Towards Approximate Spatial Queries for Large-scale Vehicle Networks

Lipeng Wan[*]
Dept. of Electrical Engineering
and Computer Science
University of Tennessee
lwan1@utk.edu

Zhibo Wang
School of Computer
Wuhan University, China
zwang32@utk.edu

Zheng Lu
Dept. of Electrical Engineering
and Computer Science
University of Tennessee
zlu12@utk.edu

Hairong Qi
Dept. of Electrical Engineering
and Computer Science
University of Tennessee
hqi@utk.edu

Wenjun Zhou
Dept. of Statistics, Operations
and Management Science
University of Tennessee
wzhou4@utk.edu

Qing Cao
Dept. of Electrical Engineering
and Computer Science
University of Tennessee
cao@utk.edu

ABSTRACT

With advances in vehicle-to-vehicle communication, future vehicles will have access to a communication channel through which messages can be sent and received when two get close to each other. This enabling technology makes it possible for authenticated users to send queries to those vehicles of interest, such as those that are located within a geographic region, over multiple hops for various application goals. However, a naive method that requires flooding the queries to each active vehicle in a region will incur a total communication overhead that is proportional to the size of the area and the density of vehicles. In this paper, we study the problem of spatial queries for vehicle networks by investigating probabilistic methods, where we only try to obtain approximate estimates within desired confidence intervals using only sublinear overheads. We consider this to be particularly useful when spatial query results can be made approximate or not precise, as is the case with many potential applications. The proposed method has been tested on snapshots from real world vehicle network traces.

Categories and Subject Descriptors

G.1.2 [**Approximation**]: Nonlinear approximation

General Terms

Algorithms

Keywords

Spatial query, Nonlinear counting

1. INTRODUCTION

In recent years, large-scale V2V (vehicle-to-vehicle) communication has been envisioned for various application needs such as ride sharing, traffic scheduling, and accident avoidance. In February 2014, the U.S. Department of Transportation's (DOT) National Highway Traffic Safety Administration (NHTSA) announced that it will begin to require vehicle-to-vehicle (V2V) communication technology to be installed for light vehicles in the coming years [1]. In such applications, frequently a user must query the status of the vehicles located within a geographic region to decide the next step of operation.

However, current approaches for this seemingly simple task are still highly inefficient, and may be slowed down when many concurrent queries are being executed. First, existing methods typically involve flooding, which is not desirable or practical when the number of vehicles becomes relatively large. Second, to obtain information on the desired phenomenon, existing methods must perform the query operations in a timely manner, or even periodically, leading to excessive communication overhead for the limited V2V communication channel.

To address these challenges, in this paper, we propose a novel mechanism that performs query operations of large-scale vehicle networks in an approximate manner. We use the measurement of cardinality as a running example, but our methods can be easily extended to queries of more general nature, such as counting only those vehicles of interest. This mechanism stems from the observation that counting the exact number of vehicles for a query, while desirable, is usually not absolutely necessary for the following two reasons. First, vehicles are inherently mobile, meaning that even an accurate answer will expire in a short period of time. Second, for many applications, obtaining an approximate estimate with a confidence level on its accuracy is sufficient. By relaxing the requirement on accuracy, we find that the approximate algorithm only requires $\mathcal{O}(\sqrt{n})$ messages to be transmitted, as opposed to $\mathcal{O}(n)$ or even $\mathcal{O}(n^{3/2})$ messages for accurate counting algorithms.

Our main idea originates from a well-known problem called *the birthday problem.* This problem considers, for a group of n randomly chosen people, the probability that at least

[*]Lipeng Wan and Zhibo Wang are co-first authors.

SIGSPATIAL'14, November 04 - 07 2014, Dallas/Fort Worth, TX, USA
Copyright 2014 ACM 978-1-4503-3131-9/14/11 ...$15.00
http://dx.doi.org/10.1145/2666310.2666490.

two have the same birthday. By the pigeonhole principle, the probability reaches 100% when the number of people reaches 367. However, with far fewer people, we can still obtain a high probability of birthday collisions.

