Probabilistic Control of Dynamic Crowds Toward Uniform Spatial-Temporal Coverage

Probabilistic Control of Dynamic Crowds Toward Uniform Spatial-Temporal Coverage

Yukio Ogawa, *Member, IEEE,* Go Hasegawa, *Member, IEEE,* and Masayuki Murata, *Member, IEEE*

Abstract—Vehicular mobility and connectivity vary significantly over space and time when vehicular crowd sensing covers a city-wide area for a long time period, but it is important to achieve sufficiently uniform data coverage to satisfy the requirements of an environmental monitoring scenario. Our goal is thus to ensure uniform spatial-temporal coverage of sensed data over a city-wide area despite such vehicle dynamics. For a large area, trajectory-based approaches must deal with a great number and variety of participant mobility patterns. Hence, we propose a probabilistic control mechanism that adaptively adjusts the incentive to each participant, without using any prior information about participants. We provide a mathematical analysis that ensures stability of the number of participants with assigned tasks (called workers), and we evaluate the mechanism's robustness by using 24-hr vehicle trace data from a city-wide area. Our results demonstrate that, when the number of participants is up to 1500 times higher than the required number of workers, sensing actions result in a distribution with a mean of about 1 and an interquartile range of around 4 for a required sensing interval; moreover, the mean increases by 2% when 30% of communication messages are randomly lost.

Index Terms—vehicular crowd sensing, environmental monitoring, spatial-temporal coverage, probabilistic control

1 Introduction

Mobile crowd sensing (MCS) is an appealing paradigm in which ordinary citizens contribute to data collection by using their own mobile devices [1]. Here, vehicles have become a promising platform for enlarging the sensing area from urban to rural scales [2], in which every vehicle has continuous network connectivity throughout the area and integrates GPS devices, smart ambient sensors [3], and occupants' smartphones [4]. Vehicles can collect real-time data from their surroundings and provide the data to, e.g., *smart cities* [5] and ITSs [6]. The geographical dispersion of vehicles might result in a mixture of sparse and dense coverage of sensing actions. However, when the data is used as basic information in public infrastructure applications, it should be obtained uniformly both across the entire area and throughout a day. For example, in the case of weather conditions in Japan, a meteorological system called *AMeDAS* [7] measures the wind direction and speed at space intervals of 17 km and time intervals of 10 min, and a radar network called *XRAIN* [8] observes the rainfall over metropolitan areas with a spatial resolution of 250 m and an update cycle of 1 min. The goal of our study is to provide such data through MCS, thus ensuring data coverage that is sufficiently uniform over space and time for applications like environmental monitoring.

A huge research effort has been devoted to the development of MCS, and urban sensing and environmental monitoring are typical MCS applications [9], [10], [11]. Case studies in this domain include monitoring of air pollutants and noise [12], road surface anomalies [13], wireless network quality [14], and rainfall intensity [15]. These kinds of sensing tasks are either conducted at a particular location with an expiration time or conducted periodically at random locations, and the spatial-temporal accuracy and granularity of the collected data are determined by the application requirements. However, every application needs to ensure completion of all tasks to guarantee the data coverage of an entire area; this is a fundamental research issue in MCS [9], [10].

Most existing works on this issue assume prior knowledge about MCS participants. A representative method is trajectory-based assignment, in which tasks are assigned to participants (called workers) that are selected by predicting their future movements or locations from their moving traces or historical locations [16], [17], [18], [19], [20]. This method increases the probability of each task being performed successfully. In contrast, we assume a city-wide scenario for a smart city. In this scenario, an MCS system must monitor an entire city-wide area, where a huge number of ordinary vehicles participates from urban, suburban, and rural areas throughout a day. Several drawbacks thus arise in this scenario:

- There are an enormous number and variety of vehicle mobility patterns in terms of vehicle speeds,

- *Y. Ogawa is with ICT Education Center, Muroran Institute of Technology, 27-1 Mizumoto-cho, Muroran, Hokkaido 050-8585, Japan. E-mail: y-ogawa@mmm.muroran-it.ac.jp*
- *G. Hasegawa is with Research Institute of Electrical Communication, Tohoku University, 2-1-1 Katahira, Aoba-ku, Sendai 980-8577, Japan.*
- *M. Murata is with the Graduate School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan.*

Manuscript received April 19, 2005; revised August 26, 2015.

travel distances, times, and routes. The local vehicle density thus significantly changes across the entire area and varies over different time periods. Furthermore, spot densities may unexpectedly change because of, e.g., traffic accidents.
- Advanced mobile networks like 5G might not yet be available in rural areas. Moreover, mobile network quality varies from location to location [21], and vehicular mobility may lead to packet loss [22]. Mobile network quality might thus add uncertainty to participants' information.
- An MCS system incurs security and privacy risks when it obtains trajectories from ordinary vehicles, except for public vehicles like buses [23].
- Even when there is a reliable, secure way to collect and store participants' trajectories, massive computational resources are needed to analyze such enormous trajectory data.

To overcome these drawbacks, we have developed a worker allocation mechanism based on a biology-inspired mathematical model called the response-threshold model [24]. This model describes the phenomena of division of labor in insect societies, and each participant reacts to an incentive (also called a stimulus) and simply decides whether to accept or refuse a task in a probabilistic manner. This lightweight approach does not require any prior knowledge about participants and can adaptively modify the behavior of a group of anonymous participants. Accordingly, the response-threshold model has the potential to solve the above problems. It has already been applied to several distributed systems to achieve self-organized control [25], [26], e.g., assignment of sensing tasks [27], routing [28], and registry service provision [29] to nodes in wireless sensor networks. It has also been used for task allocation in swarm robotic systems [30]. However, several challenges arise in using this model to achieve uniform data coverage in the above application scenario.

