PeerProbe: Estimating Vehicular Neighbor Distribution With Adaptive Compressive Sensing

Yunxiang Cai, *Member, IEEE*, Hongzi Zhu, *Member, IEEE*, Shan Chang, *Member, IEEE*, Xiao Wang, *Member, IEEE*, Jiangang Shen, *Member, IEEE*, and Minyi Guo, *Fellow, IEEE*

***Abstract*— Acquiring the geographical distribution of neighbors can support more adaptive media access control (MAC) protocols and other safety applications in Vehicular ad hoc network (VANETs). However, it is very challenging for each vehicle to estimate its own neighbor distribution in a fully distributed setting. In this paper, we propose an online distributed neighbor distribution estimation scheme, called PeerProbe, in which vehicles collaborate with each other to probe their own neighborhood via simultaneous symbol-level wireless communication. An adaptive compressive sensing algorithm is developed to recover a neighbor distribution based on a small number of random probes with non-negligible noise. Moreover, the needed number of probes adapts to the sparseness of the distribution. We implement a prototype system to verify the feasibility of PeerProbe in various typical vehicular channel conditions. We further conduct extensive simulations and the results demonstrate that PeerProbe is lightweight and can accurately recover highly dynamic neighbor distributions in critical channel conditions.**

***Index Terms*— Neighbor distribution estimation, adaptive compressive sensing, vehicular ad hoc network, OFDM, Bloom filter.**

I. Introduction

VEHICULAR ad hoc networks (VANETs) are emerging as a new landscape of mobile ad hoc networks, aiming to provide a wide spectrum of safety and comfort applications to drivers and passengers. In VANETs, vehicles equipped with wireless communication devices can exchange data with each other (vehicle-to-vehicle communications). It is essential for a vehicle to acquire the context information of the network, especially, the geographical distribution of its neighbors. We refer to the *neighbor distribution estimation problem* as the problem that a vehicle can actively estimate the respective number of neighboring vehicles within a large set of different communication ranges. Figure 1 illustrates the neighbor distribution of a vehicle within three communication ranges. Such neighbor distribution information can be utilized to design efficient media access control (MAC) protocols. For instance, a vehicle can choose proper communication ranges and broadcasting periodicities for various safety applications. Moreover, given the number of neighbors within a chosen communication range, adaptive channel allocation schemes can be achieved.

A feasible scheme for the neighbor distribution estimation problem in VANETs, however, has to meet three rigid requirements as follows: 1) due to the fast movement of vehicles, such a scheme should be fast with low computation cost; 2) considering the limited spectrum resources, the method needs to be efficient in terms of both communication costs; 3) it is often the case that there is no base station or roadside unit available, which means that such a scheme should reliably work without any centralized unit in the network; 4) such a scheme should achieve very high accuracy as the derived information could be utilized in critical driving safety applications.

In the literature, though there is no existing neighbor distribution estimation scheme available in vehicular networks, the neighbor distribution information can be derived with localization or density estimation schemes. Active localization techniques, such as via in-vehicle GPS [1], [2], cellular signal trajectory [3], range-based schemes [4] and dead reckoning [5], [6], can be used to obtain the location information for individual vehicles. To get the neighbor distribution information, vehicles need to extensively exchange their location information via broadcasting, which incurs non-negligible communication costs. Passive localization techniques, such as utilizing a surveillance system [7], ranging sensors [8] and drones [9], can obtain collective location information of vehicles within a region of interest based on infrastructure, which introduces infeasible cost for dense deployment. Density estimation schemes can count the number of neighbors within one certain region. For example, the number of neighboring vehicles within one communication range of a vehicle can be estimated by collecting the IDs or hello messages of vehicles from received packets [10]. To acquire neighbor distribution, each vehicle needs to extensively measure the density of all communication ranges of interest, leading to huge communication costs. Density information can also be acquired through fixed installed detectors, such as cameras [11]–[13], loops [14], microphones [15] and ETC readers [16], which also confronts the deployment cost issue. As a result, to the

Manuscript received 1 April 2021; revised 26 November 2021; accepted 26 January 2022; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor B. Shrader. Date of publication 14 February 2022; date of current version 18 August 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC1900700; in part by the National Natural Science Foundation of China under Grant 61772340, Grant 61672151, and Grant 61972081; in part by the Shanghai Rising-Star Program under Grant 17QA1400100; and in part by the DHU Distinguished Young Professor Program. *(Corresponding authors: Hongzi Zhu; Shan Chang.)*

Yunxiang Cai, Hongzi Zhu, Xiao Wang, Jiangang Shen, and Minyi Guo are with the Department of Computer Science and Technology, Shanghai Jiao Tong University, Shanghai 200000, China (e-mail: hongzi@cs.sjtu.edu.cn).

Shan Chang is with the School of Computer Science and Technology, Donghua University, Shanghai 201620, China (e-mail: changshan@dhu.edu.cn).

Digital Object Identifier 10.1109/TNET.2022.3149008

best of our knowledge, there is no cost-efficient scheme which can accurately obtain the neighbor distribution information in a fully distributed setting.

In this paper, we propose a fully distributed neighbor distribution estimation scheme, called *PeerProbe*, in which vehicles in vicinity collaborate with each other to probe the neighborhood via symbol-level wireless communication. Instead of letting each vehicle measure the number of neighbors within all communication ranges to get the neighbor distribution, inspired by the insight that the geographical distribution of vehicles normally could be sparse, the core idea of PeerProbe is for each vehicle to utilize compressive sensing to recover its neighbor distribution by randomly measuring the number of neighboring vehicles only within a few randomly selected communication ranges.

Three main challenges are encountered. First, it is non-trivial to count the number of neighbors of each vehicle within a specific communication range, especially when there is no central unit available to coordinate data transmission and to collect information sent from individual vehicles. To address this challenge, a Bloom filter is constructed for each vehicle in a distributed fashion by all neighbors within the communication range. More specifically, with common half-duplex radios, each vehicle randomly chooses to be a transmitter or a receiver. When a vehicle acts as a transmitter, it broadcasts an *individual bitmap* generated by making a hash of its own identity. When the vehicle acts as a receiver, it decodes superimposed signals to collect bitmaps from transmitter neighbors. This procedure repeats until each vehicle has collected enough bitmaps from most of its neighbors with high probability.

Second, it is challenging to efficiently and reliably exchange individual bitmaps among vehicles. It would be infeasible if each vehicle broadcasts individual bitmaps by packets in turn. To tackle this challenge, we use On-OFF Keying (OOK) mapping to embed the individual bitmap of a vehicle into Orthogonal Frequency Division Multiplexing (OFDM) symbols, letting each subcarrier of an OFDM symbol carry one bit. Furthermore, all transmitter neighbors simultaneously transmit their OFDM symbols while a receiver can successfully decode these superimposed OFDM symbols by an effective de-mapping scheme, deriving a *combined bitmap* which is the logical OR result of all transmitted individual bitmaps. With a series of combined bitmaps collected, each vehicle aggregates enough information to estimate the number of neighbors within a communication range.

Third, the geographical distribution of vehicles varies, which could be neither even (*e.g.*, due to traffic lights) nor sparse (*e.g.*, due to traffic jams) on a road, which makes it difficult to recover the neighbor distribution with normal compressive sensing methods. To deal with this challenge, we propose an adaptive compressive sensing scheme, in which the number of measures needed for a vehicle to recover its neighbor distribution adapts to the sparseness of its neighborhood. Moreover, probabilistic measurement noise is first filtered out via Total Variation (TV) regularization before being applied to compressive sensing to recover the final neighbor distribution.

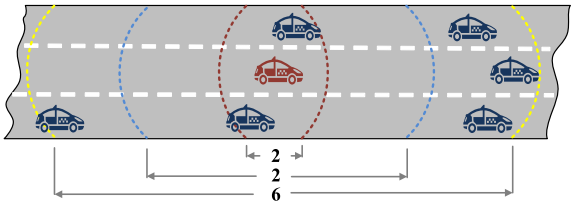


Fig. 1. Illustration of the neighbor distribution of a red vehicle, *e.g.*, the number of neighbors is 2, 2, and 6, respectively, within three consecutive communication ranges.

PeerProbe is a fully distributed scheme and requires no special hardware. We implement a prototype system, using four synchronized USRP N210 devices, to verify the feasibility of demodulating superimposed OOK-mapping OFDM symbols. Seven typical vehicular channel models are emulated and the average bit-error-rate (BER) measured in most critical vehicular channel conditions is about 2.5%. Extensive simulations are conducted. The results demonstrate that the estimation errors of the number of neighbors in one measure can reach up to 10% due to demodulation errors and the probabilistic noise. Nevertheless, the adaptive compressive sensing scheme is resilient to measurement noise and outperforms traditional compressive sensing schemes at low measurement costs.

We highlight the main contributions made in this work as follows:

- An efficient number-of-neighbors estimation method is proposed based on symbol-level wireless signals.
- An adaptive compressive sensing algorithm is developed to estimate neighbor distributions, which can handle massive measurement noise and adapt to varying densities of vehicles.
- A prototype system using four USRPs is implemented to verify the feasibility of PeerProbe by emulating typical vehicular channel conditions.
- Extensive simulations are conducted to demonstrate the efficiency of PeerProbe in achieving high accuracy neighbor distribution estimation under different channel conditions and node distributions.

