Efficient Data Gathering and Estimation for Metropolitan Air Quality Monitoring by Using Vehicular Sensor Networks

You-Chiun Wang and Guan-Wei Chen

Abstract—Owing to the rising awareness of environmental protection and health, people put a high premium on air pollution in their living environment. It thus draws considerable attention on air quality monitoring in cities. The paper suggests using a *vehicular sensor network (VSN)* to tactically monitor metropolitan air quality, and develops an *efficient data gathering and estimation (EDGE)* mechanism on VSN. EDGE has an objective to adaptively change data sampling rates of cars, such that the tradeoff between monitoring accuracy and communication cost is balanced. The monitoring accuracy is measured by the formal *air quality index (AQI)*, whereas the communication cost considers the amount of sampling data and the monetary reward given to drivers. To do so, EDGE proposes dynamic grid partition based on the variation of pollutant concentration, and computes the sampling rate by consulting car traffic in each grid. With the help of probabilistic reporting, it allows cars to collect air quality on more different positions and alleviate potential network congestion. Furthermore, EDGE applies the Delaunay triangulation to infer AQIs of the positions without any sensing data. Through *simulation of urban mobility (SUMO)* and *industrial source complex (ISC3)*, simulations are conducted based on practical metropolitan traffic and pollutant dispersion models. Experimental results demonstrate the significant effectiveness of the EDGE mechanism, under various scenarios.

Index Terms—air quality, data gathering, ISC3, SUMO, vehicular sensor network.

1 Introduction

Air pollution is caused by gaseous pollutants harmful to humans and ecosystem. Based on the report by World Health Organization [1], air pollution has been the biggest environmental health risk. The governments put much effort on monitoring such pollution, and define the *air quality index (AQI)* to evaluate pollution degree and communicate to the public. If AQI increases, air pollution becomes more severe and may cause adverse health effect or disease to people.

Air quality monitoring is usually done by placing a few monitoring stations on dedicated sites in a city [2]. However, this scheme has two limitations. First, it provides only coarse-grained monitoring, where the spatial resolution of air-pollution samplings is poor. Second, the scheme lacks flexibility. When the weather changes or the city develops, some old sites (e.g., displaced plants or new parks) may become unnecessary. To relax the limitations, some studies suggest using vehicles (e.g., cars or bikes) to carry sensors, which are called *vehicular sensor network (VSN)* [3], to provide flexible monitoring. Since vehicles have GPS navigation and they may roam through the city, VSN can tactically collect data from different locations at different times. Moreover, sensors in a VSN are able to conduct stable, long-term monitoring, as vehicles can give them an abundant supply of energy.

By using VSN to monitor air quality, this paper aims at efficiently collecting sensing data from cars and inferring absent data. We consider a set of cars equipped with gas sensors, GPS receivers, and wireless interfaces (e.g., Wi-Fi or LTE-A). They roam in the city, and periodically report sensing data along with positions to a remote server. Drivers can get monetary reward based on their reports. However, they cannot control the sampling rates of their cars. Instead, the rates are controlled by the server. Under this architecture, our objectives are to increase monitoring accuracy while decreasing the communication cost by dynamically adjusting sampling rates of different cars. We evaluate monitoring accuracy by the difference between real and estimated AQIs. The communication cost has two definitions: 1) the amount of data transmission to report air quality and 2) the monetary cost to reward drivers.

To get a good balance between the monitoring accuracy and communication cost, we keep three design considerations in mind. First, data sampling rates should be adaptively changed by referring to environmental factors like the variation of pollutant concentration and car traffic. Second, since sensing data collected on the positions in close proximity usually exhibit spatial correlation [4], [5], we should prevent cars from sampling data on nearby positions at similar time. Third, it is infeasible to collect data from all positions in the monitoring region, as there may exist obstacles (e.g., buildings). Thus, we need to estimate AQIs of such positions by exploiting sensing data that the server has received. Apparently, such estimation will affect monitoring accuracy.

Based on these considerations, this paper proposes an *efficient data gathering and estimation (EDGE)* mechanism. It employs a *region quadtree* to flexibly partition the monitoring region into heterogeneous grids and calculate data sampling rates. Then, cars adopt a *probabilistic reporting method* to avoid submitting similar data collected from close positions, which helps reduce wastage of sensing reports while alleviate network congestion. Moreover, we define various packet formats for cars to report their sensing data, so as to save data transmission. After gathering reports, EDGE finally uses the *Delaunay triangulation* [6] to infers missing data.

To evaluate EDGE's performance in a large-scale environment, we conduct simulations on practical models. We use

The authors are with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, 80424, Taiwan. E-mail: ycwang@cse.nsysu.edu.tw; m023040067@student.nsysu.edu.tw

the *simulation of urban mobility (SUMO)* [7] to generate car traffic in a city. SUMO is a popular traffic-simulation suite that supports sophisticated modeling of intermodal traffic systems such as vehicles, public transport, traffic lights, and pedestrians. We also apply the *industrial source complex (ISC3)* method [8] to simulate air pollution. ISC3 is a mature technique widely used to model pollutant diffusion from a wide variety of sources. Through different experimental scenarios, we show that EDGE can greatly save the amount of data transmission and monetary cost, while keeping monitoring accuracy. This paper contributes in developing an efficient, adaptive data gathering and estimation method for VSN to monitor metropolitan air quality, and verifying its outstanding performance via simulations considering real-life situations.

We organize the paper as follows: Section 2 introduces AQI, SUMO, and ISC3. Section 3 surveys related work, and Section 4 defines our problem. We propose EDGE in Section 5, and study simulation results in Section 6. Section 7 then draws a conclusion and discusses future work.

2 Preliminary

This section gives three models in the paper. We introduce AQI to evaluate air pollution, and discuss SUMO to simulate car traffic, followed by ISC3 to model pollutant dispersion.

2.1 Air-pollution Evaluation Model: AQI

We employ AQI defined by U.S. Environmental Protection Agency (EPA) [9] to evaluate air quality. It has six levels of air pollution in Fig. 1(a), each with one dedicated color:

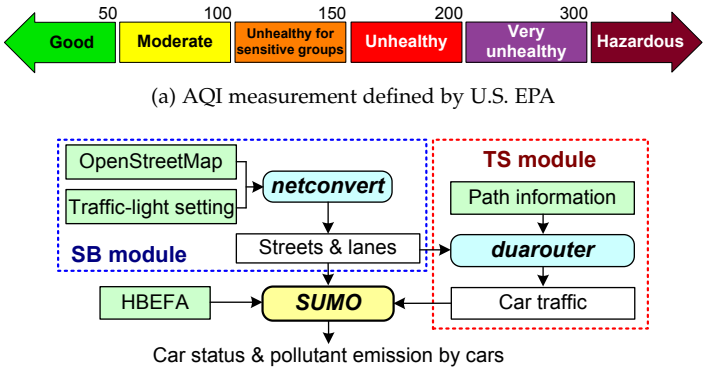
- *Good* (green, AQI $\leq$ 50): Outdoor air is safe to breathe.
- *Moderate* (yellow, AQI: 51–100): Unusually sensitive people may reduce prolonged or heavy outdoor exertion.
- *Unhealthy for sensitive groups* (orange, AQI: 101–150): Sensitive people (e.g., children, old people, outdoor workers, and patients with lung disease like asthma) need to reduce prolonged or heavy outdoor exertion.
- *Unhealthy* (red, AQI: 151–200): Sensitive people should avoid prolonged or heavy outdoor exertion. Everyone else has to reduce prolonged or heavy outdoor exertion.
- *Very unhealthy* (purple, AQI: 201–300): Sensitive people should avoid all outdoor exertion. Everyone else has to reduce outdoor exertion.
- *Hazardous* (maroon, AQI $\geq$ 301): Everyone should avoid all outdoor exertion.

EPA adopts six types of gaseous pollutants to evaluate AQI: ozone (O_3), particulate matter (i.e., PM 2.5 and PM 10), carbon monoxide (CO), sulfur dioxide (SO_2), and nitrogen dioxide (NO_2). For each pollutant, its index I_k is computed by

$$I_k = \frac{I_{\text{high}} - I_{\text{low}}}{B_{\text{high}} - B_{\text{low}}} \times (C_k - B_{\text{low}}) + I_{\text{low}}, \quad (1)$$

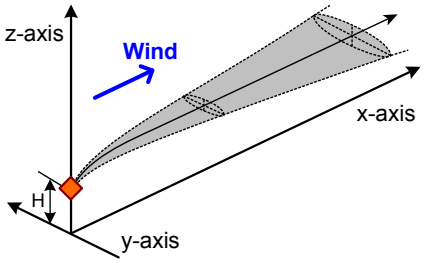
where B_{high} is a breakpoint $\geq C_k$, B_{low} is a breakpoint $\leq C_k$, and $I_{\text{high}}/I_{\text{low}}$ denote the AQI value for $B_{\text{high}}/B_{\text{low}}$. The suggested values of these parameters are given in [9].

In Eq. (1), C_k is the average concentration of pollutant k in an observing period. Based on the EPA standard, the period length is 1 hour for O_3, SO_2, and NO_2, 8 hours for O_3 and CO, and 24 hours for PM 2.5 and PM 10 [9]. Then, the overall AQI is the maximum value of I_k among all pollutants. In this paper, we use AQI to assess monitoring accuracy and determine the reward given to a driver.



(a) AQI measurement defined by U.S. EPA

(b) metropolitan traffic modeled by SUMO



(c) pollutant dispersion modeled by ISC3

Fig. 1: Three models considered in the paper.

2.2 Metropolitan Traffic Model: SUMO

SUMO is an open-source package to model real roads and car traffic, which involves high level of detail and realism [10]. It provides microscopic road imitation, including speed limit, road width, main street, minor lane, traffic light, etc. Each car is *independently* defined by multiple parameters such as identifier, departure time, velocity, and route. The mobility of cars is also delicately described. Specifically, cars stop with braking speed when meeting a red light, and resume moving with acceleration if the traffic light turns to green. With these parameters, SUMO can separately simulate the behavior of each individual car in a dynamic way. Besides, a car can be assigned to one pollutant or noise emission class to simulate air pollution or noise emitted by cars. Moreover, SUMO considers the effect of adjacent cars. For example, when two cars meet at a crossroad without traffic lights, one car will stop until another car leaves. If two cars collide, SUMO waits for a period and then removes the cars to imitate a traffic accident.

Fig. 1(b) shows the construction of traffic model by SUMO. It has two modules: *street building (SB)* and *traffic-flow setting (TS)*. The SB module gets the street map of a city from *OpenStreetMap* [11], and decides locations and red/green time of traffic lights. We can use 'netconvert', SUMO's road-network importer, to build streets and lanes on the map. The TS module defines path information for each car, including starting/terminal points, starting time, acceleration, and braking speed. Then, we use the 'duarouter' function to generate car traffic. The outputs of both modules are fed into SUMO to simulate the status of each car in the city.

HBEFA (handbook emission factors for road transport) [12] provides emission factors of gaseous pollutants for cars, which helps SUMO simulate air pollution generated by cars. It defines more than 20 vehicle emission classes to model pollutant emission of cars for a wide variety of traffic situations. Through HBEFA, SUMO can model the emission of different pollutants, including CO_2, CO, NO, NO_2, PM 2.5/10, and hydrocarbon

(HC).