We summarize the challenges and our contributions below:

- The response-threshold model describes changes in the number of workers only in the temporal domain, yet the participant density changes in both the spatial and temporal domains. Hence, we formulate a probabilistic worker allocation mechanism based on the response-threshold model to account for spatially distributed incentives and the temporal dynamics of all workers, in consideration of a less-stable network environment. This mechanism uses only the current numbers of participants and workers, the locations of sensed data, and several vehicular traffic statistics.
- Rapid changes in the number of participants could lead to instability in the number of workers, unless the participants are given optimal incentives. We thus describe temporal changes in the incentives and the number of workers as a two-dimensional nonlinear system [31]. Then, we present a mathematical analysis and specify the conditions to stabilize the number of workers.
- As far as we know, the question of whether such a probabilistic mechanism can control dynamic crowds has not yet been evaluated. Through computer simulations using city-wide vehicle trace data over a day, we demonstrate that our mechanism can overcome the above-mentioned drawbacks. The simulated spatial-temporal distribution of sensing frequencies has a mean of almost 1 and an interquartile range (IQR) around 4 for a required sensing interval, even when 30% of communication messages are randomly lost.

The rest of the paper is organized as follows. We discuss related works in Section 2. We formulate the mathematical model in Section 3 and analyze its stability in Section 4. Section 5 describes the algorithm for controlling participants. Section 6 evaluates the spatial-temporal coverage. Finally, Section 7 concludes the paper.

2 Related Work

To ensure sufficient sensing coverage, existing works have introduced a variety of techniques for MCS processes such as participant incentivization and task allocation [9], [10], [11]. As reviewed below, most of these techniques require prior knowledge about participants, such as location histories and moving traces.

Participant incentivization: There are two sensing paradigms regarding human involvement in MCS: participatory and opportunistic modes [32]. A participatory mode incentivizes participants and incorporates them into decision making, e.g., by taking a detour to a sensing location. Several studies have proposed techniques to guarantee data coverage in this mode. For example, Xu et al. [33] used the Kullback-Leibler divergence between the trajectories of incentivized vehicles and the distribution of target data to make the sensing distribution similar to the target distribution. Fan et al. [34] exploited a reverse combinatorial auction to optimize the detour cost between the task set and the participants' original trajectories. Tao and Song [35] determined workers' traveling paths through clustered tasks to balance the data quality and the workers' profits. In contrast, our MCS system operates in an opportunistic mode to support large-scale deployment [32], and participants are thus unaware of the data collection process and are not requested to deviate from their original routes.

Probabilistic task allocation: In MCS, probabilistic or nondeterministic task allocation often means that a single task at a certain location and time is assigned to multiple participants. Their probabilities of performing the task are obtained from their locations or mobility profiles [36], as well as the recruiting cost [19], [37] and social relationships [38], [39], [40], so that a single participant or multiple cooperative participants are selected to perform the task. Moreover, in [41], each task should be performed at a specific location within a range of beginning and ending times. Unlike those studies, we do not handle the

situation in which multiple participants have the possibility of simultaneously performing a single task; rather, we simply assume that a task is probabilistically allocated to a participant who happens to be at the specific location.

Task types: A worker simultaneously collects a single type or multiple types of data according to whether sensing tasks are homogeneous or heterogeneous [42]. As for our target, i.e., homogeneous tasks, Liu et al. [43] introduced reinforcement learning to identify a few subareas to be sensed in the future according to the expected coverage by participants. Cao et al. [44] used transfer learning to estimate the roads covered by new vehicular workers from those covered by previous vehicles. To handle heterogeneous tasks, Wang et al. [45] exploited the spatial-temporal correlation among multiple concurrent tasks and reused limited worker resources. Wang et al. [46] then applied Lyapunov optimization to deal with the limited sensing range of each worker.

Specific scenarios: Other works have investigated specific application scenarios. Wang et al. [47] complemented opportunistic workers moving along their routine trajectories with participatory workers deviating from their original routes. Wang et al. [48] integrated the predefined trajectories of public vehicles like buses and predicted the trajectories of ordinary vehicles. Chen et al. [49] forecasted the probabilities of routes and ride requests to optimize the sensing coverage of ridesharing vehicles.

Non-trajectory-based approach: In contrast to the above works, there have been a few works that do not analyze participants' prior information but use only their current information. For example, Zhang et al. [50] estimated the possible coverage of participants from their current locations and directions. Fiandrino et al. [51] recruited workers according to their spatial distance to tasks, their willingness to contribute, and the remaining battery charge of their devices. Similarly, our mechanism does not require participants' prior information, and it neither stores participants' current locations nor associates them with identifiers; these characteristics are favorable to promote citizen-based sensing.

Our approach: The spatial-temporal density of data acquired by opportunistic vehicular sensing is significantly affected by vehicle mobility [52]. To reduce the influence of vehicles being spaced irregularly in space and time, irregularly sensed data should be converted to uniformly distributed data [53], or uniformly distributed workers should be selected from irregularly spaced participants. We choose the latter approach and focus on how to handle the irregular spatial-temporal mobility of participants without prior knowledge about them. Our system deals only with the number of anonymous participants and does not consider who collects the data. It also does not know where participants are and when and where they start and stop traveling. Hence, our system does not assign sensing tasks deterministically to participants. Instead, all participants are continuously stimulated by incentives, and some of them probabilistically perform sensing actions, so that data is collected approximately at a required sensing interval over space and time.

To the best of our knowledge, the only work related to our application scenario is that of Montori et al. [54], who sought to control the amount of data generated by participants and proposed a distributed probabilistic algorithm based on limited feedback from a central entity. Our approach, however, is fundamentally different from theirs, because their algorithm operates only in the time domain, whereas we control both the spatial and temporal domains. Furthermore, their algorithm does not consider low-quality networks, whereas our model accounts for less-reliable message delivery [55].

3 Worker Allocation Model

This section gives an overview of our MCS shown in Fig. 1 and a mathematical model for controlling participants. In this section, the main equations are Eqs. (7) and (17); the main symbols are listed in Table 1.

