Persistent Traffic Measurement through Vehicle-to-Infrastructure Communications in Cyber-Physical Road Systems

Yu-e Sun, *Member, IEEE*, He Huang, *Member, IEEE*, Shigang Chen, *Fellow, IEEE*, Hongli Xu, *Member, IEEE*, Kai Han, *Member, IEEE*, and Yian Zhou, *Member, IEEE*

Abstract—Measuring traffic volume in a road system has important applications in transportation engineering. The connected vehicle technologies integrate wireless communications and computers into transportation systems, allowing wireless data exchanges between vehicles and road-side equipment, and enabling large-scale, sophisticated traffic measurement. This paper investigates the problem of persistent traffic measurement, which was not adequately studied in the prior art, particularly in the context of intelligent vehicular networks. We propose three estimators for privacy-preserving persistent traffic measurement: one for point traffic, one for point-to-point traffic, and another for three-point traffic. After that, we present a general framework to measure persistent traffic that go through more than three locations. The estimators are mathematically derived from the join result of traffic records, which are produced by the electronic roadside units with privacy-preserving data structures. We evaluate our estimation methods using simulations based on both real transportation traffic data and synthetic data. The numerical results demonstrate the effectiveness of the proposed methods in producing high measurement accuracy and allowing accuracy-privacy tradeoff through parameter setting.

Index Terms—Vehicular networks, traffic measurement, privacy

1 Introduction

Measuring traffic volume at points of interest in road systems provides important information for transportation engineering. These point traffic data are useful in estimating traffic link flow distribution as part of investment plan and calculating road exposure rates as part of safety analysis. Much prior research on traffic measurement collects statistics on the number of vehicles passing a certain location during a certain measurement period, often in the form of annual average daily traffic (AADT). Various predication models [4], [10], [16], [18], [23], [26] have been developed based on the data recorded by roadside units (RSU) installed at road intersections. An example is Mohammed's multiple linear regression model that incorporates demographic variables to measure AADT [16]. Another example is Lam's artificial neural network that estimates AADT based on short period counts [10]. Other research work includes the spatial statistical method by Eom et al. [4], the support vector regression model by Neto et al. [18], the absolute deviation penalty procedure by Yang et al. [26], and the regression and Bayesian model by Tsapakis et al. [23].

The emergence of connected vehicle technologies in the intelligent transportation systems promises radical changes in how transportation traffic measurement will be conducted. The trend is to integrate wireless communications and computers into vehicular cyber-physical systems for better road safety and driving experience [5], [12]. Traffic data collection will become more sophisticated with vehicular communications and networking [15], [19], [22], [33], [34], such as the Dedicated Short Range Communications standard under IEEE 802.11p [17], which supports wireless data exchanges between vehicles and RSUs.

Such automated systems have been exploited in prior research for collecting point-to-point transportation statistics, i.e., the number of vehicles traveling between any two points (locations) of interest during a certain measurement period in a road system [30], [32]. Point-to-point data provide important input to a variety of transportation studies such as identifying the real sources of traffic congestion and characterizing turning movements at intersections for signal timing determination [24]. There are two performance considerations: The obvious one is the accuracy of traffic measurement. The less obvious one is privacy concern. When vehicles are equipped for wireless communications, there are easy ways to ensure the measurement accuracy. For instance, we may require all vehicles to report their unique IDs to the RSUs that they encounter. In this way, we will be able to figure out the point-to-point traffic volume by

- *Y. Sun is with the School of Rail Transportation, Soochow University, Suzhou, Jiangsu 215006, China. E-mail: sunye12@suda.edu.cn.*
- *H. Huang is with the School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006, China. E-mail: huangh@suda.edu.cn.*
- *S. Chen is with the Department of Computer and Information of Science and Engineering, University of Florida, Gainesville, FL 32611. E-mail: sgchen@cise.ufl.edu.*
- *H. Xu and K. Han are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China. E-mail: {xuhongli, hankai}@ustc.edu.cn.*
- *Y. Zhou is with Google Inc., Mountain View, CA 94043. E-mail: yianzhou@google.com.*

Manuscript received 4 Sept. 2017; revised 29 Mar. 2018; accepted 23 July 2018. Date of publication 6 Aug. 2018; date of current version 31 May 2019. (Corresponding author: He Huang.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2018.2863296

comparing the ID sets from two RSUs. However, if a vehicle keeps transmitting its ID to RSUs, its entire moving history is recorded in great details. Such large-scale, universal tracking of movement raises privacy concern [30], [32].

In this paper, we take a step further to study a new problem of persistent traffic volume measurement. After we measure the point traffic volume at a certain location over time for many measurement periods, we naturally want to mine the data for more knowledge. Given a certain number of measurement periods, the point persistent traffic is defined as the set of vehicles that pass the location in all those periods. The rest is treated as transient traffic. For example, we may want to learn the persistent traffic volume over the workdays of a week, over the Saturdays of several weeks, or on all days in a month. Such data tells us the amount of core, stable traffic at a location, as the transient traffic varies over time. Similarly, after measuring the point-to-point traffic volume between two locations for many measurement periods, we want to know the persistent traffic volume containing common vehicles that show up during each period from one location to the other. For example, if a location is consistently congested, we may find the sources of the traffic based on point-to-point traffic measurement. In particular, the persistent point-to-point traffic tells us the minimum amount of traffic contribution that we can always expect from each of those sources. Finally, we generalize the problem to persistent multi-point traffic measurement for the volume of common vehicles that show up at multiple given locations persistently over a number of periods. This provides more detailed information about consistent traffic on given segments of a road system.

The problem of persistent traffic measurement is challenging if we want to achieve both measurement accuracy and privacy protection (which prevents even the authority from learning the trajectories of the moving vehicles). In this paper, we propose three estimators for privacy-preserving persistent traffic measurement: one for point traffic, one for point-to-point traffic, and another for three-point traffic. We then present a general framework to measure persistent traffic that goes through more than three locations. Our basic idea is for every RSU to encode the vehicles passing by during each measurement period in a privacy-preserving bitmap, called traffic record, where each vehicle is recorded by setting a randomly selected bit in the bitmap and the information of all vehicles is mixed together; vehicle identities are never transmitted or stored. To estimate persistent traffic, we combine the information of traffic records produced during different periods or locations of interest by performing bitwise AND (or bitwise OR). We observe that the volume of persistent traffic has a functional relationship with the number of ones (or zeros) in the combined bitmap after bitwise operation. We derive that function such that it can be used to estimate the volume of persistent traffic from the ones (or zeros) observed in the combined bitmap. We show that this estimation method can be effectively applied to both point persistent traffic and point-to-point persistent traffic, with varying local traffic volumes at different points (locations). We evaluate our persistent traffic estimators using simulations based on both real transportation traffic data and synthetic data. The extensive simulations demonstrate the effectiveness of the proposed methods in producing persistent traffic estimation of high accuracy and allowing accuracy-privacy tradeoff through parameter setting.

The rest of the paper is organized as follows. We first present the persistent traffic measurement model, and give the details of traffic record and vehicle encoding in Section 2. Then, we propose mechanisms for persistent point traffic measurement and persistent point-to-point traffic measurement in Sections 3 and 4, respectively. To evaluate the performance of the proposed mechanisms, we use both real transportation traffic and synthetic traffic to perform simulations in Section 7. At last, we conclude the paper in Section 9.

2 Preliminaries

2.1 Persistent Traffic Measurement

We study an intelligent transportation system with vehicle-to-infrastructure communication capability. Road-Side Units (RSUs) are deployed at locations of interest, such as street intersections. All RSUs are connected wirelessly or by wire to a central sever, where data are collected and processed for transportation traffic management functions. Each vehicle also has a unique ID. Vehicles communicate with the RSUs through DSRC [17], WiFi, or Bluetooth. For privacy protection, a vehicle should not directly transmit its ID to an RSU. In our solution, it will instead transmit a one-way hash value (which may change from location to location).

Traffic measurement is performed in each measurement period (e.g., a day), whose length is set as needed by the authority. During an arbitrary period, each RSU records the passing vehicles in a privacy-preserving data structure, called *traffic record*, without keeping any identifying information such as vehicle IDs. We study the following problems.

First, consider a single location L and a set Π of traffic records produced from the RSU at L during a number of measurement periods — for example, records from Monday through Friday of a certain week, records from Mondays of three consecutive weeks, or several records of interest based on any other criterion. A *common vehicle* refers to a vehicle that passes location L in all the measurement periods of interest. All common vehicles form the *persistent traffic*. The first problem, called *point persistent traffic measurement*, is to use the traffic records in Π to estimate the volume of persistent traffic, i.e., the number of common vehicles passing L.

Next, consider two locations, L and L'. Let Π be a set of traffic records produced by the RSU at L during a number of measurement periods of interest, and Π' be the set of traffic records produced by the RSU at L' during the same measurement periods. With respect to two locations, a *common vehicle* refers to a vehicle that passes both locations in all the measurement periods of interest, and accordingly the *point-to-point persistent traffic* is the aggregate of such vehicles. The second problem, called *point-to-point persistent traffic measurement*, is to use Π and Π' to estimate the volume of point-to-point persistent traffic, i.e., the number of common vehicles that pass both L and L'.

Finally, consider d locations, $L_1, L_2, \ldots, L_d$, and a number of measurement periods of interest. Let Π_i be the set of traffic records produced by the RSU at L_i during those periods. Define a *d-point common vehicle* as one that passes the d locations in all the measurement periods of interest, and accordingly the *d-point persistent traffic* as the aggregate of such vehicles. Our third problem, called *d-point persistent traffic measurement*, is to use $\Pi_1, \Pi_2, \ldots, \Pi_d$ to estimate the volume of d-point persistent traffic, i.e., the number of d-point common vehicles during the given periods.

2.2 Security and Threat Model

Vehicles will only interact with RSUs from trustworthy authorities. This can be easily enforced through authentication based on PKI. Communications begin with an RSU broadcast beacons, each carrying its public-key certificate, which was obtained from a trusted thirty party and was pre-installed with the RSU. When a vehicle receives a beacon, it uses its pre-installed public key of the trusted third party to verify the certificate. If not successful, the vehicle will keep silent; otherwise, it performs authentication with the RSU using the latter's public key obtained from the verified certificate. After successful authentication, it performs vehicle recording with the RSU as will be explained in the next subsection, with all data exchanges encrypted. Rogue RSUs may be deployed by non-authorities; they will fail the authentication with the vehicles, which will reject further communications.

We assume a semi-trusted model for the authorities. The transportation authority has good faith in implementing the proposed privacy-preserving methods since their goal is not to track people, but only to gather transportation traffic, which provides input for city development planning (without any real-time or short-term consequences). Their RSUs will communicate with the passing vehicles and perform all operations as expected. However, as the traffic records are produced and stored. At a later time, other people (such as police or FBI) who gain access to the records may exploit the information to track individual vehicles when they have the need to do so. For instance, if a hypothetical system design requires every vehicle to transmit its unique identifier to each encountered RSU, then these recorded identifiers may be used to track the trajectory of any vehicle. In order to prevent this from happening, it is highly desired that a vehicle will not transmit its unique ID, nor transmit any other fixed number to the RSUs that it passes.

Moreover, we assume that an anonymous MAC protocol such as SpoofMAC [20] is used to support privacy preservation such that the MAC address of a vehicle is not fixed. With such a protocol, before a vehicle communicates with an RSU, it picks a temporary MAC address randomly from a large space for one-time use, which prevents the MAC address from serving as an identifier of the vehicle.

2.3 Performance Metrics

We consider the following two performance metrics to evaluate persistent traffic measurement.

1. *Estimation Accuracy*: Let n_* be the actual volume of persistent traffic, i.e., number of common vehicles passing one location (or two locations) during the measurement periods of interest. Let $\hat{n_*}$ be the volume estimated based on the traffic records. We measure the estimation accuracy by evaluating the relative error, $\frac{|\hat{n_*}-n^*|}{n_*}$. A good traffic measurement method is expected to have close-to-zero relative errors.

