Infrastructure-Less Vehicle Traffic Density Estimation via Distributed Packet Probing in V2V Network

Christina Suyong Shin, JiHo Lee, and HyungJune Lee, *Member, IEEE*

***Abstract*—In this paper, we address the problem of vehicle traffic density estimation without relying on infrastructure cameras or sensors on the road. Previous infrastructure-less approaches still require some prior knowledge on the road infrastructure, e.g., via road topology map. We seek a lightweight estimation method based only on vehicle-to-vehicle (V2V) communication, i.e., without using any prior knowledge. The main objective of this paper is to examine traffic density through simple yet efficient packet probing within a survey time period and obtain a snapshot of the traffic density distribution map. We propose an *on-demand vehicle sampling* algorithm that makes a probing packet at a vehicle (i.e., sampler) keep sampling to explore the local traffic density on a cell basis. If a current sampler does not operate as an efficient carrier, the packet selects another one as the next sampler via *inner-relaying* and *outer-relaying* procedures. To effectively adapt the level of granularity of traffic density depending on the remaining survey time, we present an *adaptive cell sizing* algorithm. Further, we extend the sampling activity to multiple vehicle samplers by making them aggregate their collected information and also negotiate their future areas to explore. Within a designated deadline, multiple samplers collaborate for more accurate and fast traffic density estimation. By doing so by iterations till the given survey deadline, we can gather a complete view of traffic density estimates based on multiple sources where some areas have more detailed information, whereas others do less. Experiments with a real trace-driven simulation demonstrate that our proposed algorithm effectively estimates the distribution of traffic density considering local traffic conditions compared to other counterpart algorithms, with a factor of up to 9.5.**

***Index Terms*—Vehicle traffic density estimation, vehicle-to-vehicle communication, vehicular ad-hoc networks (VANETs).**

I. Introduction

IN THE era of Internet-of-Things (IoT), 10 billions of devices are expected to be connected each other by forming its own distributed network by 2020 [1], [2]. One of the most emerging trends is the advent of autonomous vehicles that requires not only to sense real-time traffic condition on the road, but also to communicate with other vehicles through its direct vehicle-to-vehicle (V2V) communication. Using its local V2V communication links in vehicular ad-hoc networks (VANETs) [3] instead of relying on infrastructure-driven centralized networking (e.g., cellular networks and pre-installed cameras and sensors), sporadic yet sometimes massive essential traffic information can be exchanged more effectively among vehicles within locally constrained areas [4].

The usage of lightweight V2V communication benefits in many aspects: even under infrastructure free or damaged environments, V2V communication enables to provide a viable self-organizing ad-hoc network. Further, location-sensitive information such as urgent accident reports can effectively be gathered by one or a group of passing on-site vehicles. The distributed self-organizing control feature on vehicles would be a necessary part of the future intelligent transport system.

The traffic density is considered as an essential road condition monitoring metric used in transport engineering. For example, this metric can give us key information in preparation for responsive traffic control, scheduling road construction, or emergency response [5]. Therefore, capturing traffic density with high fidelity plays a key role in improving the overall transport system.

The traffic density estimation problem has been actively investigated in various fields from civil engineering to computer science. Traditionally, infrastructure-based mechanisms have been proposed to solve the problem using additionally required devices such as surveillance cameras [6], [7]. Meanwhile, in [8], [9], sensor-equipped vehicles are used to collect various information from their sensors and attempt to estimate the traffic conditions. Since these studies rely on additional hardware devices, they suffer from limited scalability, and high installation and maintenance cost issues.

To address the drawback of using pre-installed hardware devices, several works have been proposed based on the idea of utilizing VANET resources such as on-board units (OBUs) and road side units (RSUs) [10], [11]. The direct V2V communication can be applied to infer local traffic condition in a rather distributed way without requiring additional resource. Researchers in [12] examine local traffic density by analyzing the relationship between the average speed of vehicles and traffic density. To infer the traffic speed, they exploit the patterns of

Manuscript received January 9, 2020; revised June 7, 2020; accepted August 19, 2020. Date of publication August 26, 2020; date of current version October 22, 2020. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) under Grant NRF-2018R1A2B6004006. The review of this article was coordinated by Prof. T. Hara. *(Corresponding author: HyungJune Lee.)*

Christina Suyong Shin is with the Department of Computer Science, University of Southern California, Los Angeles, CA 90007 USA (e-mail: cshin956@usc.edu).

JiHo Lee and HyungJune Lee are with the Department of Computer Science and Engineering, Ewha Womans University, Seoul 03760, South Korea (e-mail: jiholee@ewhain.net; hyungjune.lee@ewha.ac.kr).

Digital Object Identifier 10.1109/TVT.2020.3019783

vehicle mobility and stopping time, which are obtained from V2V communication. [13] suggests a traffic congestion detection algorithm that uses beacons to keep updating the neighbor table through periodic sampling. A fully-distributed traffic density estimation approach using a packet sampler has been proposed in [14], similar to our approach. However, these previous infrastructure-less approaches require some prior knowledge on the road infrastructure, e.g., via road topology map. Further, most of previous VANET-based research works rely on periodic broadcast-based sampling possibly with additional road topology information, causing considerable network congestion and collision.

This paper presents a lightweight traffic density estimation algorithm based on a simple packet probing approach within a permitted deadline. Our proposed algorithm calculates the traffic density through on-demand vehicle sampling and probing packet relaying at a grid cell level in a distributed fashion. To effectively control the sampling intensity and thus diversify the information level depending on traffic condition and remaining time budget, we devise an adaptive cell sizing algorithm.

Our main contributions can be summarized as follows:

- We propose a traffic density estimation algorithm only through a lightweight V2V packet probing at the vehicle side in a distributed fashion, while not requiring some additional infrastructure support during the mission.
- We design a simple yet efficient on-demand vehicle sampling and packet relaying algorithm so that we can infer the entire traffic density distribution map with different resolution by using either a single or multiple probing packets within a designated deadline.
- We suggest a unique cell split algorithm that adaptively changes the cell size and adapts the information level to balance the estimation accuracy and the given deadline.

Our paper is organized as follows: After discussing related work in Section II, we present our system model in Section III. While Section IV presents our on-demand sampling mechanism with adaptive cell sizing, Section V describes the packet relaying procedure between the old and the new vehicle samplers, and Section VI presents a domain distribution scheme among multiple vehicle samplers. We validate our work compared to other counterpart algorithms in Section VII, and then conclude this work in Section VIII.

II. Related Work

The problem of estimating traffic density has been studied, and the related work can be categorized into infrastructure-based and infrastructure-less approaches.

Traditionally, the traffic density estimation problem has been investigated mostly with infrastructure assistance. The infrastructure-based approaches tend to require pre-installed devices to detect vehicles on the road. The video monitoring and surveillance cameras have been used to detect vehicles and count the number of them [3], [6], [7], [15]–[17]. However, since these approaches are based on image processing, their accuracy degrades rapidly when there is insufficient light. Some other methods use loop detectors to check if a vehicle had stayed in one of the installed areas and count the number of vehicles [18], [19].

The aforementioned techniques require the prior hardware installation at certain areas. Therefore, it takes additional cost to install and maintain the facility. In addition, these techniques suffer from some limited coverage issues. To overcome this problem, the approach of using the communication resources in VANET has been proposed recently.

In VANET, many traffic density estimation methods have been proposed using both internal network resource such as OBU and external network resource such as RSU. OBU and RSU collaboratively gather the information about the speed and the lane change of vehicles [10], [20]. [21] divides an area into several small square regions where each region has a dedicated location server. A location server communicates with others to aggregate the vehicles' information over regions. [22] suggests a method to continuously estimate the traffic density by counting the number of beacons received at RSUs and calculating the street to junction ratio based on a road map topology. However, these approaches rely heavily on RSUs for the aggregation and propagation of the traffic information, causing its resulting communication cost and time delay.

Even in an infrastructure-less environment without RSUs, only the direct V2V communication has been used to perform the traffic density estimation. The information of neighboring vehicles can be a key information for estimating the traffic density with a certain accuracy level [12], [23]–[26]. In [27], the authors have proposed a distributed algorithm that allows neighboring vehicles to exchange their speed and estimate the traffic density based on a recording of an unusual speed. It requires the expected speed on a certain area in advance. In [28], vehicles use a device called *TrafficRep*, which is connected to a digital map database. The device reports the travel time from a segment to another when the vehicle reaches the end of the segment, and changes the device based on its V2V communication. [29] proposes a way to fuse the vehicle spacing information with the V2V communication and compute the average spacing based on a maximum likelihood estimator at a data center.

Some traffic-aware routing work suggests a way of finding out the neighboring vehicle information and uses the local neighboring vehicle information to design an effective vehicle ad-hoc routing scheme [30]–[34]. In particular, [31] selects the next-hop vehicle based on the road structure, the predicted location of neighboring vehicles, the up-to-dateness of their mobility information, and inter-vehicle communication quality. In [33], an adaptive neighbor discovery algorithm obtains the local vehicular density based on the neighbors' information as part of finding the best path for data delivery.

Most of traffic density estimation methods still rely on a segment, lane, or junction based map topology information to calculate a global traffic information [35]–[38]. Further, there is a serious error propagation issue such that a small density estimation error in a local area can lead to a large error in a the whole area of interests in case of the dense vehicle density cases [11].

Most similarly to our work, an approach using a packet sampler has been proposed in [14]. The sampler hands over a sampling kit to a new suitable vehicle when the sampler is about to leave the target area. The sampler samples the vehicle density at a random interval and calculates the instantaneous vehicle lane density.

Our work differentiates from the prior infrastructure-less approaches in that our algorithm offers an efficient coarse-to-dense traffic density information with respect to the amount of mission deadline, without using any prior knowledge on the road infrastructure. Our work suggests a clever way to utilize multiple vehicles without controlling their movements, based on a tightly coupled design of on-demand sampling, adaptive cell sizing, and priority-based relaying mechanisms.

III. System Model

We consider the problem of vehicle traffic density estimation under a time deadline constraint without relying on infrastructure cameras or sensors on the road. We seek a lightweight estimation mechanism based only on V2V packet probing and relaying. This time deadline is a user-definable parameter and can be set depending on the available observation time and the granularity level for estimating the vehicle traffic density in the region of interests (RoI). For example, a traffic control officer may want to probe traffic volume with a different resolution at each sub-region with more attention over the RoI within the next one hour, without using any prior knowledge on the road structure, such as in some disaster situations with possible road breakdown or some intermittent Internet outage. After the given time deadline, we want to obtain a map snapshot of traffic density with distinct information level for each different physical area depending on local traffic intensity and time constraint.