3.1 System Overview

System configuration: Our system consists of a central server in a data center (simply called the server hereafter), client software running on participating vehicles (simply called participants), and mobile networks and the Internet between them. The participants are ordinary vehicles with no distinctive features. The whole target area L is treated as a single sensing area and discretized into a grid of many small squares. Each square represents a discretized location denoted by $j \in L$. The server executes a sensing process at a fixed time interval called a time slot. The current time slot is denoted by t, and the length of a time slot is Δ (in seconds). On the other hand, each participant, denoted by k, executes a sensing process at an interval of ξ time slots in an independent and asynchronous manner. A worker is defined as a participant performing sensing actions and collecting data. Participants and workers that activate sensing processes at the current time slot are called active participants and active workers, respectively.

Application requirements: The only application requirement in our system is the spatial-temporal granularity, i.e., the space and time intervals of sensing actions, denoted by S (in meters) and T (in seconds), respectively. The space interval S is consistent with the grid square size for locations and is defined as $\xi\Delta$ times the participants' average velocity v:

$$S = \xi \Delta v. \tag{1}$$

The space interval S and time interval T determine the required number of workers across the area per time slot, N. Each sensing action must be performed once per T seconds, i.e., Δ/T times per time slot, at each location. The number of sensing actions equals the number of active workers, which is approximated by $1/\xi$ of the total number of workers. Thus, $\xi\Delta/T$ workers are required per time slot at each location; however, this number may be less than the maximum number of participants at a

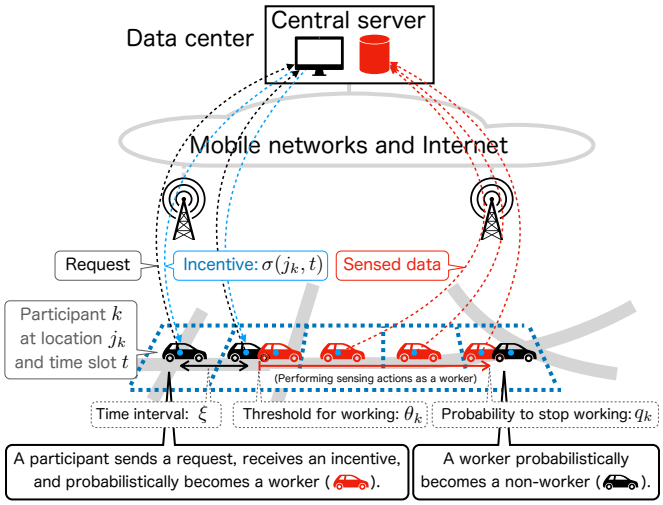


Fig. 1. System overview.

particular time slot and location, e.g., at midnight in a rural area. We thus define N by

$$N = \sum_{j \in L} \min\left\{\frac{\xi\Delta}{T},\, m_j\right\} = \sum_{j \in L} \min\left\{\frac{S}{Tv},\, m_j\right\}, \quad (2)$$

where m_j is the average number of participants per time slot at location j, which is a constant.

Network environments: At each time slot, the server estimates the total numbers of participants (including workers) and workers, m_t and $n(t)$, by counting the numbers of active participants and workers accessing the server, m'_t and $n'(t)$, respectively, and using the message loss ratio p_t $(0 \leq p_t < 1)$. Here, p_t is the ratio of the number of data instances (enveloped in messages) that the server cannot receive from active workers to the number of data instances that active workers send to the server at t. Thus, $1-p_t$ is regarded as the ratio of the number of active workers accessing the server at t, $n'(t)$, to the total number of active workers at t. Then, $n(t)$ is estimated as

$$n(t) = \frac{\xi n'(t)}{1-p_t}. \quad (3)$$

We also simply assume that m_t is similarly estimated as

$$m_t = \frac{\xi m'_t}{1-p_t}. \quad (4)$$

Note that delayed or incomplete messages are treated as lost messages, and their influence is captured by using p_t.

Remarks: The time interval ξ at which participants activate sensing processes alleviates the network bandwidth consumed by the messages sent to/from participants. Suppose that a city-wide area includes 3000 participants on average and 10000 participants at peak hours, and that the message size is 1218 bytes [56]. When ξ is set to 10, on the server side, the required bandwidth is about 3 Mbps on average and 10 Mbps at peak hours, which are not especially large values. Each worker uploads sensed data in the same manner. As workers are uniformly distributed in the area, N should be several hundred at most. For $\xi = 10$, the number of simultaneously uploaded data instances is then several tens; this number has little influence on the network load.

TABLE 1
Main symbols and definitions

S	Required space interval for sensing actions (in meters)
T	Required time interval for sensing actions (in seconds)
N	Required number of workers per time slot
Δ	Length of time slot (in seconds)
l	Ratio of number of disappearing participants to total
ξ	Interval for participant's sensing process (in time slots)
θ_k	Threshold for non-worker k to become worker
θ	Average of θ_k
q_k	Probability of worker k becoming non-worker
q	Average of q_k
m_t	Total number of participants at time slot t
m'_t	Number of active participants accessing server at t
$n(t)$	Total number of workers at time slot t
$n'(t)$	Number of active workers accessing server at t
p_t	Message loss ratio at time slot t
$\sigma(j,t)$	Incentive at location j and time slot t
$s(t)$	Base incentive at time slot t
δ_t	Magnitude of change in $s(t)$ at time slot t
ω_t	Weight factor at time slot t

3.2 Participant Behavior

Our system operates in an opportunistic mode [32], so that each participant automatically executes sensing processes without human intervention and decision-making. These processes are thus no different among individuals. Besides the time interval for each sensing process, ξ, the behavior of participant k is specified by two parameters:

- The threshold for reacting to an incentive (i.e., reward) and starting the worker role, θ_k; and
- The probability of stopping the worker role, q_k.

Participant k sets θ_k and q_k within certain ranges by considering its sensing ability, willingness to work, and so on. Let $Y_k(t) \in \{0, 1\}$ be the state of participant k at time slot t, where 0 and 1 indicate a non-worker and a worker, respectively. According to the response-threshold model [24], each participant executes the following process at every interval of ξ time slots.