2. *Preserved Privacy*: The essence of privacy preservation in transportation traffic measurement is to allow the tracker only a limited chance to identify any part of the trajectory of any vehicle. Following [30], [32], we want to make sure that anyone that possesses the traffic records cannot definitively determine any trace of any vehicle. In general terms, the traffic records may indicate that a vehicle has passed from one location to another location when the vehicle actually did not, and the records may indicate that the vehicle has not passed from one location to another location when the vehicle actually did.

As we will see shortly, the traffic records are probabilistically constructed. Consider two arbitrary locations, L and L', and an arbitrary vehicle v that is known to have passed L — for example, the vehicle is captured by a camera at the location, or it is stopped by a police car for speeding at the location. Now, the privacy concern on our traffic measurement function is that its traffic records may reveal additional information about the vehicle. More specifically, by looking for the trace that v leaves in the traffic records at L with a similar trace left at another location L', one might figure out that v has also passed L', which reveals the vehicle's moving trajectory. For example, suppose we record each vehicle by setting a bit in a bitmap maintained at L (L'), and further suppose we know which bit is set by v at L. By examining the bitmap recorded at L', if we see the bit b at the same index is also set, we may claim (or suspect) that v has also passed L'. To address this problem for better privacy protection, we want to design our traffic records in such a way that bit b at L' may be set even when v does not pass L' and it may not be set when v actually passes L'. We propose a quantitative metric, called preserved privacy, to measure the effectiveness of such a design. It is defined as $\frac{p}{p'-p}$, where p is the probability for the traffic records to claim v has passed L' even though the vehicle actually did not, and p' is the probability for the traffic records to claim that v has passed L' when it actually did so. Intuitively speaking, p is the "noise" term introduced by design to cause false claims, and $(p'-p)$ is the "information" term due to the vehicle's presence at L' after the impact of noise is removed. We use the "noise-to-information" ratio to characterize the level of privacy protection. For example, suppose $p = 10\%$ and $p' = 11\%$. The difference is just 1 percent. Whether or not the traffic records will claim v passes L' has little to do whether this actually happens. In other words, such claims are very unreliable, with the noise term (10 percent) being ten times of the information term (1 percent). In another example, if p is 0, the metric $\frac{p}{p'-p}$ will also be zero. In this case, when the traffic records claims that v has passed L', this will already be true. On the other hand, the value of this metric is large, it means that the noise term p is large, relative to the information term $(p'-p)$, and therefore when the traffic records claim the presence of v in another location, this claim is very questionable.

2.4 Traffic Record and Vehicle Encoding

Consider an arbitrary RSU installed at a certain location and an arbitrary measurement period. The data structure of traffic record is a bitmap B of m bits. Each vehicle that passes the RSU during the period is encoded by a bit, which is pseudo-randomly selected from B in a way that masks the identity of the vehicle yet leaves a probabilistic signature, allowing statistical analysis for traffic volume. The size of B, i.e., the value of m, may differ at different RSUs or at different measurement periods for the same RSU. We will come back to set m later.

The basic observation is that there is a functional relationship between the number of ones (or zeros) in B and the number of vehicles encoded—the more the number of vehicles is, the more the number of ones in B will be. Based on that function, we can estimate the number of vehicles from the number of ones. The problem of persistent traffic measurement will be more difficult as we need to combine the information from the traffic records of different periods

to figure out the number of common vehicles, which we discuss in the next two sections. Below we define how the traffic record B is constructed in each measurement period. In order to support privacy, we want to mix the information from different vehicles in B. The vehicle-encoding method should have the following properties: (1) When vehicles are encoded at a certain location, different vehicles may be probabilistically encoded by the same bit. (2) When a vehicle passes multiple locations (RSUs), it may be encoded at different bit indices. Together, they break the one-to-one association between vehicles and bits.

The traffic record is constructed as follows: At the beginning of each measurement period, the bits in B are reset to zeros. The RSU broadcasts beacons in preset intervals, such as once per second, ensuring that each passing vehicle will be able to receive a beacon, which carries the RSU's location L, its public-key certificate, and the size m of its bitmap. After a vehicle receives a beacon, it verifies the certificate and uses the public key to authenticate the RSU. After verifying that the RSU is from a trusted authority, the vehicle computes the following hash output: $h_v = H(v \oplus K_v \oplus C[H(L \oplus v) \bmod s]) \bmod m$, where H is a hash function that provides good randomness, v is the vehicle ID, K_v is a private key known only by the vehicle, L is the location of the RSU, and C is an array of s randomly selected constants. Because h_v is a function of L, its value may be different at different locations; the system parameter s controls the number of different values that h_v may take; the use of randomized constants in C helps improve the quality of input to the outer hash. The vehicle transmits h_v to the RSU, which will in turn set the bit at index h_v to one, i.e., $B[h_v] = 1$. That is the only operation of vehicle encoding. At the end of each measurement period, the RSU will send the content of the bitmap B as its traffic record to the central server, where queries may be submitted from the users to estimate persistent traffic.

The index h_v produced from a vehicle is not predictable by others because the private key K_v is not known. Moreover, the array C of constants are also known only to the vehicle. During a measurement period, many vehicles may pass an RSU. Due to vehicles' random selection of bits to set, different vehicles may choose the same bit as a result of hash collision. The same vehicle may choose different indices at different locations because the hash output is also dependent on the location L. Such mixing and variation in vehicle encoding help preserve privacy and make it harder for a tracker (including the authority) to definitively determine the trajectory of any vehicle.

Let $h_v(i) = H(v \oplus K_v \oplus C[i]) \bmod m$, where $1 \le i \le s$. We call $B[h_v(i)]$, $1 \le i \le s$, the *representative bits* of vehicle v in bitmap B. When the vehicle passes the RSU, it selects one of the representative bits uniformly at random through another hashing, $i = H(L \oplus v) \bmod s$. The size s of the array C determines the number of different representative bits from which a vehicle may choose to set. As our privacy analysis and numerical evaluation will show, this parameter controls a performance tradeoff between preserved privacy and traffic estimation accuracy.

From each bitmap B reported by an RSU, the central server can estimate the number of vehicles passing the RSU during the corresponding measurement period based on linear probabilistic counting [6], [25], [27] as follows:

$$\hat{n} = -m \ \ln \ V_0, \tag{1}$$

where V_0 is the fraction of bits in B that are zeros. Based on the historic traffic volumes, the central server will set the bitmap size at each RSU as follows:

$$m = 2^{\lceil \log_2(\bar{n} \times f) \rceil}, \tag{2}$$

where $\bar{n}$ is the expected traffic volume at the RSU during the measurement period based on historical average at the same location and the same time, and f is a system-wide load factor that specifies the ratio of the bitmap size and the expected traffic volume.

Formula (1) allows us to estimate point traffic based a single traffic record B. But we cannot apply it directly to solve the new problem of measuring *persistent* traffic across multiple traffic records. The key issues are how to combine multiple traffic records and how to derive new estimation formulas based on the combined information. Because the traffic volume $\bar{n}$ varies from place to place, the bitmap size varies accordingly. We set the value of m in (2) always as a power of two in order to facilitate joining the information of different bitmaps for persistent traffic estimation; such joining will become clear when we discuss the technical details.

Traffic measurement functions may be integrated into the future connected vehicles, which are heavily invested from both major car companies and governments. For existing vehicles, one possible way of deployment is similar to today's highway toll payment chips. Today's widespread automatic highway payment such as E-Pass [1] has demonstrated that wireless communications between E-Pass and a toll reader can be completed with vehicles travelling at highway speed. This includes the time for authentication and account information exchange. The payment chips can be modified to include the functions of traffic measurement for toll road systems. For more general deployment, one possible way is to incorporate these functions into smart vehicle plates (sometimes called electronic license plates) or to replace annual registration stickers with chips that implement such functions.

3 Measurement of Point Persistent Traffic

Given a set of t bitmaps, $\{B_1, \ldots, B_t\}$, that are measured at a certain location L of interest during t measurement periods, we want to estimate the point persistent traffic over the set as defined in Section 2.1. Let m be the largest size of all bitmaps, i.e., $m = \max\{l_1, \ldots, l_t\}$, where l_j is the number of bits in B_j, for $1 \le j \le t$.

Our basic approach is to combine B_j, for $1 \le j \le t$, through bitwise AND and establish a functional relationship between the number of persistent vehicles and the number of ones in the combined bitmap. We will then develop an estimator from the function to estimate persistent traffic based on ones in the combined bitmap. In the following, we will first introduce a technique of bitmap expansion that allows bitwise operations to be performed on bitmaps of different lengths. We will then discuss how to combine B_j, for $1 \le j \le t$, in order to properly handle transient traffic. Finally, we will derive the estimator for point persistent traffic.

3.1 Bitmap Expansion

To find the common traffic encoded by multiple bitmaps, we need to join the information from the bitmaps. Recall that each vehicle is encoded by setting a bit. If all bitmaps

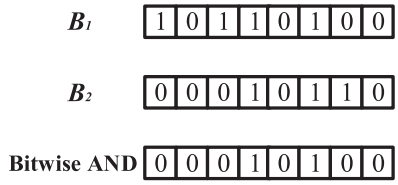


Fig. 1. An example of combining two bitmaps of the same size, B_1 and B_2, by bitwise AND.

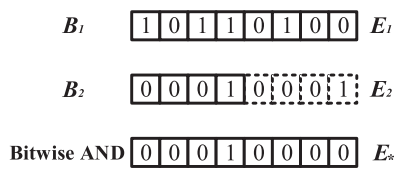


Fig. 2. An example of combining two bitmaps of different sizes by bitwise AND.

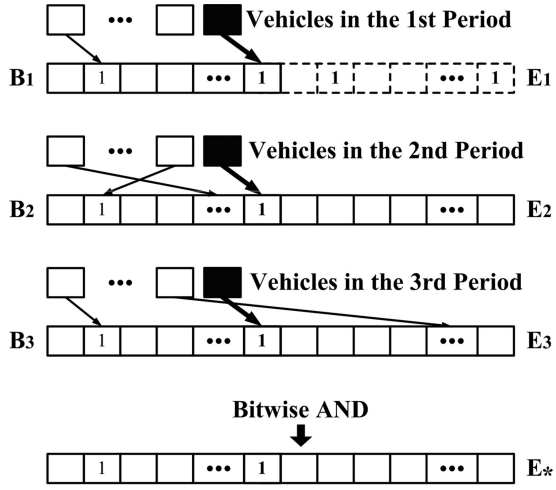


Fig. 3. An illustration for vehicle encoding in bitmaps, bitmap expansion, and bitmap joining.

have the same size, one simple approach of combining them is to perform bitwise AND, as shown by an example in Fig. 1. If a bit in the resulting bitmap is one, it means the same bit must be one in all bitmaps B_1 through B_t, indicating that there may be a common vehicle setting the bit in all those measurement periods.

However, if the bitmaps have different sizes, we will not be able to perform bitwise AND directly among them. To circumvent this problem, if the size of a bitmap B_j is smaller than m, we expand it by replicating it multiple times until its size reaches m, as shown by an example in Fig. 2, where B_2 is replicated once (dashed part). Such expansion is always possible because the sizes of all bitmaps are powers of 2. The expanded bitmap is denoted as E_j. If $l_j = m$, then E_j is simply B_j. We use Π to denote the set of expanded bitmaps (also known as traffic records). We perform bitwise AND over all expanded bitmaps in Π, and the result is denoted as E_*. Its ith bit is denoted as $E_*[i]$, $1 \le i \le m$.