We assume that vehicles are equipped with a wireless radio interface such as 802.11p, allowing them to communicate with other vehicles or RSUs. It is also assumed that vehicles are capable of tracking their own current position using the global positioning system (GPS).

We consider the following scenario: a patrol car or an RSU connected to a traffic control center drops a traffic probing packet to one of its nearby passing vehicles. A vehicle that mounts the probing packet keeps sampling local traffic density and being relayed to another vehicle over a region of interests (RoI) until the given deadline is reached. Now that the packet is finally seen and caught by a nearby RSU or any vehicle that is connected to the Internet near the deadline, we obtain local traffic density information from the probing packet that has collected all of vehicles' information (e.g., vehicle ID, location, and its peer V2V vehicle's information) for the last probing period.

By engaging multiple probing packets for which each corresponding vehicle is in charge of traffic sampling over a sub-area, an RoI area can be more effectively explored based on collaborative domain negotiation and distribution for their future sampling. For the sub-areas where some parts of vehicles have inevitably explored with duplicate samples, those samples can be used to estimate a more accurate traffic density for the sub-areas.

We tackle the traffic density estimation problem under a time constraint based on three core parts: 1) on-demand vehicle sampling, 2) probing packet relaying, and 3) adaptive cell sizing (where a cell is the basic unit for counting vehicles, and is allowed to split into multiple sub-cells). The overall procedure of our system is illustrated in Fig. 1.

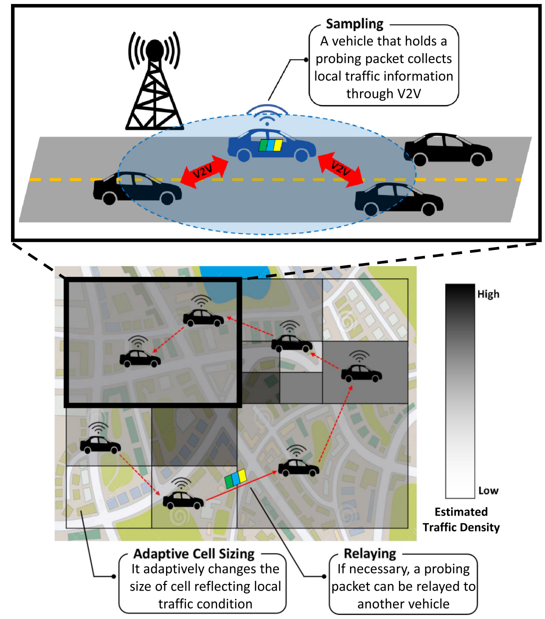

Fig. 1. Overall procedure of our proposed algorithm.

IV. On-Demand Sampling and Adaptive Cell Sizing

In infrastructure-less environments or disaster situations where pre-installed traffic monitoring devices are absent or not available to use, we consider only vehicles and their vehicle-to-vehicle communication as viable means for monitoring traffic conditions in some constrained areas. We exploit a packet as a virtual traffic monitoring agent that can sample its nearby vehicles for estimating a traffic density around its current surrounding physical area from an initiating vehicle or a designated RSU. In this way, we can secure a way of estimating traffic density for a designated area even under the worst conditions on the road.

We aim to capture traffic condition under a time deadline (e.g., within the next tens of minutes or few hours). Depending on how long the permitted traffic probing time is, we want to diversify the granularity of traffic density. For example, if a very short probing time is required to quickly obtain the current traffic status, our system offers a rough snapshot of vehicle distribution in the area. In case that a relatively longer probing time is allowed to be used, a more detailed traffic status with a higher resolution over the same area would be provided.

We define a virtual cell structure over the RoI where a cell is defined as the basic resolution unit, and the RoI is divided into $N \times N$ cells. Utilizing the given cell structure, we estimate the number of vehicles within each cell unit. A probing packet is traveled from one vehicle to another and probes its surrounding area for counting the number of vehicles within its currently visiting cell. Depending on the allowed probing time for a specific cell and the current traffic status, a cell can split into multiple sub-cells for increasing the granularity level of

No.	Sampler	Time	Location	List of neighbor ID
S1	100	13:00:30	(50,148)	120, 130, 140, 150, 160
S2	100	13:05:30	(130,120)	130, 144, 155
S3	100	13:10:30	(47,100)	130, 165
S4	100	13:15:30	(110,20)	155, 165

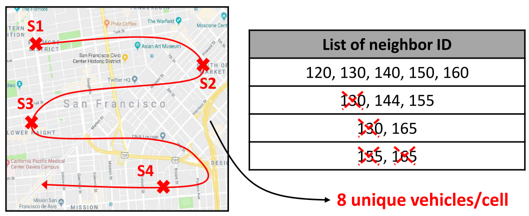

Fig. 2. An example of how a vehicle sampler samples nearby vehicles and records them in its sampling table for traffic density estimation for each cell.

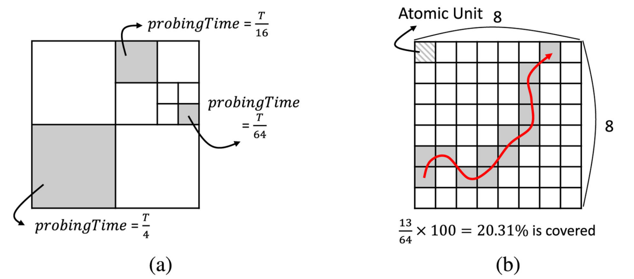

Fig. 3. Time and spatial criteria that decides whether to remain at or to leave the current cell. (a) ProbingTime as time budget to probe on a given cell. (b) CoverageRate used to check whether a cell is fairly covered.

estimation. We explain the necessary procedures of on-demand sampling via packet probing and adaptive cell sizing in the following sub-sections.

A. On-Demand Vehicle Sampling

We present an on-demand vehicle sampling mechanism such that the traffic information can be gathered via a probing packet by controlling the granularity level considering both time and space conditions. Based on the traffic density information of how many vehicles are sampled by a vehicle that currently holds a probing packet, we estimate traffic density for each cell (which is the basic sampling unit) in the RoI.

We technically define a vehicle that currently holds a probing packet as *vehicle sampler* or *sampler*. A sampler is allowed to check out the surrounding traffic conditions by sending or receiving messages. The sampler first broadcasts a `HELLO` message in a periodic manner within a given cell. Any nearby vehicles that have heard the `HELLO` message send back with `REPLY` message that embeds its own vehicle ID and the current (x, y) position (or its corresponding GPS position). Once the sampler receives all the `REPLY` messages from its neighbors at that time, it records a list of communicated vehicle IDs with the sampled time and location. The above messages and the underlying protocols are newly defined. Thus, it would be necessary to make the existing vehicle radio interface (e.g., 802.11p) compatible with our new protocol and framework by defining a special dedicated category in the packet header. As illustrated in Fig. 2, during four sampling chances, 8 unique vehicles are detected and estimated as traffic density for the given cell. The vehicle sampling procedure is described at lines 22–25 in Algorithm 1.

The current vehicle sampler carries a probing packet and conducts the sampling process on behalf of it. When the current vehicle sampler needs to be replaced by another vehicle sampler, all of the internal states for the probing packet are recorded in its packet header, and are transferred to its next vehicle sampler. Since a probing packet is mounted at a vehicle sampler at a time, we use the terms interchangeably throughout the paper.

We substantiate a cell-based probing mechanism by changing the granularity based on two criteria: 1) how long a cell is allowed to be explored from the time perspective, and 2) whether a cell is well explored with a sufficient coverage rate from the perspective of space. We balance the trade-off between time and space for providing a cost-effective traffic density estimation for a given probing time deadline.

Once a time deadline T is determined, a certain amount of time budget, defined as *probingTime* is evenly distributed among all the cells. Since our cell structure starts with $N \times N$ cells, each square cell has a probing time budget of T/N^2. For each different cell, the *probingTime* is calculated proportionally to the cell size, as in Fig. 3(a). That is, the larger a cell is, the longer it can be explored. After spending the given *probingTime* for vehicle sampling at a cell, we consider that cell to be fully probed under the time budget. Then, the probing packet leaves the current cell for a next unexplored cell by being loaded into a suitable vehicle toward the cell.

As for the space exploration criterion, we define *coverageRate* such that if the percentage of the covered pixel-like units (which is the finest fragmentation level) under the sampler's trajectory within a cell exceeds the *coverageRate* threshold, the probing packet should leave the current cell for a next unexplored cell, as illustrated in Fig. 3(b).

Using the above two criteria to determine whether a probing packet should remain, and the current vehicle sampler needs to keep sampling at its current cell, or turn toward to a next unexplored cell, our cell-based probing mechanism provides a practical balance point that satisfies both time and space constraints.

B. Adaptive Cell Sizing

We provide a traffic density estimation mechanism based on on-demand sampling using a probing packet for a given fixed cell structure. Under a given time deadline, we may need to utilize the given time effectively by adapting the information level based on the physical area or the available time. For instance, for rural areas where vehicle traffic is relatively sparse, it may be enough to have even rough snapshot of traffic density for a larger unit area. For urban areas where traffic is denser sometimes with congestion, on the other hand, we may want to acquire more detailed traffic density information for a smaller unit area.

In this section, we propose an adaptive cell sizing algorithm that makes a cell split into multiple sub-cells depending on the traffic activity degree and the allowed probing time budget.

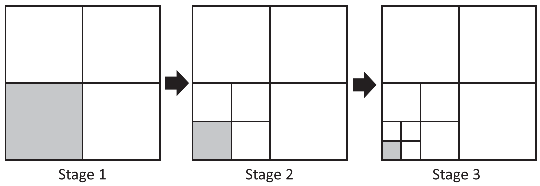

Fig. 4. An example of cell split with the initial $N \times N$ cells with $N = 2$, reaching at the atomic stage m with $m = 3$.