2.3 Pollutant Dispersion Model: ISC3

ISC3 is a steady-state Gaussian plume model used to imitate diffusion and sedimentation of pollutants in the air. It describes many features of gaseous pollutants, including 1) point, line, area, and volume sources, 2) division of point sources, 3) dry deposition and settling of particles, 4) down-wash, and 5) limited terrain adjustment. ISC3 computes pollutant concentration at a 3D position (x, y, z) by

$$C(x,y,z) = \frac{QWVD}{2\pi\mu_s\tau_y\tau_z} \times \exp\left[-\frac{1}{2}\left(\frac{y}{\tau_y}\right)^2\right], \quad (2)$$

where Q is the pollutant discharge rate (in g/s), W converts the output to concentration, V is the reflection in vertical direction (used in earth surface and atmosphere's inversion layer), D is a coefficient for disintegration (when pollutants have a half-life period, e.g., SO_2), μ_s is wind speed (in m/s), and τ_y and τ_z are coefficients of diffusion in horizontal and vertical directions, respectively. We can compute parameter V in Eq. (2) by

$$V = \exp\left[-\frac{1}{2}\left(\frac{z-H}{\tau_z}\right)^2\right] + \exp\left[-\frac{1}{2}\left(\frac{z+H}{\tau_z}\right)^2\right], \quad (3)$$

where H is the pollutant's height.

Fig. 1(c) shows ISC3, where the diamond denotes a pollutant source and wind blows along the x axis. The pollutant spreads in horizontal and vertical directions depending on τ_y and τ_z, respectively. The gray area indicates the region affected by the pollutant. The coefficients in Eqs. (2) and (3) are decided by weather and temperature, and [8] gives the suggested values. The minimum observing period of ISC3 is 1 hour, so we use Eq. (2) to update concentration in each hour.

We adopt ISC3 to model pollutant dispersion due to three reasons. First, it was considered as the most commonly used model to support various air pollution regulations by U.S. EPA [13]. Second, the accuracy of ISC3 is analyzed in large-scale geographic regions like Colombia [14] and Delhi [15]. These studies show that there exists high correlation (> 0.73) between the result computed by ISC3 and actual values measured by monitoring stations, which verifies that ISC3 can precisely describe the main factors affecting pollutant dispersion in a region. Third, ISC3 has been widely used to assess pollutant concentration from a variety of sources associated with an industrial complex. Thus, we can use ISC3 to simulate air pollution emitted from industrial zones or factories in a city.

ISC3 operates in both long-term and short-term modes. In our simulations, we use the short-term mode to imitate air pollution in the environment. Moreover, cars will also generate gaseous pollutants (by SUMO), which further improves the simulation's similarity to real scenarios.

3 Related Work

Many studies aim at monitoring air quality in a city, which are classified into two categories. One uses fixed devices (e.g., monitoring stations and static sensors) to gather environmental data. The other adopts mobile devices (e.g., smart phones and VSN) to dynamically collect air quality.

3.1 Monitoring Schemes by Fixed Devices

Using monitoring stations to measure air quality is a traditional solution, where each station provides delicate detection of gaseous pollutants. However, there are limited stations in a city, and air quality is affected by various factors like weather, car flow, people mobility, street canyons, and industrial installations [16]. To infer missing data, Zheng et al. [17] analyze past air quality by two methods. One is based on an artificial neural network to find spatial data correlation on different positions. Another uses a linear-chain conditional random field to evaluate temporal data dependency on a position.

Comparing to monitoring stations, *wireless sensor network (WSN)* provides more fine-grained monitoring of air quality. The work [18] uses a static WSN to monitor air pollution and proposes an auto-calibration method to improve data accuracy. Sensors compare the measured pollutant concentration with the data collected by their neighbors and nearby monitoring stations, so as to correct their measurement. Wang et al. [19] adopt static sensors to monitor CO and PM. To extend lifetime, sensors are powered by solar batteries, and enter the sleeping state in a suspended period. The study [20] installs sensor suites at multiple sites in a city to monitor NO_2, CO, SO_2, PM, and hydrogen sulfide (H_2S). Sensing data are mapped to a rank of AQI following the *data quality objective* regulated by European Directive [21]. Brienza et al. [22] develop a cooperative sensing system. People set up gas sensors on their houses to monitor air quality in the surroundings, and share the monitoring data via social networking.

However, the monitoring accuracy of WSN highly relies on sensor deployment [23]. Due to their small sensing range, we need to deploy a huge number of sensors to cover a large metropolitan area. Moreover, it lacks flexibility to use static WSN to collect air quality. When the monitoring mission changes (e.g., the area of interest alters), the originally installed WSN will be difficult to participate in the new mission.

3.2 Monitoring Schemes by Mobile Devices

With the popularization of smart phones, some studies use web-based frameworks or mobile applications (APP) to let people collect air quality and share data. Mun et al. [24] design a platform on mobile phones to record the impact of CO_2, PM 2.5, and smog exposure. Sensing data are uploaded to a server, which exhibits the analytical result via a web interface. In [25], participants wear air-pollution sensors and carry smart phones throughout one day, especially during rush-hour commutes. Each phone provides an APP to display recent measurement of air quality. Also, [25] develops a web-based framework to help participants report sensing data and provide access to the map overlaid with historical data. The work [26] connects a smart phone with the O_3 sensor to detect air pollution. To improve monitoring accuracy, it exploits the O_3 data collected from monitoring stations to calibrate sensors, and analyzes the effect of mobility on the accuracy of sensor readings. Cheng et al. [27] propose a cloud-based system to monitor PM 2.5, where sensors report data to a cloud server via Ethernet or mobile phones. They use the artificial neural network and Gaussian process to respectively calibrate sensing data and infer data on the positions without sensors.

Since cars move longer distances than pedestrians, using VSN to monitor air quality is more flexible. Ma et al. [28] combine VSN with a static WSN to detect air pollution. Static sensors are deployed in a grid-based manner, and act as the

backbone to relay sensing data collected by cars. The work [29] develops a VSN to collect urban CO_2 concentration. Each car regularly reports its sensing data along with position via GSM short messages. The monitoring result is displayed on Google Maps to visually show CO_2 distribution. In [30], cars are equipped with gas sensors to analyze air pollution (e.g., CO, NO_2, and O_3) in Sydney. A mobile APP is developed for drivers to upload their sensing data to a server, whose result is also shown on Google Maps. The study [31] considers a VSN architecture for air-quality monitoring, which contains both *public transportation sensing (PTS)* and *social sensing (SS)* networks. PTS network consists of buses moving along fixed routes. SS network contains passenger cars, where drivers report sensing data via mobile phones. Vagnoli et al. [32] use bikes to carry gaseous sensors to measure CO, CO_2, O_3, NO_2, and methane (CH_4). Sensing data are sent to a database via GPRS and shown by a web application. In contrast to the above studies, our work aims at developing adaptive mechanisms to gather sensing data and estimate absent data, so as to help VSN efficiently monitor air quality in a big city.

The work [33] installs *mobile agents* (i.e., migratory programs) on some cars to collect air quality. Each agent periodically checks if its car has reached the target area. If not, it checks whether the car's route is different from the expected one or the car is stuck in traffic. If so, the agent then migrates to another car. Once the agent reaches the target area, it begins collecting air quality and tries to return to the server. Obviously, this work has a different goal. Hu et al. [34] use VSN to monitor CO_2 in cities, and adjust sampling rates of cars to keep monitoring quality with less communication overhead. They divide the monitoring region into fixed grids, and propose both *variation-based (VAR)* and *gradient-based (GRA)* schemes to adjust the sampling rate in each grid by

$$R_i = a_i \times \phi_i + b_i, \tag{4}$$

where R_i is the expected number of samples, a_i and b_i are two constants based on past experience, ϕ_i is the standard deviation and gradient difference of CO_2 concentration in VAR and GRA, respectively. Comparing to [34], our EDGE mechanism flexibly adjusts sampling rates through 'dynamic' grid partition and addresses real road conditions, and it also designs an efficient solution to infer the lost AQI data. Moreover, our simulations adopt SUMO and ISC3 to model the practical metropolitan traffic and pollutant dispersion, which are not considered in [34]. Experimental results in Section 6 will show that EDGE greatly saves the communication cost and improves monitoring accuracy than both VAR and GRA.

4 Problem Definition

Let us consider a monitoring region $\mathcal{A}$ in the city which can be divided into a set $\mathcal{U}$ of *pixels*, where each pixel $u_i \in \mathcal{U}$ is the smallest unit area (e.g., $1\,m^2$) used to monitor air quality and calculate AQI. In other words, the values of AQI measured at any two positions in a pixel at the same time are viewed as *no difference*. We assume that $\mathcal{A}$ is fully covered by a wireless network such as Wi-Fi or LTE-A, so each car moving in $\mathcal{A}$ can always find a nearby base station (BS) to upload its data and receive commands. Depending on the given sampling rate, each car regularly sends out its sensing data for air quality along with the current time and position to the remote server (via the wireless network). The car's driver can be rewarded with a little money for reporting data. However, the driver cannot change the sampling rate and data content. Moreover, the driver has no idea how or where to move the car to increase reward. Thus, the car traffic in $\mathcal{A}$ is viewed as *random* (referring to Remark 1). Besides, we cut the time axis into repetitive *monitoring periods* with a length of T, depending on the type of observing pollutant (referring to Section 2.1). After collecting data from cars, the server can estimate each pixel's AQI by Eq. (1).

Let $d_i^{\mathbf{R}}$ and $d_i^{\mathbf{E}}$ be the real AQI and the AQI estimated by the remote server of each pixel $u_i \in \mathcal{U}$. Besides, we denote the set of all reporting packets from cars by $\mathcal{P}$ in a monitoring period T. Then, our problem asks how to adjust the data sampling rate of each car and estimate the lost data, such that the following three objectives can be satisfied:

$$\min \sum\nolimits_{p_j \in \mathcal{P}} \text{length}(p_j), \tag{5}$$

$$\min \sum\nolimits_{p_j \in \mathcal{P}} \text{reward}(p_j), \tag{6}$$

$$\min \frac{1}{|\mathcal{U}|} \sum\nolimits_{u_i \in \mathcal{U}} f(d_i^{\mathbf{R}}, d_i^{\mathbf{E}}), \tag{7}$$

where $\text{length}(p_j)$ and $\text{reward}(p_j)$ respectively denote the length and monetary cost of each packet $p_j \in \mathcal{P}$, $|\mathcal{U}|$ is the number of total pixels in $\mathcal{A}$, and

$$f(d_i^{\mathbf{R}}, d_i^{\mathbf{E}}) = \begin{cases} 1 & \text{if } |d_i^{\mathbf{R}} - d_i^{\mathbf{E}}| > \delta \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where δ is a tolerable deviation. Eq. (5) indicates to reduce the amount of packet transmission to report data, so as to save wireless bandwidth and avoid network congestion. Eq. (6) means to save the rewards given to drivers, so the monetary cost can decrease. Here, Eqs. (5) and (6) together minimize the total communication cost by using VSN to monitor air quality. Then, Eq. (7) improves monitoring accuracy by minimizing the ratio of pixels whose real and estimated AQIs have a difference large than the threshold δ.

Remark 1 (Driving Behavior)**.** *The work [35] points out that moving cars usually exhibit behavior unique to each driver (e.g., choosing the moving path). We develop our mechanism based on this observation, where each car moves depending on its destination, instead of exterior influence such as the reward given to a driver. Thus, we can avoid the case where some drivers* intentionally *move to or stay at certain locations in order to increase their rewards, which may substantially change car traffic or even cause congestion. Moreover, the assumption also makes sure that our mechanism can perform well under most vehicular mobility models [36].* □

Remark 2 (Rewarding Policy)**.** *We aim at adjusting sampling rates of cars to balance between the communication cost and monitoring accuracy. So, the reward is used as a metric to measure the communication cost in the paper. In Section 6, we will propose two simple rewarding policies for performance evaluation. How to design a sophisticated policy to influence the driving behavior (to facilitate gathering air quality) is out of the paper's scope, and it also conflicts with the assumption mentioned in Remark 1. Accordingly, we will leave this issue for future investigation in Section 7.* □

5 The Proposed EDGE Mechanism

We partition $\mathcal{A}$ into $K \times K$ grids, whose length is an integer multiple of pixel length. So, each pixel in $\mathcal{U}$ belongs to a single grid. The time axis is divided into *monitoring periods* with length of T, during which EDGE is executed, as Fig. 2 shows. A monitoring period is further composed of three phases.