The remainder of this paper is organized as follows. Section 2 introduces the system model and design goals. The detailed design of PeerProbe is presented in Section 3. The prototype implementation and performance evaluation are elaborated in Section 4 and Section 5, respectively. Design issues and case study are discussed in Section 6 and Section 7, respectively. Related work is presented in Section 8. In Section 9, we conclude our work and point out future direction of this work.

II. System Model and Design Goals

A. System Model

In the neighbor distribution estimation problem, we consider that there is no centralized unit in the network and vehicles are peers with equal capabilities as follows:

- **Communication:** vehicles can communicate with each other via OFDM-based wireless communication tech-

niques, such as Dedicated Short Range Communications (DSRC), which can actively control transmission power to change the communication range. Given the analysis based on real-world DSRC trace [17], we consider symmetric channels, *i.e.*, a vehicle x is in the communication range of vehicle y if and only if y is in the communication range of x when they use the same transmission power to communicate.

- **Computation:** vehicles can perform basic operations such as random number generation, hash operation, and common matrix algebra operations.
- **Synchronization:** vehicles are equipped with GPS receivers as a synchronous time reference. GPS can provide 1 pulse per second (PPS) signal with an accuracy of less than 100 ns even for low-end GPS receivers [18]. When GPS signal is lost temporally, vehicles can maintain the synchronization for a while because of the stability of the oscillator of GPS [19]. Note that we do not directly use GPS for positioning as the reported location is quite inaccurate in urban settings.

B. Design Goals

We consider the following goals in designing our distributed neighbor distribution estimation scheme:

- **High Accuracy.** The estimated neighbor distribution information could be used for driving safety applications or for MAC protocol design. These applications have the urge for high accuracy of the estimated distribution with a fine spacial granularity (*e.g.*, at meter level).
- **Cost Efficiency.** The communication cost for neighbor distribution estimation should be extremely low to reserve most channel resource for data communication. Moreover, the computation complexity of the algorithms should be low to achieve fast response time.
- **Good Reliability.** The scheme should reliably work under highly dynamic vehicular environments, for example, uneven vehicle distributions due to various traffic conditions.
- **Easy Deployment.** The scheme should have a minimal hardware requirement, consisting of only widely-available cheap sensors and commercial off-the-shelf (COTS) vehicle-to-vehicle communication radio devices, which makes it easy to deploy.

III. Design of PeerProbe

A. Overview

To facilitate individual vehicles to efficiently estimate their neighborhood, PeerProbe integrates three techniques as follows. First, a vehicle only needs to randomly measure the number of neighbors in a few communication ranges, leveraging the insight that vehicle distribution could be sparse and the power of the compressive sensing theory. Moreover, the number of measures required for a vehicle adapts to the sparseness of its neighborhood. Second, a vehicle counts the number of neighbors within a communication range with a Bloom filter, which is neatly constructed in a distributed fashion by its neighbors. Third, OFDM symbols are designed so that superimposed OFDM symbols simultaneously transmitted by multiple vehicles can be successfully demodulated, which significantly increases the utility of channel resources.

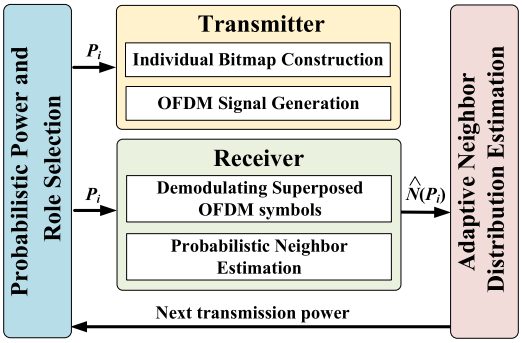


Fig. 2. The system architecture of PeerProbe.

The system architecture as depicted in Figure 2 consists of four major components:

1) Probabilistic Power and Role Selection: A vehicle randomly chooses a transmission power level P_i according to the network time. As vehicles are synchronized, it means that all vehicles would randomly select the same transmission power at a time. Given the transmission power level P_i, the vehicle chooses to be a transmitter (or a receiver) with the probability of p (or $1-p$). The probabilistic role selection repeats until a receiver has collected sufficient information from the transmitter neighbors within the communication range controlled by the transmission power level P_i.

2) Role of Transmitter: If the vehicle selects to be a transmitter, it first constructs an individual bitmap, which consists of a hashed identity (*e.g.*, using the MAC address of the vehicle) and an indication bit field and then embeds the bitmap into OFDM symbols using OOK mapping with each subcarrier conveying one bit. Finally, the vehicle transmits these OFDM symbols with the transmission power level of P_i.

3) Role of Receiver: If the vehicle selects to be a receiver, it demodulates received OFDM symbols, which are the superposition of those OFDM symbols sent from transmitter neighbors, and derives a combined bitmap. As one combined bitmap only contains the ID information of partial neighbors, the vehicle keeps collecting more combined bitmaps when it selects to be a receiver again. Finally, by aggregating the information from a series of combined bitmaps, the vehicle can estimate the number of neighbors within the communication range controlled by P_i, denoted by $\widehat{N}(P_i)$.

4) Adaptive Neighbor Distribution Estimation: Each time when a new $N(P_i)$ is estimated, the vehicle tries to recover the whole neighbor distribution using compressive sensing and compares the new result with previous result. If the difference is smaller than a given threshold, the neighbor estimation process converges and ends; otherwise, the vehicle chooses the next random transmission power level P_j and starts to estimate $N(P_j)$. Moreover, the vehicle also informs its neighbor in the communication range controlled by P_j of the new measure demand by setting the indication bit field in its individual bitmap.

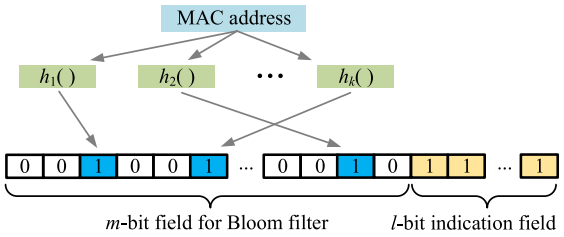


Fig. 3. An individual bitmap of a vehicle, consisting of an m-bit Bloom filter field and an l-bit indication field.

B. Probabilistic Power and Role Selection

Let $\{R_0, R_1, \cdots, R_{n-1}\}$ denote the set of communication ranges of interest in the neighbor geographical distribution estimation problem and let $\{P_0, P_1, \cdots, P_{n-1}\}$ denote the corresponding transmission power levels.[1] To randomly select a transmission power level P_i from $\{P_0, P_2, \cdots, P_{n-1}\}$, a vehicle uses the synchronized network time as the seed of a random number generator and $i = rand()\%n$.

As there is no centralized unit in the network, in order for each vehicle to collect information from neighbors within communication range R_i, we propose a probabilistic scheme, in which a vehicle v selects to be a transmitter with a probability of p and selects to be a receiver with a probability of $1-p$. When v is a transmitter, v broadcasts its own information with power P_i, and when v is a receiver, v can receive information from all transmitter neighbors within R_i. We have the following theorems:

***Theorem** 1: If each receiver vehicle expects to collect information from a proportion of λ of all its neighbors within R_i, then the probabilistic role selection procedure should be repeated at each vehicle for at least $\frac{1}{(1-p)} \cdot log_{(1-p)}(1-\lambda)$ times.*

Proof: We assume that the probabilistic role selection procedure should be repeated at least for x times and the number of neighbors is $N(P_i)$. Then, the expected number for a vehicle v to be a receiver is $x(1-p)$. When v is a receiver for the first time, the expected number of neighbors that v can receive information from is $N(P_i)p$. When v is a receiver for the second time, the expected number of new neighbors that v can receive information from is $N(P_i)(1-p)p$. Similarly, when v is a receiver for the i-th time, the expected number of new neighbors that v can receive information from is $N(P_i)(1-p)^{i-1}p$.Therefore, the total expected number of neighbors that v can receive information from is $N(P_i)(1-p)^0 p + N(P_i)(1-p)^1 p + \cdots + N(P_i)(1-p)^{x(1-p)-1}p$, which is no less than $\lambda N(P_i)$. We have

$$x \geq \frac{1}{(1-p)} \cdot log_{(1-p)}(1-\lambda), \tag{1}$$

which concludes the proof. □

Given the above theorem, x reaches the minimum when $p = 1 - \frac{1}{e}$.

[1]The specific propagation model about a transmission power level and the corresponding communication range is not the concern of this work. We hereafter use the transmission power level to refer to the corresponding communication range without the loss of generality.

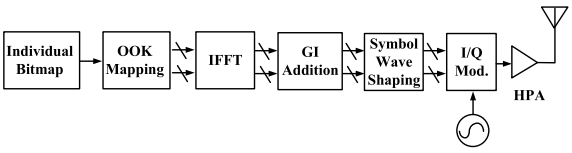


Fig. 4. The signal processing of transmitter.