1) Cell Hierarchy: We design a cell hierarchy such that a cell has a square form, and can split into 2×2 identical sub-cells in an iterative way. We allow an initial cell to split up to m iterative stages (i.e., reaching at the *atomic* stage) where the lastly split cell has the atomic size, as illustrated in Fig. 4. A cell can split into multiple cells, independently of its neighboring cell size. With this hierarchical cell structure, we can capture customized traffic density with each different information level depending on the local traffic and probing conditions.

2) Cell Split Algorithm: Our cell split algorithm determines when to split a cell by checking two conditions: 1) local traffic density and 2) allowed probing time budget. Based on these two criteria, we control necessary information level with the given probing time budget in an efficient way.

First, when the current sampler sends a HELLO message (as done in Section IV), it calculates the local density as the number of unique vehicles in the current probing cell within a certain time window, *interestWindow*. If the calculated local density at the cell exceeds a certain threshold, which is defined as $densityThreshold_k$ for the cell at stage k, the first cell split condition is satisfied. The threshold parameter $densityThreshold_k$ should be tuned with a non-increasing value with regard to k, i.e., $densityThreshold_k \geq densityThreshold_{k+1}$.The split check procedure is described at lines 27–34 in Algorithm 1.

Second, in case that any previous cell exploration has ever finished earlier by satisfying the coverage rate condition, in Section IV, we check whether all the remaining time including the saved probing time from the past exploration is enough to accommodate a finer exploration after the current cell's split. The underlying motivation is that it may well utilize some unused probing time at this level for a more detailed exploration after cell split. We calculate the estimated time required for a finer exploration considering the number of unexplored sub-cells and each probing time at its corresponding cell stage. To determine whether some of possible succeeding sub-cells are explored or not, we use the same coverage threshold criterion of *coverageRate*. If the estimated required time for a finer exploration at the next stage does not exceed the current probing time budget that has included all the accumulated unused probing time in so far, the second condition holds valid.

If both conditions are satisfied, a cell at stage k splits into 2×2 sub-cells, reaching at stage $k + 1$ (where $k = 1, \ldots, m - 1$). Using these two conditions, our mechanism provides a cost-effective way of focusing on some selected regions of higher interests through denser and more accurate estimation under a given time budget.

Algorithm 1: Traffic Density Estimation.

```
1:  Input: Time Deadline T
2:  Output: SplitCellStructure, EstimatedDensity
3:  while (Within a time deadline, T) do
4:    Initialize probing time, t, and coverage rate, c;
5:    // Start target cell probing
6:    while ((t ≤ probingTime) || (c ≤ coverageRate)) do
7:      Invoke vehicle-sampling() in every f secs;
8:      if (split-check()) then
9:        Split into sub-cells and start sub-cell probing;
10:     end if
11:     if (Currently out of the target cell) then
12:       Invoke inner-relaying via packet-relaying();
13:     end if
14:     Update probing time, t, and coverage rate, c;
15:   end while
16:   // Done for current target cell probing
17:   Invoke outer-relaying via packet-relaying();
18: end while
19: // Done for probing and return the final traffic density
    estimate for cells
20: SplitCellStructure = final cell structure at time T;
21: EstimatedDensity = the number of unique vehicles at
    each cell on the final cell structure at time T;
22: Function vehicle-sampling()
23:   Broadcast a HELLO message;
24:   Receive REPLY messages from neighboring
      vehicles;
25:   Update the sampling table with neighbors' ID and
      the current position;
26: EndFunction
27: Function split-check()
28:   localDensity = # of unique vehicles in the cell
      within interestWindow;
29:   if ((localDensity ≥ densityThreshold at current
      stage) &&
30:   (required time for the next stage ≤ current probing
      budget)) then
31:     flag = TRUE;
32:   end if
33:   Return flag;
34: EndFunction
35: Function packet-relaying()
36:   Update the neighbor list via vehicle-sampling();
37:   Select the next vehicle among candidates whose
      current location is the nearest to the target cell;
38: EndFunction
```

V. Packet Relaying

Along with the aforementioned traffic sampling and adaptive cell sizing procedure, it is essential to retain a packet relaying procedure such that a probing packet should be relayed from the currently residing vehicle to another in certain situations (as briefly described at lines 35–37 in Algorithm 1). We categorize these situations into two cases: 1) *inner-relaying* during the

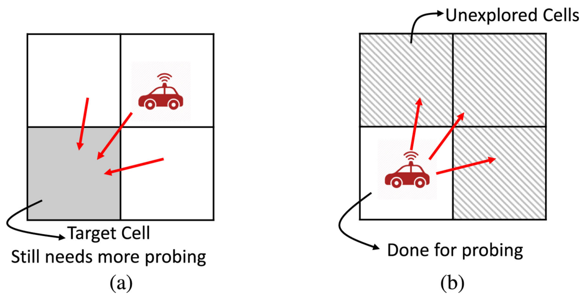

Fig. 5. Two packet relaying cases of *inner-relaying* and *outer-relaying*. (a) Inner-Relaying. (b) Outer-Relaying.

current cell exploration and 2) *outer-relaying* after the completion of the current cell exploration.

A. Inner-Relaying

Once the current sampler becomes in charge of a specific cell, the exploration of the cell should persist until a given probing time is expired, or the exploration progress reaches the coverage rate threshold. In case that the sampler unexpectedly leaves or is about to leave the cell that has not been finished yet, a new vehicle sampler should be selected. A set of vehicles that likely remain at the current cell or move toward it with the inner force can be the candidates as the next sampler. Once a next sampler is selected, the probing packet leaves the current vehicle sampler for the newly selected sampler, via *inner-relaying*, as in Fig. 5(a).

When we need the *inner-relaying*, the current sampler broadcasts a `HELLO` message to find out nearby vehicles. These vehicles are considered as candidates for the next sampler. Among these candidates, the one that is the closest to the center of the target cell is selected as the next sampler. A more advanced prediction-based vehicle selection can be applied within this framework.

B. Outer-Relaying

We describe the *outer-relaying* procedure such that the current sampler that completes the exploration of a cell relays its probing packet to another vehicle. We design this relaying scheme with two cases depending on the number of probing packets.

1) Relaying With Single Probing Packet: Once the current sampler completes the probing procedure at its current cell, the probing packet needs to move toward one of its unexplored cells. In case that the current sampler is still being in the current cell or move toward one of already-explored cells, the probing packet should leave the current sampler for a next sampler that would likely stay in an unexplored cell, via *outer-relaying*, as in Fig. 5(b).

In the *outer-relaying* case, it is necessary to determine the next target cell and the next sampler together. We first select the next target cell. The current sampler broadcasts a `HELLO` message and its replying neighboring vehicles' information is utilized. Among all the unexplored cells based on the collected vehicle and its residing cell information, a cell in which the largest number of replying vehicles are located becomes the next target cell. The underlying reason is that the more neighbors in a cell, the higher possibility the cell is well explored. If there are multiple cells with the same largest number of vehicles, the one that is the closest from the current sampler is selected. Once the next target cell is determined, we select one vehicle that is the closest to the center of the selected next target cell, among vehicles that have responded within the next target cell.

2) Priority-Based Relaying With Multiple Probing Packets: Although one probing packet can cover the whole RoI for traffic density estimation within a given deadline, we extend the packet probing to multiple packets. We aim to execute more accurate and fine traffic estimation during a relatively longer probing time at each probing packet via cooperative probing by multiple packets within the total deadline.

Since multiple probing packets explore the RoI area in a distributed manner, an individual probing packet is not aware of others unless they become adjacent within the communication range. If two probing packets, i.e., two vehicle samplers come across at some point, they perform a domain negotiation and decision procedure. To determine their future domains to explore, they exchange the explored cell, unexplored cell, and investigating cell lists.

Once the cell information is shared between two adjacent probing packets, their respective vehicle samplers run a cell prioritizing algorithm. Based on the recently updated cell information, each vehicle calculates a priority for each individual cell under the current cell structure. Using the cell priority information, it selects a target cell to examine after fully probing the current cell.

Regarding the *Inner-Relaying* for multiple probing packets, the same relaying procedure for the single probing packet case is executed. For the *Outer-Relaying* procedure with multiple probing packets, each vehicle sampler selects a next target cell based on the calculated priority.

We consider three metrics of cell state, distance, and cell size to prioritize the cells for choosing the most urgent cell to explore among them as follows.

1) Cell State: We define four cell states with '*Unexplored*,' '*Expected*,' '*Considered*,' and '*Explored*' depending on whether a cell is explored or not, and its probing priority level. The *Unexplored* cell is the cell that has not been explored yet by any probing packets. The *Expected* cell is the cell that turns out to be scheduled for probing by another probing packet after communication among probing packets. The *Considered* cell is the cell with different cell structure that two probing packets retain. When the probing progress is shared between two packets, a cell may fully be probed by one probing packet, whereas it has partially been probed by another with a different cell size. The *Explored* cell is the cell that has fully been probed by a probing packet. Since an *Unexplored* cell needs to be covered by a vehicle sampler with the highest priority, its corresponding weight, $w_{unexplored}$ attains the highest value. On the other hand, already-explored cells need to have the least priority for sampling attention with the lowest weight value, $w_{explored}$. Considering the cell definition and its probing priority, its corresponding weight value w_{state}

is assigned as follows: $0 \leq w_{explored} \leq w_{considered} \leq w_{expected} \leq w_{unexplored} \leq 1$. The suitable weight values can be tuned in experiments.

2) Distance: If a candidate cell is located relatively near the probing packet itself, it would likely move to the cell more easily compared to other candidate cells. We consider a relative distance measure d_{ratio} to be the distance from the probing packet to a candidate cell, which is normalized by the diagonal length of the entire RoI. Then, we define $w_{distance}$ as $1 - d_{ratio}$ so that the distance-related priority weight value can be closer to 1 for a candidate cell closer to the current probing packet.

3) Cell Size: We consider the cell size to calculate the net weight value for a candidate cell to probe. If a candidate cell has a larger size to be probed by a probing packet, its priority should also be proportional to its size. We define a relative cell size as the normalized cell size by the maximum cell size in the cell structure and use the measure as its corresponding weight value w_{size}.

We finally calculate the net priority level for each candidate cell, p_{cell} that reflects all of the three weight values from cell state, distance, and cell size as follows:

$$p_{cell} = (\alpha \cdot w_{state} + \beta \cdot w_{distance}) \cdot w_{size} \tag{1}$$

where α and β are weighting factors whose summation should be 1.