Related work. In the domain of vehicle networks, previous research has studied multiple challenges such as trajectory mining [6,8], neighbor query and routing [3,4], and ride sharing [5]. Query vehicles in a region, however, has been relying mostly on video processing techniques [2, 7]. Such methods usually take advantage of recent progress in computer vision, and are able to detect specific image features such as plate numbers from captured images. Therefore, compared to our approach, such methods are completely different in that they are passive and require the deployment of large-scale infrastructures.

2. PROBLEM FORMULATION AND PROTOCOL DESIGN

In this section, we describe the problem formulation and provide an overview of the protocol. We assume that at the moment of query, vehicle nodes of interest are distributed in a flat, geographic region with a known boundary. We do not make assumptions on the geographic distribution of vehicles, i.e., they have irregular densities due to the road and traffic conditions. We assume that each vehicle is equipped with a transmitter that allows it to communicate with vehicles within its proximity.

We consider in a region, there are an unknown number of vehicles that have a certain feature of interest. Our goal is to query this region to find this unknown number. We denote the total number of nodes as f. Our goal is to obtain its approximation $\tilde{f}$ within a confidence interval ϵ and an error probability of δ, so that $Pr\{|\tilde{f} - f| \leq \epsilon f\} \geq 1 - \delta$.

The core of our protocol is an algorithm that consists of two stages: in the first stage, the query node sends probes to the network via *random-cast*, which ensures that the probes are distributed in an uniform manner. Every probe contains a round number, a sequence number, and a reporting period. When a destination node receives two or more probes with the same round number, it will send back a response to the query node containing the number of collisions it has detected at the end of the reporting period through geographic forwarding. The query node keeps track of the reported collisions, as well as the number of probes it has sent so far. At the end of each reporting period, the query node updates the estimate and the confidence interval. If the confidence interval satisfies the requirement, the query node stops sending probes. Otherwise, the query node will send additional probes with extended reporting periods to trigger more collisions, so that the estimation accuracy can be improved.

A random-cast is a communication process where the sender sends a packet to a receiver that is randomly and uniformly distributed in a set of potential candidates, C, such that each candidate has the same probability of receiving this packet. Our design of the random-cast protocol combines geographic forwarding with random walking, which leverages the spatial and topological distribution of the nodes to achieve the overall uniformity. The algorithm steps are shown in Algorithm 1.

One critical component of this algorithm is to perform random-walking after geographic forwarding. Specifically, random-walking serves the goal that the walk must terminate at a node i belonging to the network with an equal probability.

Algorithm 1 The Random-cast algorithm

```
assumption: node deployment boundary $S$ is known
query node randomly selects a target location $L$ in $S$
query node sends the probe to the node $N$ nearest to $L$
$current\_node \leftarrow N$
$walking\_hops \leftarrow 0$
while $walking\_hops < Max$ do
    $current\_node$ picks the next hop $next\_node$ with the RWD method
    $current\_node$ sends the packet to $next\_node$
    $walking\_hops \leftarrow walking\_hops + 1$
    $current\_node \leftarrow next\_node$
end while
```

Note that the key advantage of this random-walking based design is that it is not affected by the real topology of nodes in an area. Even in the case that nodes are densely packed on a road, the random nature of the delivery process ensures that packet reception is decoupled with the actual density of nodes. Hence, it is sufficiently robust to irregularities in road and traffic conditions.

3. CARDINALITY ESTIMATION IN ASQ

In this section, we describe the estimation step of the ASQ protocol.

3.1 Expected Collisions and Its Variance

We first derive the expected value of collisions and its variance, so that we can develop an unbiased estimator for the total number of nodes. Assume that we have t probes as sequentially distributed into the network. Each probe may arrive at any of the f nodes. The number of collisions that are introduced when a probe lands at a node, which is a random variable, as C_i, for the ith probe. The probability that $C_i = 1$ is:

$$P(C_i = 1) = 1 - (1 - \frac{1}{f})^{i-1} \tag{1}$$

Therefore, the expected total number of collisions for these t probes is:

$$\mu = \sum_{i=1}^{t} C_i = t - f + \left(1 - \frac{1}{f}\right)^{t} f \tag{2}$$

Note that we are treating the collision between a new probe and each of the previous probes individually. That is, suppose that two previous probes both landed on the same node. If a third probe lands on the same node, the number of collisions for this new probe should be counted as 2 instead of 1.