Consider an arbitrary common vehicle v. Its hash output, $h_v = H(v \oplus K_v \oplus C[H(L \oplus v) \bmod s]) \bmod m$, gives the index of the bit in E_* that the vehicle is mapped to. In order to make sure that all common vehicles are recorded by E^*, the following property should hold after bitwise AND: $E_*[h_v] = 1$, for any common vehicle v. It is obvious that this property holds in the special case where all original bitmaps B_j, $1 \le j \le t$, have the same size m. In the general case, thanks to the design fact that the bitmap sizes are two's powers, we prove the property as follows: Consider an arbitrary bitmap B_j of size l_j. The bit set to one by v is at index $h_v \bmod l_j$. After l_j is expanded to m, all the bits in E_j at indices $(h_v \bmod l_j) + kl_j$, $0 \le k < \frac{m}{l_j}$, are ones. Because both l_j and m are powers of 2 and $m \ge l_j$, we know that $\frac{m}{l_j}$ is a positive integer. Hence, $h_v \bmod m = (h_v \bmod l_j) + k'l_j$, for a certain integer $k' \in [0, \frac{m}{l_j})$. Therefore, the bit in E_j at index $(h_v \bmod m)$ must be one. Since this holds for all expanded bitmaps $E_j, 1 \le j \le t$, we conclude that $E_*[h_v] = 1$.

3.2 Handling Transient Traffic

Can we simply estimate the number of common vehicles based on the number of ones in E_*? The answer is no because transient vehicles can also cause bits in E_* to be ones. See Fig. 3 for example, which shows three bitmaps, B_1, B_2 and B_3, collected from the same location. The vehicles that appear in each measurement period are shown above each bitmap. A black box indicates a common vehicle that appears in all measurement periods at the location. A white box indicates a transient vehicle. Each vehicle sets a bit to one, as the arrows in the figure show. The size of B_1 is half of the other bitmaps' size. We expand B_1 to E_1 by doubling its size, as shown in the figure with dashed lines. E_*, which is the bitwise AND of the three bitmaps, is at the bottom of the figure. We only show the values of two bits in E_*; both are ones. The first bit of one is caused by transient vehicles, which are different cars but happen to set bits at the same index due to hash collision. The second bit of one is caused by a common vehicle. We also want to point out that two common vehicles may set the same bit to one due to hash collision. Therefore, a bit of one in E_* may indicate zero, one or multiple common vehicles. Estimating common vehicles solely based on ones in E_* will be inaccurate. But if we combine information in Π into more than one bitmap and use that information jointly with E_*, we will be able to gain enough differentiation through probabilistic derivation, which in turn allows us to make meaningful estimation. So, in addition to E_*, we produce two more combined bitmaps below.

We divide Π into two subsets, $\Pi_a = \{E_1, \ldots, E_{\lceil t/2 \rceil}\}$ and $\Pi_b = \{E_{\lceil t/2 \rceil + 1}, \ldots, E_t\}$. Let E_a be the join of bitmaps in Π_a by bitwise AND, E_b the join of bitmaps in Π_b by bitwise AND, and E_* the join of E_a and E_b by bitwise AND.

E_a (or E_b) encodes both the set of common vehicles and possibly some transient vehicles. From (1), we compute the number of independent vehicles that would have produce the bitmap E_a (or E_b):

$$n_a = \frac{\ln V_{a,0}}{\ln(1 - \frac{1}{m})}, \qquad n_b = \frac{\ln V_{b,0}}{\ln(1 - \frac{1}{m})}, \tag{3}$$

where $V_{a,0}$ ($V_{b,0}$) is the fraction of zeros in E_a (E_b). Essentially we use an abstract set of n_a vehicles to produce the same effect as what all vehicles in Π_a jointly produce in E_a. This abstraction relieves us from the dependency within Π_a. Because the bits of ones in E_a retain the information from the common vehicles, the n_a vehicles contain the set of common vehicles. Similarly we use an abstract set of n_b vehicles to summarize the effect of Π_b. While dividing Π into more than two sets is possible, we find the two-set solution is not only simple but works effectively.

3.3 Deriving an Estimator for Persistent Traffic

For an arbitrary bit in E_*, its value can be modeled as a random binary variable whose value is probabilistically determined as the vehicles randomly choose their bits to set. Let n_* be the number of common vehicles. The probability P_* for at least one of the common vehicles to set the bit is

$$P_* = 1 - \left(1 - \frac{1}{m}\right)^{n_*}. \tag{4}$$

The probability for this bit to be set by a transient vehicle in E_a (or E_b) is

$$P_a = 1 - \left(1 - \frac{1}{m}\right)^{n_a - n_*}, P_b = 1 - \left(1 - \frac{1}{m}\right)^{n_b - n_*}. \tag{5}$$

Let $X_{i,1}$, $1 \le i \le m$, be the event that the ith bit in E_* becomes one. Combining the above analysis, the probability for $X_{i,1}$, $1 \le i \le m$, to occur is

$$\begin{aligned} &Prob\{X_{i,1}\} = P_* + (1 - P_*)P_aP_b \\ &= 1 - \left(1 - \frac{1}{m}\right)^{n_*} + \left(1 - \frac{1}{m}\right)^{n_*} \times \left(1 - \left(1 - \frac{1}{m}\right)^{n_a - n_*}\right) \\ &\quad \times \left(1 - \left(1 - \frac{1}{m}\right)^{n_b - n_*}\right) \\ &= 1 - \left(1 - \frac{1}{m}\right)^{n_a} - \left(1 - \frac{1}{m}\right)^{n_b} + \left(1 - \frac{1}{m}\right)^{n_a + n_b - n_*}. \end{aligned} \tag{6}$$

Transforming (3) to $V_{a,0} = (1 - \frac{1}{m})^{n_a}$ and $V_{b,0} = (1 - \frac{1}{m})^{n_b}$, we have

$$Prob\{X_{i,1}\} = 1 - V_{a,0} - V_{b,0} + V_{a,0}V_{b,0}\left(1 - \frac{1}{m}\right)^{-n_*} \tag{7}$$

Let $V_{*,1}$ be a random variable for the fraction of bits in E_* that are ones. We can measure an instance value of $V_{*,1}$ from E_*. This instance value will be used in the estimator derived later. We have

$$V_{*,1} = \frac{1}{m}\sum_{i=1}^{m} I_{X_{i,1}}, \tag{8}$$

where $I_{X_{i,1}}$ be the indicator variable of $X_{i,1}$, whose value is 1 when the event $X_{i,1}$ occurs and 0 otherwise. Clearly, $E(I_{X_{i,1}}) = Prob\{X_{i,1}\}$. Hence,

$$E(V_{*,1}) = \frac{1}{m}\sum_{i=1}^{m} E(I_{X_{i,1}}) = \frac{1}{m}\sum_{i=1}^{m} Prob\{X_{i,1}\}. \tag{9}$$

Because $Prob\{X_{i,1}\}$, $1 \le i \le m$, has the same value in (7), we have

$$E(V_{*,1}) = 1 - V_{a,0} - V_{b,0} + V_{a,0}V_{b,0}(1 - \frac{1}{m})^{-n_*}. \tag{10}$$

Solving the equation for n_*, we have

$$n_* = \frac{\ln V_{a,0} + \ln V_{b,0} - \ln(E(V_{*,1}) + V_{a,0} + V_{b,0} - 1)}{\ln(1 - \frac{1}{m})}. \tag{11}$$

Replacing the expected value $E(V_{*,1})$ with the instance value $V_{*,1}$ measured from E_*, we have the following formula for an estimated value $\hat{n_*}$ of the number of common vehicles.

$$\hat{n_*} = \frac{\ln V_{a,0} + \ln V_{b,0} - \ln(V_{*,1} + V_{a,0} + V_{b,0} - 1)}{\ln(1 - \frac{1}{m})}. \tag{12}$$

where $V_{a,0}$, $V_{b,0}$ and $V_{*,1}$ are measured from E_a and E_b and E_*, respectively.

4 Measurement of Point-to-Point Persistent Traffic

Consider two locations of interest, L and L'. Let $\{B_1, \ldots, B_t\}$ and $\{B'_1, \ldots, B'_t\}$ be the sets of bitmaps measured during the same periods at L and L', respectively. We want to estimate the point-to-point persistent traffic between the locations as defined in Section 2.1. Let m (m') be the largest size of all bitmaps from L (L'). Without loss of generality, assume $m \le m'$.

To find point-to-point persistent traffic, we need to combine the traffic records (bitmaps) at each location first and then further combine the resulting bitmaps from two locations. Each step uses the technique of expansion to make sure that bitmaps are of the same length so that bitwise operations can be performed. We will then derive an estimator for persistent traffic based on information in the combined bitmaps.

4.1 Two-Level Bitmap Expansion and Joining

The first level of bitmap expansion and joining are performed among the bitmaps from a single location. Consider the bitmaps from L. For each bitmap B_j, $1 \le j \le t$, if its size is smaller than m, we expand it by replicating it multiple times until its size reaches m. We then perform bitwise AND over all expanded bitmaps from L. The resulting bitmap is denoted as E_*, whose size is m. As we have explained in Section 3.1, the bitmap E_* encodes the set C of common vehicles appearing at one location L during t measurement periods. Besides that, E_* also encodes transient vehicles, e.g., those vehicles that set the first bit of one in E_* in Fig. 3.

Similarly, we expand each bitmap from L' to the size of m' and perform bitwise AND over all expanded bitmaps from L'. The result is denoted as E'_*, which encodes the set C' of common vehicles appearing at L' during the t measurement periods, as well as transient vehicles. What we are interested here is not $|C|$ or $|C'|$; they are the subject of the previous section. Let $C'' = C \bigcap C'$. We want to know $|C''|$, the number of common vehicles that pass both L and L' during the t measurement periods. When we discuss the point-to-point common vehicles in C'', the vehicles in C or C'' but not in C'' will also be referred to as transient vehicles.

The second level of bitmap expansion and joining are performed between two locations. If $m < m'$, we expand E_* by replicating it multiple times until its size research m'. The expanded bitmap is denoted as S_*, where we use S to signify this is the Second level expansion. If $m = m'$, S_* is simply E_*. We join the expanded S_* with E'_* by bitwise OR and the resulting bitmap is denoted as E''_*. (The reason for bitwise OR instead of bitwise AND is that the probabilistic analysis for deriving an estimator based on the result of bitwise AND is extremely difficult, whereas bitwise OR gives a closed-form formula.)

In the following, we will analysis the zero ratio of E''_* to derive an estimator for point-to-point persistent traffic. Note that we divide the bitmap set Π into two subsets in the point model, which can greatly help us to filter the noise

bits of ones introduced by vehicles. However, we should use the total number of independent vehicles that would have produced the bitmap E_* and E'_* to analyze the probability of one bit in E''_* is zero; otherwise, our probabilistic analysis will be wrong. Therefore, Π (Π') will not be divided into two subsets in the point-to-point model.

4.2 Deriving an Estimator for Point-to-Point Persistent Traffic

In Section 3.3, a common vehicle always sets bits in E_a and E_b at the same index, which makes probabilistic analysis much simpler. For persistent point-to-point traffic measurement, a common vehicle may set bits in E_* and E'_* at difference indices, $h_v = H(v \oplus K_v \oplus C[H(L \oplus v) \bmod s]) \bmod m$ and $h'_v = H(v \oplus K_v \oplus C[H(L' \oplus v) \bmod s]) \bmod m'$, which are dependent on location coordinates, L and L'. This makes the problem much harder because a common vehicle does not necessarily set bits in E_* and E'_* at the same index. It only has a certain probability to do so.

E_* (or E'_*) encodes both the set C'' of common vehicles and possibly some transient vehicles. Again based on (1), we compute the number of independent vehicles that would have produced the bitmap E_* (or E'_*):

$$n = \frac{\ln V_{*,0}}{\ln(1-\frac{1}{m})}, \qquad n' = \frac{\ln V'_{*,0}}{\ln(1-\frac{1}{m'})}, \tag{13}$$

where $V_{*,0}$ ($V'_{*,0}$) is the fraction of zeros in E_* (E'_*). Similar to Section 3.3, we use an abstract set of n independent vehicles to produce the same effect as what all vehicles that pass L will jointly produce in E_*. Yet the bits of ones in E_* retain all information from the common vehicles. We also use an abstract set of n' independent vehicles to summarize what the vehicles passing $L's$ will produce in E'_*.