The larger p_{cell} value a candidate cell has, the higher priority it is scheduled to be probed by a probing packet at its vehicle sampler. The current vehicle sampler selects a cell with the largest p_{cell} value as its next target cell, and moves toward the cell.

The above cell prioritizing algorithm runs in two conditions: 1) after a probing packet has fully probed a cell and 2) when two vehicle samplers happen to communicate with their communication range and try to negotiate their future cells to probe.

According to the cell prioritizing algorithm, a probing packet focuses on cells with the higher priority in terms of cell state, distance, and cell size. Thanks to this stage, multiple probing packets collaborate to probe over a large RoI area in an effective manner, while even some inevitably overlapped cell information can also turn into a more detailed traffic density estimation with a higher fidelity.

As an exception handling case, if a vehicle turns out to stay at the same location for a certain *stagnation* period, our mechanism lets the probing packet be relayed to another vehicle according to one between *inner-relaying* and *outer-relaying*. If there does not exist any available next vehicle to relay, the current sampler keeps holding the probing packet and attempts to relay to another vehicle at the next time.

VI. Advanced Multiple Packet Probing

Multiple vehicle samplers that hold their own probing packets can participate in the traffic density estimation in a collaborative manner. For some overlapped regions, we can obtain a more accurate density estimate via information aggregation by multiple samplers. For non-overlapped regions, multiple samplers can be

Algorithm 2: Advanced Traffic Density Estimation.

Input: Time Deadline *T*, # of Probing Packets *N*
Output: *SplitCellStructure, EstimatedDensity*
while (Within a time deadline, T) **do**
  Initialize the probing time t, and the coverage rate c;
  // Start target cell probing
  for ($n = 1$ to N) **do**
    Invoke vehicle-sampling() in every f secs;
    if (split-check()) **then**
      Split into sub-cells and start sub-cell probing;
    end if
    if (negotiation-check()) **then**
      Integrate two cell structures and cell states;
      Distribute the candidate cells into *cell window* per packet;
      Update each packet's cell state;
    end if
    if (($t \leq$ *probingTime*) $\|$ ($c \leq$ *coverageRate*)) **then**
      if (Currently out of the target cell) **then**
        Invoke inner-relaying via packet-relaying();
      end if
    else if ($t >$ *probingTime*) $\|$ ($c >$ *coverageRate*) **then**
      // Done for current target cell probing.
      Update the cell state;
      select-target();
      flag-outer = TRUE;
    end if
    if (*flag-outer*) **then**
      Invoke outer-relaying via packet-relaying();
    end if
    Update the probing time t, and the coverage rate c;
  end for
end while
Aggregate N packets' cell structures to one final cell structure
// Done for probing and return the final traffic density estimate for cells
SplitCellStructure = final cell structure at time T;
EstimatedDensity = the number of unique vehicles at each cell on the final cell structure at time T;
Function vehicle-sampling()
  Broadcast a `HELLO` message;
  Receive `REPLY` messages from neighboring vehicles;
  Update the sampling table with neighbor ID & the current location;
EndFunction
Function split-check()
  localDensity = # of unique vehicles in the cell within *interestWindow*;
  if ((*localDensity* $\geq$ *densityThreshold* at current stage) &&
  (required time for the next stage $\leq$ current probing budget)) **then**
    flag = TRUE;

```
46:     end if
47:     Return flag;
48:  EndFunction
49:  Function negotiation-check()
50:     if ((Two packets meet in the range of
        communication) && (Both packets are not in the
        middle of negotiation)) then
51:         flag = TRUE;
52:     end if
53:     Return flag;
54:  EndFunction
55:  Function packet-relaying()
56:     Update the neighbor list via vehicle-sampling();
57:     Select the next vehicle among candidates whose
        current priority is the highest;
58:  EndFunction
```

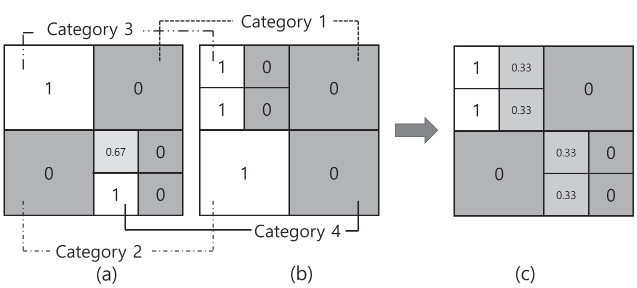

Fig. 6. An example of priority-based relaying after probing a cell. (a) Cell state. (b) Priority-based Relaying.

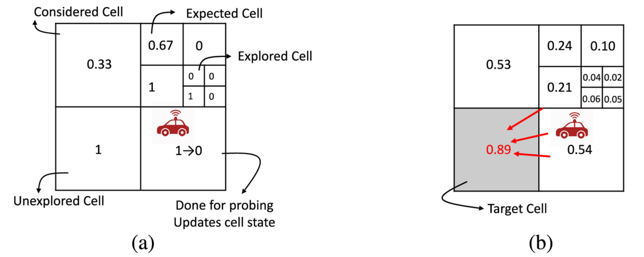

Fig. 7. Illustrative example of domain aggregation between two probing packets with the cell state of 1 for unexplored cells, 0.67 for expected cells, 0.33 for considered cells, and 0 for explored cells. (a) Packet A. (b) Packet B. (c) Aggregated Cell.

dispatched to each separate cell, achieving a faster exploration over the total remaining area to visit. By having a balance between the overlapped cells and the non-overlapped cells, we aim to capture more accurate and faster traffic density estimation by utilizing multiple vehicle samplers.

We present a domain aggregation and distribution algorithm. Once multiple vehicle samplers become adjacent within the radio range and exchange their collected information, they reflect the newly acquired information at their own cell structure. After this procedure, they negotiate and determine where to explore in the near future under a given remaining deadline (as described at lines 11–14 in Algorithm 2).

A. Domain Aggregation

When two vehicle samplers are within the radio range, they exchange and merge the collected traffic information of the already-explored cells and their own cell structure. In case that their cell structures have all the same cell split hierarchy in terms of cell stage, each cell state is updated by reflecting the latest exploration. If two probing packets retain each different cell structures with different stage levels for certain cells, they integrate them by adjusting to the finer cell stage.

As described in Section V, we assign a priority level to every candidate cell. We classify the domain aggregation cases into four categories as illustrated in Fig. 7. In category 1 where the cell structure and its governing cell's state is the same for two probing packets, the cell state keeps remained. In category 2 where the cell state is different for a cell at each probing packet with the same cell structure, the cell state is updated with the lower cell state weight value between them. In category 3 where the cell state is the same at two probing packets even with each different cell stage on a cell, the cell state keeps remained. In category 4 where the cell state of one probing packet is different from that of the other probing packet under each different cell structure, all of its corresponding cells become *considered* cells.

The cell distance weight value $w_{distance}$ is recalculated with the average location of all communicating probing packets. The cell size weight value w_{size} is updated with the smaller one from the values from probing packets (i.e., $\min_k w_{size_{(k)}}$ where $w_{size_{(k)}}$ is the cell size value by the kth probing packet). Based on the aggregated cell information from each probing packet, its vehicle sampler recalculates the net priority level p_{cell} to update the latest cell information after domain aggregation.

B. Domain Distribution

After aggregating the cell information from other probing packets via their vehicle samplers, each *vehicle sampler* determines its future cell probing plan. While increasing the accuracy of the traffic density estimation by utilizing some necessary coverage redundancy, we need to moderately control the redundancy rate to avoid excessive coverage overlap and spread them over the RoI for efficiency.

To effectively control the cell coverage redundancy, we introduce a concept of *cell window*, which is the future visiting cell list with $N_{cell-window}$ cells for a probing packet. If two probing packets need to distribute their future cells to visit, they pre-select $(2N_{cell-window})$ candidate cells in the descending order of p_{cell}. After making all possible combinations of two group pairs with $\binom{2N_{cell-window}}{N_{cell-window}}$ cases where one group consists of $N_{cell-window}$ cells, we select one pair case that shows the minimum difference in the cumulative net priority level within a group, i.e., $\min |\sum_{i \in G_a} p_{cell_i} - \sum_{i \in G_b} p_{cell_i}|$ for groups G_a and G_b.

Once two group lists G_a and G_b are determined, each list needs to be assigned to each probing packet. We select one cell with the highest net priority level from two lists. If the net priority level of one cell from one group list is the same as that of the other

TABLE I
COMPARISON BETWEEN SINGLE AND ADVANCED MULTIPLE PACKET PROBING

Property \ Type	Single Packet Probing	Advanced Multiple Packet Probing
Sampling Process	Sequential	Parallel
Information Source	Single probing packet	Multiple probing packets with information exchange and merge
Cell Priority	Distance	Cell state, Distance, Cell size
Domain Distribution	X	O
Revisiting Frequency	Low	Relatively high
Final Density Estimation	Single result from single packet	Average result from multiple packets

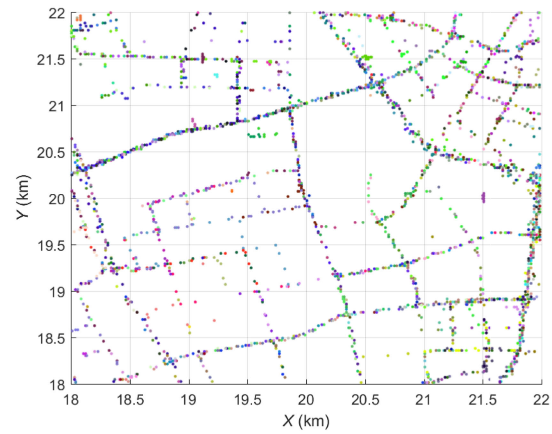

Fig. 8. Geographical distribution of all the vehicles from 12am to 1am, Jan. 31, 2007 within 4×4 km^2 RoI in *SUVnet-Trace* Shanghai Taxi Trace.

cell from the other group list, we randomly assign each group to a probing packet. We compare the coverage rate on the cell by two packets, and the packet with a higher coverage rate becomes finally in charge of the group list that includes the highest priority cell. In case that both packets have the same coverage rate, we randomly assign each group to a probing packet.

After the negotiation procedure, a cell that is determined to be probed by another packet is updated to an *Expected* cell. By reflecting some potential opportunity to revisit the *Expected* cell in the near future, multiple probing packets can prioritize more urgent cells to probe, while allowing some level of coverage overlap for maintaining a high estimation fidelity.