To simplify our notations, we define $\tau = t/f$. Considering that f is relatively large, we can write the result above as:

$$\mu \simeq f(e^{-\tau} - 1) + t \tag{3}$$

or:

$$\mu/f = e^{-\tau} + \tau - 1 \tag{4}$$

Similarly we can decide the variance of C_i as:

$$Var(C_i) = (1 - \frac{1}{f})^{i-1} \times (1 - (1 - \frac{1}{f})^{i-1}) \tag{5}$$

The total variance is determined to be:

$$\begin{aligned}\sigma^2 &= \sum_{i=1}^{t} Var(C_i) \\ &= \frac{\left(-1+\left(\frac{-1+f}{f}\right)^t\right) f \left(1+\left(-1+\left(\frac{-1+f}{f}\right)^t\right) f\right)}{-1+2f}\end{aligned} \tag{6}$$

Following a similar procedure above, by replacing $\left(\frac{-1+f}{f}\right)^t$ by $e^{-\tau}$, we can have:

$$\sigma^2 = \frac{f\left(e^{-\tau}-1\right)\left(1-f+fe^{-\tau}\right)}{2f-1} \tag{7}$$

Observe that based on Eq. (3), we can obtain an estimate of f given a known t. However, this equation does not give us a closed-form estimator for f due to its non-linear nature. Therefore, simple numerical methods such as Newton's can be used.

3.2 Estimator Accuracy Analysis

According to the central limit theorem, the sum of independent variables can be approximated by normal distributions. Thus we know that approximately, if we denote the number of collisions as random variable X, we have $X \sim \aleph(\mu, \sigma^2)$, leading to the following theorem:

Theorem 3.1. (Collision Estimation) *In a network of f nodes, let the query node send t packets following the randomcast protocol, and let N represent the total number of collisions (note that if multiple packets are received at the same node, this is counted multiple times), if $t/f = \tau$, we have $N \sim \aleph(\mu, \sigma^2)$ where*

$$\mu = f(e^{-\tau} + \tau - 1) \tag{8}$$

and

$$\sigma^2 = \frac{f\left(e^{-\tau}-1\right)\left(1-f+fe^{-\tau}\right)}{2f-1} \tag{9}$$

Recall our goal is to estimate f from the observed number of collisions. Note that the expectation of number of collisions, μ, is monotonically decreasing with f. If we define the inverse function of Eq. (3) as $g()$, we have $g(\mu(f)) = f$. Similarly, when f increases and t fixed, σ decreases. Finally, as the number of nodes increases, if the number of probes remains the same, the inaccuracy will monotonically grow, i.e., the ratio between σ and μ increases.

Based on these results, we can predict the operation of the ASQ protocol as follows: as the query node continues to send more probes, more collisions will occur, leading to lower ratios between σ and μ, which indicates more accurate estimations. Beyond a certain point, the query node will always stop sending new probes after it obtains an estimate that is accurate enough.

3.3 Extreme Cases and Closed-form Results

Although in the general case, we cannot derive a closed-form estimator for the total number of nodes f, we can obtain an approximate estimation of it by observing that f is assumed to be quite large in the network. Therefore, we can approximately replace it with ∞, and derive the limit for μ. However, the limit of Eq. (3) does not give us a closed-form result either. To get a closed-form result, we have to define yet another auxiliary parameter, ρ, as follows:

$$\rho = \frac{t}{\sqrt{f}}$$

The meaning of ρ is that we can consider the value of t in the order of $\mathcal{O}(\sqrt{f})$, and ρ represents the constant factor of this order. Hence, we can obtain the following limits for μ and σ:

$$\lim_{f\to\infty} \mu = \rho^2/2 \tag{10}$$

$$\lim_{f\to\infty} \sigma^2 = \rho^2/2 \tag{11}$$

According to Theorem 3.1, we know that the number of collisions follows the normal distribution $X \sim \aleph(\mu, \sigma^2)$. Through approximation, this shows that when the number of nodes is sufficiently large, the normal distribution becomes $X \sim \aleph(\rho^2/2, \rho^2/2)$.