Non-worker: When the network with the server is available, non-worker k at location j_k and time slot t requests the server to return an incentive of amount $\sigma(j_k, t)$. It then becomes a worker with the following probability [24]:

$$P(Y_k(t+\xi) = 1 | Y_k(t) = 0) = \frac{\sigma^2(j_k,t)}{\sigma^2(j_k,t) + \theta_k^2}. \quad (5)$$

Upon becoming a worker, k starts sensing actions at the beginning of the ξ-time-slot interval.

Worker: Worker k performs sensing actions (i.e., collects data) regardless of whether the network is available. When it is available, k uploads sensed data with its location and time information. The data sensed at t is uploaded at $t+\xi$. Then, k stops its sensing actions and becomes a non-worker with the following probability:

$$P(Y_k(t+\xi)=0|Y_k(t)=1)=q_k. \tag{6}$$

Equation (6) means each worker continues sensing actions for an average period of $1/q_k$ times ξ time slots.

When the network is not available, non-workers and workers cannot send requests and sensed data, respectively. Accordingly, during a network failure, they are not allowed to change their roles: non-workers cannot become workers, while workers continue their sensing actions and cannot return to being non-workers. Our model accounts for these behaviors via the message loss ratio p_t, though the server only estimates the current numbers of workers and participants by Eqs. (3) and (4). All the data sensed during a network failure are sent to the server as soon as the network becomes available.

In addition, when workers leave the sensing area or stop traveling, then they send their data to the server before leaving or stopping.

3.3 Incentive Model

The amount of incentive at each location and each time slot is determined to ensure two properties:

- The total number of workers at the current time slot is close to that at the previous time slot; and
- The current amount of data at a location is close to that at other locations.

Let $s(t)$ be the base incentive at time slot t; $s(t)$ is independent of locations. The server increases $s(t)$ at time slot $t+1$ when the current number of workers, $n(t)$, is smaller than the required number of workers, N, and it decreases $s(t)$ at $t+1$ when $n(t)$ is larger than N. This procedure is given as follows:

$$s(t+1)=s(t)+\delta_t\left(1-\frac{n(t)}{N}\right), \tag{7}$$

where δ_t is the magnitude of the change in $s(t)$.

The amount of incentive at location j and time slot t, $\sigma(j,t)$, is computed from $s(t)$ as follows:

$$\sigma(j,t)=w_{j,t}s(t), \tag{8}$$

where $w_{j,t}$ is a weight coefficient for location j and time slot t. Here, $w_{j,t}$ is a function of the ratio of the cumulative amount of data collected at j and t to the space-averaged cumulative amount of data at collected t (denoted by $r_{j,t}$):

$$w_{j,t}=f(r_{j,t}) \tag{9}$$

$$r_{j,t}=\frac{\sum_{\tau=t-t_w+1}^{t} g_{j,\tau}}{\frac{1}{|L|}\sum_{j\in L}\sum_{\tau=t-t_w+1}^{t} g_{j,\tau}}, \tag{10}$$

where f is a function, $g_{j,\tau}$ is the amount of data at j and τ, and t_w is the width of a moving time window. To decrease differences among the cumulative amounts of data at different locations, f should assign a larger (smaller) weight to a location where the cumulative amount of data is smaller (larger) than the average. Though f depends on the spatial distribution of vehicles in the area, we assume that it is expressed in terms of powers of $1/r_{j,t}$:

$$f(r_{j,t})=\alpha\left(\frac{1}{r_{j,t}}\right)^{\beta}, \tag{11}$$

where α and β are constants.

3.4 Worker Dynamics

The state of each worker is probabilistically determined by Eqs. (5) and (6). The expected total number of workers in the area is formulated in the following way. At time slot t, m'_t active participants access the server, and $n'(t)$ of them are active workers. Thus, $m'_t-n'(t)$ active non-workers access the server. Hence, the number of workers at time slot $t+1$, $n(t+1)$, is calculated from the number of workers at time slot t, $n(t)$, as follows:

$$n(t+1)+n^*(t+1)=n(t)+\nu(t)(m'_t-n'(t))-qn'(t). \tag{12}$$

Here, $\nu(t)(m'_t-n'(t))$ represents the number of non-workers that will start working, and $qn'(t)$ represents the number of workers that will stop working, where q is the mean of q_k (the probability of a worker becoming a non-worker).

In Eq. (12), $n^*(t+1)$ represents the number of workers that disappear from the area between time slots t and $t+1$. First, we determine the number of disappearing participants, which is the sum of the number of participants that leave the area and the number of participants that stop traveling in the area. To conveniently count the number of participants that disappear at $t+1$, we assume it is roughly proportional to the total number of participants in the area at $t+1$, and we denote the proportionality coefficient by l. We further assume l approximates the ratio of the number of disappearing workers, $n^*(t+1)$, to the sum of the numbers of workers and disappearing workers, $n(t+1)+n^*(t+1)$. Then, $n^*(t+1)$ is given by

$$n^*(t+1)=\frac{l}{1-l}n(t+1). \tag{13}$$

In Eq. (12), $\nu(t)$ represents the average probability of a non-worker becoming a worker at t, which is defined by using Eq. (5) as follows:

$$\nu(t)=\frac{1}{m'_t-n'(t)}\sum_{k\in U(t)}\frac{\sigma^2(j_k,t)}{\sigma^2(j_k,t)+\theta_k^2}, \tag{14}$$

where $U(t)$ is the set of active non-workers accessing the server at t. By using Eq. (8), $\nu(t)$ is approximated as a function of the base incentive at t, $s(t)$:

$$\nu(t) = \frac{1}{m'_t - n'(t)} \sum_{k \in U(t)} \frac{(w_{j_k,t} s(t))^2}{(w_{j_k,t} s(t))^2 + \theta_k^2} \sim \frac{(\omega_t s(t))^2}{(\omega_t s(t))^2 + \theta^2}, \tag{15}$$

where θ is the mean of θ_k (the threshold for each non-worker becoming a worker). We introduce a weight factor at t, ω_t, which is defined as

$$\omega_t = \frac{\theta}{s(t)} \sqrt{\frac{\nu(t)}{1 - \nu(t)}}. \tag{16}$$

This enables us to describe the temporal dynamics of all the spatially distributed workers, instead of the specific temporal dynamics of each worker.