***Theorem** 2: If each transmitter vehicle expects to inform a proportion of γ of all its neighbors within R_i, then the probabilistic role selection procedure should be repeated at each vehicle for at least $\frac{1}{p} \cdot log_p(1-\gamma)$ times.*

Proof: The proof is similar as that of Theorem 1. We assume that the probabilistic role selection procedure should be repeated at least for x times and the number of neighbors is $N(P_i)$. Then, the expected number for a vehicle v to be a transmitter is xp. When v is a transmitter for the first time, the expected number of neighbors that can receive v's information is $N(P_i)(1-p)$. When v is a transmitter for the second time, the expected number of new neighbors that can receive v's information is $N(P_i)p(1-p)$. Similarly, when v is a transmitter for the i-th time, the expected number of new neighbors that can receive information from v is $N(P_i)(p)^{i-1}(1-p)$. Therefore, the total expected number of neighbors that can receive information from v is $N(P_i)(p)^0(1-p) + N(P_i)(p)^1(1-p) + \cdots + N(P_i)(p)^{xp-1}(1-p)$, which is no less than $\lambda N(P_i)$. We have

$$x \geq \frac{1}{p} \cdot log_p(1-\lambda), \tag{2}$$

which concludes the proof. □

Given the above theorem, x reaches the minimum when $p = \frac{1}{e}$.

Considering both Theorem1 and Theorem 2, we set $p = 0.5$. As a result, if we expect each receiver vehicle to collect information from $\lambda = 95\%$ of its neighbors and each transmitter vehicle to inform $\gamma = 95\%$ of its neighbors within the communication range R_i, each vehicle needs to repeat the probabilistic role selection procedure for nine times.

C. Role of Transmitter

When a vehicle selects to be a transmitter, it helps other receiver neighbors to construct the bitmap of a Bloom filter by broadcasting its own individual bitmap through OFDM symbols with the transmission power level P_i.

1) Individual Bitmap Construction: As depicted in Figure 3, an individual bitmap consists of two bit fields, *i.e.*, an m-bit Bloom filter field and an l-bit indication field. A transmitter vehicle sets up the m-bit Bloom filter field according to the results of a group of k hash functions, calculated with its MAC address. The vehicle sets the indication bits to inform other vehicles that it also expects to collect information from other neighbors in this communication range. Note that l bits are set at the same time to mitigate the impact of demodulation error (see more details in § 3.4.1 and § 3.5.4).

2) OFDM Symbol Generation: We adopt OOK mapping to embed each bit in an individual bitmap into one corresponding subcarrier of an OFDM symbol. Specifically, bit 1 and

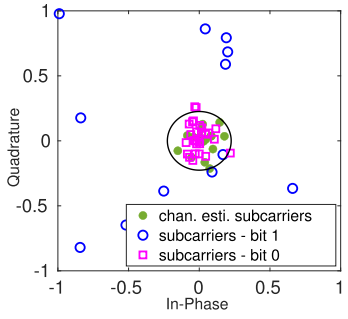


Fig. 5. Demodulation algorithm, where the Euclidean norm of a complex number is used to determine bit 1 or bit 0.

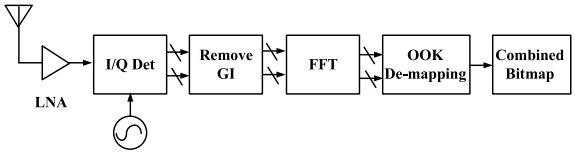


Fig. 6. The signal processing of receiver.

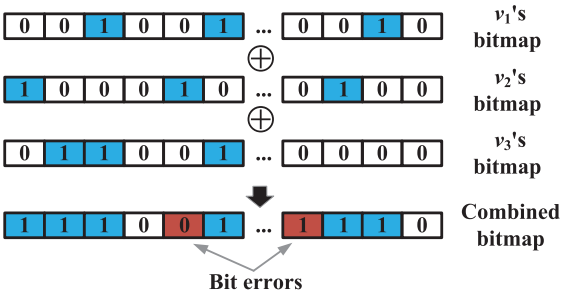


Fig. 7. An example of derived combined bitmap, which is actually the logical OR result of three individual bitmaps.

bit 0 are mapped to the symbol (1, 0) and (0, 0) in the constellation diagram, respectively. As depicted in Figure 4, the signal processing of a transmitter is compatible with IEEE 802.11p radios. Specifically, 48 subcarriers with index $[-26 : -22, -20 : -8, -6 : -1, 1 : 6, 8 : 20, 22 : 26]$ are used for bitmap transmission and the rest 16 subcarriers, including four pilot subcarriers, eleven null subcarriers and the DC subcarrier, always transmit bit 0, *i.e.*, symbol (0, 0), for channel estimation. Consequently, when the length of an individual bitmap is larger than 48, the bitmap is divided into multiple OFDM symbols. With the generated OFDM symbols, a transmitter vehicle broadcasts each OFDM symbol with the chosen transmission power level P_i according to the synchronized network time.

D. Role of Receiver

When a vehicle v selects to be a receiver, it collects combined individual bitmaps from transmitter neighbors and estimates the number of neighbors when enough combined bitmaps are collected.

1) Demodulating Superposed OFDM Symbols: Given the number of neighbors $N(P_i)$, there are on average $pN(P_i)$ transmitter neighbors, simultaneously broadcasting their individual bitmaps via OFDM symbols. These OFDM symbol signals are superimposed before they hit the antenna of v. Figure plots the signal processing of a receiver. To demodulate superimposed OFDM symbols, after FFT, a complex number $z_i = I_i + jQ_i$ can be obtained from subcarrier i. We introduce a simple yet effective OOK de-mapping algorithm. Specifically, the Euclidean norm of the complex on each channel estimation subcarrier is calculated, and the maximal norm is used as the threshold to determine bit 1 or bit 0 on other data subcarriers.

For example, Figure 5 plots the constellation diagram for one superimposed OFDM symbol through channel. The dark circle in the figure indicates the calculated threshold. It can be seen that most bits can be correctly demodulated except that there are three bits being wrong. As a result, the demodulated bitmap is actually the result of logical OR operation of those individual bitmaps. Moreover, bit errors would happen in the combined bitmap due to channel impairments. Figure 7 illustrates an example of an extracted combined bitmap from superimposed individual bitmaps.

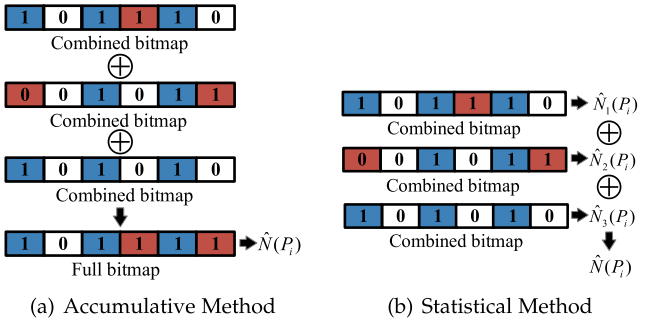


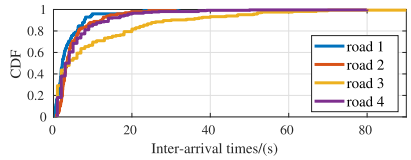
Fig. 8. (a) Accumulative method where a vehicle aggregates the ID information from neighbors by performing the logical OR operation on all combined bitmaps to obtain full bitmap and then using the full bitmap to estimate the number of neighbors. (b) Statistical Method where a vehicle estimates the number of neighbors by using each combined bitmap and averages a series of estimation results to get final neighbor number estimation.

2) Probabilistic Neighbor Estimation: According to Theorem 1 and Theorem 2, each vehicle with common half duplex communication ability needs to repeat the probabilistic role selection procedure at least $max\{\frac{1}{(1-p)} \cdot log_{(1-p)}(1-\lambda), \frac{1}{(p)} \cdot log_{(p)}(1-\lambda)\}$ times to collect enough information for neighbor number estimation within a communication range. With a series of collected combined bitmaps, v has two methods to aggregate the collected combined bitmaps from neighbors.

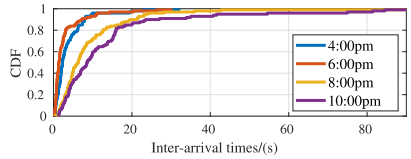
(a) *Accumulative Method.* As one combined bitmap only contains the ID information of partial neighbors, one straightforward scheme is to aggregate these combined bitmaps to obtain a full bitmap of a Bloom filter [20], which contains enough ID information of neighbors. As depicted in 8(a), the full bitmap is the result of logical OR operation of combined bitmaps. With the aggregated full bitmap, v can estimate the number of neighbors as the following equation [20]:

$$\widehat{N}(P_i) = -\frac{mln(1 - \frac{c(Z)}{m})}{k} \tag{3}$$

where m is the length of the bitmap; k is the number of hash functions and the function $c(\cdot)$ counts the number of ones within the bitmap.



(a) Inter-arrival time at difference places



(b) Inter-arrival time at difference times

Fig. 9. Illustration of observation 1. The geographical distribution of vehicles on a surface road could be sparse and the sparseness of the distribution varies over space and time.

The accumulative scheme is easy to implement and works well only in good communication channel conditions (or BER is low). As illustrated in Figure 8(a), the reason is that the error bits accumulate when the number of collected combined bitmaps increases.