For an arbitrary bit $E''_*[i]$, $1 \le i \le m'$, whose value is the OR of $S_*[i]$ and $E'_*[i]$. We derive the probability for $E''_*[i]$ to be zero. For this to happen, no common/transient vehicle should set $S_*[i]$ or $E'_*[i]$ to one. Let n'' be the number of common vehicles in C''.

First, consider an arbitrary common vehicle in C''. If the vehicle sets $E_*[i \bmod m]$ to one, then after expansion $S_*[i]$ will be one. Let $E_*[i' \bmod m]$, $1 \le i' \le m'$, be the bit that the vehicle sets at L. The probability for $E_*[i' \bmod m]$ to be different from $E_*[i \bmod m]$ is $1 - \frac{1}{m}$. In this case, $S_*[i]$ is not set by the vehicle. Under this condition ($i' \bmod m \neq i \bmod m$), we analyze the probability of the vehicle not setting $E'_*[i]$ at location L'. The vehicle will be mapped at L' to one of its s representative bits, including $E'_*[i']$ with probability $\frac{1}{s}$ — in which case, $E'_*[i]$ is not set because $i' \neq i$. With probability $1 - \frac{1}{s}$, the vehicle is mapped to a bit other than $E'_*[i']$, and that bit has a chance of $\frac{1}{m'}$ to happen to be $E'_*[i]$. In summary, the probability for any common vehicle not to set either $S_*[i]$ or $E'_*[i]$ is $(1-\frac{1}{m})(\frac{1}{s} + (1-\frac{1}{s})(1-\frac{1}{m'}))$. The probability for none of the common vehicles to set either $S_*[i]$ or $E'_*[i]$ is

$$P_1 = \left(1-\frac{1}{m}\right)^{n''} \left(\frac{1}{s} + \left(1-\frac{1}{s}\right)\left(1-\frac{1}{m'}\right)\right)^{n''}. \tag{14}$$

Second, there are $n - n''$ transient vehicles passing location L. The probability for none of these transient vehicles to set $E_*(i \bmod m)$ is $(1-\frac{1}{m})^{n-n''}$. Similarly, there are $n' - n''$ transient vehicles passing location L'. The probability for none of these transient vehicles to set $E'_*[i]$ is $(1-\frac{1}{m'})^{n'-n''}$.

Now we model the value of the ith bit in E''_* as a binary random variable, where $1 \le i \le m'$. Let $Y_{i,0}$, $1 \le i \le m'$, be the event that the ith bit remains zero. Combining the above analysis, we have

$$\begin{aligned} Prob\{Y_{i,0}\} &= P_1 \left(1-\frac{1}{m}\right)^{n-n''} \left(1-\frac{1}{m'}\right)^{n'-n''} \\ &= \left(1+\frac{1}{sm'-s}\right)^{n''} \left(1-\frac{1}{m}\right)^{n} \left(1-\frac{1}{m'}\right)^{n'}. \end{aligned} \tag{15}$$

Applying (13), we have

$$Prob\{Y_{i,0}\} = \left(1+\frac{1}{sm'-s}\right)^{n''} V_{*,0} V'_{*,0}. \tag{16}$$

Let $V''_{*,0}$ be a random variable for the fraction of bits in E''_* that are zeros. We have

$$V''_{*,0} = \frac{1}{m'} \sum_{i=1}^{m'} I_{Y_{i,0}}, \tag{17}$$

where $I_{Y_{i,0}}$ be the indicator variable of $Y_{i,0}$, whose value is 1 when the event $Y_{i,0}$ occurs and 0 otherwise. Clearly, $E(I_{Y_{i,0}}) = Prob\{Y_{i,0}\}$. Hence,

$$E(V''_{*,0}) = \frac{1}{m'} \sum_{i=1}^{m'} E(I_{Y_{i,0}}) = \frac{1}{m'} \sum_{i=1}^{m'} Prob\{Y_{i,0}\}. \tag{18}$$

Because $Prob\{Y_{i,0}\}$, $1 \le i \le m'$, has the same value in (16), we have

$$E(V''_{*,0}) = \left(1+\frac{1}{sm'-s}\right)^{n''} V_{*,0} V'_{*,0}. \tag{19}$$

Suppose m' is large. Solving the equation for n'' and apply $\ln(1+x) \approx x$ when x is small, we have

$$n'' \approx sm'(\ln E(V''_{*,0}) - \ln V_{*,0} - \ln V'_{*,0}). \tag{20}$$

Replacing the expected value $E(V''_{*,0})$ with the instance value $V''_{*,0}$ measured from E''_*, we have the following formula for an estimated value $\hat{n''}$ of the number of common vehicles passing both locations L and L' during all t measurement periods.

$$\hat{n''} = sm'(\ln V''_{*,0} - \ln V_{*,0} - \ln V'_{*,0}). \tag{21}$$

where $V_{*,0}$, $V'_{*,0}$ and $V''_{*,0}$ are measured from E_* and E'_* and E''_*, respectively.

5 Measurement of Multi-Point Persistent Traffic

In this section, we generalize our method for multi-point persistent traffic measurement. Consider a set of locations of interest, $L_1, \ldots, L_d$. Let $\{B_{1,1}, \ldots, B_{1,t}\}, \ldots, \{B_{d,1}, \ldots, B_{d,t}\}$ be the sets of bitmaps measured during the same periods at $L_1, \ldots, L_d$, respectively. Our aim is to estimate the multi-point persistent traffic as defined in Section 2.1. Let m_i be the largest size of all bitmaps from L_i. Without loss of generality, we assume that $m_1 \le m_2 \le \ldots \le m_d$. Below we will first show how to extend our design for three-point persistent traffic measurement, and then present a general framework for multi-point persistent traffic measurement.

5.1 Three-Point Persistent Traffic Measurement

Similar to the point-to-point case, we first perform the first-level bitmap expansion and join among the bitmaps from each location L_i, i.e., expending the bitmaps in $\{B_{i,1}, \ldots, B_{i,t}\}$ to the size of m_i and then performing bitwise AND over all the expended bitmaps. The resulting bitmap is denoted by E_i.

Then, we perform the second-level bitmap expansion and join among bitmaps E_1, E_2 and E_3. As $m_1 \leq m_2 \leq m_3$, we expand E_1 and E_2 (if necessary) by replicating them multiple times until each of them reaches the size of m_3. We use S_i to denote the expanded bitmap of E_i. Note that S_3 equals to E_3. The result of bitwise OR over S_1, S_2 and S_3 is denoted as S_*. According to [25], the number of vehicles estimated based on the bitmap E_i (S_*) is

$$N_i = \frac{\ln V_{i,0}}{\ln(1-\frac{1}{m_i})},$$

$$N_* = \frac{\ln V_{*,0}}{\ln(1-\frac{1}{m_3})}, \tag{22}$$

where $V_{i,0}$ and $V_{*,0}$ are the fractions of zeros in E_i and S_*, respectively, for $1 \leq i \leq 3$. We cannot use N_* as the estimation for the three-point persistent traffic volume because bitmap expansion distorts vehicle recording. After E_i is expanded to S_i, as the bitmap size is increased from m_i to m_3 and the fraction of zeros stays the same as $V_{i,0}$, the estimated number of vehicles becomes $\frac{\ln V_{i,0}}{\ln(1-\frac{1}{m_3})} \approx \frac{m_3}{m_i} N_i$ because $\ln(1-\frac{1}{x}) \approx \frac{1}{x}$ when x is large. Observed from a different angle, each vehicle that sets a single bit of one in E_i causes $\frac{m_3}{m_i}$ bits in S_* to be set as ones, which in turn results in an *overestimation factor* of $\frac{m_3}{m_i}$ (for this vehicle) in (22).

Let W_i be the set of vehicles that persistently pass location L_i, n_i be the number of vehicles that persistently pass location L_i, and $n_{i,j}$ be the number of point-to-point persistent vehicles that pass location L_i and L_j. Our target is to estimate the number of three-point persistent vehicles, which is denoted as $n_{1,2,3} = |W_1 \cap W_2 \cap W_3|$. The persistent vehicles that pass location L_1 can be partitioned into four sets: $W_1 \cap W_2 \cap W_3$, $W_1 \cap W_2 - W_3$, $W_1 \cap W_3 - W_2$, and $W_1 - W_2 - W_3$. Further consider three other sets of vehicles, $W_2 \cap W_3 - W_1$, $W_2 - W_1 - W_3$, and $W_3 - W_1 - W_2$, which pass locations L_2 and L_3, only L_2, and only L_3, respectively. Below we derive the overestimation factor of vehicles in each of the above sets.

1) Consider a vehicle $v \in W_1 \cap W_2 \cap W_3$. It pseudo-randomly selects a bit in $E_1[i \bmod m_1]$ ($1 \leq i \leq m_3$) and sets it to one. After the second level bitmap expansion, v sets bits $S_1[k * (i \bmod m_1)]$, $1 \leq k \leq \frac{m_3}{m_1}$, to ones. As a result, $\frac{m_3}{m_1}$ bits in S_*, i.e., $S_*[k * (i \mod m_1)]$, $1 \leq k \leq \frac{m_3}{m_1}$, are set to ones by v. Let $E_2[i' \bmod m_2]$, $1 \leq i' \leq m_3$, be the bit that v sets to one at L_2. There are two cases: $i \bmod m_1 = i' \bmod m_1$ and $i \bmod m_1 \neq i' \bmod m_1$. In the former case, v does not introduce any additional bit of one in S_* due to its appearance at location L_2. In the latter case, v will introduce $\frac{m_3}{m_2}$ more bits of ones in S_*. With a probability of $1-\frac{1}{s}$, v chooses a different bit from its s representative bits at L_2 than the bit chosen at L_1, and the conditional probability for $i \bmod m_1 \neq i' \bmod m_1$ is $1-\frac{1}{m_1}$. Therefore, the expected number of bits of ones that vehicle v sets in S_* due to its appearance at L_1 and L_2 is $\frac{m_3}{m_1} + \frac{m_3}{m_2}(1-\frac{1}{s})(1-\frac{1}{m_1})$.

Next, we consider location L_3. There are two cases that v may bring more ones in S_* due to its appearance at location L_3. Let $E_3[i'' \bmod m_3]$, $1 \leq i'' \leq m_3$, be the bit that v sets at L_3. The first case is $i'' \bmod m_1 \neq i' \bmod m_1 \neq i \bmod m_1$. Note that the probability of $i \bmod m_1 \neq i' \bmod m_1$ is $(1-\frac{1}{s})(1-\frac{1}{m_1})$, and the conditional probability for v to choose a different bit from its s representative bits at L_3 than those at L_1 and L_2 is $1-\frac{2}{s}$, so the conditional probability for $E_3[i'']$ being neither $S_1[\frac{m_3}{m_1}(i \bmod m_1)]$ nor $S_2[\frac{m_3}{m_2}(i' \bmod m_2)]$ is $1-\frac{m_3/m_1+m_3/m_2}{m_3}$. The second case is $i'' \bmod m_1 \neq i' \bmod m_1 = i \bmod m_1$. There are two subcases depending on whether v chooses two different bits from its logical bit array or not. Similar to the analysis for the first case, the probabilities for these two subcases to happen are $\frac{1}{s}(1-\frac{1}{s})(1-\frac{1}{m_1})$ and $(1-\frac{1}{s})\frac{1}{m_1}(1-\frac{2}{s})(1-\frac{1}{m_1})$, respectively.