Finally, by substituting Eqs. (3), (4), (13), and (15) into Eq. (12), we get the temporal dynamics of the workers:

$$n(t+1) = (1-l)\Bigg\{ n(t) + \frac{1-p_t}{\xi} \Bigg\{ \frac{(\omega_t s(t))^2}{(\omega_t s(t))^2 + \theta^2} (m_t - n(t)) - q n(t) \Bigg\} \Bigg\}. \tag{17}$$

4 Stability Analysis of Worker Dynamics

The combination of Eqs. (7) and (17) in Section 3 is a nonlinear system of two-dimensional first-order difference equations, for which this section presents mathematical analysis [57] to obtain stability criteria.

4.1 Number of Workers in Steady State

To simplify the notation of Eqs. (7) and (17), we use vector notation with $\boldsymbol{z}(t) = [s(t) \;\; n(t)]^{\mathrm{T}}$ and a map $\boldsymbol{f} : \boldsymbol{z}(t) \mapsto \boldsymbol{z}(t+1)$. Then, we obtain a first-order difference equation:

$$\boldsymbol{z}(t+1) = \boldsymbol{f}(\boldsymbol{z}(t)). \tag{18}$$

We derive a steady-state solution $\bar{\boldsymbol{z}} = [\bar{s} \;\; \bar{n}]^{\mathrm{T}}$ that satisfies $\bar{\boldsymbol{z}} = \boldsymbol{f}(\bar{\boldsymbol{z}})$. The amount of incentive given to participants in the steady state, $\bar{s}$, and the number of workers in that state, $\bar{n}$, are calculated from Eqs. (7) and (17):

$$\begin{cases} \bar{s} = \dfrac{\theta}{\omega_t} \sqrt{\dfrac{(1-\chi_t)N}{\kappa_t m_t - (1 - \chi_t + \kappa_t)N}}, & (19) \\ \bar{n} = N, & (20) \end{cases}$$

$$\chi_t = 1 - l - \frac{(1-l)(1-p_t)q}{\xi}, \tag{21}$$

$$\kappa_t = \frac{(1-l)(1-p_t)}{\xi}. \tag{22}$$

Equation (20) indicates that $\bar{n}$ equals the required number of workers, N. In Eq. (19), $\bar{s}$ is a positive real number and $1 - \chi_t > 0$ holds. Thereby, $\kappa_t m_t - (1 - \chi_t + \kappa_t)N > 0$ must hold; this gives the following relationship between N and m_t:

$$N < \frac{\kappa_t}{1 - \chi_t + \kappa_t} m_t. \tag{23}$$

4.2 Stability Conditions for Number of Workers

Let $\boldsymbol{x}(t) = [x_{\mathrm{s}}(t) \;\; x_{\mathrm{n}}(t)]^{\mathrm{T}}$ be the difference between $\boldsymbol{z}(t)$ and $\bar{\boldsymbol{z}}$. Then, $\boldsymbol{z}(t)$ is expressed as $\boldsymbol{z}(t) = \bar{\boldsymbol{z}} + \boldsymbol{x}(t)$, and $\boldsymbol{z}(t+1)$ satisfies $\boldsymbol{z}(t+1) = \bar{\boldsymbol{z}} + \boldsymbol{x}(t+1) = \boldsymbol{f}(\bar{\boldsymbol{z}} + \boldsymbol{x}(t))$. A first-order Taylor expansion approximates Eq. (18):

$$\boldsymbol{x}(t+1) = A\boldsymbol{x}(t), \tag{24}$$

$$A = \left. \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}} \right|_{\boldsymbol{z} = \bar{\boldsymbol{z}}}, \tag{25}$$

where A is the Jacobian matrix given by

$$A = \begin{bmatrix} 1 & -\dfrac{\delta_t}{N} \\ 2\kappa_t(m_t - \bar{n}) \dfrac{{\omega_t}^2 \bar{s} \theta^2}{\{(\omega_t \bar{s})^2 + \theta^2\}^2} & \chi_t - \kappa_t \dfrac{(\omega_t \bar{s})^2}{(\omega_t \bar{s})^2 + \theta^2} \end{bmatrix}. \tag{26}$$

To simplify the notation, we write this matrix as

$$A = \begin{bmatrix} 1 & -a \\ b & 1 - 2c \end{bmatrix}, \tag{27}$$

where a, b, and c are constants. These are obtained by substituting Eqs. (19) and (20) into the elements of matrix (26), as follows:

$$\begin{cases} a = \dfrac{\delta_t}{N}, & (28) \\ b = \dfrac{2\omega_t \sqrt{(1-\chi_t)N\{\kappa_t m_t - (1 - \chi_t + \kappa_t)N\}^3}}{\kappa_t \theta (m_t - N)}, & (29) \\ c = \dfrac{(1-\chi_t) m_t}{2(m_t - N)}. & (30) \end{cases}$$

Suppose that Eq. (24) satisfies a given initial condition, $\boldsymbol{x}(0) = \boldsymbol{x}_0$; then, Eq. (24) is expressed as

$$\boldsymbol{x}(t) = A^t \boldsymbol{x}_0. \tag{31}$$

We assume that

$$\boldsymbol{x}(t) = \lambda^t \boldsymbol{h} \tag{32}$$

is a solution of Eq. (31), and by substituting it into Eq. (24), we have

$$\lambda \boldsymbol{h} = A\boldsymbol{h}. \tag{33}$$

This means that the solution $\lambda^t \boldsymbol{h}$ in Eq. (32) indeed solves Eq. (24) when $\boldsymbol{h}$ is an eigenvector of matrix (27) with an associated eigenvalue λ. The eigenvector $\boldsymbol{h}$ is a nonzero solution of Eq. (33), and the associated eigenvalue λ should satisfy the characteristic polynomial of matrix (27):

$$\det(\lambda I - A) = \lambda^2 - 2(1-c)\lambda + ab - 2c + 1 = 0. \tag{34}$$

The discriminant of the characteristic polynomial is denoted as D and defined as

$$D = c^2 - ab. \tag{35}$$

In the following, we analyze the stability of the steady-state solution $\bar{\boldsymbol{z}}$ by classifying the roots of the characteristic polynomial (34) according to the sign of the discriminant (35).