We compare two probing methods of *Single Packet Probing* and *Advanced Multiple Packet Probing* as summarized in Table I.

VII. EVALUATION

We validate our proposed algorithm in a real trace-driven network simulation based on MATLAB. We use real-world taxi trace *SUVnet-Trace* dataset collected in Shanghai, China [39]. In the dataset, 4,000 taxis equipped with GPS are traced over Shanghai, China from Jan. 31 2007 to Mar. 1 2007, and has the following five features: taxi ID, timestamp, latitude, longitude, and geographic angle. We focus on 10-day taxi trace (from Jan. 31 2007 to Feb. 9 2007) over 4×4 km^2. as in Fig. 8. In our experiments, we run total 240 cases on the hour for every 24 hour over 10 days and take the average performance while showing the standard deviation wherever applicable. The RoI of 4×4 km^2 is initially divided into 2×2 cells, and a cell is allowed to split up to stage 3 where $N = 2, m = 3$. The sampling period of 30 seconds, the *coverageRate* threshold of 70% and the *stagnation* window of 120 sec are used. For the cell split condition, the time window of *interestWindow* is set to 120 sec, and the values of *densityThreshold*$_k$ is tuned to be 10, 5, and 2 for stage k.

For the advanced multiple packet probing case, the parameter values of $w_{explored} = 0, w_{considered} = 0.33, w_{expected} = 0.67$, and $w_{unexplored} = 1, \alpha = 0.4$, and $N_{cell-window} = 4$ have been tuned to be used. The cell state was set proportionally. α and $N_{cell-window}$ that were robust in our algorithm were selected through the experiments. Three probing packets have been used in experiments, unless otherwise noted.

We vary the time deadline T from 30 minutes to 2 hours (for most cases, one hour is used by default unless otherwise noted). For a given time deadline, our algorithm offers the traffic density estimations for each resulting split cell at the end. To fairly compare our algorithm with its ground-truth, the ground-truth is calculated from the raw data to have the average number of actual vehicles during the sampling period for each cell size over T in the same cell hierarchy as ours.

To quantify how our traffic density estimate is far from the ground-truth, we adopt a distance metric of *Earth Mover's Distance* (EMD) [40]. The EMD metric is a distance metric between two distributions, as the incurred cost for transforming one distribution to the other. We calculate the EMD between our estimated distribution of traffic density and its original ground-truth, implying that the lower EMD value is, the more accurate performance is achieved.

A. Parameter Selection

We discuss how design parameters affect our algorithms in Fig. 9. Among those, we choose some major design parameters that can cause performance change more dynamically: $N_{cell-window}$, α, *densityThreshold*$_k$, and *interestWindow*. We consider some major parameters to examine their effect on estimation accuracy and packet overhead. As shown in Fig. 9(a), both the highest accuracy and the lowest packet overhead were achieved at $N_{cell-window} = 4$. This implies that making a series of the future cell visit decisions on up to 4 cells offers a balanced point between coverage overlap and broad spread over the RoI to capture both accuracy and efficiency.

In *Advanced Multiple Packet Probing*, we introduced α that controls the relative weight between the cell state, and the distance between the packet and a candidate cell to visit in Section V as in Fig. 9(b). As varying α from 0 to 1 we explore the effect of the correlation between *weight_state* and *weight_d*. As α increases from 0 to 0.4, the estimation accuracy is significantly improved with a factor of 3.14. Beyond the point at $\alpha = 0.4$, the

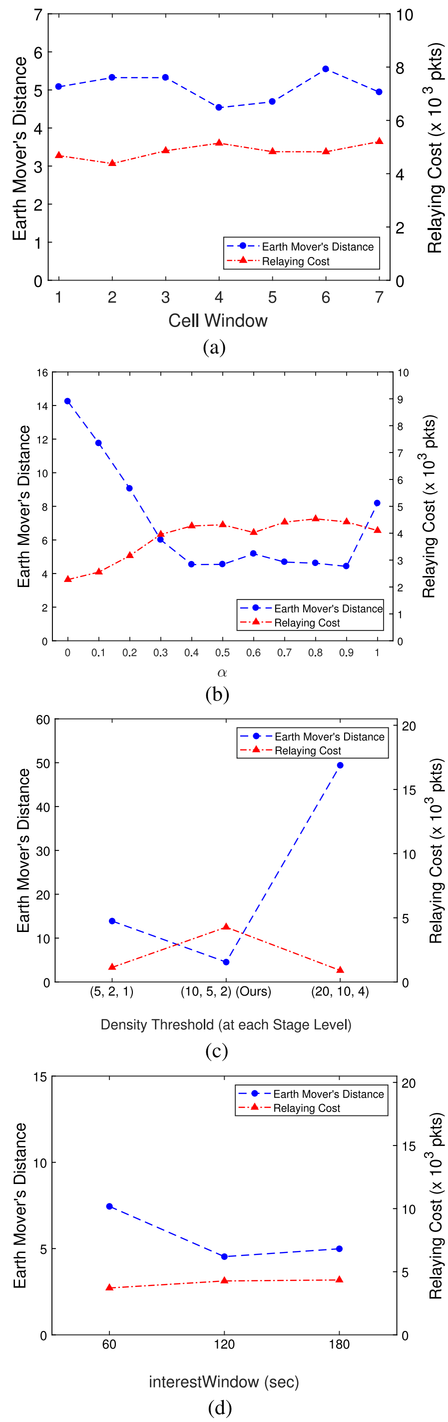

Fig. 9. Effect of $N_{cell-window}$, α, *densityThreshold*$_k$, and *interestWindow* on traffic density estimation and relaying cost performance. (a) Effect of *Ncell-window*. (b) Effect of α parameter in Advanced Multiple Packet Probing. (c) Effect of *densityThresholdk* parameters in Advanced Multiple Packet Probing. (d) Effect of *interestWindow* parameters in Advanced Multiple Packet Probing.

accuracy is saturated until $\alpha = 0.9$. $\alpha = 0.4$ is selected for its highest accuracy in the experiments. This means that a well mixture of cell state and distance plays an important role in achieving a reliable traffic density estimate.

We examine how the selection of *densityThreshold*$_k$ parameters at each stage level k affects performance as shown in Fig. 9(c). If the density threshold is too low, e.g., (5, 2, 1), it makes a cell split relatively easy even under some sporadic traffic condition, and thus the estimation accuracy is relatively low. In case of the high density threshold criteria used in the experiments, e.g., (20, 10, 4), it is difficult to have the cells split, critically degrading the estimation accuracy. The usage of (10, 5, 2) as the density thresholds across stages turns out to be effective in the experimental environment. This result implies that a suitable cell split condition is a key factor to optimize traffic estimation performance.

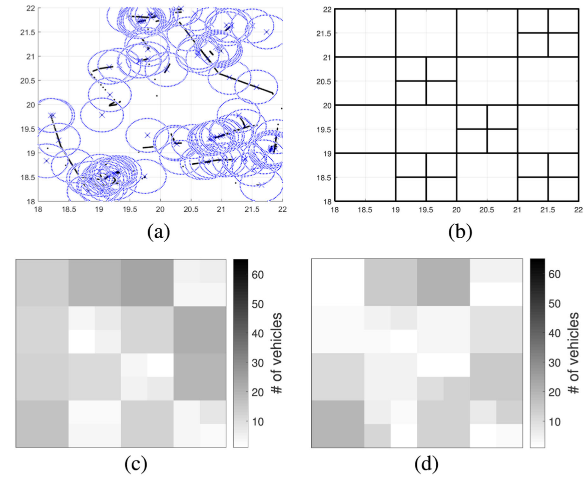

Fig. 10. *Best* case performance of single probing packet(with the EMD of 3.1 at 7pm, Jan. 31, 2007). (a) Probing packet trajectory. (b) Final split cell structure.(c) Ground-truth. (d) Our algorithm.

We look into the effect of *interestWindow* as in Fig. 9(d). The *interestWindow* of 120 seconds turned out to be effective. If *interestWindow* is too short, the cell split would rarely occur, and this degrades the estimation accuracy. On the other hand, if *interestWindow* is too long, the cell split occurs in a more relaxed condition, e.g., even under a sporadic traffic condition.

B. Overall Performance

We explore the traffic density estimation performance of our algorithm. We show three representative *Best*, *Worst*, and *Average* cases, which are selected from all running test cases, in terms of EMD with its ground-truth. We visualize each experiment result with physical trajectory of a probing packet up to the given deadline, final split cell structure, and height map of vehicle density for the ground-truth versus our algorithm's estimate.

1) Single Packet Probing: For the *Best* case as in Fig. 10, the probing packet evenly covers the RoI area by finding a suitable vehicle at a time (as shown in Fig. 10(a)) while some cells split into the atomic cell, reaching at stage 3 as in Fig. 10(b). As the EMD is the smallest among all the test cases, the visual comparison between two height maps of ground-truth (Fig. 10(c)) and our algorithm (Fig. 10(d)) also shows small discrepancy.

For the *Worst* case as in Fig. 11, a probing packet does not well cover the RoI area due to both insufficient vehicle samplers and insufficient vehicles under monitor at the time of selecting a next sampler and probing nearby vehicles to cover unexplored cells

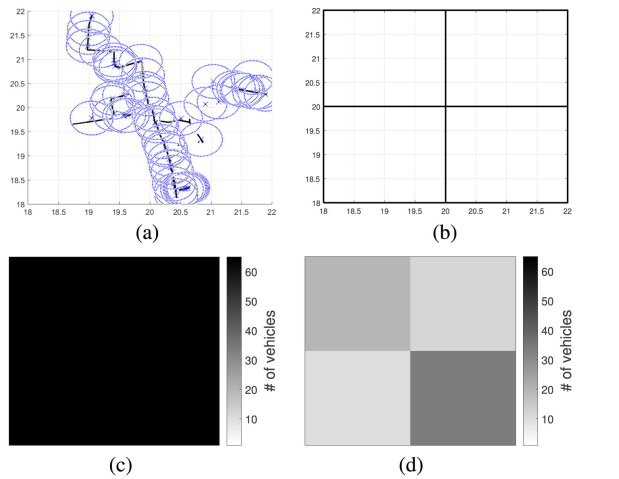

Fig. 11. *Worst* case performance of single probing packet(with the EMD of 72.5 at 10am, Feb. 4, 2007). (a) Probing packet trajectory. (b) Final split cell structure.(c) Ground-truth. (d) Our algorithm.