(b) *Statistical Method*. Each combined bitmap contains ID informations of sampled neighbors and can be used to estimate the number of transmitters $N_j(P_i)$ with a factor $\frac{1}{p}$,where p denotes the probability of being a transmitter, *i.e.*,

$$\widehat{N_j}(P_i) = -\frac{mln(1-\frac{c(Z)}{m})}{k}\frac{1}{p} \tag{4}$$

As illustrated in Figure 8, because the estimation based on one single combined bitmap is not stable, we calculate the average of $\widehat{N_j}(P_i)$ based on a series of n combined bitmaps,

$$\widehat{N}(P_i) = \frac{1}{n}\sum_{j=1}^{n}\widehat{N_j}(P_i). \tag{5}$$

E. Adaptive Neighbor Distribution Estimation

In order to fully understand the characteristics of distribution of vehicles, we record the inter-arrival time of two successive vehicles at some place. The sparsity of spatial distribution can be estimated with the inter-arrival time. Given a certain speed, larger inter-arrival time means sparser the distribution is. We randomly selected 4 surface roads in city, which are thought to be crowded and often in traffic jams. At each road we record the arrival time of each vehicle from 4:00 pm to 10:00 pm with an interval of 2 hours and plot the CDF (Cumulative Distribution Function) of inter-arrival times in Figure 9(a) and 9(b). Although the measurements include the evening rush hours, there are still more than 20% of inter-arrival times larger than 10s, which means that the distance between cars is about 200 meters with the average speed of 10m/s. On the other hand, if there is a traffic jam, the road will become crowded. As for the geographical distribution of vehicles, we have the following observation.

***Observation** 1: The geographical distribution of vehicles on a surface road could be sparse and the sparseness of the distribution varies over space and time.*

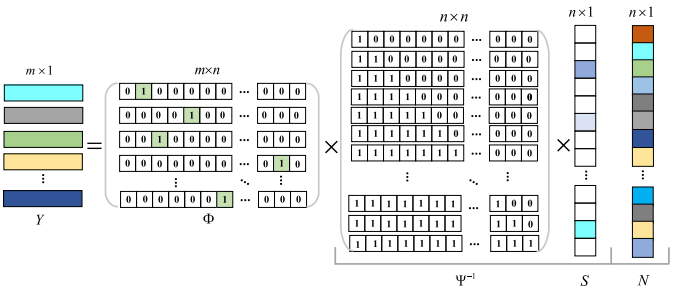


Fig. 10. Illustration of compressive sensing, where Y is the measurement vector; Φ is an $m \times n$ measurement matrix; Ψ is an $n \times n$ differential basis; S is the transformed vector in a sparse domain; N is the expected neighbor distribution vector.

Recent advances in the field of compressive sensing have developed reliable recovery algorithms for inferring sparse representations if one can randomly measure arbitrary linear combinations of the signal. Then the signal could be reliably reconstructed through solving an optimization problem [21]. Given the above observation, we leverage compressive sensing theory to recover the neighbor distribution of individual vehicles at the cost of a small number of random measures. Moreover, the number of measures needed for a vehicle adapts to the sparseness changes of its neighbor distribution.

1) Determining the Representation Basis: We define the geographical neighbor distribution of a vehicle v as a sequence of two-tuples $\{(P_0, N(P_0)), (P_1, N(P_1)), (P_2, N(P_2)), \cdots, (P_{n-1}, N(P_{n-1}))\}$, where $P_0 < P_1 < \cdots < P_{n-1}$. It is clear that $N(P_0) \leq N(P_1) \leq \cdots \leq N(P_{n-1})$ and the vector $N = (N(P_0), N(P_1), \cdots, N(P_{n-1}))$ is not sparse. However, as depicted in Figure 10, N could be transformed to a sparse vector S, *i.e.*, $S = \Psi N$ and $N = \Psi^{-1}S$, where Ψ is an $n \times n$ differential basis and Ψ^{-1} is a lower triangular matrix,

$$\Psi = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}, \tag{6}$$

$$\Psi^{-1} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix}. \tag{7}$$

For instance, $N = (2, 2, 6)$ and $S = (2, 0, 4)$ in the example of Figure 1.

2) Constructing the Measurement Matrix: The measurement matrix Φ is an $m \times n$ matrix, where m equals to the number of measures and each row corresponds to the transmit power levels from P_0 to P_{n-1}. As described in § III-B, each vehicle randomly chooses a transmission power level $P_i \in \{P_0, P_1, \cdots, P_{n-1}\}$ according to the network time for m times and estimates the corresponding number of neighbors $\widehat{N}(P_i)$. As shown in Figure 10, if P_j is chosen in the i-th measure, then Φ_{ij} is set to 1. Let Y denotes the vector of the

TABLE I
VEHICULAR CHANNEL MODELS

Channel Model	Doppler Shift (Hz)	Path Delays (ns)	Average Path Gains (dB)
Ch1: VET Express Oncoming	1000-1200	[0, 1, 2, 100, 101, 200, 201, 202, 300, 301, 302]	[0, 0, 0, -6.3, -6.3, -25.1, -25.1, -25.1, -22.7, -22.7, -22.7]
Ch2: RTV Expressway	600-700	[0, 1, 2, 100, 101, 102, 200, 201, 300, 301, 400, 401]	[0, 0, 0, -9.3, -9.3, -9.3, -20.3, -20.3, -21.3, -21.3, -28.8, -28.8]
Ch3: VTV Express SDWW	900-1150	[0, 1, 100, 101, 200, 300, 400, 401, 500, 600, 700, 701]	[0, 0, -11.2, -11.2, -19, -21.9, -25.3, -25.3, -24.4, -28, -26.1, -26.1]
Ch4: RTV Urban Canyon	300	[0, 1, 2, 100, 101, 102, 200, 201, 300, 301, 500, 501]	[0, 0, 0, -11.5, -11.5, -11.5, -19, -19, -25.6, -25.6, -28.1, -28.1]
Ch5: RTV suburban street	300-500	[0, 1, 100, 101, 200, 201, 300, 301, 400, 500, 600, 700]	[0, 0, -9.3, -9.3, -14, -14, -18, -18, -19.4, -24.9, -27.5, -29.8]
Ch6: VTV Urban Canyon Oncoming	400-500	[0, 1, 100, 101, 102, 200, 201, 202, 300, 301, 400, 401]	[0, 0, -10, -10, -10, -17.8, -17.8, -17.8, -17.8, -21.1, -21.1, -26.3, -26.3]
Ch7: HIPERLAN-E	1000-1200	[0, 10, 20, 40, 70, 100, 140, 190, 240, 320, 430, 560, 710, 880, 1070, 1280]	[–4.9, -5.1, -5.2, -0.8, -1.3, -1.9, -0.3, -1.2, -2.1, -0, -1.9, -2.8, -5.4, -7.3, -10.6, -13.4, -17.4, -20.9]

m measures. Then we have:

$$Y = \Phi N = \Phi \Psi^{-1} S, \tag{8}$$

Since the neighbor estimation under certain communiucation range P_i has non-negligible errors, each measure $\widehat{N}(P_i)$ has noise η, Y can be represented as follows:

$$Y = \Phi N + \eta = \Phi \Psi^{-1} S + \eta, \tag{9}$$

where η denotes the vector of estimation errors.

3) Recovering the Neighbor Distribution Signal: According to (9), to recover N in the sparse domain is to solve the following l_0 optimization problem:

$$\widehat{S} = \arg\min_{S} \|S\|_0 \quad \text{s.t. } Y = AS \tag{10}$$

where Y and $A = \Phi\Psi^{-1}$ are known. The above minimization problem can be resolved with orthogonal matching pursuit (OMP) algorithms [22], [23] [24]. We adopt the CoSaMP algorithm [25] as it can identify many components during each iteration, which allows the algorithm to run faster for many types of signals. By the way, besides these two typical algorithm, there is a lot of on-going work studied to conquer this optimization problem but this topic is out of scope of this paper.

According to the compressive sampling theory, a K-sparse signal can be reconstructed from m measurements, if m satisfies $m \geq b \cdot \mu^2(\Phi, \Psi) \cdot K \cdot logn$ where b is a positive constant, and $\mu(\Phi, \Psi)$ is the coherence between measurement matrix Φ and representation basis Ψ [26]. The coherence metric measures the largest correlation between any two element of Φ and Ψ, defined as: $\mu(\Phi, \Psi) = \sqrt{n} \cdot \max_{1 \leq i,j \leq n} |\langle \phi_i, \psi_j \rangle|$. We can see that the smaller the coherence between Φ and Ψ is, the less measurements are needed to reconstruct the signal. Because Φ is randomly generated, Φ is largely incoherent with any fixed representation basis Ψ. $m = 3K$ to $4K$ is usually sufficient to perfectly recover the signal.

4) Adapting to Varying Sparseness: Different vehicles have distinct neighbor distributions, *i.e.*, sparseness of N varies among vehicles, which means each vehicle needs to conduct different number of measures. To this end, an adaptive neighbor distribution estimation algorithm is proposed. Specifically, after z ($z \geq 1$) measures have been conducted, a vehicle v ranks the z measures according to the corresponding transmission power levels and gets measurement vector

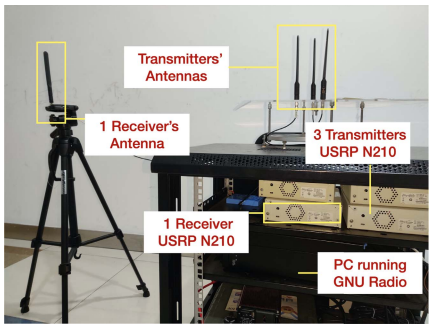


Fig. 11. A prototype implementation of PeerProbe, using four USRPs.