In the case of $D > 0$: Matrix (27) has a pair of distinct real eigenvalues. Let λ_1 and λ_2 ($\lambda_1 \neq \lambda_2$) denote the eigenvalues, and let $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ denote their associated eigenvectors. The general solution of Eq. (24) is given by

$$\boldsymbol{x}(t) = d_1 {\lambda_1}^t \boldsymbol{h}_1 + d_2 \lambda_2^t \boldsymbol{h}_2, \tag{36}$$

where d_1 and d_2 are constants that depend on the initial condition $\boldsymbol{x}(0)$. We compute λ_1 and λ_2 according to the characteristic polynomial (34), and then $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ by using Eq. (33). The general solution (36) is thus expressed as follows:

$$\begin{aligned}\begin{bmatrix} x_{\mathrm{s}}(t) \\ x_{\mathrm{n}}(t) \end{bmatrix} &= d_1 \left(1 - c + \sqrt{c^2 - ab}\right)^t \begin{bmatrix} -a \\ -c + \sqrt{c^2 - ab} \end{bmatrix} \\ &+ d_2 \left(1 - c - \sqrt{c^2 - ab}\right)^t \begin{bmatrix} -a \\ -c - \sqrt{c^2 - ab} \end{bmatrix}. \end{aligned} \tag{37}$$

The steady-state solution $\bar{\boldsymbol{z}}$ is asymptotically stable; that is, we have $\lim_{t\to\infty} \boldsymbol{z}(t) = \bar{\boldsymbol{z}}$ when $|\lambda_1| < 1$ and $|\lambda_2| < 1$, which gives

$$\left|1 - c + \sqrt{c^2 - ab}\right| < 1 \text{ and } \left|1 - c - \sqrt{c^2 - ab}\right| < 1. \tag{38}$$

In the case of $D < 0$: The eigenvalues of matrix (27) are complex conjugates. We denote them by λ_1 and $\bar{\lambda}_1$ and their associated eigenvectors by $\boldsymbol{h}_1$ and $\bar{\boldsymbol{h}}_1$. The general solution of Eq. (24) is then given by

$$\boldsymbol{x}(t) = d\lambda_1^t \boldsymbol{h}_1 + \bar{d}\,\bar{\lambda}_1^t\,\bar{\boldsymbol{h}}_1 = 2\Re(d\lambda_1^t \boldsymbol{h}_1), \tag{39}$$

where d and $\bar{d}$ are a complex constant and its complex conjugate, respectively, and $\Re$ indicates the real part. Furthermore, λ_1 and $\boldsymbol{h}_1$ are respectively obtained as

$$\lambda_1 = 1 - c + i\sqrt{ab - c^2} = \sqrt{ab - 2c + 1}\,\mathrm{e}^{i\phi}, \tag{40}$$

$$\phi = \tan^{-1} \frac{\sqrt{ab - c^2}}{1 - c}, \tag{41}$$

$$\boldsymbol{h}_1 = \begin{bmatrix} -a \\ -c + i\sqrt{ab - c^2} \end{bmatrix} = \begin{bmatrix} -a \\ -\sqrt{ab}\,\mathrm{e}^{-i\psi} \end{bmatrix}, \tag{42}$$

$$\psi = \tan^{-1} \frac{\sqrt{ab - c^2}}{c}. \tag{43}$$

Here, i denotes the imaginary unit. Hence, the general solution (39) is computed as

$$\begin{aligned}\begin{bmatrix} x_{\mathrm{s}}(t) \\ x_{\mathrm{n}}(t) \end{bmatrix} &= d_1 \left(\sqrt{ab - 2c + 1}\right)^t \begin{bmatrix} a\cos\phi t \\ \sqrt{ab}\,\cos(\phi t - \psi) \end{bmatrix} \\ &+ d_2 \left(\sqrt{ab - 2c + 1}\right)^t \begin{bmatrix} a\sin\phi t \\ \sqrt{ab}\,\sin(\phi t - \psi) \end{bmatrix}, \end{aligned} \tag{44}$$

where d_1 and d_2 are constants. The steady-state solution $\bar{\boldsymbol{z}}$ is thus asymptotically stable when we have

$$0 < ab - 2c + 1 < 1. \tag{45}$$

In the case of $D = 0$: The eigenvalues of matrix (27) are real and equal. We denote them by λ_1 and λ_2, and they are expressed as

$$\lambda_1 = \lambda_2 = 1 - c. \tag{46}$$

In this case, the general solution of Eq. (24) is directly obtained from Eq. (31). A^t in Eq. (31) is computed as

$$A^t = (1 - c)^t \begin{bmatrix} 1 + \frac{ct}{1-c} & -\frac{at}{1-c} \\ \frac{bt}{1-c} & 1 - \frac{ct}{1-c} \end{bmatrix}. \tag{47}$$

The general solution is thus obtained as

$$\begin{aligned}\begin{bmatrix} x_{\mathrm{s}}(t) \\ x_{\mathrm{n}}(t) \end{bmatrix} &= (1 - c)^t \begin{bmatrix} 1 + \frac{ct}{1-c} & -\frac{at}{1-c} \\ \frac{bt}{1-c} & 1 - \frac{ct}{1-c} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \\ &= d_1 (1 - c)^t \begin{bmatrix} 1 + \frac{ct}{1-c} \\ \frac{bt}{1-c} \end{bmatrix} + d_2 (1 - c)^t \begin{bmatrix} -\frac{at}{1-c} \\ 1 - \frac{ct}{1-c} \end{bmatrix}, \end{aligned} \tag{48}$$

where d_1 and d_2 are constants. The steady-state solution $\bar{\boldsymbol{z}}$ is asymptotically stable when

$$|1 - c| < 1. \tag{49}$$

5 Algorithm for Allocation Control

This section explains how the server calculates an optimal incentive and controls the participants' behavior.