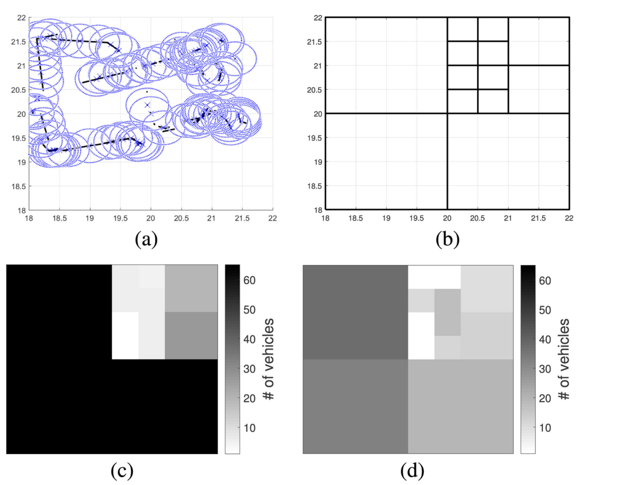

Fig. 12. *Average* case performance of single probing packet(with the EMD of 19.8 at 1pm, Feb. 2, 2007). (a) Probing packet trajectory. (b) Final split cell structure.(c) Ground-truth. (d) Our algorithm.

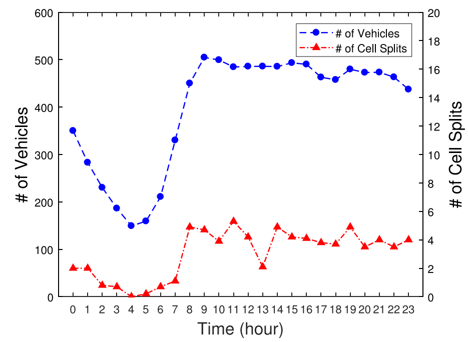

Fig. 13. Empirical relationship between traffic density and cell split frequency over time.

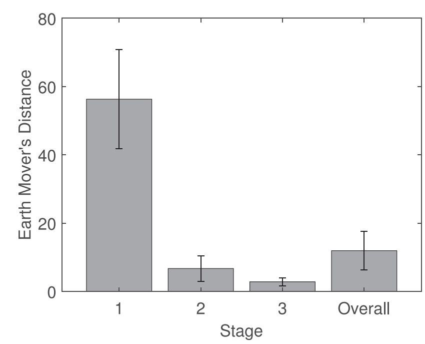

Fig. 14. Traffic density estimation performance with respect to stage progress.

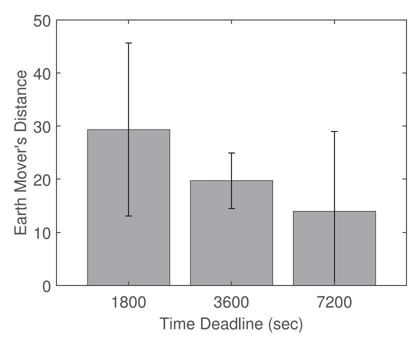

Fig. 15. Traffic density estimation performance with respect to time deadline T.

and relay to, as shown in Fig. 11(a). For this reason, the initial cells are not split at all as in Fig. 11(b), showing only the roughest estimate based on the initial cell size. The visual comparison between Fig. 11(c) and Fig. 11(d) indicates that our algorithm underestimates the traffic density due to the aforementioned reason.

The *Average* case as in Fig. 12 shows less coverage of the probing packet in Fig. 12(a) while a less number of cell split occurs as in Fig. 12(b), compared to the *Best* case (i.e., 9 cell splits for *Best* vs. 3 cell splits for *Average*). This implies that the more cell split occurs due to sufficient coverage by a denser traffic condition, the more accurate traffic estimation is achieved. This observation is supported by an experiment that measures both the number of vehicles and the number of cell splits over time, as in Fig. 13. Both measures are highly correlated each other, and thus, our algorithm greatly utilizes a dense traffic situation by inducing more frequent cell splits for finer and more accurate traffic density estimation.

We validate that our algorithm offers a more accurate traffic density estimation for a smaller cell at a further stage progress in Fig. 14. We categorize all the cell cases depending on cell stage at the end of each experiment. As a cell splits toward a later stage, our algorithm provides a more accurate estimate over a smaller area compared to its ground-truth.

We vary the time deadline T from 30 minutes up to 2 hours as in Fig. 15. As the time deadline gets relaxed, a probing packet can be exposed to the same RoI area longer, increasing the chance to encounter nearby vehicles. Further, the longer time deadline allows a cell to split into sub-cells with higher probability, improving the accuracy of traffic density estimation.

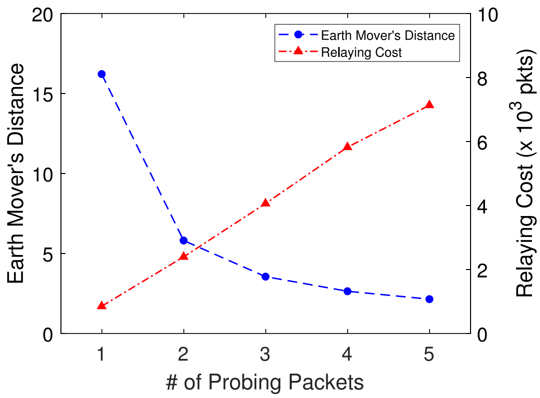

Fig. 16. Traffic density estimation performance with respect to the number of probing packets.

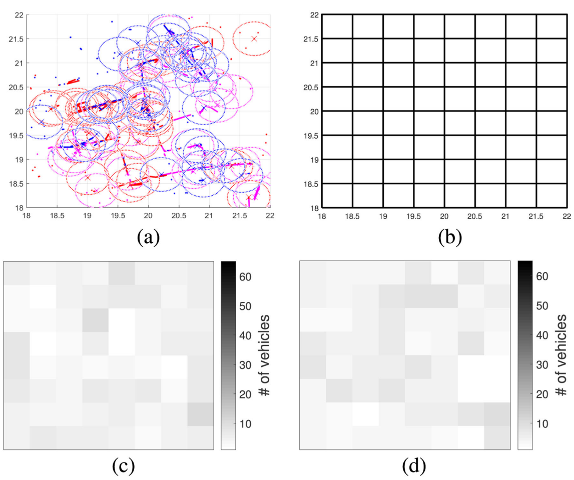

Fig. 17. *Best* case performance of advanced multiple probing packet(with the EMD of 1.4 at 9am, Feb. 1, 2007). (a) Probing packet trajectory. (b) Final split cell structure.(c) Ground-truth. (d) Our algorithm.

2) *Advanced Multiple Packet Probing:* If multiple packets are used to probe over the same area, we evaluate how the additional packet resource affects the accuracy and the overhead of traffic density estimation, as in Fig. 16. As a larger number of probing packets become engaged, the EMD metric starts being reduced significantly, particularly from one packet to two packets, while in return, its relaying cost linearly increases. As a trade-off between accuracy and overhead, we believe that choosing three probing packets provide a well-balanced environment setting where its EMD is 3.56.

As conducted in Figs. 10, 11, and 12 for the single packet probing case, we show three representative visualizations among all running cases. As illustrated in Figs. 17 and 19 for the advanced multiple packet probing, The *Best* and *Average* cases with multiple probing packets split the cell structure in a relatively finer level, leading to more accurate traffic estimation results. The *Best* case as in Fig. 17 shows the fully split cell structure with the best performance, whereas the *Worst* case as in Fig. 18 has the initial intact cell structure without split. This demonstrates that also for the advanced multiple packet probing, as a cell structure for probing evolves into a finer level, its resulting traffic density estimation performance becomes more improved.

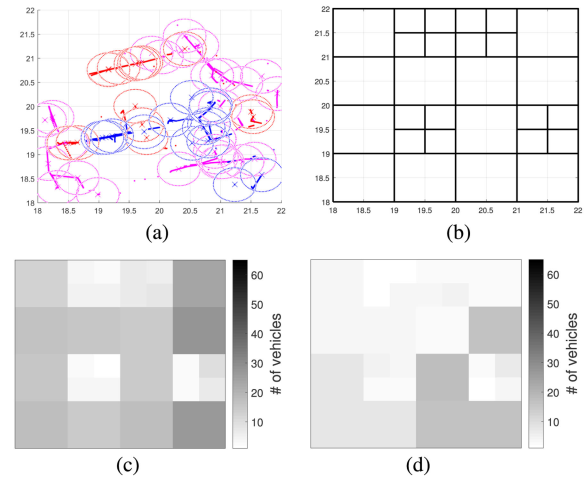

Fig. 18. *Worst* case performance of advanced multiple probing packet (with the EMD of 31.8 at 1am, Jan. 31, 2007). (a) Probing packet trajectory. (b) Final split cell structure. (c) Ground-truth. (d) Our algorithm.

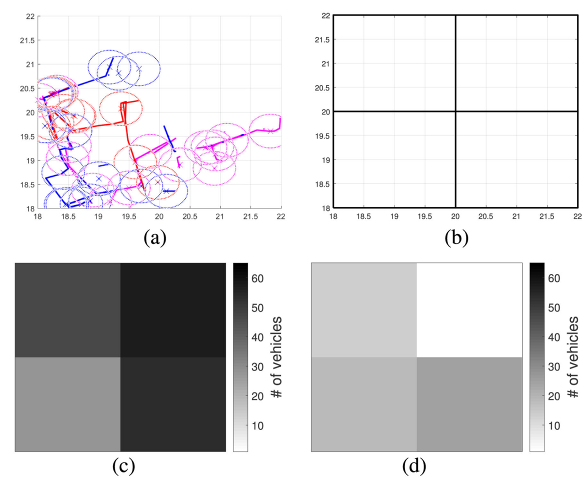

Fig. 19. *Average* case performance of advanced multiple probing packet (with the EMD of 4.8 at 5pm, Feb. 8, 2007). (a) Probing packet trajectory. (b) Final split cell structure.(c) Ground-truth. (d) Our algorithm.