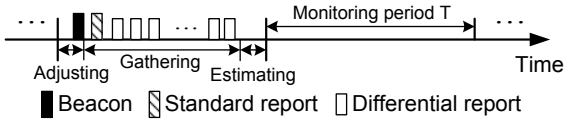
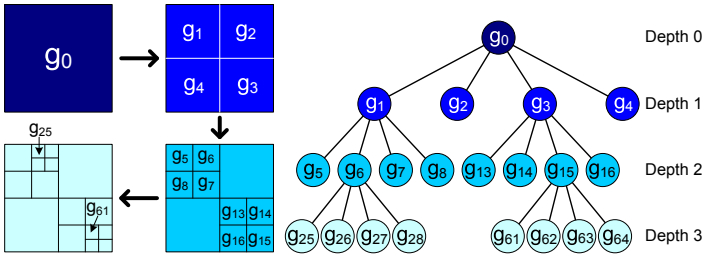


Fig. 2: Time structure of our EDGE mechanism.



(a) grid partition (b) region quadtree

Fig. 3: Using a region quadtree to maintain the dynamic grid partition.

Adjusting phase: Based on the sensing data collected in the previous period, the remote server modifies the grid partition by cutting a grid that has diverse data, or merging adjacent grids with similar data. To do so, EDGE uses a *region quadtree* to keep 'dynamic' grid partition. Then, it recomputes the sampling rate of each grid, and sends *beacons* containing the new partition along with sampling rates to cars via BSs.

Gathering phase: On hearing a beacon, each car decides its interval between two reports. To avoid some cars simultaneously sampling data on close positions, which results in redundant data, we propose a *probabilistic reporting method* to let cars sample data from different positions. Moreover, two types of packet formats, called *standard* and *differential* reports, are designed to provide flexible reporting for cars. As shown in Fig. 2, each car first sends a standard report to give complete information, followed by successive differential reports to save the amount of data transmission.

Estimating phase: After gathering data from cars, the remote server obeys the guideline in Section 2.1 to compute each pixel's AQI. However, some pixels may have no data due to lack of cars. Thus, we adopt the *Delaunay triangulation* to estimate AQIs for such pixels.

The gathering phase dominates a monitoring period, as adjusting and estimating phases are done by the remote server in the background. Each car has to keep track of beacons to update the grid partition and sampling rate. In case of missing the beacon, a car sends a *probing packet* to its BS to get the necessary information (discussed in Section 5.2.2).

In Fig. 2, T decides the updating frequency of grid partition. Hence, we prefer the length of pollutant observing period to be several times of T (e.g., 5 or 6 times), so the update of grid partition can catch the concentration variation in an observing period. As mentioned in Section 2.1, the minimum observing period is 1 hour. Thus, we set T to 10 minutes in the simulations. Below, we detail our design in each phase.

5.1 Adjusting Phase

5.1.1 Dynamic Grid Partition

We adopt a *region quadtree* for space indexing. It is a popular data structure that can easily describe a partition of 2D space by iteratively decomposing the space into four equal quadrants, as shown in Fig. 3. Each tree node either has exactly four children (i.e., an internal node), or has no children (i.e., a leaf). We choose to use the region quadtree due to three reasons:

- The region quadtree is suitable to maintain our grid partition, where the root denotes $\mathcal{A}$, and each leaf corresponds to a grid where all cars share the same sampling rate.
- Let n be the depth of a region quadtree. Then, $\mathcal{A}$ can be divided into at most $2^n \times 2^n$ grids. Thus, we can control the number of grids in $\mathcal{A}$ by simply adjusting tree depth.
- By assigning each node of a *perfect region quadtree*[1] with an identification (ID) in sequence, where the root is given with ID g_0, the IDs of four children of a node with ID g_i must be $4g_i + 1, \cdots, 4g_i + 4$. Fig. 3 gives an example, where the IDs of four children of node g_6 are $g_{25}, \cdots, g_{28}$. Thus, it is fast to map any grid in $\mathcal{A}$ to the corresponding node in the region quadtree by using IDs.

As we will mention in Section 5.2.2, using the region quadtree helps simplify packet formats by avoiding recording too many positioning data. Moreover, the value of K can be decided by the depth of the region quadtree, as discussed in Remark 3.

Remark 3 (Determining tree depth and K)**.** *To avoid cutting the monitoring region $\mathcal{A}$ into too small grids, we should limit the maximum depth n of the region quadtree by*

$$\sigma \times \sqrt{\frac{|\mathcal{A}|}{2^n \times 2^n}} \geq v_{\max} \times t_{\min}, \tag{9}$$

where σ is a ratio, $|\mathcal{A}|$ denotes the area of $\mathcal{A}$, $v_{\max}$ is the maximum velocity of cars, and $t_{\min}$ is the minimum expected time that each car should stay in a grid. Here, the left part of Eq. (9) computes the average length of moving paths in the smallest grid. Since roads may not be necessarily parallel to the grid edge, the grid length must not be the minimum path in a grid. That is why we multiply a ratio of σ to the grid length to calculate the average length. (Remark 5 will discuss how to find σ.) The right part of Eq. (9) gives the shortest moving distance that a car is expected to pass through the smallest grid. By using Eq. (9), we can ensure that a car will averagely stay in the smallest grid for at least $t_{\min}$ time to monitor air quality. It also prevents cars from frequently changing their sampling rates in a monitoring period when they move fast.

Besides, $\mathcal{A}$ is divided into $K \times K$ grids in the beginning. We can set $K = 2^{\lceil \frac{n}{2} \rceil}$ such that the initial tree depth is nearly a half of the maximum tree depth. It could thus help fast adjust the grid partition of $\mathcal{A}$ in the initial stage. □

Definition 1. *Let $I_1, I_2, \cdots, I_k$ be AQIs of sensing data $\{s_1, s_2, \cdots, s_k\}$, respectively. Then, $s_1, s_2, \cdots, s_k$ are called λ-similar if 1) $I_1, I_2, \cdots, I_k$ belong to the same AQI level in Section 2.1 and 2) $\max_{j=1..k}\{I_j\} - \min_{j=1..k}\{I_j\} \leq \lambda$.* □

Given the grid partition, cars in each grid can report their sensing data following the sampling rate in that grid. Then, based on the data collected from the previous monitoring period, EDGE adaptively adjusts the grid partition by the four cases blow. Fig. 4 also gives an example to show these cases, where we have seven grids g_1, g_2, g_4, g_{17}, g_{18}, g_{19}, and g_{20}.

Case of no-change: If all sensing data sampled in a grid g_i are λ-similar, it means that the pollutant concentration remains

1. It is a region quadtree tree in which all internal nodes have four children and all leaves have the same depth.

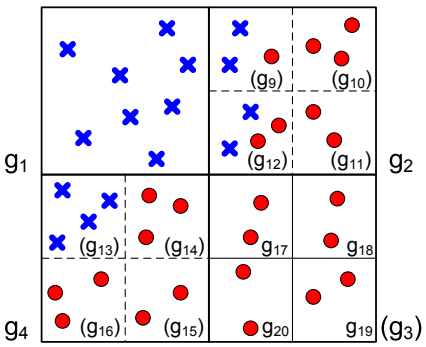


Fig. 4: Four cases to adjust the grid partition, where there are two groups of λ-similar data (marked by crosses and circles), and the IDs in brackets indicate that the corresponding grids do not appear in the current grid partition.

TABLE 1: Detection range and sensitivity of gaseous sensors.

pollutant	detection range	sensitivity	reference
O_3	10–1000 ppb	1.5	[37]
CO	30–1000 ppm	0.13	[38]
SO_2	1–200 ppm	3	[39]
NO_2	50–5000 ppb	6	[40]
PM 2.5/10	0–999 $\mu g/m^3$	5% (error)	[41]

ppb: parts per billion, ppm: parts per million

steady in g_i. So, there is no need to modify such a grid. Grid g_1 in Fig. 4 shows this case.

Case of dividing: When some child grids of a grid g_i contain sensing data that are not λ-similar, the pollutant concentration would significantly vary in g_i. Therefore, we should divide g_i to get a more fine-grained observation. Grid g_2 in Fig. 4 gives an example. Since its child grids g_9 and g_{12} have non-λ-similar data, g_2 should be further partitioned.

Case of merging: It is a special condition of the *case of no-change*. In particular, a grid g_i and its three sibling grids (i.e., they share the same parent in the region quadtree) contain only λ-similar data. This case implies that the current grid partition is too narrow. Thus, we merge g_i with its three sibling grids. Fig. 4 presents an example, where grids $g_{17} \sim g_{20}$ will be merged into a large grid g_3 due to their similar data.

Case of marking: It is a special condition of the *case of dividing*. Specifically, a grid g_i contains non-λ-similar data, but each of its child grids has only λ-similar data. Grid g_4 in Fig. 4 shows an example, where it has two groups of λ-similar data, but its child grids g_{13}, g_{14}, g_{15}, and g_{16} each has only one group of data. However, if we simply divide the grid, each subgrid will satisfy the *case of no-change* but all subgrids share the same sampling rate. Obviously, such division is useless. Thus, we 'mark' g_i (without dividing it) and increase its sampling rate accordingly (discussed in Section 5.1.2).

Since EDGE aims at long-term monitoring, we amend the region quadtree smoothly and iteratively. Thus, each grid is examined once in a monitoring period. In this way, the depth of the region quadtree is either increased/decreased by one (due to the cases of *dividing* and *merging*) or has no change in every period. Besides, if a grid satisfies the *case of dividing* but its tree depth reaches the maximum threshold by Eq. (9), we apply the *case of marking* to that grid to avoid generating too small grids (as discussed in Remark 3). Remark 4 comments on the four cases and λ.

Remark 4 (Four cases and λ)**.** *When the pollutant concentration of a grid is steady, there might exist little fluctuation in sensor readings.*[2] *Also, due to sensor calibration or some circumstances (e.g., temporarily following a truck), it may be difficult to find a grid whose sensing data are the same. Thus, if we simply decide whether to keep or split a grid by checking if all of its sensing data have an 'equal' AQI, the cases of* no-change *and* merging *would never occur. Besides, the* case of dividing *may frequently happen, making $\mathcal{A}$ be partitioned into many small grids. To solve the problem, we use Definition 1 by allowing a small tolerance λ on the AQI values of sensing data. Thus, the occurring probability of cases of* no-change *and* merging *can be substantially increased.*

To find λ's value, we refer to past statistics of pollutant concentration from monitoring stations, and take 3% of concentration range, for instance, to be λ. We take Kaohsiung city as an example, where the minimum and maximum O_3 concentration is 4.77 and 72.96 ppb, respectively [42]. Thus, we set $\lambda = (72.96 - 4.77) \times 3\% \approx 2$ ppb. When more sophisticated sensors are used, we can lower λ's value accordingly (e.g., taking 1% of concentration range). □

5.1.2 Calculating data sampling rates

After deciding the grid partition, EDGE finds the sampling rate $\hat{r}_i$ of each grid g_i by

$$\hat{r}_i = \varepsilon \times \frac{B(g_i)}{F(g_i) \times t_{\text{avg}}(g_i)}, \tag{10}$$

where ε controls data-sampling speed, $B(g_i)$ is a baseline for the number of sensing data to be collected in g_i, $F(g_i)$ is car traffic in g_i, and $t_{\text{avg}}(g_i)$ is average time that cars stay in g_i. Here, $B(g_i)$ is a constant depending on the application, so $\hat{r}_i$ is affected by other parameters. We set coefficient ε as follows:

$$\varepsilon = \begin{cases} 1/2 & \text{if all sensing data in } g_i \text{ are } \lambda\text{-similar} \\ 2 & \text{otherwise.} \end{cases} \tag{11}$$

The design consideration behind Eq. (11) is to adaptively adjust the sampling rate based on the data variation in each grid. Specifically, when the pollutant concentration remains steady, it is unnecessary to collect many similar data in the grid. Thus, we halve the sampling rate by taking $\varepsilon = 1/2$. On the contrary, when there is significant variation in concentration, we should double the sampling rate (by taking $\varepsilon = 2$) to capture such high variation. Fig. 4 gives an example. We slow down the sampling rates of grids g_1, g_3, g_{10}, and g_{11}, as they cover the area where the pollutant concentration keeps stable. For grids g_4, g_9, and g_{12}, we speed up their sampling rates to react to the significant change in concentration.