3.4 Estimation of Confidence Intervals

We next describe how we obtain the confidence interval estimation with only one single measurement of the collision count. This problem is challenging due to that we can only observe the collision count once for overhead reasons. We use $Z_{\alpha/2}$ to denote the α percentile for the unit normal distribution. For example, if $\alpha = 95\%$, we have $Z_{\alpha/2} = 1.96$.

Intuitively, the estimation for confidence intervals works as follows: given an observed N_c as the collision number and the number of sent probes as t, we know that it follows the distribution of $\aleph(\mu(f), \sigma^2(f))$. Therefore, with a confidence of α, the observed value is located within the range of $[\mu(f) - Z_{\alpha/2}\sigma(f), \mu(f) + Z_{\alpha/2}\sigma(f)]$. Because μ and σ are both monotonically decreasing with f, we only need to obtain f_{max} and f_{min}, where:

$$\mu(f_{min}) - Z_{\alpha/2}\sigma(f_{min}) = N_c \tag{12}$$

$$\mu(f_{max}) + Z_{\alpha/2}\sigma(f_{max}) = N_c \tag{13}$$

This gives the upper and lower bounds for estimating f, which can be calculated according to Eqs. (8) and (9). Moreover, the best estimate of f itself can be calculated by

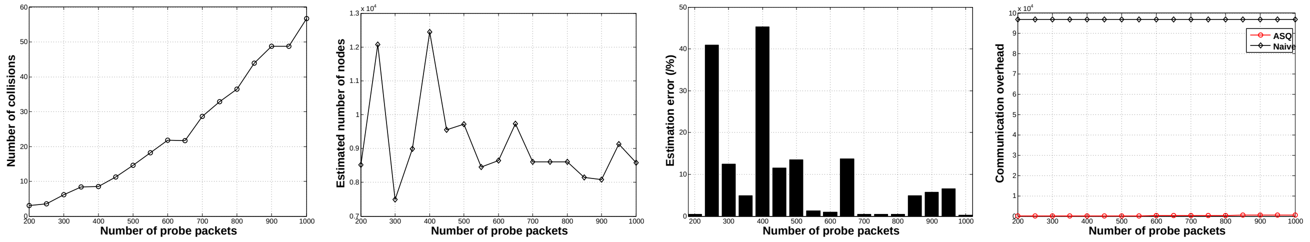
$$\mu(\tilde{f}) = N_c \tag{14}$$

In our algorithm, confidence intervals were used to determine when to stop sending probe packets. However, since the true number of nodes, f, is always unknown to us, it is impossible to calculate the accurate confidence interval. To solve this problem, we use $(f_{max} - f_{min})/\tilde{f}$, referred to as "confidence interval ratio", to help decide whether more probe packets should be sent or not.

4. PERFORMANCE EVALUATION

4.1 Parameter settings

In this section, we conduct extensive simulations to evaluate the cardinality estimation. We use a graph derived from snapshots of real, open source taxi traces in Beijing, China [9]. There are 8563 vehicle nodes, and the communication range is set to 1000m. The topology of this snapshot is shown in Fig. 1. We use the following metrics to evaluate the performance of the proposed algorithm.



(a) Number of Collisions (b) Estimated Cardinality (c) Estimation Error (d) Communication Overhead

Figure 2: Performance of ASQ.

