not increase infinitely and has a peak value. With time passes by, the speed will reach the peak value and after that, it begins to slow down.

The conclusion of Lemma 2 can be used to estimate the density of vehicles if it is unknown. The idea is that if current density is unknown, the BS can set the TTL of the data collection process to a conservative huge value.
and starts the collection. After that, the BS waits for data to be transferred back. Meanwhile, it calculate the data collection speed. When the peak value occurs, the density can be get by solving Formula 14. In this case, all parameters except $\theta$ are all known.

The density estimation algorithm is shown in Algorithm 2. Line 2 and 3 are used to count the number of vehicles whose data have been received in the current duty cycle. Line 5 calculates the differential vehicle data received between current duty cycle and previous duty cycle. If the differential value is negative, it means the collection speed has arrived at the maximum value, the condition is tested in line 11. After that, the vehicle density can be estimated by the solving Equation 14.

Algorithm 2 Vehicle density estimation

Input: The received package: pkg

1: repeat
2: if $\text{CurrentTime} - t_{\text{Start}} < t_s$ then
3: $\text{cur} \leftarrow \text{cur} + \text{pkg}.\text{Datacount}$
4: else
5: $\text{delta} \leftarrow \text{cur} - \text{pre}$
6: $tLen \leftarrow tLen + 1$
7: $tStart \leftarrow \text{CurrentTime}$
8: $\text{pre} \leftarrow \text{cur}$
9: $\text{cur} \leftarrow 0$
10: end if
11: until $\text{delta} \geq 0$
12: Solve Equation 14 using $tLen$ to get $\theta$
13: return $\theta$

C. TTL Adjustment

Before sending the request, the BS need to estimate the time of the collection process. The the life time of the collection process is controlled by TTL. TTL is calculated and then put into the TTL field of the broadcast package. The value of TTL decreases 1 after each duty cycle. When TTL value reaches zero, the algorithm stops running at the same time on both the BS and the vehicles.

The data collection requirement of applications are defined as collecting data from certain number or ratio of vehicles. Notice that if vehicle density is known, absolute number and relative number can be transformed from each other. Here we use relative number for demonstration. To calculate the relationship between collection ratio and the required time, the only thing we need to do is solving equation $\frac{N(t)}{t} = p$, where $p$ is the percentage that required by the application. The solution is:

$$ t = \frac{d}{r} - \frac{\ln\left(\frac{p - 1}{r - \pi^2 \theta p}\right)}{\pi r^2 \theta - 1} \quad (15) $$

If the application does not provide such specific requirement, our algorithm will use the average data collection speed for metrics. The algorithm will try to collect certain amount of data to maximize the average data collection speed. The average data collection speed is defined as:

$$ E(t) = \frac{N(t)}{t} = \frac{\theta \pi d^2}{t(1 + (\theta \pi d^2 - 1)e^{-(\theta \pi d^2 - 1)(t - \frac{d}{r})})} \quad (16) $$

The average data collection speed $E(t)$ is a transcendental equation, its plot is illustrated in Fig. 5. From the plot, we can see that it indeed has a maximum value. However, The solution for $t$ to maximize $E(t)$ can not be represented as a elementary function. The solution can be expressed as:

$$ t = \frac{1 - W_{-1}\left(\frac{e}{1 - \pi r^2 \theta}\right)}{\pi r^2 \theta - 1} + \frac{d}{r} \quad (17) $$

Where $W_k$ is the Lambert W function [15] and $k$ means the $k$-th branch of this multi-valued function. In Equation 17, $W_{-1}$ means $k$ is -1.

Using Equation 15 and 17, the algorithm can estimate the TTL adaptively according to different requirements. Both Equation 15 and 17 rely on only three variables : $\theta$, $r$ and $d$. Among the three variables, transmission range $r$ is a fixed value and only depends on the capability of the network hardware. The collection range $d$ depends on the application, but it is an input of the algorithm and can also be treated as preknowledge. The only variable which depends on outside environment is the vehicle density $\theta$. That is the main reason why the algorithm is adjusted according to the vehicle density. In our solution, vehicle density can be estimated using Algorithm 2. As an alternative, it is also possible to use the historical data to estimate vehicle density.

V. EVALUATION

A. Experiment setup

To evaluate the performance of the proposed algorithm and the accuracy of the model analysis, we’ve implemented our solution and tested it using network simulator NS-3 [16].