In Eq. (10), $F(g_i)$ is defined by the number of cars in g_i during unit time, which is estimated by the number of sensing data collected in the previous monitoring period. When g_i has large car traffic, we can slow down its sampling rate because more cars help collect g_i's air quality, and vice versa. To alleviate the effect of extreme situations (e.g., g_i contains very few cars or traffic congestion occurs in g_i), making $\hat{r}_i$ become unreasonably large or small, we use two thresholds $F_{\min}$ and $F_{\max}$ to limit the range of $F(g_i)$. When $F(g_i) < F_{\min}$, we set $F(g_i) = F_{\min}$. Once $F(g_i) > F_{\max}$, we set $F(g_i) = F_{\max}$. To determine $F_{\min}$ and $F_{\max}$, we can gather statistics of car traffic in $\mathcal{A}$, and take the average of smallest and largest 20% of past data to be the values of $F_{\min}$ and $F_{\max}$, respectively. For example, by using SUMO to imitate car traffic in Kaohsiung city, we set $F_{\min} = 5$ and $F_{\max} = 20$ in our simulations.

2. For example, Table 1 shows the detection range and sensitivity of practical gaseous sensors. The sensitivity indicates how much a sensor's output changes when the input quantity being measured alters, and it limits the range of fluctuation in sensor readings.

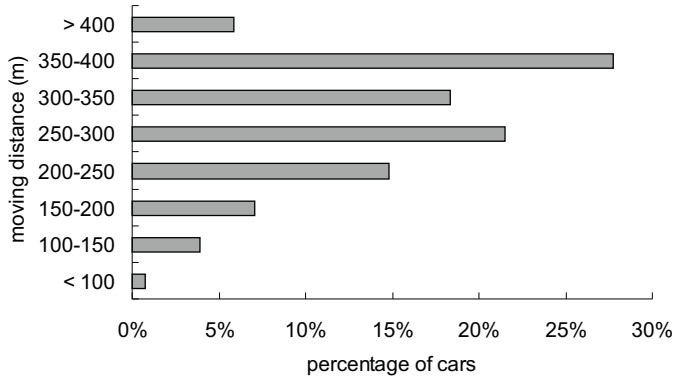


Fig. 5: The percentage of different moving distances for cars to pass through one grid, where we take Kaohsiung city as an example.

The sampling rate $\hat{r}_i$ is inversely proportional to $t_{\text{avg}}(g_i)$ in Eq. (10). Obviously, if a car stays in g_i for longer time, it can collect more data from g_i, and vice versa. The average staying time $t_{\text{avg}}(g_i)$ is derived by $\sigma \times L(g_i)/v$, where $L(g_i)$ denotes the length of grid g_i and v is the average velocity of that car. Remark 5 discusses how to choose a proper ratio σ to find the average distance required by cars to pass through a grid.

Remark 5 (The choice of σ)**.** *To find σ, we can employ SUMO to estimate the moving distance of each car to pass through one grid in $\mathcal{A}$. For instance, Fig. 5 gives the percentage of different moving distances of cars in Kaohsiung city, and we divide $\mathcal{A}$ into 16×16 grids. The grid length is 386 meters. Since each grid has different street topologies and traffic amount, cars spend different distances to move through a grid. From Fig. 5, we observe that most cars move a distance of $200 \sim 400$ meters to pass through one grid, and their average moving distance is 322 meters. Thus, we set $\sigma = 322/386 \approx 0.834$ in the example. For other cities, we can use the same way to calculate their σ values.* □

It is worth mentioning that a large $F(g_i)$ value (i.e., higher car traffic) does not necessarily imply a large $t_{\text{avg}}(g_i)$ value (i.e., longer staying time of cars). One example is that many cars in a grid move through unobstructed highways, so each car will not stay in the grid for a long time. Another example is that just few cars in a grid are impeded by a serial of red lights. In this example, we have a small $F(g_i)$ value but a large $t_{\text{avg}}(g_i)$ value. That is why we consider both parameters $F(g_i)$ and $t_{\text{avg}}(g_i)$ in Eq. (10).

5.2 Gathering Phase

When the adjusting phase ends, the remote server announces the new grid partition and the data sampling rate in each grid to cars via beacons (broadcasted by BSs). Therefore, once a car enters a new grid g_i with sampling rate $\hat{r}_i$ derived from Eq. (10), it decides the time interval between two successive reports in that grid by $f_{\mathbf{T}}(1/\hat{r}_i)$, where $f_{\mathbf{T}}(\cdot)$ is a translation function on the basis of slot time. For example, supposing that $\hat{r}_i = 20$ samples/h and each slot is 1 minute, the time interval will be $f_{\mathbf{T}}(1/20) = 1/20 \times 60 = 3$ slots (i.e., minutes). To conduct sampling, each car in g_i keeps a counter initially set to $f_{\mathbf{T}}(1/\hat{r}_i)$. When the counter reaches to zero, the car measures air quality on the current position, sends its sensing data to the server, and resets the counter again. This mechanism is straightforward but it leaves three questions.

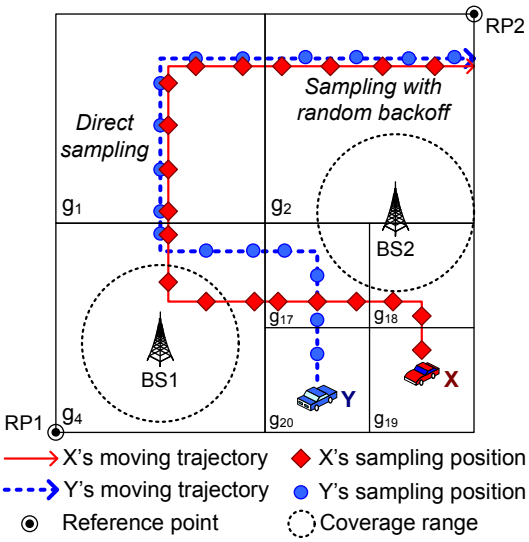


Fig. 6: Using random backoff to let cars sample data on different positions.

Question 1: Because there could exist traffic lights in a grid and the velocities of cars may vary, how do we control the decrement of each car's counter for such situations?

Question 2: Since cars in a grid share the same sampling rate, and some cars may move along the same road, how do we prevent them from collecting data on similar positions?

Question 3: As a grid may contain crowded streets, how do we avoid network congestion due to the transmission of overwhelming sensing data?

To solve these questions, we develop a probabilistic reporting method for cars to efficiently transmit their sensing data.

5.2.1 Probabilistic Reporting Method

For *Question 1*, if we simply decrease the counter of a car in a constant speed, when the car stops due to the red light or other situations (e.g., traffic congestion), it may sample data on some positions close to each other. Even worse, the car may collect multiple (and possibly redundant) data on the same position when it waits for a long red-light time. Thus, even if the car sends a lot of sensing data, these data cover a very small part in the grid. To address this issue, we propose one improvement as follows:

Improvement 1: We freeze a car's counter when it stops, and resume the counter again when the car begins to move.

Since most cars are equipped with GPS receivers, each car can easily detect its movement and thus start/stop its counter accordingly. With the above improvement, a car is able to collect air quality from more different positions in a grid. Moreover, because all cars in a grid have the same sampling rate, the number of sensing data reported by a car is inversely proportional to its velocity. In particular, a slow moving car can sample more data, while a fast moving car just samples few data. To balance the amount of sensing data reported by each car in the same grid, we propose another improvement:

Improvement 2: Each car decreases its counter only when keeps moving. Instead of constantly decreasing the counter, we multiply the decreasing speed by a scaling factor v_j/v_{avg}, where v_j is the car's velocity and v_{avg} is the average (or expected) velocity of cars in a grid.

Fig. 6 shows *Question 2*, where two cars X and Y enter grid g_1 at near time and move along the same road. As they share the same rate $\hat{r}_1$, their counters are set equally. Both X and

Y may sample data on close positions, so their sensing data would exhibit high correlation [4], [5]. Moreover, it is redundant to sample multiple data on a position simultaneously, as one report is enough to compute the corresponding AQI. To solve the question, we set a *random backoff time* for each car whenever it moves into a new grid g_i by:

$$t_B = f_{\mathbf{T}}\left(\mathrm{rand}(0,1)/\hat{r}_i\right), \tag{12}$$

where $\mathrm{rand}(0,1)$ is a random value in $[0,1]$. The first time to set the car's counter becomes $f_{\mathbf{T}}(1/\hat{r}_i)+t_B$, and the counter is reset to $f_{\mathbf{T}}(1/\hat{r}_i)$ when countdown completes. Grid g_2 in Fig. 6 gives an example. With different t_B values, cars X and Y can sample data on different positions in g_2 even if they move closely. It helps the remote server obtain sensing data from more positions in a grid. By Eq. (12), a car defers its sampling for at most $f_{\mathbf{T}}(1/\hat{r}_i)$ time when moving into a grid, so the car will miss no more than one report in each grid.

Question 3 occurs when a street contains volumes of car traffic. Though we set $F(g_i) = F_{\max}$ in Eq. (10) to handle the case, a large number of cars may still sample data in a small area. It would cause network congestion and waste wireless bandwidth. To conquer this question, we use a probability p_S to help each car decide whether to sample data when its counter reaches to zero:

$$p_S = \begin{cases} F_{\max}/F(g_i) & \text{if } F(g_i) > F_{\max} \\ 1 & \text{otherwise.} \end{cases} \tag{13}$$

In Eq. (13), we use probabilistic sampling only when car traffic exceeds the threshold $F_{\max}$. In other (normal) situations, each car still keeps regular data sampling by setting its probability to one. Remark 6 gives a comment on the design of our probabilistic reporting method.