5.1 Optimal Incentive Calculation

We define the amount of the base incentive at time slot t, $s(t)$, in a stable situation to satisfy two conditions:

- **Condition 1:** The number of workers, $n(t)$, is in a state of stabilizing over time. That is, either Eq. (38), (45), or (49) is satisfied.
- **Condition 2:** The number of workers, $n(t)$ is in a state of approaching a steady state, $\bar{n}\,(= N)$, at the highest rate. That is, $x_{\mathrm{n}}(t) \to 0$ at the highest rate as $t \to \infty$ in either Eq. (37), (44), or (48).

For Condition 2, we define $x_{\mathrm{n}}(t) \to 0$ to mean that $|x_{\mathrm{n}}(t)| < \epsilon$, where ϵ is a small constant. We refer to the time until $|x_{\mathrm{n}}(t)| < \epsilon$ is satisfied as the convergence time (in time slots), and we denote it by G:

$$G = \{t \in \mathbb{Z} \mid \forall \tau \in \mathbb{Z}, \tau \geq t, |x_{\mathrm{n}}(\tau)| < \epsilon\}. \tag{50}$$

Condition 2 (as well as Condition 1) is a complex function of the parameters δ_t, N, l, ξ, θ, q, ω_t, m_t, and p_t in Eqs. (37), (38), (44), (45), (48), and (49). However, G depends only on δ_t (the magnitude of change in the amount of the base incentive at t), because N, l, ξ, θ, and q are given in advance, while ω_t, m_t, and p_t are calculated at each time slot from information sent by the participants. We thus define the minimum convergence time C by using the optimal value of δ_t:

$$C = \min_{\delta_t \in \mathbb{R}} G. \tag{51}$$

Algorithm 1 Participant k's Sensing Process

```
1: procedure PARTICIPANT(ξ, θ_min, θ_max, q_min, q_max)
2:     Set θ_k (θ_min < θ_k < θ_max) and q_k (q_min < q_k < q_max)
3:     Set t_0, participant k's starting time slot
4:     for t = t_0, t_0 + ξ, t_0 + 2ξ, ... do
5:         if Y_k(t) = 1 then
6:             Sense data at location j_k and store it
7:         end if
8:         if k can access central server then
9:             if Y_k(t) = 0 then
10:                Send θ_k; request and receive σ(j_k, t)
11:                Decide whether to start via Eq. (5)
12:                Set 1 to Y_k(t + ξ) when k starts working
13:            else if Y_k(t) = 1 then
14:                Send q_k; send stored data with j_k and t
15:                Decide whether to stop via Eq. (6)
16:                Set 0 to Y_k(t + ξ) when k stops working
17:            end if
18:        end if
19:    end for
20: end procedure
```

Algorithm 2 Central Server's Control Process

```
1: procedure SERVER(N, l, ξ, θ_min, θ_max, q_min, q_max)
2:     Initialize ω_t, p_t, and s(t)
3:     for t = 0, 1, 2, ... do
4:         for k ∈ all connected participants do
5:             if Y_k(t) = 0 then
6:                 Receive θ_k; return σ(j_k, t) to k
7:             else if Y_k(t) = 1 then
8:                 Receive q_k; receive data with j_k and t
9:             end if
10:        end for
11:        for j ∈ L do
12:            Update g_{j,t} and w_{j,t} via Eqs. (9) and (10)
13:        end for
14:        Estimate m_t via Eq. (4) and n(t) via Eq. (3)
15:        if N does not satisfy inequality (23) then
16:            N ← γ N_max
17:        end if
18:        Calculate θ from θ_k's and q from q_k's
19:        Calculate δ_t via Eq. (52) and s(t+1) via Eq. (7)
20:        Calculate ν(t) via Eq. (14) and ω_t via Eq. (16)
21:        ω_{t+1} ← ω_t
22:        p_{t+1} ← estimated p_t
23:    end for
24: end procedure
```

Hence, we determine the optimal value of δ_t to attain the minimum convergence time, as follows:

$$\delta_t = \arg\min_{\delta_t \in \mathbb{R}} G. \tag{52}$$

Note that, to reduce the computing time, we substitute the following equation for Eq. (52) in the evaluation described in Section 6:

$$\delta_t = \arg\min_{\delta_t \in \mathbb{R}} \{|x_{\rm n}(t)| \mid t \in \mathbb{Z}, \forall \tau \in \mathbb{Z}, \tau \geq t, |x_{\rm n}(\tau)| < \epsilon\}. \tag{53}$$

5.2 Worker Allocation Algorithm

As shown in Algorithm 1, each participant first gets the parameters ξ, the range of θ_k, $[\theta_{\min}, \theta_{\max}]$, and the range of q_k, $[q_{\min}, q_{\max}]$, from the server. Then, at every interval of ξ time slots, it executes the sensing process, which is, in turn, controlled by the server via Algorithm 2.

The server initially gets the parameters, N, l, ξ, and the ranges of θ_k and q_k from an administrator. The server executes the following control process at every time slot.