The solution is implemented as an application protocol. It is built on top of the UDP and then IP protocol. In MAC and PHY layer, 802.11p is used, which is designed dedicatedly for vehicular network. Table II shows the parameters we used during the simulation. To connect
vehicles in a peer to peer manner, the MAC type is set to Adhoc. Considering the signal interference by the buildings and vehicles, the network transmission range is set to a conservative value, 70 meters. The length of duty cycle is set to 1 second, which means vehicles and BS transmit only once in a second. The build-in mobility model for NS-3 are not suitable for urban traffic. To make the simulation realistic, we implemented the Manhattan mobility model [17], which utilizes a common seen grid road topology and therefore more applicable for our scenario.

B. Result analysis

Firstly, we’re interested in the accuracy of the mathematical model. To verify the accuracy, we use different parameter configurations to run the simulation, and compare the simulation result with the analytical result. The vehicle density $\theta$ and radius of ROI $d$ are variables and different values will be used. For density, we use 0.0001 and 0.0002, which means there are 1000 or 2000 vehicles per square kilometer. For ROI radius, we use two values: 1000m and 2000m.

Fig. 6 depicts the comparison between simulation result and analytical result for data collection request dissemination phase with different parameter settings. It can be seen that the simulation curves fit the analytical function curves under the three preferences. It is also an interesting observation that increasing the size of ROI will not increase the data collection time significantly. The conclusion can be made from the first and the third plot, in which the radius of ROI $d$ is to 1000 and 2000 correspondingly. This is a major difference between networks with immobile and mobile nodes. It has been studied in [18].

Fig. 7 depicts the comparison between simulation result and analytical result for both dissemination phase and aggregation phase. Compared with previous experiment, the differences between simulation and analytical values are larger. The cause of the differences is the way we modeling the aggregation phase. The aggregation delay is modeled using mathematical expectation or mean value. However, mean values can not reflect the individual behavior of each single vehicle. The differences are significant especially at the beginning phase of data collection. For the analytical model, the first data arrives at the BS after $\frac{2d}{\theta}$. However, in real word, the first data will arrive much earlier than it. Despite of this, the overall collection time will not be affected and the result can prove this.

In addition, the overall length of the network package is another interesting metrics. In our solution, we keep appending data to the end of the broadcast package, if a vehicle carries too much data, it may cause some other problems like transmission delay and package fragmentation. In our experiment, we monitor the vehicles communicating with the BS, if the vehicles send a package with appended data, we’ll log the transfer and the size of package. Table III shows the result. The average package size varies from 0.6KB to 2KB. Considering the geographical routing algorithm, the closer to the BS, the larger will the package size be. It is an acceptable value.

C. Comparison study

To study the performance of the proposed data collection, we also compare our solution with classical AODV [19], which has also been implemented in NS-3. In data dissemination phase, we use the same implementation. After a node received the broadcast, we compare to use AODV and our solution to send the data back to BS. We focus on the total data transferred over VSNs in both cases.

Fig. 8 shows the comparison results in terms of total data sent. From the figure, we can see that the data amount of both protocols increases with number of vehicles.
However, in all cases, our proposed solution works better than AODV. This is mainly because in our solution, we utilize the broadcast in dissemination phase for data aggregation. The total amount of data transferred are reduced significantly. In AODV, data dissemination and data aggregation are completed separated. Resulting in additional overhead.

VI. CONCLUSION

In this work, we have presented a data collection method for VSNs. In our algorithm, we focus on how to collect required amount of data in minimum time. The algorithm is self-reliance since it can estimate current vehicle density as collecting data and then use it to adjust the further collection time. The experiment have shown that our analytical model match the experiment result well. Evaluation results also show that our solution outperforms AODV based solution in terms of total amount of data transferred via VSNs.

REFERENCES


Zongjian He received his B.S. degree in software engineering from Tongji University, China in June 2004 and M.S. degree in computer science from Tongji University, China in April 2007. He is now working at School of Software Engineering, Tongji University, China. His current research interests include vehicular networks, community sensing and mobile computing.

Huijuan Zhang received the B.S., M.S. and Ph.D. degree from Xidian University, China in 1993, 1996 and 2005, respectively. She is now an associate professor at School of Software Engineering, Tongji University, China. Her research interests include game theory, wireless networks and embedded systems.