Remark 6 (Rationale of probabilistic reporting)**.** *The objective of probabilistic reporting is to make cars sample data from more pixels in a grid. As we will discuss in Section 5.3, the remote server requires the collected data to derive AQIs of the pixels without reports. If we simply let some cars sample data on close or the same pixels, even though their sensor readings are different, it cannot improve the accuracy of estimating lost data. Instead, it only causes wastage in communication, as one report is enough to compute the pixel's AQI. Thus, the probabilistic reporting method asks cars in the same grid to sample data on different pixels by counter decrement and random backoff. Moreover, it adopts a probability idea to reduce the amount of data transmission in a small area where traffic jam occurs, thereby alleviating network congestion.* □

5.2.2 Packet Formats in EDGE

We adopt the Internet protocol (IP) to transmit packets, and skip the IP header in our discussion. When the remote server decides the grid partition and data sampling rates, it asks each BS to broadcast a *beacon* with the following payload:

$$(\mathrm{Type}=0, \mathrm{RP1}, \mathrm{RP2}, N_G, \mathrm{GID}_1, \hat{r}_1, \cdots, \mathrm{GID}_{N_G}, \hat{r}_{N_G}),$$

where Type is an indicator to depict the functionality; RP1 and RP2 are *reference points* respectively located on $\mathcal{A}$'s left-down and right-up positions, as Fig. 6 shows. We use the national marine electronics association (NMEA) format [43] to describe the latitude and longitude of a position. As the grid partition may not exactly fit to a BS's coverage range, we use field N_G to record the number of grids covered by each BS. In the beacon, GID_k and $\hat{r}_k$ denote the *grid ID* and *sampling rate* of the kth grid covered by a BS, respectively. Fig. 6 gives two examples, where BS1 sends a beacon '(0, 02228.5000N, 12010.3200E, 02302.8030N, 12085.8030E, 1, 4, 20)'. Thus, the positions of $\mathcal{A}$'s reference points are (N22°28′50″, E120°10′32″) and (N23°2′80.3″, E120°85′80.3″), and BS1 covers grid g_4 with a sampling rate of 20 samples/h. BS2 sends a beacon '(0, 02228.5000N, 12010.3200E, 02302.8030N, 12085.8030E, 3, 2, 25, 17, 30, 18, 20)'. So, it covers grids g_2, g_{17}, and g_{18}, whose sampling rates are 25, 30, and 20 samples/h, respectively.

Thanks to the region quadtree, we need not record much positioning information in beacons. Instead, we only indicate the positions of two reference points in $\mathcal{A}$, and the grid partition can be quickly obtained by using grid IDs of the region quadtree. It thus greatly simplifies the beacon format. Also, by changing the two reference points in beacons, we can easily change the monitoring region $\mathcal{A}$ without much effort. Notice that when a car misses the beacon, it sends a *probing packet* with the payload of '$(\mathrm{Type}=1)$' to its BS. Then, the BS will unicast the beacon to that car.

To save data transmission, we define two kinds of reports for cars. Specifically, a *standard report* has the following format:

$$(\mathrm{Type}=2, \mathrm{Time}, \mathrm{Reading}, \mathrm{Latitude}, \mathrm{Longitude}),$$

where the Time, $\mathrm{Reading}$, $\mathrm{Latitude}$, and $\mathrm{Longitude}$ fields have lengths of 32, 16, 88, and 88 bits, respectively. Besides, the format of a *differential report* is given as follows:

$$(\mathrm{Type}=3, D_{\mathrm{Time}}, D_{\mathrm{Reading}}, D_{\mathrm{Latitude}}, D_{\mathrm{Longitude}}),$$

where D_{Time}, D_{Reading}, D_{Latitude}, and $D_{\mathrm{Longitude}}$ are the differences of time, sensing data, latitude, and longitude between the current and previous reports, respectively. Their length are 9, 8, 20, and 20 bits[3], so a differential report further saves 167 bits. We give an example, where a car first sends a standard report '(2, 184013, 800, 02478.8722N, 12099.8483E)'. It means that the car samples a value of 800 on 18:40:13 at the position (N24°78′87.22″, E120°99′84.83″). Then, the car sends a differential report '(3, 32, -10, 33596, -2559)', which indicates that it samples a value of 790 on 18:40:45 at the new position (N24°82′23.18″, E120°99′59.24″).

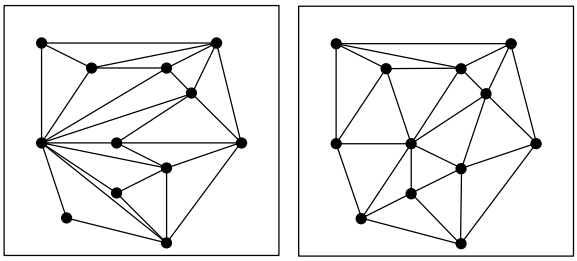
There are four cases that a car should send a standard report: 1) a new monitoring period starts, 2) the car enters a new grid, 3) it handovers to another BS, and 4) the difference between the current and previous sensing values is out of range $[-128, 127]$ (due to the length of D_{Reading}). Here, case 4) implies that the car locates in an area where the pollutant concentration significantly varies, so it is better to report the complete information. Except for these cases, the car can send differential reports to save the amount of data transmission.

5.3 Estimating Phase

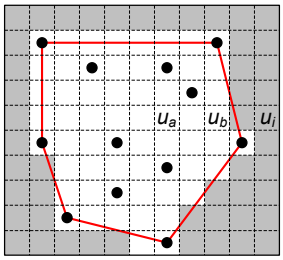
After collecting data from cars, the remote server uses the scheme in Section 2.1 to compute AQI for each sampling position. If multiple cars have measured air quality on the same position, the server takes the report with the largest timestamp. However, since it is not possible to control the movement of cars (referring to Remark 1), we may not collect data for every pixel of $\mathcal{U}$ in the gathering phase. Therefore, we need to estimate lost data in the final phase.

Let $\mathcal{U}_{\mathrm{data}}$ and $\mathcal{U}_{\mathrm{loss}}$ be the sets of pixels with and without sensing data, respectively, so $\mathcal{U}_{\mathrm{data}} \cup \mathcal{U}_{\mathrm{loss}} = \mathcal{U}$ and $\mathcal{U}_{\mathrm{data}} \cap \mathcal{U}_{\mathrm{loss}} = \emptyset$. Our idea is to do triangulation by using

3. D_{Reading}, D_{Latitude}, and $D_{\mathrm{Longitude}}$ each contains a sign bit.



(a) simple triangulation (b) Delaunay triangulation



(c) $\mathcal{U}^I_{\text{loss}}$ and $\mathcal{U}^O_{\text{loss}}$

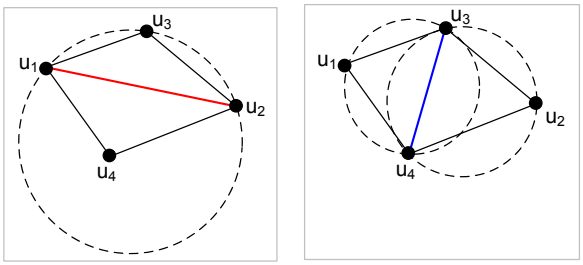
Fig. 7: Estimating the lost data by triangulation.

all pixels in $\mathcal{U}_{\text{data}}$, and then estimate the data of each pixel in $\mathcal{U}_{\text{loss}}$ based on the triangulation. However, if we use a *simple triangulation* on $\mathcal{U}_{\text{data}}$, it may result in many 'skinny' triangles. In this case, the average distance between any point inside a skinny triangle to each triangle vertex will become longer, which increases the inaccuracy of estimation. Thus, we suggest using the *Delaunay triangulation* on $\mathcal{U}_{\text{data}}$, whose definition is that no pixel in $\mathcal{U}_{\text{data}}$ will locate inside the circumcircle of any forming triangle. The Delaunay triangulation is able to maximize the smallest angle of all the angles of the forming triangles, so it can avoid generating many skinny triangles. Fig. 7(a) and (b) give an example, where dots indicate the pixels in $\mathcal{U}_{\text{data}}$.

To find the Delaunay triangulation, we borrow the concept from [44]. Given a simple triangulation on $\mathcal{U}_{\text{data}}$, we check if every forming triangle meets the definition of Delaunay triangulation. Let us denote by $\mathcal{E}$ the set of edges found by the simple triangulation. For each edge, say, $\overline{u_iu_j}$ in $\mathcal{E}$, we examine each of the four vertices of the two triangles sharing $\overline{u_iu_j}$, and check whether it locates inside the circumcircle of any triangle. If so, it means that $\overline{u_iu_j}$ is illegal and we exchange $\overline{u_iu_j}$ by the 'opposite' edge in $\mathcal{E}$. Fig. 8 presents an example, where we check edge $\overline{u_1u_2}$. Since vertex u_4 locates inside the circumcircle of triangle $\triangle u_1u_2u_3$, $\overline{u_1u_2}$ is illegal. In this case, we replace it by the opposite edge $\overline{u_3u_4}$. Thus, the four vertices u_1, u_2, u_3, and u_4 all locate on the peripheries of circumcircles. We can iteratively check (and possibly exchange) every edge in $\mathcal{E}$, and the final result will be the Delaunay triangulation.

After doing triangulation, we further divide $\mathcal{U}_{\text{loss}}$ into two non-overlapped subsets $\mathcal{U}^I_{\text{loss}}$ and $\mathcal{U}^O_{\text{loss}}$. For each pixel u_i in $\mathcal{U}_{\text{loss}}$, if at least one half of u_i is covered by the polygon formed by the Delaunay triangulation, then u_i belongs to $\mathcal{U}^I_{\text{loss}}$. Otherwise, u_i belongs to $\mathcal{U}^O_{\text{loss}}$. Fig. 7(c) shows an example, where each small square denotes a pixel, and the pixels with dots belong to $\mathcal{U}_{\text{data}}$. In Fig. 7(c), the pixels in $\mathcal{U}^I_{\text{loss}}$ and $\mathcal{U}^O_{\text{loss}}$ are colored by white and gray, respectively.

Then, we apply the technique of *triangulated irregular network* [45] to estimate AQI of each pixel in $\mathcal{U}^I_{\text{loss}}$. Specifically, let us consider a pixel $u_i \in \mathcal{U}^I_{\text{loss}}$ whose coordinates are (x_i, y_i).



(a) (b)

Fig. 8: Finding the Delaunay triangulation: (a) checking edge $\overline{u_1u_2}$ and (b) exchanging $\overline{u_1u_2}$ by the new edge $\overline{u_3u_4}$.

Here, we take the central point in a pixel to represent its coordinates. Suppose that u_i locates inside a triangle with three vertices (i.e., pixels) u_a, u_b, and u_c whose coordinates are (x_a, y_a), (x_b, y_b), and (x_c, y_c), respectively. In addition, the AQIs of u_a, u_b, and u_c are w_a, w_b, and w_c, respectively. Then, we compute u_i's AQI by

$$w_i = (-\alpha x_i - \beta y_i - \Phi)/\gamma, \tag{14}$$

where

$$\alpha = y_a(w_b - w_c) + y_b(w_c - w_a) + y_c(w_a - w_b), \tag{15}$$
$$\beta = w_a(x_b - x_c) + w_b(x_c - x_a) + w_c(x_a - x_b), \tag{16}$$
$$\gamma = x_a(y_b - y_c) + x_b(y_c - y_a) + x_c(y_a - y_b), \tag{17}$$
$$\Phi = (-\alpha x_a - \beta y_a)/(1 + w_a). \tag{18}$$

In this way, we can derive the AQI of each white pixel in Fig. 7(c). Then, we update $\mathcal{U}_{\text{data}}$ by $\mathcal{U}_{\text{data}} \cup \mathcal{U}^I_{\text{loss}}$.