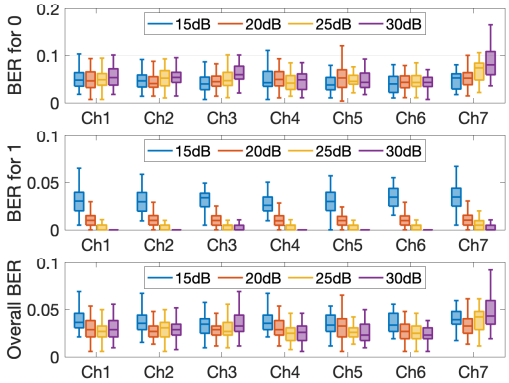


Fig. 12. BERs using seven typical vehicular channel models.

$M = \{\widehat{N}(P_{i0}), \widehat{N}(P_{i1}), \cdots, \widehat{N}(P_{iz})\}, P_{i0} < P_{i1} < \cdots < P_{iz}$. As the probabilistic neighbor estimation would introduce non-negligible measurement noise, M is first de-noised using the TV regularization [27]. Then, the de-noised z measures are used to recover the neighbor distribution, denoted as $\widehat{N}^{v,z}$. Similarly, when vehicle v continues to perform the $(z+1)$-th measure and it can get an updated recovered neighbor distribution $\widehat{N}^{v,z+1}$. Vehicle v calculates an *indicator* as

$$indicator = \frac{\|\hat{N}^{v,z+1} - \hat{N}^{v,z}\|_2}{\|\hat{N}^{v,z}\|_2} \tag{11}$$

If the indicator is smaller than a threshold ϵ, the neighbor distribution estimation process ends. In this case, vehicle v clears the l-bit indication field in its individual bitmap in all future broadcastings, but still needs to repeat the probabilistic role selection procedure as described in § 3.2 until the indication fields in all received combined bitmaps are cleared[2] (*i.e.*, the neighbor estimation process ends for all its neighbors).

IV. PROTOTYPE IMPLEMENTATION

We implement a prototype system of PeerProbe, using four Ettus USRP N210s and a PC computer. Each USRP is associated with a CBX daughterboard and a GPSDO module which is used to synchronize the USRP N210 reference clock to within 0.01 ppm of the worldwide GPS standard. The PC runs Ubuntu 16.04 LTS 64-bit as the operating system

[2]To mitigate the impact of demodulation errors, the indication field of a combined bitmap is considered clear if there are more than $\frac{l}{2}$ bits are zero.

and has one Intel i7-8700 CPU and 16GB memory. All four USRPs are divided into two groups and connected to the PC through two separate 1Gbps Ethernet ports for better throughput. We implement OFDM transceivers as described in § 3.3.2 and § 3.4.1 with GNU Radio and set three USRPs as transmitters and set one USRP as a receiver as illustrated in Figure. The central frequency and the bandwidth are 5.89GHz and 10MHz, respectively. The sample rate for each USRP is 10M samples per second, *i.e.*, 80Mbps for 8-bit I/Q samples.

To verify the feasibility of demodulating superimposed OFDM symbols in more practical conditions, we emulate vehicle-to-vehicle communication using seven typical vehicular channel models [28] according to the characteristics of each channel model as listed in Table I. These channel models are based on standard Tapped Delay Line (TDL) channel model, consisting seven common scenarios in VANETs. For each channel model, each transmitter transmits one million bits. The receiver demodulates the superimposed OFDM symbols and calculates the bit error rate (BER) for bit 0, BER for bit 1 and the overall BER. Figure 12 plots the BERs obtained with different signal-to-noise ratio (SNR). We have leant the following lessons. First, BER for bit 1 is usually higher than BER for bit 0 with the current de-mapping scheme. Second, as SNR increases, BER for bit 1 dramatically drops while BER for bit 0 increases. The reason is that the threshold used to distinguish bit 0 or bit 1 shrinks when SNR is good, which makes it more easier for a bit 0 to be recognized as a bit 1. Third, the overall BER is relative low with an average of 2.5% even for the most critical channel conditions. It is possible to devise an adaptive de-mapping scheme based on the SNR in the future.

V. Performance Evaluation

A. Methodology

We conduct extensive simulations using the VENUS simulator (http://lion.sjtu.edu.cn/project/projectDetail?id = 17) to evaluate the performance of PeerProbe with respect to various vehicle distributions and critical channel conditions. Specifically, we randomly generate a neighbor distribution, following three types of vehicle distributions, *i.e.*, uniform distribution, Gaussian distribution, and Poisson distribution. For each distribution type, we vary the total number of vehicles, ranging from 20 to 100. We consider a set of distinct BERs to reflect the impact of particular channel models. In addition, we also evaluate PeerProbe on a real-world trace, referred to as the highD dataset [9], which is collected on German highways using a drone and consists of more than 110,500 vehicles recored at six locations.

We define two accuracy metrics as follows:

- **Neighbor Estimation Error Ratio (NEER)**: refers to the ratio of the absolute error of the estimated number of neighbors with certain transmission power level P_i to the ground truth, calculated as $\frac{|\widehat{N}(P_i)-N(P_i)|}{N(P_i)}$.
- **Neighbor Distribution Estimation Error (NDEE)**: we measure the difference between the estimated neighbor distribution of a vehicle v and the ground truth using RMS error, calculated as $\sqrt{\frac{\sum_{i=1}^{n}(\widehat{N}_i^v-N_i^v)^2}{n}}$.

We compare the two aggregation methods named accumulated (acc) method and statistical (sta) method as introduced in § 3.4.2 to validate different parameters that affect NEER and compare our scheme with the following four candidate schemes by examining the NDEE metric:

- **Orthogonal Matching Pursuit (OMP)**: OMP [22], [23] [24] is a classic compressive sensing algorithm which leverages an iterative greedy algorithm for sparse signal recovery. OMP updates approximation for $\hat{S}$ in each iteration by adding components that are most strongly correlated with the remaining part of the signal.
- **Compressive Sampling Matching Pursuit (CoSaMP)**: CoSaMP [25] is an improved version of OMP which provides a theoretical reconstruction guarantee by using a backtracking framework.
- **NN interpolation (NN)**: is the nearest neighbor interpolation algorithm which selects the value of the nearest point, yielding a piecewise-constant interpolant.
- **Linear interpolation (Linear)**: is the linear interpolation algorithm which uses linear polynomials to construct new data points within the range of a discrete set of known data points.

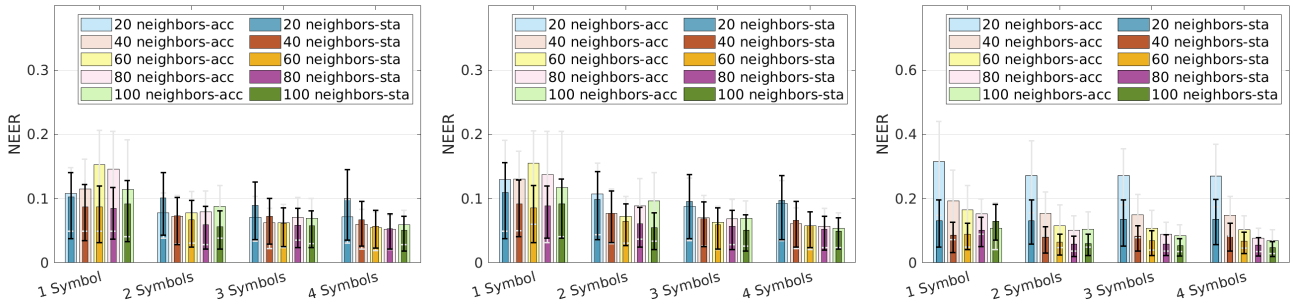
B. Accuracy of Estimating the Number of Neighbors

1) Effect of Individual Bitmap Length: We randomly generate uniform neighbor distributions with the number of neighbors varying from 20 to 100 with an interval of 20. We set the repeating times of the probabilistic role selection to nine and use two hash functions. We vary the length of individual bitmaps, ranging from using one OFDM symbol (*i.e.*, 48 bits) up to using four OFDM symbols (*i.e.*, 192 bits) and study the NEER under three BERs, *i.e.*, 0.1%, 1% and 5%. For each setting, we run the experiment for 500 times.

Figure 13 plots the NEER as the function of individual bitmap length with accumulated (acc) method and statistical (sta) method. It is clear to see that the NEER decreases as the bitmap length increases and the variance of NEER decreases as the bitmap length increases. Moreover, it can also be seen that a long bitmap can deal with large BERs and a dense neighbor distribution. In practice, low response time is essential for vehicular applications, which limits the maximal length of individual bitmaps. Therefore, the strategy is to choose the longest bitmap as long as the Quality of Service (QoS) of application scenarios can be guaranteed.

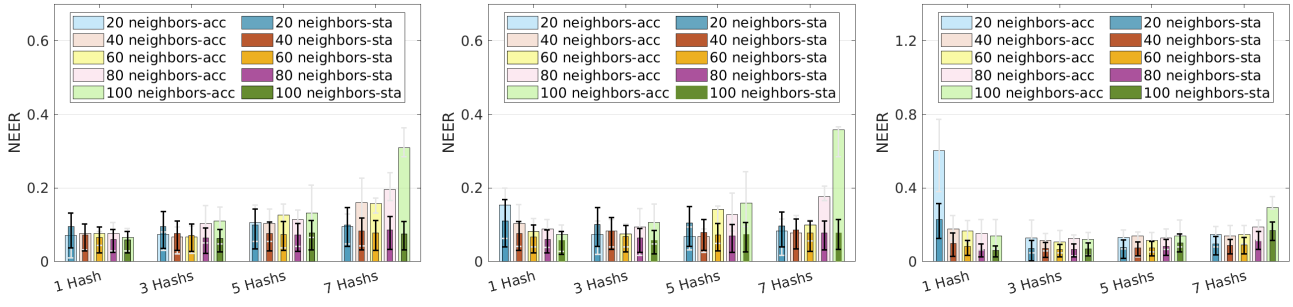
2) Effect of the Number of Hash Functions: The settings of this experiment are similar to the above experiment except that we fix the length of individual bitmaps using two OFDM symbols. We vary the number of hash functions, ranging from one to seven with an interval of two and study the NEER under the same four BERs. For each setting, we run the experiment for 500 times.