Combining the above analysis, we have that the overestimation factor, i.e., the expected number of bits of ones that vehicle v sets in S_*, is

$$Q_1 = \frac{m_3}{m_1} + \left(1-\frac{1}{s}\right)\left(1-\frac{1}{m_1}\right)\left[\frac{m_3}{m_2} + \frac{1}{s} + \left(1-\frac{1}{m_2}\right)\left(1-\frac{2}{s}\right)\right]. \tag{23}$$

The number of vehicles in $W_1 \cap W_2 \cap W_3$ is $n_{1,2,3}$. Therefore, all vehicles in $W_1 \cap W_2 \cap W_3$ are overestimated as $Q_1 * n_{1,2,3}$ vehicles in the estimation of (22).

2) Consider the vehicle set $W_1 \cap W_2 - W_3$. Following an analysis similar to the case of $v \in W_1 \cap W_2 \cap W_3$, we have that the expected number of bits of ones that vehicle v sets in S_* at L_1 and L_2 is

$$Q_2 = \frac{m_3}{m_1} + \frac{m_3}{m_2}\left(1-\frac{1}{s}\right)\left(1-\frac{1}{m_1}\right). \tag{24}$$

The cardinality of $W_1 \cap W_2 - W_3$ is $n_{1,2} - n_{1,2,3}$. Thus, all vehicles in $W_1 \cap W_2 - W_3$ are overestimated as $Q_2 * (n_{1,2} - n_{1,2,3})$ vehicles in (22).

3) Consider the vehicle set $W_1 \cap W_3 - W_2$. Similar to the case of $v \in W_1 \cap W_2 - W_3$, the expected number bits of ones that vehicle v set in S_* at L_1 and L_3 is

$$Q_3 = \frac{m_3}{m_1} + \left(1-\frac{1}{s}\right)\left(1-\frac{1}{m_1}\right). \tag{25}$$

The cardinality of $W_1 \cap W_2 - W_3$ is $n_{1,3} - n_{1,2,3}$. Hence, all vehicles in $W_1 \cap W_3 - W_2$ are overestimated as $Q_3 * (n_{1,3} - n_{1,2,3})$ vehicles in (22).

4) Consider the vehicle set $W_2 \cap W_3 - W_1$. Similar to the case of $v \in W_1 \cap W_2 - W_3$, the expected number of bits of ones that vehicle v sets in S_* at location L_2 and L_3 is

$$Q_4 = \frac{m_3}{m_2} + \left(1-\frac{1}{s}\right)\left(1-\frac{1}{m_2}\right). \tag{26}$$

The cardinality of $W_2 \cap W_3 - W_1$ is $n_{2,3} - n_{1,2,3}$. Hence, all vehicles in $W_2 \cap W_3 - W_1$ are overestimated as $Q_4 * (n_{2,3} - n_{1,2,3})$ vehicles in (22).

5) Consider a vehicle in $v \in W_1 - W_2 - W_3$. After the second level bitmap expansion, v sets $\frac{m_3}{m_1}$ bits in S_* to ones. The cardinality of $W_1 - W_2 - W_3$ is $n_1 - n_{1,2} - n_{1,3} + n_{1,2,3}$. Thus, all vehicles in $W_1 - W_2 - W_3$ are overestimated as $\frac{m_3}{m_1}(n_1 - n_{1,2} - n_{1,3} + n_{1,2,3})$ vehicles in (22).

6) Consider a vehicle $v \in W_2 - W_1 - W_3$. After the second level bitmap expansion, v sets $\frac{m_3}{m_2}$ bits in S_* to ones. The

cardinality of $W_2 - W_1 - W_3$ is $n_2 - n_{1,2} - n_{2,3} + n_{1,2,3}$. Thus, all vehicles in $W_2 - W_1 - W_3$ are overestimated as $\frac{m_3}{m_2}(n_2 - n_{1,2} - n_{2,3} + n_{1,2,3})$ vehicles in (22).

7) Consider a vehicle $v \in W_3 - W_1 - W_2$. After the second level bitmap expansion, v sets only one bit in S_* to one. The cardinality of $W_3 - W_1 - W_2$ is $n_3 - n_{1,3} - n_{2,3} + n_{1,2,3}$. Thus, all vehicles in $W_3 - W_1 - W_2$ are estimated as $n_3 - n_{1,3} - n_{2,3} + n_{1,2,3}$ vehicles in (22).

According to the above analysis, we have

$$\begin{aligned} N_* = {} & Q_1 n_{1,2,3} + Q_2(n_{1,2} - n_{1,2,3}) \\ & + Q_3(n_{1,3} - n_{1,2,3}) + Q_4(n_{2,3} - n_{1,2,3}) \\ & + \frac{m_3}{m_1}(n_1 - n_{1,2} - n_{1,3} + n_{1,2,3}) \\ & + \frac{m_3}{m_2}(n_2 - n_{1,2} - n_{2,3} + n_{1,2,3}) \\ & + (n_3 - n_{1,3} - n_{2,3} + n_{1,2,3}). \end{aligned} \tag{27}$$

Finally, we get the estimator $\hat{n}_{1,2,3}$ by solving (27):

$$\hat{n}_{1,2,3} = \frac{N_* - Q_2 n_{1,2} - Q_3 n_{1,3} - Q_4 n_{2,3} - \frac{m_3 n_1'}{m_1} - \frac{m_3 n_2'}{m_2} - n_3'}{Q_1 - Q_2 - Q_3 - Q_4 + \frac{m_3}{m_1} + \frac{m_3}{m_2} + 1}, \tag{28}$$

where $n_1' = n_1 - n_{1,2} - n_{1,3}$, $n_2' = n_2 - n_{1,2} - n_{2,3}$ and $n_3' = n_3 - n_{1,3} - n_{2,3}$. The values of n_i and $n_{i,j}$, for $1 \le i, j \le 3$, $i \ne j$, can be estimated by the approaches in Section 3.3 and Section 4, respectively.

5.2 Multi-Point Persistent Traffic Measurement

The analysis of multi-point scheme is similar to that of the three-point scheme. We first perform the first-level bitmap expansion and join to get the resulting bitmap E_i for each location L_i, and then perform the second-level bitmap expansion for each E_i (if necessary) to get the expended bitmap S_i. After that, we perform bitwise OR over S_1, $S_2, \ldots, S_d$ to get the resulting bitmap S_*. Note that the sizes of $S_1, S_2, \ldots, S_d$ are all m_d.

We can follow the similar reasoning as that for the three-point scheme to derive the d-point persistent traffic estimator, i.e., partitioning the persistent vehicles into multiple sets and then analyzing the overestimation factor of vehicles in each of the partitioned sets. We denote the set of d locations as $\mathcal{L}_d = \{L_1, L_2, \ldots, L_d\}$, and the set of vehicles that just pass all the locations in $\mathcal{L}$ as $\mathcal{V}_{\mathcal{L}}$. Thus, all the persistent vehicles can be partitioned into sets $\mathcal{V}_{\mathcal{L}_d}$, $\mathcal{V}_{\mathcal{L}_d - \{L_i\}} (1 \le i \le d)$, $\mathcal{V}_{\mathcal{L}_d - \{L_i, L_j\}} (1 \le i < j \le d), \ldots, \mathcal{V}_{\{L_i\}} (1 \le i \le d)$.

The d-point persistent traffic estimator can be inductively derived, i.e., deriving the k-point persistent traffic estimators from $k = 3$ to d one by one. Suppose that we have derived the estimators for the k-point $(1 \le k \le d-1)$ persistent traffic, the details for deriving the d-point persistent traffic are described as follows:

Step 1: For each partition $\mathcal{V}_{\mathcal{L}}$, we obtain the overestimation factor of vehicles in partition $\mathcal{V}_{\mathcal{L}}$ by using the similar reasoning as that for the three-point scheme.

Step 2: Construct the functions for N_* and the cardinality of partitioned sets:

$$\begin{aligned} N_* = {} & Q_{\mathcal{L}_d} |\mathcal{V}_{\mathcal{L}_d}| + \sum_{i=1}^{d} Q_{\mathcal{L}_d - \{L_i\}} |\mathcal{V}_{\mathcal{L}_d - \{L_i\}}| \\ & + \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} Q_{\mathcal{L}_d - \{L_i, L_j\}} |\mathcal{V}_{\mathcal{L}_d - \{L_i, L_j\}}| \\ & + \cdots + \sum_{i=1}^{d} Q_{\{L_i\}} |\mathcal{V}_{\{L_i\}}|. \end{aligned} \tag{29}$$

where $Q_{\mathcal{L}}$ is the overestimation factor of vehicles in partition $\mathcal{V}_{\mathcal{L}}$, and $|\mathcal{V}_{\mathcal{L}}|$ is the cardinality of $\mathcal{V}_{\mathcal{L}}$. Note that N_* is the number of independent vehicles that would have produced the bitmap S_*:

$$N_* = \frac{\ln V_{*,0}}{\ln\left(1 - \frac{1}{m_d}\right)}, \tag{30}$$

where $V_{*,0}$ is the fraction of zeros in S_*. The set cardinality $|\mathcal{V}_{\mathcal{L}}|$ can be derived from the *inclusion-exclusion principle*. The number of vehicles that persistently pass less than d locations can be measured by the k-point $(1 \le k \le d-1)$ persistent estimators. Thus, there is only one unknown variable $n_{1,2,\ldots,d}$ in equation (29), which is the number of vehicles that persistently pass d locations.

Step 3: Combining (29) and (30), we get the estimator for d-point persistent traffic.

However, the computation overhead of the central server grows exponentially as d increases: to measure the volume of d-point persistent traffic, the central server needs to compute the k-point persistent traffic volume for each combination of $k(1 \le k \le d-1)$ locations. Fortunately, our general scheme is still sufficiently efficient for most transportation applications, as d is usually small (e.g., 2 or 3) in reality.

6 Privacy Analysis

When a vehicle passes an RSU, the only thing that a vehicle does is to set a bit in the RSU's bitmap to one at an index that may vary from location to location. Moreover, different vehicles may choose the same indices. What each RSU gathers is a bitmap, with each bit of one suggesting the passage of at least one vehicle. Therefore, the tracker may possibly identify the trajectory of a common vehicle through the observation that bits with the same index at two different locations are both ones. Below, we analyze privacy preservation of our persistent-traffic measurement design in terms of the probabilistic noise-to-information ratio as defined in Section 2.3.

When a vehicle v passes a location L, it sends an index value i to the RSU, which set the bit at the index in the bitmap B to one, i.e., $B[i] = 1$. Let m be the size of B. Suppose the authority is able to associate the index i with the vehicle v at L, for example, when the vehicle is stopped by a police for speeding, there is no other vehicle around, and the police informs the authority. Now if the authority finds at a different location L' that the bit at the same index in the bitmap B' is also one, i.e., $B'[i] = 1$, can it assert that the vehicle v has moved from L' to L, thus revealing the partial trajectory of the vehicle?

Recall that other vehicles may choose the same index and the same vehicle may choose difference indices at different locations. The bit $B'[i]$ may have been set by other vehicles passing L'; in this case, the above assertion about the

trajectory of v will be wrong. Let p be the probability that $B'[i]$ is set to one by other vehicles even if v does not pass L'. Let n' be the number of vehicles passing L', each having a probability of $\frac{1}{m'}$ to set $B'[i]$. Therefore,

$$p = 1 - \left(1 - \frac{1}{m'}\right)^{n'}. \tag{31}$$

Let p' be the probability that $B'[i]$ is set to one when v does pass L'. According to Section 2.4, the same vehicle may set bits at different indices at different locations. In particular, it has s representative bits and randomly selects one to set at L'. Therefore, the probability for v to set $B'[i]$ to one is $\frac{1}{s}$. We know by (31) that other vehicles will set $B'[i]$ with probability p. Hence,

$$p' = p + (1 - p)\frac{1}{s}. \tag{32}$$

The probabilistic noise-to-information ratio is therefore

$$\frac{p}{p' - p} = \frac{1 - (1 - \frac{1}{m'})^{n'}}{(1 - \frac{1}{m'})^{n'}\frac{1}{s}}. \tag{33}$$

To protect the trajectory privacy of vehicles, we expect this ratio to be at least greater than one, and the larger the better. From (33), we found that the probabilistic noise-to-information ratio is determined by s and f. Our simulation results show that the relative error of our estimators will increase with the increase of s and decease with the increase of f. However, the probabilistic noise-to-information ratio is the opposite. Therefore, there is a tradeoff between accuracy and privacy.