On the other hand, we use *linear extrapolation* to infer the AQI of a pixel $u_i \in \mathcal{U}^O_{loss}$. For each such u_i, we take two pixels u_a and u_b from $\mathcal{U}_{\text{data}}$ for reference, such that u_a, u_b, and u_i all locate in the same row (or column). Suppose that u_b is closer to u_i than u_a. Then, u_i's AQI is calculated by

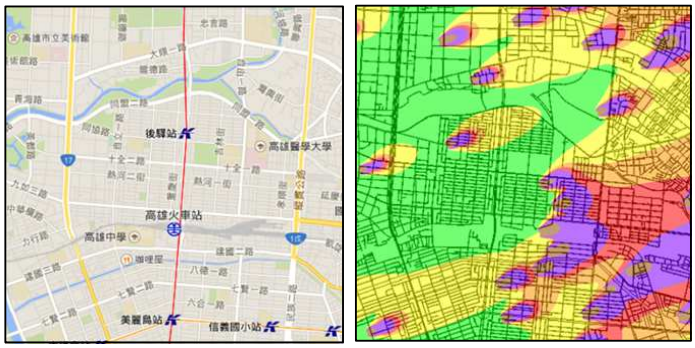
$$w_i = w_a + (w_b - w_a) \times \frac{\mathbf{D}(u_i, u_a)}{\mathbf{D}(u_a, u_b)}, \tag{19}$$

where $\mathbf{D}(\cdot, \cdot)$ is the distance between two pixels. Fig. 7(c) gives an example, where u_a, u_b, and u_i locate in the same row. Then, we move u_i from $\mathcal{U}^O_{loss}$ to $\mathcal{U}_{\text{data}}$. The above iteration is repeated until $\mathcal{U}^O_{loss}$ becomes empty.

5.4 Discussion

We then discuss two issues. First, one may argue that cars also contribute air pollution. However, as mentioned in Remark 1, cars move merely depending on their destinations, instead of exterior influence (e.g., rewards given to drivers). In other words, our EDGE mechanism works based on 'original' car traffic, so it will not ask cars to move to certain positions and thus generate 'extra' air pollution. Moreover, air pollution generated by cars should be viewed as an impartible part of environmental air pollution. Even if monitoring stations or static WSNs are used, these cars still generate the same amount of air pollution. Thus, it is unnecessary to distinguish pollutants caused by cars from environmental pollutants.

The next issue is data retransmission due to network congestion or temporal failure. However, EDGE works on the application layer and data retransmission is the job of underlying layers (e.g., TCP). Specifically, if a car finds that it fails to send reports to the associated BS, the car can keep reports in the



(a) (b)

Fig. 9: The monitoring region $\mathcal{A}$ in our simulator: (a) street map of downtown Kaohsiung and (b) dispersion model of CO pollutant generated by ISC3.

TABLE 2: Simulation parameters.

Cars' parameters:	
region length	2, 3, 4, 5, 6 km
number of cars	100, 300, 500, 700, 900, 1100
speed limit	10, 30, 50, 70, 90, 120 km/h
braking speed	$4.2 \sim 4.8$ m/s
acceleration	$2.0 \sim 2.4$ m/s
air pollution	HBEFA (version 3.1)
Pollutant's parameters:	
gaseous pollutant	CO
pixel size	1 m $\times$ 1 m
number of sources	25, 50, 75, 100
duration of emission	$2 \sim 4$ hours
amount of emission	$250 \sim 750$ grams
Other parameters:	
monitoring period	600 seconds
initial grid partition	16×16
monetary reward	\$0.05, \$1.0, \$0.15

buffer and send them out later (maybe to another BS). This is done by underlying layers, not EDGE itself. Moreover, based on the time structure of EDGE in Fig. 2, the report is valid if it can be received by the remote server during a monitoring period (even though the report is retransmitted several times). Besides, the monitoring period (e.g., 10 minutes in simulations) is much longer than the retransmission duration of underlying layers (e.g., 75 seconds in TCP). In other words, data retransmission would have litter effect on the error rate. Thus, we do not deal with data retransmission in EDGE but leave it to underlying-layer protocols.

6 Experimental Results

To evaluate EDGE's performance, we develop a simulator with the models discussed in Section 2. It selects the downtown area of Kaohsiung city[4] in Fig. 9(a) as the monitoring region $\mathcal{A}$, where we use VSN to collect air quality and exploit AQI in Section 2.1 to measure air pollution. As mentioned in Section 2.2, SUMO is used to generate car traffic in $\mathcal{A}$. Each street has a speed limit, which decides the velocity of cars moving along it. If a car reaches its destination or moves out of $\mathcal{A}$, it stops collecting air quality. Since new cars will move into $\mathcal{A}$, we can keep the average number of cars. Also, by using the HBEFA vehicle emission class, each car generates different amount of air pollution. Moreover, we adopt ISC3 in Section 2.3 to model the dispersion of CO pollutant in the environment. Specifically, we pick a number of source locations in $\mathcal{A}$ to emit CO gas (e.g., they may be the sites of factories). Each source emits various amount of CO gas with a different duration. Besides, the source locations would change as time goes by. Fig. 9(b) gives a snapshot of CO dispersion by ISC3, where colors represent different concentration of CO gas. Table 2 summarizes the parameters used in our simulator.

We compare EDGE with VAR and GRA schemes in [34], which rely on Eq. (4) to compute the sampling rate in each grid. They work on the premise that grid partition is fixed, and use linear interpolation/extrapolation to estimate lost data. We divide $\mathcal{A}$ into 16×16 grids for both schemes. EDGE also starts from the same partition, but it dynamically amends the partition via a region quadtree whose maximum depth is 4. In EDGE, we set $\lambda = 5$, $F_{\min} = 5$ and $F_{\max} = 20$ for car traffic, and $\sigma = 0.834$ as discussed in Remark 5.

4. Kaohsiung is the second largest city in Taiwan and has the population of more than 2.7 millions.

According to Section 4, three metrics are used to measure average performance of each scheme in a monitoring period.

Data transmission: We evaluate the amount of sensing data sent from cars, which is denoted by '(sample)' in simulation figures. Moreover, we measure the amount of total data transmitted in the network (including control messages broadcasted by each BS), which is denoted by '(total)' in figures.

Monetary cost: Two scenarios are considered. Each driver is given \$0.1 for a report in the *constant reward (CR)* scenario. The *variable reward (VR)* scenario classifies AQI in Fig. 1(a) into three groups: *harmless* (AQI ≤ 100, good and moderate levels), *normal* (AQI: 101–200, unhealthy sensitive group and unhealthful levels), and *critical* (AQI ≥ 201, very unhealthful and hazardous levels). A driver is given \$0.05, \$0.1, and \$0.15 for each harmless, normal, and critical report, respectively. Remark 7 comments on both scenarios.

Error rate: Since VAR, GRA, and EDGE contain gathering and estimating phases, we observe the impact of each phase. For the gathering phase (denoted by '(G)' in figures), if a pixel has no car to report sensing data, its value is *null*. Thus, the error rate will be the ratio of pixels with null values. For the estimating phase (denoted by '(E)' in figures), the server infers the sensing data of those pixels whose values are null. We set $\delta = 5$ and use Eq. (7) to find the error rate.

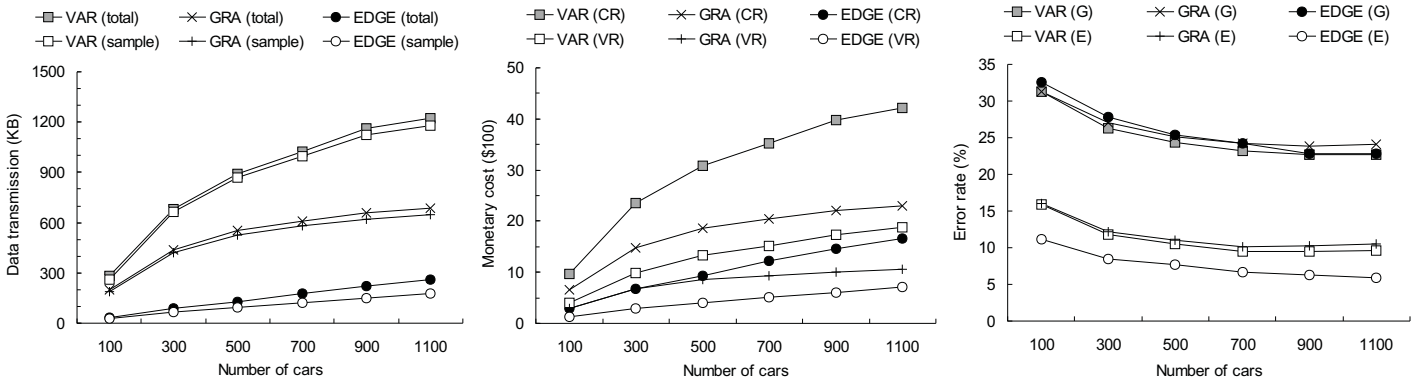
We also vary the simulation parameters from three aspects.

Car: We adjust *car traffic* in a monitoring region to observe the effect of car density. It helps measure system performance in different types of regions (e.g., suburb or downtown). Besides, we alter the *speed limit* to check if car speeds will affect performance. The result can be also used in various applications like monitoring in congested roads or highways.

Environment: We vary the *regional size* to evaluate different methods in small- and large-scale environments. This aspect is useful because the monitoring mission may be conducted in only small regions (e.g., one district of the city) or the whole metropolitan area.

Pollutant: By changing the number of *pollutant sources*, we can measure whether each method is able to save both data transmission and monetary cost when the pollutant concentration keeps relatively steady, and lower down the error rate as concentration changes drastically.

Remark 7 (CR and VR scenarios)**.** *In Eq. (6), the monetary cost is decided by the number of packets in $\mathcal{P}$ and reward(p_i). In the CR scenario, minimizing the monetary cost may not be necessarily equal to minimizing the amount of data transmission due to the design of different reporting formats in Section 5.2.2. We consider*



(a) data transmission (b) monetary cost (c) error rate

Fig. 10: Effect of car traffic on average performance of each scheme.

an example where two methods make cars send the equal number of reports in $\mathcal{P}$. One method only uses standard reports, while the other lets cars frequently send differential reports. In this case, both methods have the same amount of data transmission but they result in different monetary costs. Consequently, we use the CR scenario to demonstrate the benefit of using differential reports.

The VR scenario exhibits the concept of 'differentiating' sensing data. As mentioned in Section 2.1, the government uses AQI to describe the degree of air pollution. It will have more interesting in the positions with large AQI values, as there may exist abnormal events (e.g., gas leak). Thus, we prefer giving the drivers who find such positions (i.e., by sending critical reports) a higher reward. On the other hand, by giving the drivers who send harmless reports a lower reward, we can decrease the overall monetary cost when the air pollination is light. Thus, the VR scenario also gives an example of adjusting reward(p_i) to save the monetary cost. □

6.1 Effect of Car Traffic

We first evaluate the effect of car traffic on different schemes, where the number of cars in $\mathcal{A}$ ranges from 100 to 1100. Fig. 10(a) gives the amount of data transmission, where sensing data reported from cars dominate all transmission in VSN. Since EDGE keeps dynamic grid partition, it requires slightly more control messages (i.e., beacons) to announce the grid structure. It is shown in [34] that GRA greatly reduces the amount of data reports than VAR. However, both schemes do not consider the extreme case where a street is congested by many cars. Thus, their amount of data transmission significantly increases as the number of cars grows. On the contrary, EDGE not only uses an upper threshold $F_{\max}$ to limit car traffic $F(g_i)$ in Eq. (10), but also proposes a probabilistic sampling method by Eq. (13) to deal with this case. Therefore, it avoids drastically increasing data transmission when there are more cars in $\mathcal{A}$. Averagely, EDGE reduces 83.7% and 73.1% of data transmission than VAR and GRA, respectively. Also, it reduces 88.2% and 80.1% of sensing reports comparing with VAR and GRA, respectively.