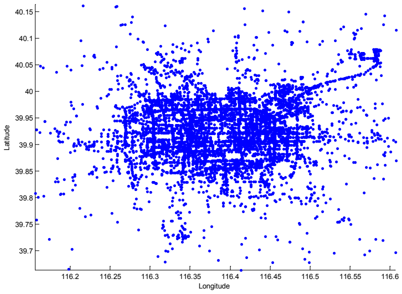


Figure 1: Topology of a snapshot of a real vehicle network (location: Beijing)

- Estimated cardinality $\tilde{f}$: the estimated number of nodes in the network;
- Estimation error: $|\tilde{f} - f|/f * 100\%$;
- Number of collisions: the total number of collisions obtained at the query node for all the probe packets;
- Communication overhead for reporting: the total number of packets needed to report collisions to the query node.

Note that each data point in the following figures is an average of 10 experiments.

4.2 Evaluation results

Fig. 2 shows the experimental results of using the ASQ to estimate the cardinality for the snapshot graph.

As shown in the first part of Fig. 2, the number of collisions increases when more probe packets are sent, which improves the estimation accuracy of the cardinality.

The second and third part of Fig. 2 show the estimated cardinality and its corresponding estimation error for ASQ, respectively. We can observe that ASQ achieves much better estimation results when the number of probe packets is large. Due to the lack of collisions, when the number of probe packets is very low, the performance of ASQ is compromised. It is worth noting that the ASQ can achieve a very low estimation error with only $\rho\sqrt{f}$ probe packets, where ρ represents a constant factor that can be empirically set, which is consistent with our analysis on extreme cases in Section 3.3.

The ASQ has very low communication overhead compared to naive approach, where each node reports a message to the query node (which has a communication overhead about the same as the total number of nodes). As shown in the last part of Fig. 2, due to the increase of collisions, ASQ consumes larger communication overhead. However, note that even with large number of collisions, ASQ still consume much smaller communication overhead than the naive approach, Therefore, the ASQ is energy efficient and can achieve very high estimation accuracy.

5. CONCLUSIONS

In this paper, we presented the design, analysis, and evaluation of ASQ, an approximate spatial query protocol for large scale vehicle networks based on only V2V communication. Our results show that this method achieves overhead that is an order of magnitude lower than the naive method that requires all nodes to report their presence, yet still provides sufficiently accurate estimates for most application uses.

Acknowledgements

The work reported in this paper was supported in part by the National Science Foundation grant CNS-0953238 and CNS-1239478.

6. REFERENCES

[1] DOT Website. http://www.nhtsa.gov/About+NHTSA/Press+Releases/2014/USDOT+to+Move+Forward+with+Vehicle-to-Vehicle+Communication+Technology+for+Light+Vehicles.

[2] Erhan Bas, A. Murat Tekalp, and F. Sibel Salman. Automatic Vehicle Counting from Video for Traffic Flow Analysis. *Intelligent Vehicles Symposium*, 2007.

[3] Parisa Ghaemi, Kaveh Shahabi, John P. Wilson, and Farnoush Banaei Kashani. Continuous maximal reverse nearest neighbor query on spatial networks. In *SIGSPATIAL/GIS*, pages 61–70, 2012.

[4] Abdeltawab M. A. Hendawi and Mohamed F. Mokbel. Panda: a predictive spatio-temporal query processor. In *SIGSPATIAL/GIS*, pages 13–22, 2012.

[5] Shuo Ma and Ouri Wolfson. Analysis and evaluation of the slugging form of ridesharing. In *SIGSPATIAL/GIS*, pages 64–73, 2013.

[6] Leon Stenneth, Ouri Wolfson, Philip S. Yu, and Bo Xu. Transportation mode detection using mobile phones and gis information. In *GIS*, pages 54–63, 2011.

[7] Kunfeng Wang, Zhenjiang Li, Qingming Yao, Wuling Huang, and Fei-Yue Wang. An Automated Vehicle Counting System for Traffic Surveillance. *Vehicular Electronics and Safety, IEEE International Conference on*, 2007.

[8] Josh Jia-Ching Ying, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S. Tseng. Semantic trajectory mining for location prediction. In *GIS*, pages 34–43, 2011.

[9] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. T-drive: driving directions based on taxi trajectories. *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 99–108, 2010.