7 Simulation

In this section, we perform simulations to evaluate the performance of our proposed persistent traffic estimators in terms of estimation accuracy and preserved privacy under different parameter settings. We use both real transportation traffic and synthetic traffic. To the best of our knowledge, this is the first work that studies persistent traffic measurement (as defined in Section 2.1) through vehicle-to-infrastructure communications. There are prior privacy-preserving approaches for measuring point-to-point traffic [30], [32] or measuring travel time [7]. But there is no prior work that measures *persistent* point-to-point traffic under the same model in Section 2.2. We stress these are very different problems. Therefore, we will compare the proposed estimators with some benchmark methods of simpler designs to demonstrate the effectiveness of the proposed design.

7.1 Simulation Results Based on Real Traffic Data

First, we use the real-world vehicle trip table measured at the city of Sioux Falls, South Dakota; the data can be found in [11], which contains the actual traffic volume from one point to another in the city. In our simulation, we generate the point-to-point common vehicles between two locations L and L' based on the number n'' from the vehicle trip table, and then randomly generate $n - n''$ transient vehicles for L and $n' - n''$ transient vehicles for L', where n (n') is the total traffic volume, i.e., the sum of all entries in the trip table involving L (L'). *For each generated vehicle, we randomly select its vehicle ID and private key that are needed for vehicle recording in Section 2.4.* The bitmap size m (m') is computed from n (n') and f; see Section 2.4. In the simulation, we let L' be the location with the largest total traffic volume of all, with $n' = 451000$. We randomly select 8 other locations as L.

We simulate 10 measurement periods with randomly generated transient vehicles. The performance of our estimator on point-to-point persistent traffic between L and L' is shown in Table 1-3, where we vary s from 2 to 3 to 5 and set f to 2. The impact of different f values will be shown later. The results are the average of 1000 simulation runs. It can be seen from the fourth row that $m' \neq m$ and their ratio ranges from 2 to 16. The 6th-9th rows present the relative error (defined in Section 2.3) when $t = 3, 5, 7, 10$, respectively. In Table 1, we can see that the estimation error is mostly small. The error is higher when $L = 8$, where the number of common vehicles is just 3,000, comparing with 451,000 vehicles passing L' and 28,000 vehicles passing L; in this case, the noise generated from the transient vehicles is high, relative to the number of common vehicles. Similar observations can be made in other tables. We also note that the relative error increases when s increases, which means we should pick a small value of s. But s also has an impact on the privacy: the larger it is, the better the privacy becomes, as we will demonstrate later. We recommend to choose $s = 3$ as a compromise.

We include the last line in the tables for a benchmark comparison with a simpler design where we set $m' = m$ and m is determined by n and f, which is to ensure the privacy of the vehicles pass location L. Everything else stays the same as described in the paper. The relative error in the last line is larger than that in the 7th line (which is also bolded); in both lines, $t = 5$. For example, when $n'' = 3,000$ in the last column, the relative error of the proposed estimator is just 0.0585, whereas the relative error of the same-size design is 1.3749 when $s = 3$.

The Sioux Falls data can only support limited evaluation. We resort to synthetic data for other simulations.

7.2 Simulation Results Based on Synthetic Traffic

Next, we evaluate the proposed estimators based on synthetic traffic data. For point persistent traffic measurement, the number of vehicles that passes L during each measurement period is randomly generated from the range of $(3000, 10000]$. Let n_{min} be the minimum number of generated vehicles that pass location L in any measurement period. We set the number of common vehicles n_* at L during all measurement periods from $0.01n_{min}$ to $0.5n_{min}$, with steps of $0.01n_{min}$. We set $s = 3$ and $f = 2$. We compare the proposed estimator (Section 3) with a benchmark method of a simpler design that estimates directly from E_* with $\hat{n}_* = \frac{lnV_{*,0}}{ln(1-1/m)}$ [6], [25], [27], where E_* is the bitwise AND of all t bitmaps from L.

The simulation results are presented in Fig. 4, where the horizontal axis represents the actual persistent traffic volume and the vertical axis represents the relative error. The left plot is the comparison between the proposed estimator and the benchmark when $t = 5$, the right plot is the comparison when $t = 10$. In both cases, the proposed estimator significantly outperforms the benchmark, particularly when the persistent traffic volume is relatively small. The relative error becomes much smaller when t is increased from 5 to 10. That is because the AND join of more bitmaps helps filter out the ones produced by transient vehicles (which are noise).

TABLE 1
Relative Error of Point-to-Point Persistent Traffic Volume Estimation in the Sioux Falls Network ($s = 2$)

L	1	2	3	4	5	6	7	8
n	213000	140000	121000	78000	76000	47000	40000	28000
m	524288	524288	262144	262144	262144	131072	131072	65536
m'/m	2	2	4	4	4	8	8	16
n''	40000	20000	19000	8000	8000	7000	6000	3000
relative error ($t = 3$)	0.0070	0.0112	0.0151	0.0232	0.0215	0.0271	0.0281	0.0637
relative error ($t = 5$)	**0.0066**	**0.0098**	**0.0114**	**0.0180**	**0.0184**	**0.0189**	**0.0176**	**0.0392**
relative error ($t = 7$)	0.0063	0.0090	0.0119	0.0159	0.0159	0.0173	0.0186	0.0311
relative error ($t = 10$)	0.0069	0.0091	0.0118	0.0157	0.0171	0.0190	0.0179	0.0329
same-size bitmaps ($t = 5$)	**0.0065**	**0.0103**	**0.0174**	**0.0333**	**0.0343**	**0.0756**	**0.0826**	**0.8484**

TABLE 2
Relative Error of Point-to-Point Persistent Traffic Volume Estimation in the Sioux Falls Network ($s = 3$)

L	1	2	3	4	5	6	7	8
n	213000	140000	121000	78000	76000	47000	40000	28000
m	524288	524288	262144	262144	262144	131072	131072	65536
m'/m	2	2	4	4	4	8	8	16
n''	40000	20000	19000	8000	8000	7000	6000	3000
relative error ($t = 3$)	0.0122	0.0167	0.0210	0.0369	0.0361	0.0398	0.0438	0.0948
relative error ($t = 5$)	**0.0101**	**0.0144**	**0.0169**	**0.0252**	**0.0267**	**0.0284**	**0.0265**	**0.0585**
relative error ($t = 7$)	0.0111	0.0151	0.0171	0.0257	0.0241	0.0279	0.0251	0.0518
relative error ($t = 10$)	0.0104	0.0139	0.0172	0.0258	0.0256	0.0261	0.0234	0.0497
same-size bitmaps ($t = 5$)	**0.0110**	**0.0172**	**0.0267**	**0.0510**	**0.0491**	**0.1271**	**0.1305**	**1.3749**

TABLE 3
Relative Error of Point-to-Point Persistent Traffic Volume Estimation in the Sioux Falls Network ($s = 5$)

L	1	2	3	4	5	6	7	8
n	213000	140000	121000	78000	76000	47000	40000	28000
m	524288	524288	262144	262144	262144	131072	131072	65536
m'/m	2	2	4	4	4	8	8	16
n''	40000	20000	19000	8000	8000	7000	6000	3000
relative error ($t = 3$)	0.0194	0.0271	0.0361	0.0556	0.0585	0.0737	0.0726	0.1603
relative error ($t = 5$)	**0.0190**	**0.0235**	**0.0264**	**0.0473**	**0.0460**	**0.0448**	**0.0501**	**0.0904**
relative error ($t = 7$)	0.0176	0.0219	0.0260	0.0437	0.0393	0.0477	0.0432	0.0847
relative error ($t = 10$)	0.0180	0.0213	0.0274	0.0434	0.0410	0.0418	0.0428	0.0809
same-size bitmaps ($t = 5$)	**0.0175**	**0.0257**	**0.0430**	**0.0772**	**0.0824**	**0.2187**	**0.2314**	**2.1002**

For point-to-point persistent traffic measurement, we study the relative estimation error with respect to the actual persistent traffic volume. The number of vehicles that passes L (or L') is randomly generated from $(3000, 10000]$, and thus the two locations have the same average traffic. Suppose n_{min} (n'_{min}) is the minimum number of generated vehicles that passed L (L') during any measurement period. Let $n''_{min} = \min\{n_{min}, n'_{min}\}$. We set the number n'' of common vehicles from $0.01n''_{min}$ to $0.5n''_{min}$, with step size of $0.01n''_{min}$. The simulation results are presented in the left plot of Fig. 5, which shows that the relative error decreases as the actual persistent traffic volume increases. The reason is that as we increase the persistent traffic while the noise (transient traffic) stays the same, the relative error caused by noise to the persist traffic will decrease. We also see that the error decreases as s increases, agreeing with the results in Tables 1, 2, and 3. In the right plot, we generate the number of vehicles that passes L' from $(30000, 100000]$. Therefore, the average traffic volume at L' is 10 times of that at L, which means m' is much larger than m. Similar results are observed, which means that the proposed estimator produces stable performance when the traffic volumes at the two locations vary greatly.

For the three-point persistent traffic measurement, the number of vehicles passing through L_i during each measurement period is randomly generated from the range $(3000, 10000]$. Thus, the three locations have the same average traffic. Suppose that $n_{i,min}$ is the minimum number of generated vehicles that pass through location L_i during any measurement period, for $1 \leq i \leq 3$. Let $n^*_{min} = \min\{n_{1,min}, n_{2,min}, n_{3,min}\}$. We set the number $n_{1,2,3}$ of the three-point common vehicles from $0.01n^*_{min}$ to $0.5n^*_{min}$, with step size of $0.01n^*_{min}$. The simulation results are shown in the third points in Figs. 6, 7, and 8.

Under different values of f, Figs. 6, 7, and 8 present the measurement accuracy in a different form, with each point in the figures representing a measurement, where the x-coordinate of the point is the actual persistent traffic volume and the y-coordinate is the estimated volume. We also draw the equality line $y = x$. The closer the points are to the line, the better the measurement accuracy will be. As the points cluster around the equality line, three plots in each figure confirm that the proposed estimators produce good measurement accuracy for point, point-to-point and three-point persistent traffic. When we increase the value of f from 1 to

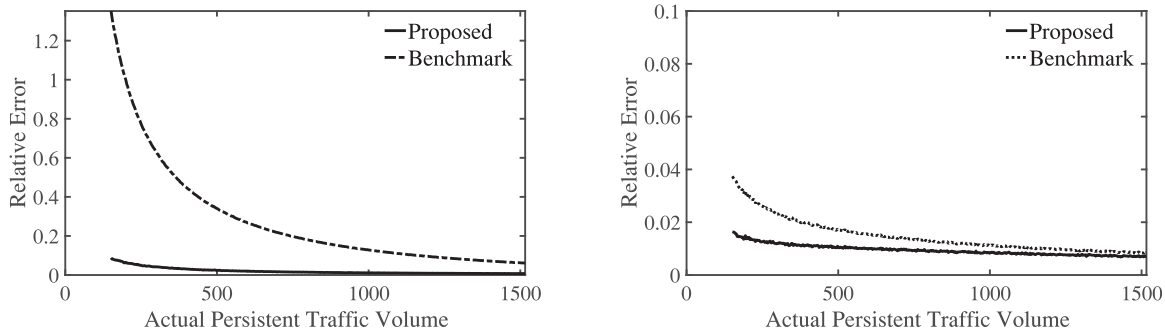


Fig. 4. Relative error of point persistent traffic estimation. Left plot: $t = 5$; right plot: $t = 10$.