Fig. 10(b) shows the monetary cost, where all schemes have higher costs in the CR scenario. The reason is that a driver is given with the same reward (i.e., \$0.1) for every report, while it is possible to give a driver less reward (i.e., \$0.05) when the report falls into the harmless degree. From Fig. 10(a), because VAR generates the largest amount of data transmission, it requires the highest monetary cost. Thanks to the dynamic grid partition, EDGE can result in the lowest monetary cost. On the average, EDGE respectively saves 66.6% and 43.5% of the monetary cost than VAR and GRA under the CR scenario. Besides, under the VR scenario, it saves averagely 67.0% and 46.9% of the monetary cost than VAR and GRA, respectively.

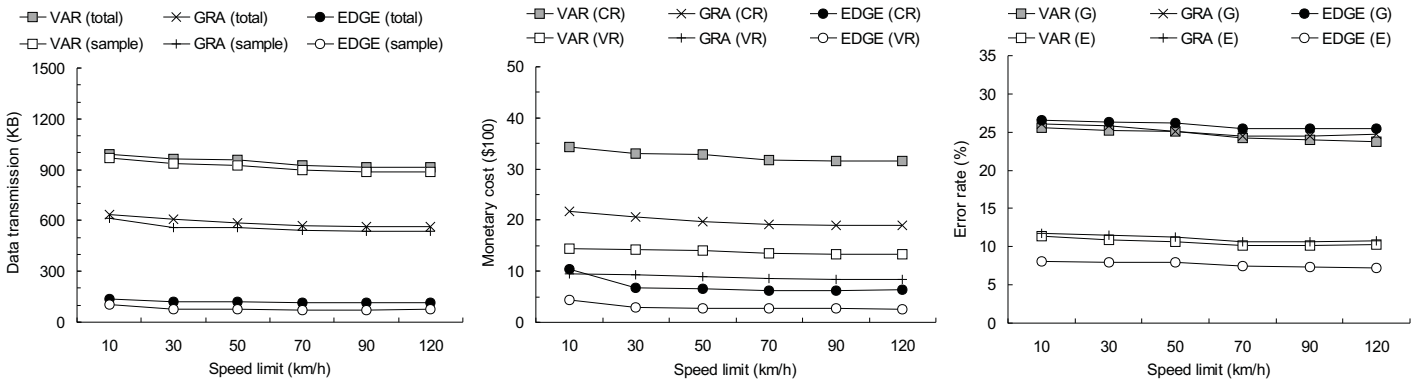
Fig. 10(c) presents the error rate. Although more cars result in more air pollution, they also help collect more data from $\mathcal{A}$. Thus, all schemes have more pixel values to infer lost data, so their error rates decrease accordingly. When we consider only the gathering phase, the average error rate of each scheme is close (i.e., VAR $\approx$ 25.1%, GRA $\approx$ 25.9%, and EDGE $\approx$ 25.9%)[5]. Through probabilistic reporting, EDGE prevents cars from sampling data on too close pixels. Thus, even if cars generate fewer reports in EDGE, it still has a similar error rate with others. This demonstrates the benefit of using probabilistic reporting to let cars collect data on diverse positions. Moreover, when we apply the estimating phase, the error rate obviously reduces. Since EDGE uses sophisticated methods including the Delaunay triangulation and triangulated irregular network to infer lost data, as compared with simple interpolation and extrapolation in both VAR and GRA, it thus has the lowest error rate. In particular, EDGE improves 31.1% and 34.6% of monitoring accuracy than VAR and GRA when $\delta = 5$, respectively.

6.2 Effect of Speed Limit

We then measure the effect of different speed limits (from 10 to 120 km/h), where the results are shown in Fig. 11. The speed limit decides the maximum velocity of cars moving along a street, but cars may not keep such a velocity due to traffic lights or road conditions (e.g., cars slow down as traffic congests). When cars move pretty slowly (i.e., $\leq$ 30 km/h), they might generate more pollution. Thus, both data transmission and monetary cost become slightly larger when the speed limit is very low. With flexible grid structure and probabilistic reporting, EDGE greatly reduces the amount of data transmission and monetary cost. Specifically, it saves 87.2%, 91.3%, 78.3%, and 78.3% of data transmission, sensing reports, CR monetary cost, and VR monetary cost than VAR, respectively. Comparing to GRA, EDGE reduces 79.4%, 85.8%, 64.5%, and 66.2% of data transmission, sensing reports, CR monetary cost, and VR monetary cost, respectively.

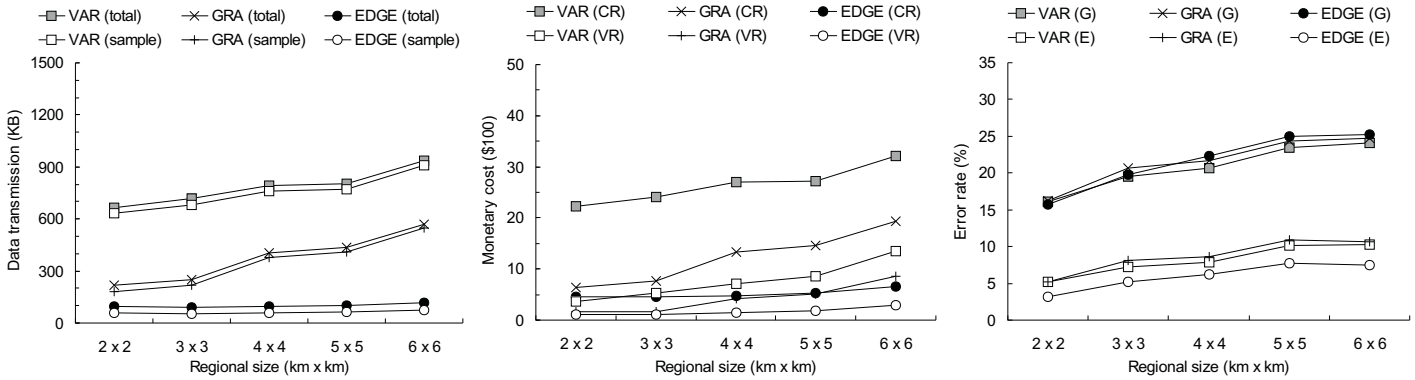
Similarly, when cars move in a low speed, the error rate lightly increases (due to more pollutant emitted by cars), as

5. In Fig. 9(b), there are around 20% of pixels that cars cannot pass through. These pixels may contain buildings, parks, or railways. That is why all methods encounter high error rates when they do not use the estimating phase.



(a) data transmission (b) monetary cost (c) error rate

Fig. 11: Effect of speed limit on average performance of each scheme.



(a) data transmission (b) monetary cost (c) error rate

Fig. 12: Effect of regional size on average performance of each scheme.

Fig. 11(c) shows. Since the number of cars is fixed to 500, EDGE results in slightly more pixels with null values than VAR and GRA when only the gathering phase is adopted. However, by applying the estimating phase (with $\delta = 5$), EDGE can significantly decrease the error rate. On the average, EDGE improves 27.5% and 30.8% of monitoring accuracy comparing with VAR and GRA, respectively.

6.3 Effect of Regional Size

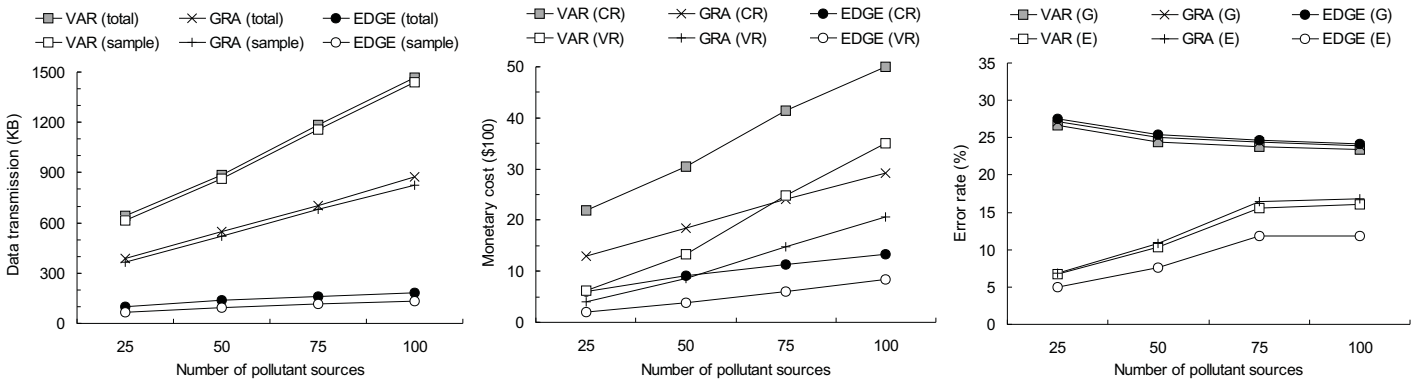
Next, we investigate the effect of $\mathcal{A}$'s size on performance. We generate car traffic and pollutant dispersion on a $6 \times 6\,\text{km}^2$ area, and select its central part to be $\mathcal{A}$, where the size of $\mathcal{A}$ increases from 2×2 to $6 \times 6\,\text{km}^2$. Fig. 12(a) and (b) show the communication cost. Apparently, the remote server receives more sensing reports from cars in a larger region, and thus requires a higher monetary cost. On the average, EDGE reduces 87.0%/91.6% and 70.1%/79.4% of the transmission amount of total/sensing data than VAR and GRA, respectively. Besides, it saves 80.6%/77.0% and 52.4%/50.6% of the monetary cost than VAR and GRA under the CR/VR scenario, respectively.

Fig. 12(c) gives the error rates of VAR, GRA, and EDGE. Since the conditions of cars (i.e., density, speed limit, air pollution emitted) do not change, enlarging $\mathcal{A}$ will also increase the error rate. Such a phenomenon is more obvious when we merely use the gathering phase. In this case, EDGE has a slightly larger error rate than others. With the help of estimating phase, EDGE is able to reverse the situation. Specifically, it further enhances 27.6% and 32.3% of monitoring accuracy than VAR and GRA when $\delta = 5$, respectively.

6.4 Effect of Pollutant Source

Finally, we study the effect of pollutant source, where the number of sources in $\mathcal{A}$ is set to 25, 50, 75, and 100. In this experiment, the number of cars is kept in 500 and the speed limit is 50 km/h, so the amount of pollution generated by cars does not change. Thus, the overall air quality is decided by the number of pollutant sources, and Fig. 13 gives the simulation result. Both VAR and GRA observe the difference among the values of sensing data in a grid, and compute sampling rates by Eq. (4). However, since the grid size is fixed, they enlarge the sampling rate in a grid when the number of sources grows. Thus, not only the amount of data transmission but also the monetary cost will substantially increase. In contrast, EDGE uses heterogeneous grids to fine tune sampling rates, so it increases the sampling rate only when required. Thus, its data transmission and monetary cost can smoothly increase when there are more sources in $\mathcal{A}$. On the average, EDGE saves 85.9%, 89.7%, 72.2%, and 72.9% of data transmission, sensing reports, CR monetary cost, and VR monetary cost than VAR, respectively. Besides, comparing to GRA, EDGE reduces 76.7%, 82.5%, 52.9%, and 55.9% of data transmission, sensing reports, CR monetary cost, and VR monetary cost, respectively.

From Fig. 13(c), we observe that the error rate smoothly decreases as the number of pollutant sources increases, when each scheme has only the gathering phase. The reason is that the monitoring region $\mathcal{A}$ and cars' conditions (e.g., number and speed) do not change. When there are more sources, all schemes will prefer increasing the sampling rates of cars. In this case, more pixels could be monitored by cars, thereby reducing the error rate. On the other hand, when the estimat-



(a) data transmission (b) monetary cost (c) error rate

Fig. 13: Effect of pollutant source on average performance of each scheme.

ing phase is conducted, all schemes can greatly decrease their error rates. However, the error rate increases as the number of sources grows. It is because the pollutant concentration varies drastically, making the estimation become more inaccurate. Average speaking, EDGE can reduce 25.8% and 28.9% of the error rate than VAR and GRA, respectively, when $\delta = 5$.