Figure 14 plots the NEER as the function of the number of hash functions. It can be seen that, on one hand, according to (3) the estimated $\widehat{N}(P_i)$ is inversely proportional to the number of hash functions, which can tolerate more bit errors



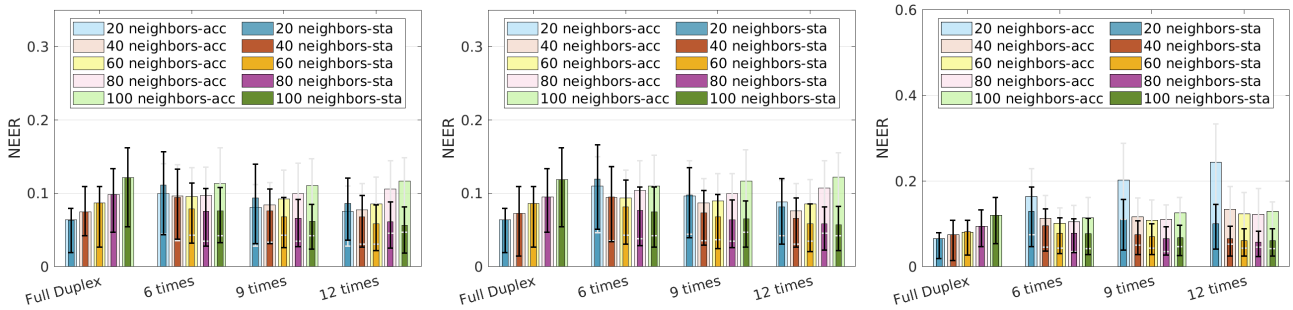
(a) BER=0.1% (b) BER=1% (c) BER=5%

Fig. 13. Neighbor estimation error ratio using different bitmap lengths under different BERs.



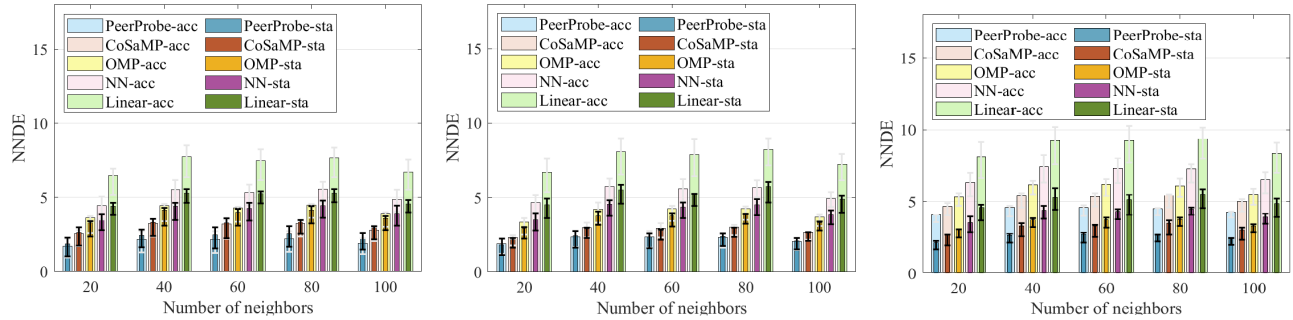
(a) BER=0.1% (b) BER=1% (c) BER=5%

Fig. 14. Neighbor estimation error ratio using different number of hash functions under different BERs.



(a) BER=0.1% (b) BER=1% (c) BER=5%

Fig. 15. Neighbor estimation error ratio using different role selection times under different BERs.



(a) BER=0.1% (b) BER=1% (c) BER=5%

Fig. 16. Impact of the neighbor density under different BERs.

when more hash functions are involved (*e.g.*, when BER is large, the NEER first drops as the number of hash functions increases); on the other hand, more hash functions will saturate the bitmap, which decreases the estimation accuracy of a Bloom filter (*e.g.*, when BER is large, the NEER increases as the number of hash functions is larger than three as shown in

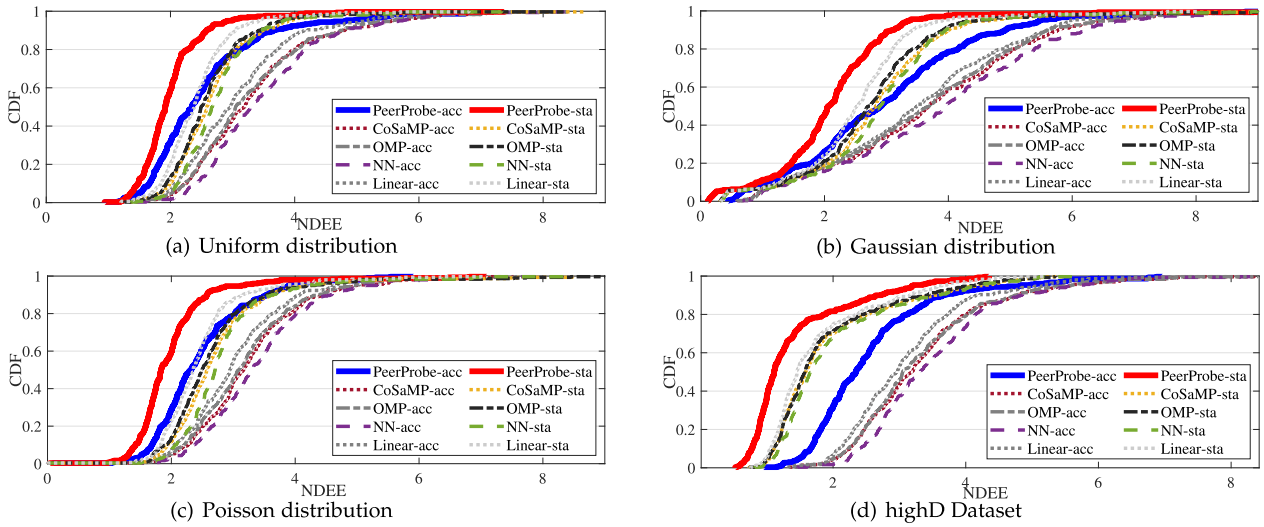


Fig. 17. Impact of distinct vehicle distribution types, *i.e.*, uniform distributions, Gaussian distributions, and Poisson distributions and a real-world vehicle trajectory dataset.

Figure 14). Given the length of individual bitmaps, the optimal number of hash functions can be determined.

3) Effect of Probabilistic Role Selection Times: The experiment settings are similar to the above experiment except that we fix the bitmap length using two OFDM symbols and use three hash functions. We vary the number of repeating times of probabilistic role selection, ranging from six to twelve with an interval of three and study the NEER under the same four BERs. In addition, we also consider full duplex radios (*i.e.*, the bitmap of a Bloom filter can be directly built by a receiver) for comparison. For each setting, we run the experiment for 500 times.

Figure 15 plots the NEER as the function of the repeating times of the probabilistic role selection with the two aggregation methods. The statistical method is better than accumulated method in almost all case. If we use accumulated method, according to Theorem 1, a receiver cannot collect bitmaps from enough neighbors when the repeating time is smaller than nine, leading to underestimated results. When the repeating time is larger than nine, more combined bitmaps are aggregated, leading to more accumulative bit errors and inaccurate estimation. In contrast, the statistical method performs better with the repeating time increases. So the strategy is to choose the larger repeating time as long as the QoS can be guaranteed.

C. Accuracy of Estimating Neighbor Distribution

1) Impact of Neighbor Density: Based on previous experiments, the bitmap length is fixed using two OFDM symbols (*i.e.*, 96-bit long) and three hash functions. The number of repeating times of probabilistic role selection is set to nine. We refer to the number of controllable transmission power levels as the dimension of neighbor distributions and set the dimension of neighbor distributions to 100. ϵ takes an empirical threshold of 0.03. We randomly generate uniform neighbor distributions with the number of neighbors varying form 20 to 100 with an interval of 20 and study the NDEE under three BERs, *i.e.*, 0.1%, 1% and 5%. For each setting, we run the experiment for 500 times.

Figure 16 plots the NDEE of five candidate estimation methods as the function of neighbor density. It can be seen that, in general, all methods can achieve stable performance when the neighbor density changes. BER has significant influence on the ultimate neighbor distribution estimation accuracy. NDEE increases as the neighbor density and BER increase. PeerProbe-sta outwits other methods.