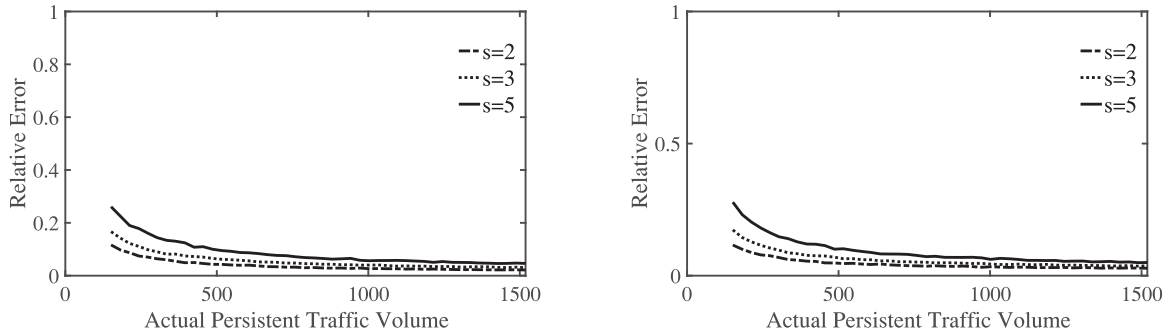


Fig. 5. Relative error of point-to-point persistent traffic estimation when $t = 7, f = 2$. Left plot: same average traffic at L and L'; right plot: average traffic at L' is 10 times of that at L.

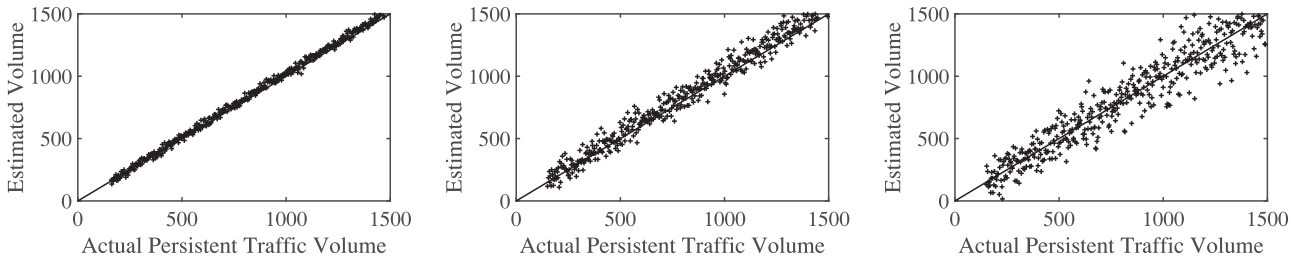


Fig. 6. Measurement accuracy of persistent traffic volume when $t = 5, s = 3$, and $f = 1$. First Plot: point persistent traffic; second Plot: point-to-point persistent traffic; third Plot: three-point persistent traffic.

2 to 3, the estimation accuracy is visibly better. Recall that f is the ratio of the bitmap size and the expected traffic volume. Increasing f means that the bitmap size is increased, which reduces the mixing of information from different vehicles, thus improving accuracy, but in the meantime reducing privacy protection, as we will see next.

In Fig. 9, we vary the traffic volume for three locations. Let $n_{i,max}$ be the maximum number of vehicles that pass L_i in one measurement period. Then, we set $n_{1,max} = 10000$, and the relationship of $n_{1,max} : n_{2,max} : n_{3,max}$ in the three sub-figures of Fig. 9 are set to $1 : 2 : 4$, $1 : 4 : 16$ and $1 : 8 : 64$ respectively. Correspondingly, the values of m_i's satisfy $m_1 : m_2 : m_3 = 1 : 2 : 4$, $1 : 4 : 16$ and $1 : 8 : 64$, respectively. The simulation results are similar with those for the point-to-point scheme. The gap of the traffic volume among different locations only slightly reflect the estimation accuracy, which indicates that our bitmap expansion works well. Even under the case of $m_1 : m_2 : m_3 = 1 : 8 : 64$, our scheme still has a high estimation accuracy.

7.3 Preserved Privacy

In Table 4, we examine privacy protection by measuring the probabilistic noise-to-information ratio with respect to f and s. We know that the larger this ratio is, the better the privacy protection will be, because it will become increasingly uncertain to use the traffic records to track individual vehicles. We want this ratio to be at least greater than 1. We see from the table that the ratio increases when f decreases or s increases. Earlier we have observed that the estimation accuracy moves in the opposite direction: it decreases when f decreases or s increases. So there is a tradeoff between accuracy and privacy. Based on all our numerical evaluations, we believe $f = 2$ and $s = 3$ make a good compromise between the two. Under these parameters, our accuracy evaluation has consistently produced good results, and the probabilistic noise-to-information ratio is about 2 as shown in the table. In the last row, we also give the noise probability p that the traffic records will show a vehicle passes both locations even when it actually does not. The value of p only depends on f. It is about 40 percent when $f = 2$. A noise-to-information ratio of 2 implies a probability of 60 percent that the traffic records will show a vehicle passes both locations when it does, including the noise contribution of 40 percent. Noise (40 percent) overwhelms information (20 percent) by a ratio of 2 to 1, making any tracking result very questionable.

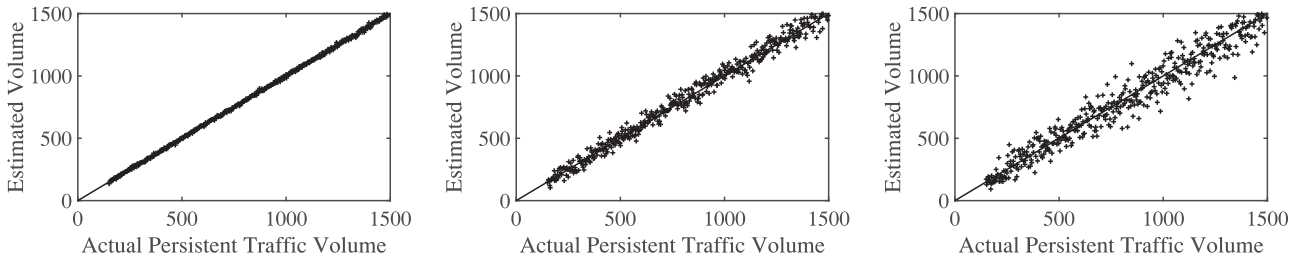


Fig. 7. Measurement accuracy of persistent traffic volume when $t = 5, s = 3$, and $f = 2$. First Plot: point persistent traffic; second Plot: point-to-point persistent traffic; third Plot: three-point persistent traffic.

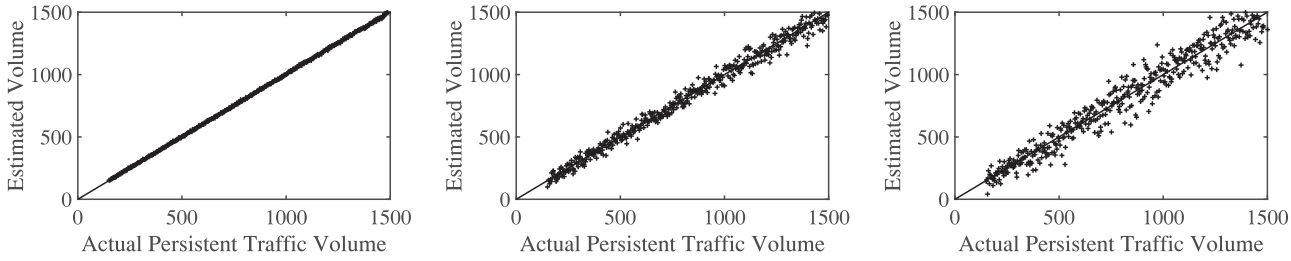


Fig. 8. Measurement accuracy of persistent traffic volume when $t = 5, s = 3$, and $f = 3$. First Plot: point persistent traffic; second Plot: point-to-point persistent traffic; third Plot: three-point persistent traffic.

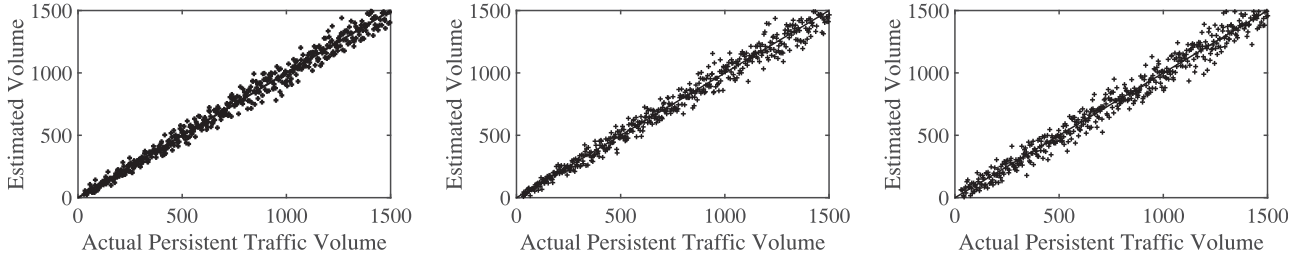


Fig. 9. Measurement accuracy of three-point persistent traffic volume when $t = 5, s = 3$, and $f = 2$. First plot: $m_1 : m_2 : m_3 = 1 : 2 : 4$; second plot: $m_1 : m_2 : m_3 = 1 : 4 : 16$; third plot: $m_1 : m_2 : m_3 = 1 : 8 : 64$.

8 Related Work

Transportation traffic measurement refers to the estimation of the point or multi-point traffic volume during a particular measurement period. Note that, we use point traffic to denote the vehicles that traversing a particular geographical location (e.g., road intersection). Hence, point traffic measurement often returns in the form of annual average daily traffic (AADT). Various prediction methods [18], [23], [26] have been proposed to measure single-point traffic volume based on the data recorded by automatic traffic recorders (ATR) installed at road intersections. For example, [18] employs a support vector regression model to evaluate the volume, [26] introduces an absolute deviation penalty procedure, and [23] uses the regression and Bayesian model to solve this problem. Different from point traffic, multi-point traffic refers to the common vehicles that travel through multiple particular independent locations. In fact, the passing locations of a specific vehicle are part of its trajectory. Therefore, we should consider the privacy protection issues in the multi-point traffic measurement. Efficient mechanisms (such as [21]) have been proposed to achieve differential privacy for cardinality estimation between two data sets. However, those two data sets do not exist in our work since all RSUs belong to the same authority. Therefore, we prevent RSUs from learning the IDs of vehicles by encoding them in anonymous bitmaps, and further study how to protect the trajectories of vehicles even when the point location privacy of vehicle has been leaked. Various methods are proposed to estimate the multi-point traffic. Lou and Yin proposed a model to infer point-to-point statistics from single-point data [14], but it has limited practicability for its high computation overhead. To solve this problem, Zhou et al. studied the point-to-point and multi-point traffic measurement problems respectively in [31] and [30], which can balance the privacy preservation and measurement accuracy. We study a new problem in this work, which aims to measure the persistent traffic volume. To solve this problem, we need to join not only the bitmaps of different locations like the existing studies but also the bitmaps of different measurement periods. Therefore, this problem meets more challenges if we want to achieve both measurement accuracy and privacy.

Network traffic measurement is another branch of the traffic measurement, which is similar to transportation traffic

TABLE 4
Privacy Preserving: The Probabilistic Noise-to-Information Ratio and Noise p

s \ f	$f = 1$	$f = 1.5$	$f = 2$	$f = 2.5$	$f = 3$	$f = 3.5$	$f = 4$
$s = 2$	3.4368	1.8956	1.2975	0.9837	0.7912	0.6614	0.5681
$s = 3$	5.1553	2.8433	1.9462	1.4755	1.1869	0.9922	0.852
$s = 4$	6.8737	3.7911	2.5950	1.9673	1.5825	1.3229	1.1361
$s = 5$	8.5921	4.7389	3.2437	2.4592	1.9781	1.6536	1.4201
p	0.6321	0.4866	0.3935	0.3297	0.2835	0.2485	0.2212

measurement. Various methods [2], [3], [8], [13], [28], [29] have been proposed for network traffic measurement, which is to measure the network traffic in a network router. However, we need to protect the trajectory privacy of vehicles in transportation traffic measurement, which is not considered in the network traffic measurement. Thus, none of these network traffic measurement methods can be directly employed in the scenario of transportation traffic measurement.