7 Conclusion and Future Work

Using VSN to monitor metropolitan air quality has received considerable attention. This paper develops a 3-phase EDGE mechanism to efficiently gather sensing data from cars and estimate absent data. The goal is to balance between communication overhead and monitoring accuracy. EDGE proposes dynamic grid partition and probabilistic reporting methods to adjust sampling rates and avoid redundant data, respectively. Different formats of sensing reports are proposed to save data transmission. With the Delaunay triangulation, EDGE can infer lost data more precisely. By using simulations built on SUMO and ISC3 models and a real street map, we compare EDGE with VAR and GRA schemes. Experimental results show that EDGE reduces 79–92% of sensing reports and saves 44–81% of monetary cost. When only the gathering phase is used, EDGE has a similar error rate with VAR and GRA. However, by applying the estimating phase, EDGE further improves 26–35% of monitoring accuracy.

In Remark 2, we leave an issue of designing the rewarding policy for future study. When the reward given to a driver is considerable, the rewarding policy may influence driving behavior, and could facilitate air-quality monitoring by VSN. For example, we can give a higher reward to encourage drivers to collect data on locations where pollutant concentration changes drastically, while lower the reward when cars report sensing data with little variation. However, we should deal with the case where numerous drivers move to the same area in order to increase the reward, which will cause traffic congestion in that area. Thus, it also deserves further investigation to conduct real experiments to measure the effect of rewarding policy for such scenarios.

References

[1] World Health Organization (WHO), "7 million premature deaths annually linked to air pollution," Mar. 2014. [Online]. Available: http://www.who.int/mediacentre/news/releases/2014/air-pollution/en/

[2] B. Zou, J.G. Wilson, F.B. Zhan, and Y.N. Zeng, "Air pollution exposure assessment methods utilized in epidemiological studies," *Journal of Environmental Monitoring*, vol. 11, no. 3, pp. 475–490, 2009.

[3] Y.C. Wang, "Mobile sensor networks: system hardware and dispatch software," *ACM Computing Surveys*, vol. 47, no. 1, pp. 12:1–12:36, 2014.

[4] Y.C. Wang, Y.Y. Hsieh, and Y.C. Tseng, "Multiresolution spatial and temporal coding in a wireless sensor network for long-term monitoring applications," *IEEE Trans. Computers*, vol. 58, no. 6, pp. 827–838, 2009.

[5] P. Wang and I.F. Akyildiz, "Spatial correlation and mobility-aware traffic modeling for wireless sensor networks," *IEEE/ACM Trans. Networking*, vol. 19, no. 6, pp. 1860–1873, 2011.

[6] F. Aurenhammer, R. Klein, and D.T. Lee, *Voronoi Diagrams and Delaunay Triangulations*. Singapore: World Scientific Publishing, 2013.

[7] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO – Simulation of Urban MObility," *Int'l Journal on Advances in Systems and Measurements*, vol. 5, no. 3 & 4, pp. 128–138, 2012.

[8] U.S. Environmental Protection Agency, *Industrial Source Complex (ISC) Dispersion Model User's Guide*. Columbus: BiblioGov Publishing, 2013.

[9] U.S. Environmental Protection Agency, "Technical assistance document for the reporting of daily air quality – the air quality index (AQI)," Dec. 2013. [Online]. Available: http://www3.epa.gov/airnow/aqi-technical-assistance-document-dec2013.pdf

[10] L. Bedogni, M. Gramaglia, A. Vesco, M. Fiore, J. Harri, and F. Ferrero, "The Bologna ringway dataset: improving road network conversion in SUMO and validating urban mobility via navigation services," *IEEE Trans. Vehicular Technology*, vol. 64, no. 12, pp. 5464–5476, 2015.

[11] OpenStreetMap. [Online]. Available: https://www.openstreetmap.org/

[12] HBEFA. [Online]. Available: http://www.hbefa.net/e/index.html

[13] U.S. Environmental Protection Agency, "Air quality modeling." [Online]. Available: https://www3.epa.gov/airquality/modeling.html

[14] J.I. Huertas, M.E. Huertas, S. Izquierdo, and E.D. Gonzalez, "Air quality impact assessment of multiple open pit coal mines in northern Colombia," *Journal of Environmental Management*, vol. 93, no. 1, pp. 121–129, 2012.

[15] S. Gulia, S.M.S. Nagendra, and M. Khare, "Performance evaluation of ISCST3, ADMS-urban and AERMOD for urban air quality management in a mega city of India," *Int'l Journal of Sustainable Development and Planning*, vol. 9, no. 6, pp. 778–793, 2014.

[16] S. Vardoulakisa, B.E.A. Fisherb, K. Pericleousa, and N. Gonzalez-Flescac, "Modelling air quality in street canyons: a review," *Atmospheric Environment*, vol. 37, no. 2, pp. 155–182, 2003.

[17] Y. Zheng, F. Liu, and H.P. Hsieh, "U-Air: when urban air quality inference meets big data," *Proc. ACM SIGKDD Int'l Confe. Knowledge Discovery and Data Mining*, 2013, pp. 1436–1444.

[18] W. Tsujitaa, A. Yoshinoa, H. Ishidab, and T. Moriizumi, "Gas sensor network for air-pollution monitoring," *Sensors and Actuators B: Chemical*, vol. 110, no. 2, pp. 304–311, 2005.

[19] C.H. Wang, Y.K. Huang, X.Y. Zheng, T.S. Lin, C.L. Chuang, and J. A. Jiang, "A self sustainable air quality monitoring system using WSN," *Proc. IEEE Int'l Conf. Service-Oriented Computing and Applications*, 2012, pp. 1–6.

[20] M. Penza, D. Suriano, M.G. Villani, L. Spinelle, and M. Gerboles, "Towards air quality indices in smart cities by calibrated low-cost

sensors applied to networks," *Proc. IEEE Conf. Sensors*, 2014, pp. 2012–2017.
[21] European Parliament, "Directive 2008/50/EC of the European Parliament and of the Council of 21 May 2008 on ambient air quality and cleaner air for Europe," May 2008. [Online]. Available: http://ec.europa.eu/environment/air/quality/legislation/existing_leg.htm
[22] S. Brienza, A. Galli, G. Anastasi, and P. Bruschi, "A cooperative sensing system for air quality monitoring in urban areas," *Proc. Int'l Conf. Smart Computing*, 2014, pp. 15–20.
[23] Y.C. Wang, C.C. Hu, and Y.C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network," *IEEE Trans. Mobile Computing*, vol. 7, no. 2, pp. 262–274, 2008.
[24] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research," *Proc. ACM Int'l Conf. Mobile Systems, Applications, and Services*, 2009, pp. 55–68.
[25] N. Nikzad, N. Verma, C. Ziftci, E. Bales, N. Quick, P. Zappi, K. Patrick, S. Dasgupta, I. Krueger, T. S. Rosing, and W. G. Griswold, "CitiSense: improving geospatial environmental assessment of air quality using a wireless personal exposure monitoring system," *Proc. ACM Conf. Wireless Health*, 2012, pp. 1–8.
[26] D. Hasenfratz, O. Saukh, S. Sturzenegger, and L. Thiele, "Participatory air pollution monitoring using smartphones," *Proc. Int'l Workshop on Mobile Sensing*, 2012, pp. 1–5.
[27] Y. Cheng, X. Li, Z. Li, S. Jiang, Y. Li, J. Jia, and X. Jiang, "AirCloud: a cloud-based air-quality monitoring system for everyone," *Proc. ACM Conf. Embedded Network Sensor Systems*, 2014, pp. 251–265.
[28] Y. Ma, M. Richards, M. Ghanem, Y. Guo, and J. Hassard, "Air pollution monitoring and mining based on sensor grid in London," *Sensors*, vol. 8, no. 6, pp. 3601–3623, 2008.
[29] S.C. Hu, Y.C. Wang, C.Y. Huang, and Y.C. Tseng, "A vehicular wireless sensor network for CO_2 monitoring," *Proc. IEEE Conf. Sensors*, 2009, pp. 1498–1501.
[30] V. Sivaraman, J. Carrapetta, K. Hu, and B.G. Luxan, "HazeWatch: A participatory sensor system for monitoring air pollution in Sydney," *Proc. IEEE Conf. Local Computer Networks*, 2013, pp. 56–64.
[31] S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, and B. Nath, "Real-time air quality monitoring through mobile sensing in metropolitan areas," *Proc. ACM SIGKDD Int'l Workshop on Urban Computing*, 2013, pp. 1–8.
[32] C. Vagnoli, F. Martelli, T.D. Filippis, S.D. Lonardo, B. Gioli, G. Gualtieri, A. Matese, L. Rocchi, P. Toscano, and A. Zaldei, "The SensorWebBike for air quality monitoring in a smart city," *Proc. IET Conf. Future Intelligent Cities*, 2014, pp. 1–4.
[33] G. Mitra, C. Chowdhury, and S. Neogy, "Application of mobile agent in VANET for measuring environmental data," *Proc. Int'l Conf. Applications and Innovations in Mobile Computing*, 2014, pp. 48–53.
[34] S.C. Hu, Y.C. Wang, C.Y. Huang, and Y.C. Tseng, "Measuring air quality in city areas by vehicular wireless sensor networks," *Journal of Systems and Software*, vol. 84, no. 11, pp. 2005–2012, 2011.
[35] B. Higgs and M. Abbas, "Segmentation and clustering of car-following behavior: recognition of driving patterns," *IEEE Trans. Intelligent Transportation Systems*, vol. 16, no. 1, pp. 81–90, 2015.
[36] J. Harri, F. Filali, and C. Bonnet, "Mobility models for vehicular ad hoc networks: a survey and taxonomy," *IEEE Comm. Surveys & Tutorials*, vol. 11, no. 4, pp. 19–41, 2009.
[37] MiCS-2610 O_3 sensor. [Online]. Available: https://www.cdiweb.com/datasheets/e2v/mics-2610.pdf
[38] TGS 2442–for the detection of carbon monoxide. [Online]. Available: https://www.soselectronic.com/productdata/52/85/5/52855/TGS2442.pdf
[39] MQ136 semiconductor sensor for sulfur dioxide. [Online]. Available: http://www.china-total.com/Product/meter/gas-sensor/MQ136.pdf
[40] MiCS-2710 NO_2 sensor. [Online]. Available: https://www.cdiweb.com/datasheets/e2v/mics-2710.pdf
[41] Laser PM2.5 sensor specification. [Online]. Available: http://breathe.indiaspend.org/wp-content/uploads/2015/11/nova_laser_sensor.pdf
[42] M.H. Cheng, H.F. Chiu, and C.Y. Yang, "Coarse particulate air pollution associated with increased risk of hospital admissions for respiratory diseases in a tropical city, Kaohsiung, Taiwan," *Int'l Journal of Environmental Research and Public Health*, vol. 12, no. 10, pp. 13053–13068, 2015.
[43] NMEA data. [Online]. Available: http://www.gpsinformation.org/dale/nmea.htm
[44] L. Yonghe, F. Jinming, and S. Yuehong, "A simple sweep-line Delaunay triangulation algorithm," *Journal of Algorithms and Optimization*, vol. 1, no. 1, pp. 30–38, 2013.
[45] T.K. Poiker, "Triangulated irregular network (TIN) data model," in *Encyclopedia of Geography*. New York: SAGE Publications, 2010.