Now we investigate the effect of vehicle interaction, splitted cell stage level, and time deadline on the performance improvement under multiple packet probing scenarios. We collect all of the corresponding cases from the simulated dataset depending on each criterion. As in Fig. 20, as the number of negotiations among encountered probing vehicles increases, the traffic estimation accuracy significantly gets improved with a factor of 2.4 and higher in EMD with only 3 and more negotiations compared to zero negotiation. In return, due to the increased vehicle interactions, the relaying cost has slightly increased. It is interesting to see that for the zero negotiation case where three probing packets have never met till deadline, the accuracy has still been improved compared to the single packet probing case due to the usage of more probing resources under the same

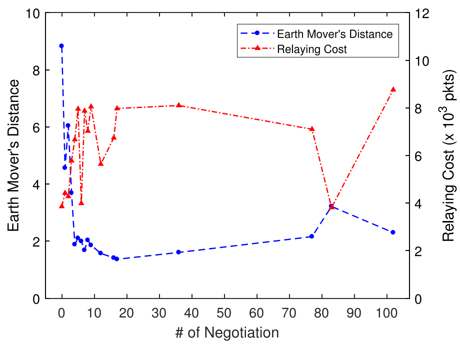

Fig. 20. Traffic density estimation performance with respect to the number of negotiations in case of multiple packet probing cases.

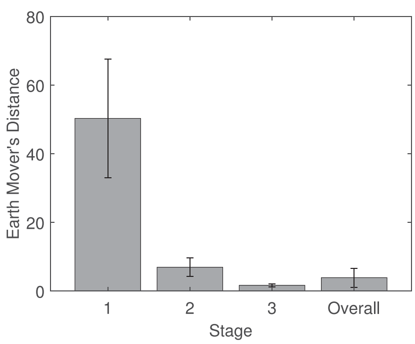

Fig. 21. Traffic density estimation performance with respect to stage progress in case of multiple packet probing cases.

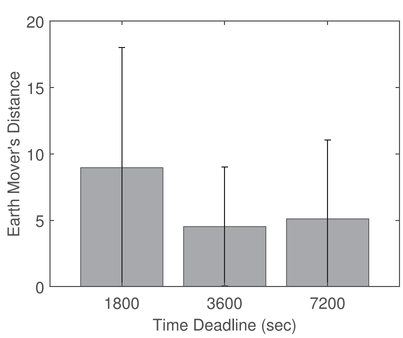

Fig. 22. Traffic density estimation performance with respect to time deadline T in case of multiple packet probing cases.

fixed deadline. This implies that not only more physical probing resource, but also its synergetic interaction for collaborative probing planning have indeed helped to improve the estimation accuracy.

We reconfirm that the traffic estimation accuracy gets improved as cells split at more fine cell stages, and the time deadline gets relaxed, as in Figs. 21 and 22, as similarly validated for the single probing case in Figs. 14 and 15.

We investigate whether our priority-based relaying mechanism has indeed contributed to the performance on traffic density estimation and relaying cost as in Fig. 23. For a fair comparison, we compare *Single Packet* and *Advanced Multiple Packet Probing* with one packet so that the latter one works without

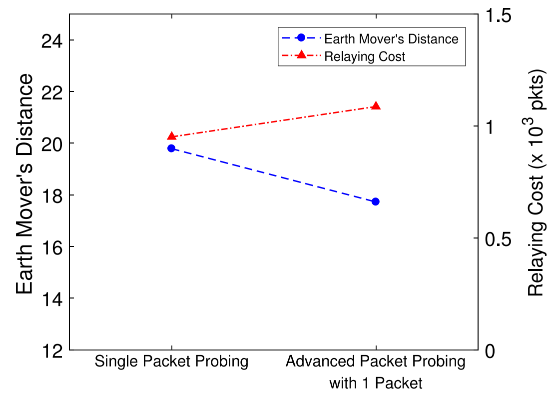

Fig. 23. Effect of priority-based relaying.

the domain distribution. Our priority-based relaying improves the accuracy with 10.4%, while consuming more relaying cost similarly with more than 14.2% in return. This means that a next cell selection should be made based on the distance from the current cell to the next one, and cell state and size altogether.

We validate how our traffic density estimation algorithms of *Single Packet* and *Advanced 3-Packet* perform as the operation time passes by under two exemplary cases of EMD = 10 and EMD = 20 under one hour deadline. Within one hour, *Single Packet* reaches the EMD of 10 and 20 with 35.2% and 60.4% case fraction, whereas *Advanced 3-Packet* does with 95.2% and 95.6%. This implies that *Advanced Packet* fully utilizes more physical resource and its organized usage via negotiation, reaching a traffic density level more quickly.

We compare our *Single Packet Probing* and *Advanced Multiple Packet Probing (Advanced N-Packet)* algorithms against three different baseline algorithms. The first baseline algorithm is *Naive* where a next vehicle sampler is randomly chosen while the same cell split algorithm as ours is used. The second algorithm is *N-Packet* where N number of packets probe the area with neither domain distribution nor priority-based relaying. The third algorithm is *Broadcast*, a broadcast-based traffic estimation method that keeps increasing the number of probing packets via broadcast on the smallest atomic cell structure without using any permitted deadline margin. We quantify the EMD and the transmission cost for packet relaying for these five algorithms as in Fig. 25.

For the comparison between *Naive* and *Single Packet Probing* algorithm, both algorithms use only one probing packet with the same cell split algorithm whereas only the packet relaying algorithm is different. When relying on very lightweight communication resource, i.e., one probing packet, the effective sampler selection via *inner-relaying* and *outer-relaying* plays a very critical role in accomplishing the final traffic density estimation.

Regarding the comparison between *N-Packet* and *Advanced N-Packet*, both algorithms use three probing packets, while our *Advanced N-Packet* exploits a domain distribution and priority-based relaying algorithm. For the multiple packet probing, the usage of both components has indeed improved the accuracy with a factor of 1.51, while consuming its resulting increased

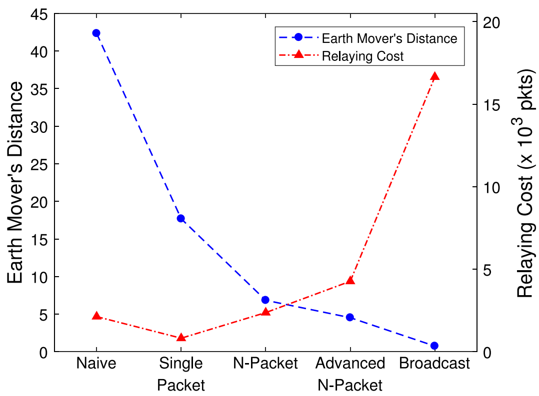

Fig. 24. Cumulative distribution of elapsed time of *Single Packet* and *Advanced 3-Packet* under one hour deadline.

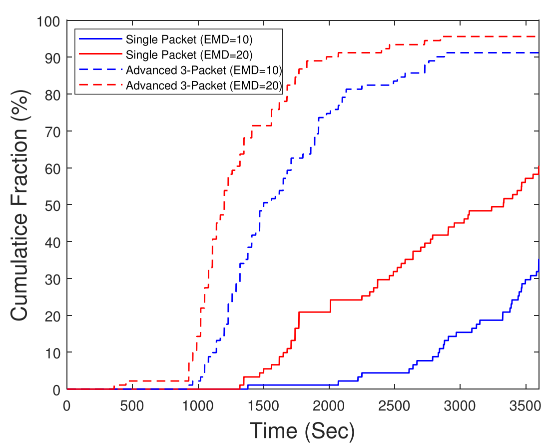

Fig. 25. Overall performance comparison against three baseline algorithms where our *Advanced N-Packet* outperforms the baseline algorithms with a factor of up to 9.3 in accuracy.

relaying cost with a factor of 1.80. This demonstrates that collaborative packet probing has achieved more accurate traffic density estimation with a reasonable increase in relaying cost.

Upon comparing with *Broadcast*, *Broadcast* provides the least discrepancy on the smallest cell structure for traffic density estimation. However, it spreads a huge amount of probing packets to numerous vehicles in the RoI, consuming tremendously higher relaying cost with a factor of 19.4. Our algorithm provides a practically effective solution with good approximates for traffic density using only one simple lightweight probing packet with selective packet relaying to selected vehicles that we cover evenly throughout the RoI.

This result implies that three components of 1) careful vehicle sampler selection (from *Naive* to *Single Packet*), 2) additional physical probing resource (from *Single Packet* to *N-Packet*), and 3) a tightly-coupled domain distribution and priority-based relaying strategy (from *N-Packet* to *Advanced N-Packet*) have achieved contributed to improving the traffic density estimation accuracy with a factor of up to 9.3 (from *Naive* to *Advanced N-Packet*) in an infrastructure-less VANET environment.

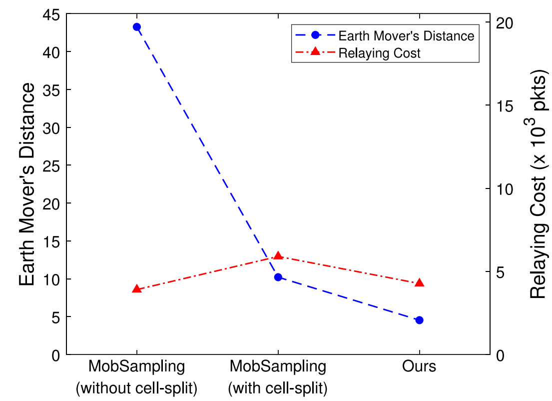

Fig. 26. Performance comparison against a counterpart algorithm.

Finally, we compare the *Advanced 3-Packet* algorithm against to a previous work, *MobSampling* [14], which is most closely related to our work. Since *MobSampling* operates based on the lane infrastructure information, and the exact system-level comparison in the same environment is not feasible, we implemented only the core relaying method for the comparison study. We implemented two variants of as-is *MobSampling* and an improved *MobSampling* that embeds the crucial *adaptive cell sizing* algorithm called *cell-split* of ours, for fair comparison. It is because *MobSampling* does not have the concept of adaptive granularity control for sampling that has turned out to be very important to achieve higher accuracy.

As shown in Fig. 26, our algorithm is more accurate than both *MobSampling* without cell-split and with cell-split with a factor of 9.5 and 2.3, respectively. The comparison result against *MobSampling* without cell-split demonstrates that our *Adaptive cell sizing* algorithm is an effective component to have a reliable estimation by varying the information resolution depending on the detected local vehicle density level. Upon comparing with *MobSampling* with cell-split, the purely packet-based *Inner-Relaying* and *Outer-Relaying* of ours without any prior knowledge of road infrastructure outperforms an infrastructure-aware relaying scheme.