9 Conclusion and Future Work

This paper studies the new problem of persistent traffic measurement in the context of intelligent vehicular networks, where the vehicles can communicate with the RSUs wirelessly. We present the operation protocol that the RSUs use to encode the vehicles in their traffic records. We first propose three novel estimators for measuring point persistent traffic volume, point-to-point persistent traffic volume and three-point persistent traffic volume, and then generalize our method for multi-point persistent traffic measurement. The estimator design considers both measurement accuracy and privacy preservation. We analyze the preserved privacy of the estimators. The numerical evaluation demonstrates the effectiveness of the proposed methods in producing high measurement accuracy and allowing accuracy-privacy tradeoff through parameter setting.

In this paper, for point-to-point persistent traffic measurement, we do not consider the directionality of traffic. The estimated result includes traffic in both directions. In our future work, we will study directional persistent traffic measurement, which provides more detailed information on persistent traffic that travels from one location to another location of interest.

Acknowledgements

The research of authors is partially supported by National Science Foundation (NSF) NeTS 1719222, NSF NeTS 1718708, National Natural Science Foundation of China (NSFC) under Grant No. 61672369, No. 61572342, and the Natural Science Foundation of Jiangsu Province under Grant No. BK20161258. The research of Kai Han is partially supported by NSFC under Grant No. 61772491, No. 61472460, and NSF of Jiangsu Province under Grant No. BK20161256. The research of Hongli Xu is supported by the NSFC under Grant No. 61472383, U1709217, and 61728207, and the Natural Science Foundation of Jiangsu Province in China under No. BK20161257. This work is also supported by the grant from the Florida Cybersecurity Center. The preliminary version of this paper appeared in IEEE ICDCS 2017 [9].

References

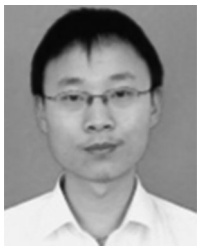
[1] [Online]. Available: https://www.cfxway.com
[2] J. Cao, A. Chen, and T. Bu, "A quasi-likelihood approach for accurate traffic matrix estimation in a high speed network," in *Proc. 27th Conf. Comput. Commun.*, 2008, pp. 21–25.
[3] H. Dai, M. Shahzad, A. X. Liu, and Y. Zhong, "Finding persistent items in data streams," *Proc. VLDB Endowment*, vol. 10, no. 4, pp. 289–300, 2016.
[4] J. K. Eom, M. S. Park, T. Heo, and L. F. Huntsinger, "Improving the prediction of annual average daily traffic for nonfreeway facilities by applying a spatial statistical method," *J. Transp. Res. Board*, vol. 1968, pp. 20–29, 2006.
[5] J. Eriksson and H. Balakrishnan, "Cabernet: Vehicular content delivery using WiFi," *Proc. 14th ACM Int. Conf. Mobile Comput. Netw.*, 2008, pp. 199–210.
[6] C. Estan, G. Varghese, and M. Fish, "Bitmap algorithms for counting active flows on high-speed links," *IEEE/ACM Trans. Netw.*, vol. 14, no. 5, pp. 925–937, Oct. 2006.
[7] P. Fuxjaeger, S. Ruehrup, T. Paulin, and B. Rainer, "Towards privacy-preserving wi-fi monitoring for road traffic analysis," *IEEE Intell. Transp. Syst. Mag.*, vol. 8, no. 3, pp. 63–74, Fall 2016.
[8] H. Huang, Y.-E Sun, S. Chen, S. Tang, K. Han, J. Yuan, and W. Yang, "You can drop but you can't hide: K-persistent spread estimation in high-speed networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2018, pp. 1889–1897.
[9] H. Huang, Y.-E Sun, S. Chen, H. Xu, and Y. Zhou, "Persistent traffic measurement through vehicle-to-infrastructure communications," in *Proc. 37th Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 394–403.
[10] W. Lam and J. Xu, "Estimation of AADT from short period counts in Hong Kong – A comparison between neural network method and regression analysis," *J. Adv. Transp.*, vol. 34, pp. 249–268, 2000.
[11] L. J. LeBlanc, E. K. Morlok, and W. P Pierskalla, "An efficient approach to solving the road network equilibrium traffic assignment problem," *Transp. Res.*, vol. 9, no. 5, pp. 309–318, 1975.
[12] U. Lee, J. Lee, J. Park, and M. Gerla, "FleaNet: A virtual market place on vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 344–355, Jan. 2010.
[13] T. Li, S. Chen, and Y. Qiao, "Origin-destination flow measurement in high-speed networks," in *Proc. IEEE INFOCOM*, 2012, pp. 2526–2530.
[14] Y. Lou and Y. Yin, "A decomposition scheme for estimating dynamic origin-destination flows on actuation-controlled signalized arterials," *Transp. Res. Part C*, vol. 18, pp. 643–655, 2010.
[15] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, Aug. 2014.
[16] D. Mohamad, K. C. Sinha, T. Kuczek, and C. F. Scholer, "Annual average daily traffic prediction model for county roads," *J. Transp. Res. Board*, vol. 1617, pp. 69–77, 1998.
[17] Y. L. Morgan, "Notes on DSRC & WAVE standards suite," *IEEE Comm. Surveys Tutorials*, vol. 12, no. 4, pp. 504–518, Oct. 2010.
[18] M. C. Neto, Y. Jeong, M. K. Jeong, and Lee D. Han, "AADT prediction using support vector regression with data-dependent parameters," *Expert Syst. Appl.*, vol. 36, pp. 2979–2986, Mar. 2009.
[19] M. Pan, P. Li, and Y. Fang, "Cooperative communication aware link scheduling for cognitive vehicular ad-hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 4, pp. 760–768, May 2012.
[20] SpoofMAC, Spoof your MAC address, 2015, https://github.com/feross/SpoofMAC
[21] R. Stanojevic, M. Nabeel, and T. Yu, "Distributed cardinality estimation of set operations with differential privacy," in *Proc. IEEE Symp. Privacy-Aware Comput.*, 2017, pp. 37–48.
[22] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, "An identity-based security system for user privacy in vehicular Ad Hoc networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 9, pp. 1227–1239, Sep. 2010.
[23] I. Tsapakis, W. H. Schneider, and A. Nichols, "A bayesian analysis of the effect of estimating annual average daily traffic for heavy-duty trucks using training and validation Data-Sets," *Transp. Planning Technol.*, vol. 36, pp. 201–217, 2013.
[24] USDOT, Traffic Monitoring Guide, 2013, http://www.fhwa.dot.gov/policyinformation/tmguide
[25] K.-Y. Whang, B. T. Vander-Zanden, and H. M. Taylor, "A linear-time probabilistic counting algorithm for database applications," *ACM Trans. Database Syst.*, vol. 15, no. 2, pp. 208–229, Jun. 1990.
[26] B. Yang, Y. Wang, S. Wang, and Y. Bao, "Efficient local AADT estimation via SCAD variable selection based on regression models," in *Proc. Chinese Control Decision Conf*, 2011, pp. 1898–1902.
[27] M. Yoon, T. Li, S. Chen, and J. Peir, "Fit a compact spread estimator in small high-speed memory," *IEEE/ACM Trans. Netw.*, vol. 19, no. 5, pp. 1253–1264, Oct. 2011.
[28] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Modeling Comput. Syst.*, 2003, pp. 206–217.
[29] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An information-theoretic approach to traffic matrix estimation," in *Proc. Conf. Appl. Technol. Architectures Protocols Comput. Commun.*, 2003, pp. 301–312.
[30] Y. Zhou, S. Chen, Z. Mo, and Q. Xiao, "Point-to-Point traffic volume measurement through variable-length bit array masking in vehicular cyber-physical systems," in *Proc. IEEE 35th Int. Conf. Distrib. Comput. Syst.*, 2015, pp. 51–60.

[31] Y. Zhou, S. Chen, Z. Mo, and Y. Yin, "Privacy preserving origin-destination flow measurement in vehicular cyber-physical systems," in *Proc. IEEE 1st Int. Conf. Cyber-Physical Syst. Netw. Appl.*, 2013, pp. 32–37.
[32] Y. Zhou, Z. Mo, Q. Xiao, S. Chen, and Y. Ying, "Privacy-preserving transportation traffic measurement in intelligent cyber-physical road systems," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3749–3759, May 2016.
[33] X. Zhu, S. Jiang, L. Wang, and H. Li, "Efficient privacy-preserving authentication for vehicular Ad Hoc networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 907–919, Feb. 2014.
[34] Y. Zhu, Y. Wu, and B. Li, "Vehicular Ad Hoc networks and trajectory-based routing," *Internet Things*, vol. 9, pp. 143–167, 2014.

Yu-e Sun received the PhD degree from the Shenyang Institute of Computing Technology, Chinese Academy of Science. She is an associate professor with the School of Rail Transportation, Soochow University, P.R. China. Her current research interests cnclude span traffic measurement, privacy preserving, algorithm design, and analysis for wireless networks. She is a member of the IEEE and ACM.

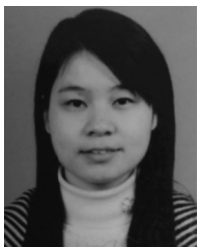
He Huang received the PhD degree from the Department of Computer Science and Technology, University of Science and Technology of China (USTC), in 2011. He is an associate professor in the School of Computer Science and Technology, Soochow University, P.R. China. His current research interests include traffic measurement, spectrum auction, privacy preserving in auction, and algorithmic game theory. He is a member of the IEEE and the ACM.

Shigang Chen received the BS degree in computer science from the University of Science and Technology of China in 1993, and the MS and PhD degrees in computer science from the University of Illinois at Urbana-Champaign, in 1996 and 1999, respectively. He is a professor with the Department of Computer and Information Science and Engineering, University of Florida. After graduating from UIUC, he was with Cisco Systems on network security for three years and helped start a network security company, Protego Networks. He joined the University of Florida as an assistant professor in 2002, and was promoted to associate professor in 2008 and to professor in 2013. He received the IEEE Communications Society Best Tutorial Paper Award in 1999, NSF CAREER Award in 2007, and the Cisco University Research Award in 2007 and 2012. He published more than 190 peer-reviewed journal/conference papers and received 12 US patents. He holds the University of Florida Research Foundation Professorship in 2017-2020 and the University of Florida Term Professorship in 2017-2020. He is a fellow of the IEEE, an ACM Distinguished member, and an IEEE ComSoc Distinguished lecturer.



Hongli Xu received the BS degree in computer science from the University of Science and Technology of China, in 2002, and the PhD degree in computer software and theory from the University of Science and Technology of China, in 2007. He is an associated professor with the School of Computer Science and Technology, University of Science and Technology of China. He has published more than 60 papers, and held about 20 patents. His main research interests include software defined networks, cooperative communication, and vehicular ad hoc network. He is a member of the IEEE.

Kai Han received the BS and PhD degrees in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 1997 and 2004, respectively. He is currently a professor with the School of Computer Science and Technology, USTC. His research interests include wireless ad hoc and sensor networks, mobile and cloud computing, combinatorial and stochastic optimization, algorithmic game theory, as well as machine learning. He is a member of the IEEE and ACM.



Yian Zhou received the BS degree in computer science and economics from the Peking University of China in 2010, and the PhD degree in computer and information science and engineering from the University of Florida, Gainesville, FL, in 2015, under the supervision of Prof. S. Chen. She is currently a software engineer with Google Inc. Her current research interests include traffic flow measurement, cyber-physical systems, RFID systems, big network data, and cloud computing. She is a member of the IEEE.