2) Impact of Vehicle Distribution Types: We examine the impact of different vehicle distributions to the performance of PeerProbe. Specifically, we generate three typical types of geographical distributions of vehicles, *i.e.*, uniform distributions, Gaussian distributions, and Poisson distributions, using 2000 vehicles on a 4-lane road of 10km at a granularity of one meter. Furthermore, a real-world vehicle location dataset, called highD, is also considered. For uniform distributions, vehicles are uniformly distributed. For Gaussian distributions, the distribution parameter μ and σ are set to 5 km and 2.5 km, respectively. For Poisson distributions, the distribution parameter λ is set to 0.2. For all vehicle distribution types and the real-world vehicle trajectory trace highD, we estimate the neighbor distribution of each vehicle, using all candidate methods for comparison.

Figure 17 plot the CDFs of NDEE for all vehicles obtained in all distributions. It can be seen that PeerProbe can always achieve higher estimation accuracy, especially on the highD dataset. In addition, for all vehicle distributions, 90% of vehicles require less than 40 measures to recover a neighbor distribution of 100 ranges using PeerProbe. Figure 18 illustrates an example of recovered neighbor distribution using PeerProbe and all other methods.

D. Communication and Computation Costs

The communication delay equals to the needed number of OFDM symbols per bitmap $\times$ repeating times of role selection $\times$ the needed number of measures $\times$ the duration

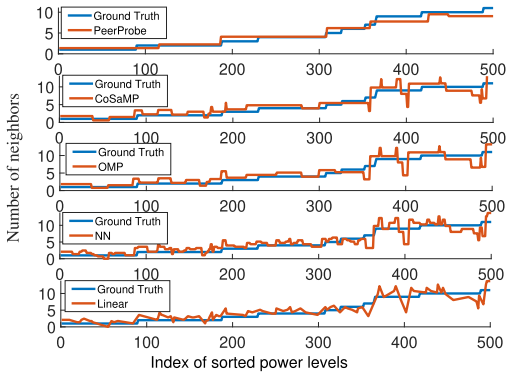


Fig. 18. An illustration of recovered neighbor distributions using different candidate methods.

of an OFDM symbol. For instance, for 2-symbol bitmaps, nine repeating times, 40 measures and 8 μs OFDM symbol duration, the communication delay for estimating one neighbor distribution is about 5.7 ms. The main computational cost stems from the CoSaMP algorithm which has a computational complexity of $O(K \cdot m^2 \cdot n)$ where K, m and n are the sparseness of the signal, the number of measures and the dimension of the signal. For example, the average running time of CoSaMP algorithm on a laptop with a 4-core 1.4GHz Intel Core i5 CPU and a 16GB memory is about 9ms in the experiment described § 5.3.2.

VI. Discussion of Design Issues

A. Doppler Effect

As vehicles are moving, the Doppler effect may cause a frequency shift at the receiver end. The relationship between received frequency f_{rx} and transmitted frequency f_{tx} can be characterized by

$$f_{rx} = (1 + \frac{\Delta v}{c}) f_{tx}, \tag{12}$$

where Δv is the relative speed between a transmitter and receiver pair and c is the speed of light. The frequency shift $\Delta f = \frac{\Delta v}{c} f_{tx}$ will destroy the orthogonality of OFDM subcarriers and cause inter-carrier interference, affecting the demodulation of superimposed OFDM symbols. From our emulation results, it can be seen that the influence of the Doppler effect is much smaller than that of the noise level. The reason is that the OOK modulation can effectively conquer the rotation of symbols caused by the Doppler effect in the constellation.

B. Malicious Vehicles

It is essential for all vehicles to follow the protocols of PeerProbe before accurate neighbor distribution information can be obtained at individual vehicles. Malicious vehicle can severely affect the performance of PeerProbe in multiple ways. For example, if a selfish vehicle simply keeps silent when it is supposed to transmit its bitmap, other vehicles cannot hear from this vehicle and therefore would underestimate the number of neighbors. To make things worse, a malicious vehicle can actively broadcast a bitmap set with all ones all the time, which makes other vehicles cannot estimate the number of neighbors through combined bitmaps at all. In fact, such deny-of-service attacks are easy to launch in any vehicular network applications. Though security issues are not concerned in the current design of PeerProbe, it would be necessary to integrate a secure authentication technique in the system. In this way, each vehicle are authenticated before performing PeerProbe protocols. Moreover, vehicles with misbehavior can be identified through offline analysis.

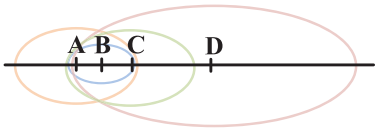


Fig. 19. An illustration of flexible broadcasting with each vehicle adopting different transmission power.

VII. Case Study

This section introduces two cases to demonstrate the potential of PeerProbe for solving problems in VANETs.

A. Flexible VANET MAC Protocols

In VANETs, vehicles need to periodically broadcast safety messages for safety applications. For instance, vehicles are required to exchange messages every 100ms using a fixed transmission power, leading to inefficient wireless spectrum utilization. Due to the highly dynamic nature of VANETs, to design a reliable and efficient MAC protocol in VANETs is very challenging [29], [30]. Given the neighbor distribution, both the optimal communication range and broadcasting periodicity of each vehicle can be determined.

With the help of PeerProbe, each vehicle can set a minimized communication range and periodicity according to the current neighbor distribution information rather than takes a fixed setting. For example, given the geographical distribution of four vehicles A, B, C and D as shown in Figure 19, each vehicle can estimate its neighborhood and adjust their transmission power to cover two of their nearest neighbors. In this case, if the MAC is based on TDMA, each vehicle can adopt a different transmission power (as indicated by the ellipses of different colors in Figure 19) and transmit their safety messages in allocated time slots. Therefore, the space utilization of wireless channels is improved.

B. Traffic Accident Monitoring

With the information of neighbor distribution estimation, PeerProbe can also be utilized to monitor traffic accidents. For instance, an RSU can obtain the vehicle distribution in its neighborhood via PeerProbe at a low deployment cost. As illustrated in the upper plot of Figure 20, if there is no traffic accident, the number of neighboring vehicles gradually increases with the distance from the RSU. In contrast, as shown in the lower plot of Figure 20, a surge of the number of vehicles at certain distance indicates a traffic jam,

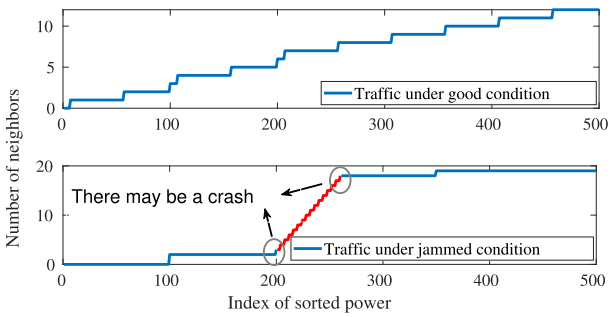


Fig. 20. An illustration of traffic jam monitoring.

probably caused by a traffic accident or a traffic light (always at given location). As a result, the traffic administration agency can quickly respond according to this information. Moreover, individual vehicles can report a potential traffic jam to the authority based on their own neighbor distribution patterns. Such a traffic monitoring scheme would be cost efficient and would not have issues commonly met in image-based solutions, such as privacy and bad weather or poor light conditions.

VIII. Related Work

We classify the existing work that is relevant to our work into two categories, *i.e.*, localization schemes aiming to estimate locations of vehicles over time and density estimation schemes aiming to estimate vehicle density in certain areas.

A. Localization Schemes

Localization schemes can be further divided into active localization techniques and passive localization techniques. In active localization schemes, vehicles actively obtain their individual locations. For example, vehicles equipped with GPS modules can directly get the latitude and longitude coordinates [1], [2]. Since commercial GPS devices may report errors on the order of ten meters in urban setting [31], [32], Xue *et al.* refine the location by matching its cellular-aware trajectory with a pre-constructed map [3]. Some methods combine GPS information and dead-reckoning sensors to reduce position errors [5], [6]. Some schemes utilize ranging sensors [8] to calibrate GPS positions received from neighbors. Physical layer information can also be utilized to refine location by determining the relative distance and angle of vehicles [33], [34]. With these schemes, vehicles need to extensively exchange their location information via broadcasting, which incurs non-negligible communication costs.

Passive localization techniques obtain collective location information of vehicles within a region of interest based on infrastructure. For instance, US Department of Transportation implements multiple synchronized digital video cameras mounted on top of buildings to record vehicle positions [7]. Robert *et al.* utilize computer vision method to record vehicle positions with drones [9]. These methods introduce infeasible costs for dense deployment and maintenance.

B. Density Estimation Schemes

Density estimation schemes can be divided into infrastructure based methods and infrastructure free methods. For infrastructure-based methods, various sensors are deployed to estimate the number of neighbors within one certain region. Vehicles in videos acquired by traffic surveillance systems can be detected and counted by computer vision methods [11]–[13]. Karandeep *et al.* utilize information provided by loop detectors to estimate traffic densities by using a Markov model approach. Vivek *et al.* implement a roadside microphone and estimate vehicle density with acoustic signal [15]. Mao *et al.* establish a maximum likelihood estimator to estimate vehicle density with ETC data [16]. These methods confront the deployment cost issue.

For infrastructure-free methods, the number of neighboring vehicles within one communication range of a vehicle can be estimated by collecting the IDs of neighbors or hello messages [10] from received packets. The most relevant work to our scheme is CoReCast [35] where a bloom filter constructed at packet level is used to count the number of neighbors of a vehicle within the communication range. These methods need to extensively measure the density of all communication ranges of interest, causing large communication costs. Besides, they need reliable packet communication and do not study the neighbor distribution. In contrast, PeerProbe is a distributed neighbor distribution estimation scheme without packet communication or any other prior assumption.