VIII. Conclusion

We have presented a traffic density estimation algorithm that aims to obtain a snapshot of the traffic density distribution map during a survey time period. Our algorithm consists of three parts: on-demand vehicle sampling, probing packet relaying, and adaptive cell sizing. Based on our on-demand vehicle sampling algorithm, a probing packet keeps sampling while being embedded into a vehicle (i.e., sampler) to collect local traffic information. When the packet notices that its current sampler is no longer an efficient carrier, it is relayed to another vehicle based on our packet relaying algorithm. Our adaptive cell sizing algorithm controls the size of a cell, which is the basic unit for density estimation. By adaptively changing the cell size, we can achieve the necessary level of information within the remaining time budget. As a result, a single probing packet plays an effective role in gathering local traffic information with high fidelity. Further, extending the probing by using multiple packets along with multiple vehicles leads to an even more accurate traffic estimation based on cooperative area exploration decision. We have validated that our algorithm effectively obtains the traffic density distribution using V2V packet probing through balancing the level of estimation accuracy and the survey time budget.

For future work, we may adaptively increase and decrease the number of probing packets depending on various traffic conditions under a shorter survey time for more accurate estimation. It would be interesting to assign each different survey time to a cell depending on the local traffic density level by exploiting some historical pattern and allowing to access some degree of prior knowledge on the infrastructure in parts. Also, we may devise a dynamic sampling mechanism based on adaptive sampling frequency selection to reduce some redundant communication cost.

References

[1] I. Lee and K. Lee, "The internet of things (IoT): Applications, investments, and challenges for enterprises," *Bus. Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[2] G. Yang *et al.*, "A health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2180–2191, Nov. 2014.

[3] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *Proc. IEEE World Forum Internet Things* , 2014, pp. 241–246.

[4] L. Kong, M. K. Khan, F. Wu, G. Chen, and P. Zeng, "Millimeter-wave wireless communications for IoT-cloud supported autonomous vehicles: Overview, design, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 62–68, Jan. 2017.

[5] E. Hossain *et al.*, "Vehicular telematics over heterogeneous wireless networks: A survey," *Comput. Commun.*, vol. 33, no. 7, pp. 775–793, 2010.

[6] 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.

[7] 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, 2009.

[8] C. P. van Hinsbergen, T. Schreiter, F. S. Zuurbier, J. Van Lint, and H. J. Van Zuylen, "Localized extended Kalman filter for scalable real-time traffic state estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 385–394, Mar. 2012.

[9] Y. Yuan, J. Van Lint, R. E. Wilson, F. van Wageningen-Kessels, and S. P. Hoogendoorn, "Real-time lagrangian traffic state estimator for freeways," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 59–70, Mar. 2012.

[10] J. Barrachina *et al.*, "A V2I-based real-time traffic density estimation system in urban scenarios," *Wireless Pers. Commun.*, vol. 83, no. 1, pp. 259–280, 2015.

[11] T. Darwish and K. A. Bakar, "Traffic density estimation in vehicular ad hoc networks: A review," *Ad Hoc Netw.*, vol. 24, pp. 337–351, 2015.

[12] M. Artimy, "Local density estimation and dynamic transmission-range assignment in vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 400–412, Sep. 2007.

[13] R. Bauza and J. Gozálvez, "Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications," *J. Netw. Comput. Appl.*, vol. 36, no. 5, pp. 1295–1307, 2013.

[14] L. Garelli, C. Casetti, C.-F. Chiasserini, and M. Fiore, "Mobsampling: V2v communications for traffic density estimation," in *Proc. IEEE 73rd Veh. Technol. Conf.*, 2011, pp. 1–5.

[15] M. Balcilar and A. C. Sonmez, "Extracting vehicle density from background estimation using Kalman filter," in *Proc. IEEE 23rd Int. Symp. Comput. Inform. Sci.*, 2008, pp. 1–5.

[16] X. Li, Y. She, D. Luo, and Z. Yu, "A traffic state detection tool for freeway video surveillance system," *Procedia-Social Behavioral Sci.*, vol. 96, pp. 2453–2461, 2013.

[17] S. Zhang, G. Wu, J. P. Costeira, and J. M. Moura, "Understanding traffic density from large-scale web camera data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5898–5907.

[18] W. Deng, H. Lei, and X. Zhou, "Traffic state estimation and uncertainty quantification based on heterogeneous data sources: A three detector approach," *Transp. Res. Part B: Methodol.*, vol. 57, pp. 132–157, 2013.

[19] T. Z. Qiu, X.-Y. Lu, A. H. Chow, and S. E. Shladover, "Estimation of freeway traffic density with loop detector and probe vehicle data," *Transp. Res. Rec.*, vol. 2178, no. 1, pp. 21–29, 2010.

[20] Y. Ma, M. Chowdhury, A. Sadek, and M. Jeihani, "Real-time highway traffic condition assessment framework using vehicle–infrastructure integration (vii) with artificial intelligence (AI)," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 615–627, Dec. 2009.

[21] X.-Y. Bai, X.-M. Ye, H. Jiang, and J. Li, "A novel traffic information system for vanet based on location service," in *Proc. 16th IEEE Int. Conf. Netw.*, 2008, pp. 1–6.

[22] J. Barrachina *et al.*, "I-VDE: A novel approach to estimate vehicular density by using vehicular networks," in *Proc. Int. Conf. Ad-Hoc Netw. Wireless*, 2013, pp. 63–74.

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

[24] S. Panichpapiboon and P. Leakkaw, "Traffic density estimation: A mobile sensing approach," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 126–131, Dec. 2017.

[25] R. Shirani, F. Hendessi, and T. A. Gulliver, "Store-carry-forward message dissemination in vehicular ad-hoc networks with local density estimation," in *Proc. IEEE 70th Veh. Technol. Conf. Fall*, 2009, pp. 1–6.

[26] A. Thakur and R. Malekian, "Fog computing for detecting vehicular congestion, an internet of vehicles based approach: A review," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 2, pp. 8–16, Summer 2019.

[27] S. Dornbush and A. Joshi, "Streetsmart traffic: Discovering and disseminating automobile congestion using VANET's," in *Proc. IEEE 65th Veh. Technol. Conf.*, 2007, pp. 11–15.

[28] S. Goel, T. Imielinski, and K. Ozbay, "Ascertaining viability of wifi based vehicle-to-vehicle network for traffic information dissemination," in *Proc. 7th Int. IEEE Conf. Intell. Transp. Syst.*, 2004, pp. 1086–1091.

[29] J. Wang, Y. Huang, Z. Feng, C. Jiang, H. Zhang, and V. C. Leung, "Reliable traffic density estimation in vehicular network," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6424–6437, Jul. 2018.

[30] T. Darwish and K. A. Bakar, "Traffic aware routing in vehicular ad hoc networks: Characteristics and challenges," *Telecommun. Syst.*, vol. 61, no. 3, pp. 489–513, 2016.

[31] T. S. Darwish, K. A. Bakar, and K. Haseeb, "Reliable intersection-based traffic aware routing protocol for urban areas vehicular ad hoc networks," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 1, pp. 60–73, Spring 2018.

[32] K. N. Qureshi, A. H. Abdullah, and A. Altameem, "Road aware geographical routing protocol coupled with distance, direction and traffic density metrics for urban vehicular ad hoc networks," *Wireless Pers. Commun.*, vol. 92, no. 3, pp. 1251–1270, 2017.

[33] Y. Sun, S. Luo, Q. Dai, and Y. Ji, "An adaptive routing protocol based on QoS and vehicular density in urban vanets," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 6, 2015, Art. no. 631092.

[34] C. Zhao, C. Li, L. Zhu, H. Lin, and J. Li, "A vehicle density and load aware routing protocol for VANETs in city scenarios," in *Proc. IEEE Int. Conf. Wireless Commun. Signal Process.*, 2012, pp. 1–6.

[35] M. Jerbi, S.-M. Senouci, T. Rasheed, and Y. Ghamri-Doudane, "An infrastructure-free traffic information system for vehicular networks," in *Proc. IEEE 66th Veh. Technol. Conf.*, 2007, pp. 2086–2090.

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

[37] S. Panichpapiboon and W. Pattara-Atikom, "Exploiting wireless communication in vehicle density estimation," *IEEE Trans. Veh. Technol.*, vol. 60, no. 6, pp. 2742–2751, Jul. 2011.

[38] M. D. Venkata, M. M. Pai, R. M. Pai, and J. Mouzna, "Traffic monitoring and routing in VANETs—A cluster based approach," in *Proc. IEEE 11th Int. Conf. ITS Telecommun.*, 2011, pp. 27–32.

[39] WnSN, "Suvnet-trace data," [Online]. Available: http://wirelesslab.sjtu.edu.cn/taxi_trace_data.html. Accessed on: Oct. 24, 2018.

[40] Y. Rubner, C. Tomasi, and L. J. Guibas, "The Earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, 2000.

Christina Suyong Shin received the B.S. and M.S. degrees in computer science and engineering from Ewha Womans University, in 2017 and 2019, respectively. She is currently working toward the Ph.D. degree at University of Southern California. She is broadly interested in wireless sensor networks and vehicular ad-hoc networks. Her current research mainly focuses on mobile network systems, autonomous vehicles.

JiHo Lee received the B.S. degree in software science from Dankook University, Yongin, Republic of Korea, in 2019. She is currently working toward the M.S. degree in computer science and engineering at Ewha Womans University, Seoul, Republic of Korea. Her current research interests include VANET, wireless ad-hoc networks, edge computing, and deep learning.

HyungJune Lee received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 2001, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 2006 and 2010, respectively. He joined Broadcom as a Senior Staff Scientist for working on research and development of 60 GHz 802.11ad SoC MAC. Also, he worked for AT&T Labs as a Principal Member of Technical Staff with the involvement of LTE overload estimation, LTE-WiFi interworking, and heterogeneous networks. He is currently an Associate Professor with the Computer Science and Engineering Department, Ewha Womans University. His current research interests include future wireless networks on the IoT, fog computing, VANET, and machine learning-driven network system design.