IX. Conclusion and Future Work

In this work, a distributed neighbor distribution estimation scheme PeerProbe has been developed in VANETs. In PeerProbe, vehicles can efficiently exchange hashed ID information through superimposed OFDM symbols. With a few rough measures on the number of neighbors in randomly selected communication range, a vehicle can accurately recover the neighbor geographical distribution, leveraging the power of the combination of TV de-noising technique and the compressive sensing theory. PeerProbe needs no centralized unit or any prior knowledge about vehicle distributions. Moreover, PeerProbe is lightweight and easy to deploy with a minimal requirement on hardware. We have implemented a prototype of PeerProbe and conducted extensive simulations. The results demonstrate the efficacy of PeerProbe. In the future, we plan to extend PeerProbe to other OFDM based vehicular communication technologies such as C-V2X or 5G-based C-V2X. Another direction is to evaluate PeerProbe in various real-world scenarios especially the reliability of demodulating superposed OFDM symbols.

References

[1] M. Rohani, D. Gingras, V. Vigneron, and D. Gruyer, "A new decentralized Bayesian approach for cooperative vehicle localization based on fusion of GPS and VANET based inter-vehicle distance measurement," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 2, pp. 85–95, Apr. 2015.

[2] S. B. Cruz, T. E. Abrudan, Z. Xiao, N. Trigoni, and J. Barros, "Neighbor-aided localization in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2693–2702, Oct. 2017.

[3] H. Xue, H. Zhu, S. Cao, S. Chang, and J. Cao, "UPS: Combatting urban vehicle localization with cellular-aware trajectories," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–7.

[4] C.-H. Ou, "A roadside unit-based localization scheme for vehicular ad hoc networks," *Int. J. Commun. Syst.*, vol. 27, no. 1, pp. 135–150, 2014.

[5] A. A. Wahab, A. Khattab, and Y. A. Fahmy, "Two-way TOA with limited dead reckoning for GPS-free vehicle localization using single RSU," in *Proc. 13th Int. Conf. ITS Telecommun. (ITST)*, Nov. 2013.

[6] W.-W. Kao, "Integration of GPS and dead-reckoning navigation systems," in *Proc. Vehicle Navigat. Inf. Syst. Conf.*, vol. 2, Oct. 1991, pp. 635–643.

[7] U.S. Department of Transportation. (2007). *NGSIM–Next Generation Simulation*. [Online]. Available: http://www.ngsim.fhwa.dot.gov

[8] E.-K. Lee, S. Y. Oh, and M. Gerl, "RFID assisted vehicle positioning in VANETs," *Pervasive Mobile Comput.*, vol. 8, no. 2, pp. 167–179, 2012.

[9] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2118–2125.

[10] S. Kuklinski and G. Wolny, "Density based clustering algorithm for VANETs," in *Proc. 5th Int. Conf. Testbeds Res. Infrastructures Develop. Netw. Communities Workshops*, Apr. 2009, pp. 1–6.

[11] S. Zhang, G. Wu, J. P. Costeira, and J. M. F. Moura, "FCN-rLSTM: Deep spatio-temporal neural networks for vehicle counting in city cameras," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3667–3676.

[12] C. Ozkurt and F. Camci, "Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks," *Math. Comput. Appl.*, vol. 14, no. 3, pp. 187–196, Dec. 2009.

[13] O. Asmaa, K. Mokhtar, and O. Abdelaziz, "Road traffic density estimation using microscopic and macroscopic parameters," *Image Vis. Comput.*, vol. 31, no. 11, pp. 887–894, 2013.

[14] K. Singh and B. Li, "Estimation of traffic densities for multilane roadways using a Markov model approach," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4369–4376, Nov. 2012.

[15] V. Tyagi, S. Kalyanaraman, and R. Krishnapuram, "Vehicular traffic density state estimation based on cumulative road acoustics," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1156–1166, Mar. 2012.

[16] R. Mao and G. Mao, "Road traffic density estimation in vehicular networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2013, pp. 4653–4658.

[17] F. Bai, D. D. Stancil, and H. Krishnan, "Toward understanding characteristics of dedicated short range communications (DSRC) from a perspective of vehicular network engineers," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2010, pp. 329–340.

[18] H. A. Omar, W. Zhuang, and L. Li, "VeMAC: A TDMA-based MAC protocol for reliable broadcast in VANETs," *IEEE Trans. Mobile Comput.*, vol. 12, no. 9, pp. 1724–1736, Sep. 2013.

[19] W. Franz and H. Hartenstein, *Inter-Vehicle-Communications Based on Ad Hoc Networking Principles: The FleetNet Project*, M. Mauve, Ed. Germany: Universitätsverlag Karlsruhe, 2005, p. 287, doi: 10.5445/KSP/1000003684.

[20] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, Jul. 1970.

[21] L. Yang, Y. Li, Q. Lin, H. Jia, X.-Y. Li, and Y. Liu, "Tagbeat: Sensing mechanical vibration period with COTS RFID systems," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3823–3835, Dec. 2017.

[22] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[23] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conf. Signals, Syst. Comput.*, Nov. 1993, pp. 40–44.

[24] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[25] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.*, vol. 26, no. 3, pp. 301–321, 2009.

[26] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[27] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D, Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, 1992.

[28] T. Wang, A. Hussain, Y. Cao, and S. Gulomjon, "An improved channel estimation technique for IEEE 802.11p standard in vehicular communications," *Sensors*, vol. 19, no. 1, p. 98, 2019.

[29] F. Lyu *et al.*, "MoMAC: Mobility-aware and collision-avoidance MAC for safety applications in VANETs," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10590–10602, Nov. 2018.

[30] F. Lyu *et al.*, "SS-MAC: A novel time slot-sharing MAC for safety messages broadcasting in VANETs," *IEEE Trans. Veh. Technol.*, vol. 67, no. 4, pp. 3586–3597, Apr. 2018.

[31] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Vehicular ad hoc networks: A new challenge for localization-based systems," *Comput. Commun.*, vol. 31, no. 12, pp. 2838–2849, 2008.

[32] R. Schubert, M. Schlingelhof, H. Cramer, and G. Wanielik, "Accurate positioning for vehicular safety applications–The SAFESPOT approach," in *Proc. IEEE 65th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2007, pp. 2506–2510.

[33] D. Li, T. Bansal, Z. Lu, and P. Sinha, "MARVEL: Multiple antenna based relative vehicle localizer," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2012, pp. 245–256.

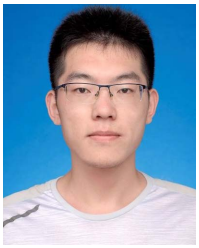
[34] Y. Zhu, Y. Cai, H. Zhu, and S. Chang, "DeepAoA: Online vehicular direction finding based on a deep learning method," in *Proc. IEEE 25th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2019, pp. 782–789.

[35] T. Das *et al.*, "CoReCast: Collision resilient broadcasting in vehicular networks," in *Proc. 16th Annu. Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2018, pp. 217–229.

Yunxiang Cai (Member, IEEE) received the B.S. degree in physics from the Ocean University of China in 2018. He is currently pursuing the master's degree with the Department of Computer Science and Technology, Shanghai Jiao Tong University, under the supervision of Prof. H. Zhu. His research interests include vehicular *ad-hoc* networks and mobile computing.

Hongzi Zhu (Member, IEEE) received the Ph.D. degree in computer science from Shanghai Jiao Tong University in 2009. He was a Post-Doctoral Fellow with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, and the Department of Electrical and Computer Engineering, University of Waterloo, in 2009 and 2010. He is now an Associate Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include vehicular networks, mobile sensing, and computing. He received the Best Paper Award from IEEE GLOBECOM 2016. He is a member of the IEEE Computer Society, the IEEE Communication Society, and the IEEE Vehicular Technology Society. He was a leading Guest Editor of *IEEE Network* magazine. He is an Associate Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. For more information, please visit http://lion.sjtu.edu.cn.

Shan Chang (Member, IEEE) received the Ph.D. degree in computer software and theory from Xian Jiaotong University in 2013. From 2009 to 2010, she was a Visiting Scholar with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. She was also a Visiting Scholar with the BBCR Research Laboratory, University of Waterloo, from 2010 to 2011. She is now an Associate Professor with the Department of Computer Science and Technology, Donghua University, Shanghai. Her research interests include security and privacy in mobile networks and sensor networks.



Xiao Wang (Member, IEEE) received the B.S. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, in 2019, where he is currently pursuing the master's degree with UM-SJTU Joint Institute. His research interests mainly focus on mobile computing and big data analysis.

Jiangang Shen (Member, IEEE) received the B.S. degree from Shanghai Jiao Tong University, Shanghai, in 2019, where he is currently pursuing the master's degree with the Department of Computer Science and Technology. His research interests include the Internet of Things and wireless networks.

Minyi Guo (Fellow, IEEE) received the Ph.D. degree in computer science from the University of Tsukuba, Japan. He is currently a Zhiyuan Chair Professor and the Head of the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His present research interests include parallel/distributed computing, compiler optimizations, embedded systems, pervasive computing, big data, and cloud computing. He is a Fellow of CCF. He is currently on the editorial board of IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON CLOUD COMPUTING, and *Journal of Parallel and Distributed Computing*.