- When accessed by participant k, the server either returns an incentive of $\sigma(j_k, t)$ to a non-worker k or receives sensed data and its corresponding location and time, j_k and t, from worker k (Lines 4 - 10).
- The server then updates the amounts of collected data, $g_{j,t}$, and the weight coefficient $w_{j,t}$ in accordance with Eqs. (9) and (10) for every location (Lines 11 - 13).
- The server estimates the numbers of workers and participants, $n(t)$ and m_t, respectively (Line 14).
- When N does not satisfy Eq. (23), it is temporarily replaced by a maximum value (Lines 15 - 17).
- The server calculates the means of θ_k and q_k, i.e., θ and q. It then calculates δ_t by Eq. (52) and the next base incentive $s(t+1)$ by Eq. (7) (Lines 18 - 19).
- It computes the average probability of a non-worker becoming a worker, $\nu(t)$, by Eq. (14), and the weight factor ω_t by Eq. (16); then, it substitutes ω_t for the next value, ω_{t+1} (Lines 20 - 21).
- It estimates the message loss ratio p_t and substitutes it for the next value, p_{t+1} (Line 22).

In Line 14, the numbers of participants and workers are respectively estimated by

$$m_t = \sum_{\tau=t-\xi+1}^{t} \frac{m'_\tau}{1-p_\tau}, \tag{54}$$

$$n(t) = \sum_{\tau=t-\xi+1}^{t} \frac{n'(\tau)}{1-p_\tau}, \tag{55}$$

to average the estimation errors of Eqs. (4) and (3).

In Line 16, a temporal maximum number of required workers is given by $\gamma N_{\max}$, where $N_{\max} = \left\lfloor \frac{\kappa_t}{1-\chi_t+\kappa_t} m_t \right\rfloor$ from Eq. (23) and γ is a constant for setting a surplus.

In Line 19, $x_{\rm n}(t)$ in Eqs. (52) and (53) is a nonlinear function. We approximate the optimal value of δ_t through a numerical algorithm such as Newton's method [58].

In Line 22, the message loss ratio p_t is estimated from the number of active workers accessing the server, $n'(t)$. When a worker does not upload sensed data because of a network failure, it retransmits the data when the network recovers. All sensed data is eventually sent to the server, and each data instance includes a time stamp of when

it was sensed. The server thus counts the number of active workers at a past time slot from the number of data instances sensed at that time slot. Then, at time slot t, the server calculates the message loss ratio at time slot $t-\eta$ by

$$p_{t-\eta} = \sum_{\tau=t-\eta-\xi+1}^{t-\eta} \frac{n'(\tau)}{\sum_{j\in L} g_{j,\tau}}, \quad (56)$$

where η is the maximum period for which a worker cannot access the server, and $\sum_{j\in L} g_{j,\tau}$ is the total number of data instances sensed at time slot τ. The server uses $p_{t-\eta}$ as p_t and the next p_{t+1}.

6 Evaluation

This section describes the effectiveness of our proposed mechanism by using 24-hr, city-wide vehicle trace data.

6.1 Dataset and Parameter Settings

We used the largest-scale open dataset for urban vehicular mobility, from the *TAPASCologne* project [59]. This dataset describes 24-hr vehicular traffic in a 400-km^2 region around Köln, Germany. We generated traces containing the location and timestamp of every vehicle at every time slot from the TAPASCologne scenario by using Simulation of Urban MObility (*SUMO*) [60]. We assumed 30% of all vehicles participated in MCS, and we scaled the vehicular traffic demand by 0.3 (see *Caution* in the TAPASCologne Scenario [61]). Figure 2 shows the spatial density distribution of participants in the 24-hr period. The main parameter settings are listed in Table 2.

Application requirements: The space interval S was set to 125, 250, 500, or 1000 m. The time interval T was set to 300, 600, 1800, or 3600 s. Then, Eq. (2) determined the required number of workers, N, whose time average ranged from 10 (at $S = 1000$ m and $T = 3600$ s) to 328 (at $S = 125$ m and $T = 360$ s). Note that Eq. (2) computed N every hour, because m_j in Eq. (2) was defined with hourly granularity from the dataset.

Network environments: We assumed two network situations. In the first situation, vehicles moved throughout a low-quality mobile network environment, and messages between participants and the server were randomly lost over the area. In this situation, the message loss ratio p_t was set to 0.1, 0.3, or 0.5. In the second situation, a network outage occurred in a local area, where all messages were lost. Figure 2 shows the local outage areas where p_t was set to about 0.1 or 0.3. Each value equaled the ratio of the outage area to the total area, for consistency with the uniformly distributed workers, and the values differed a little depending on the grid square size, which was S.

Participant behavior: The time interval for each participant's sensing actions, ξ, was fixed at 10 time slots. The threshold for each non-worker to become a worker, θ_k, was randomly selected within $[1, 3]$ so that the mean θ was nearly 2. The probability for each worker to become a non-worker, q_k, was randomly selected within $[0.7, 0.9]$ so that the mean q was nearly 0.8. We will explain the reasons for these values in Section 6.2. Note that the ranges of θ_k and q_k have little impact on the sensing frequency distributions described in Section 6.3.

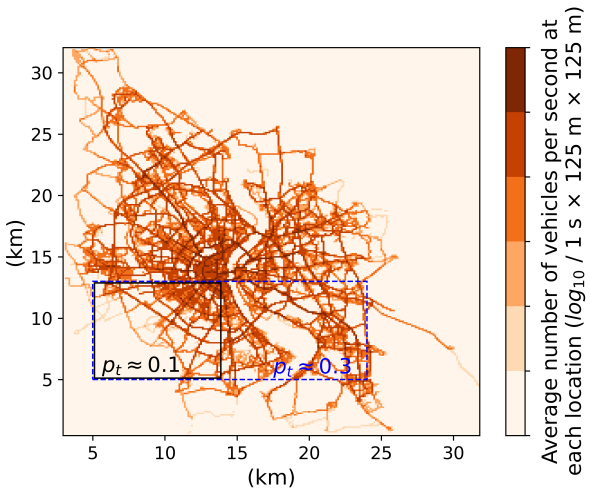


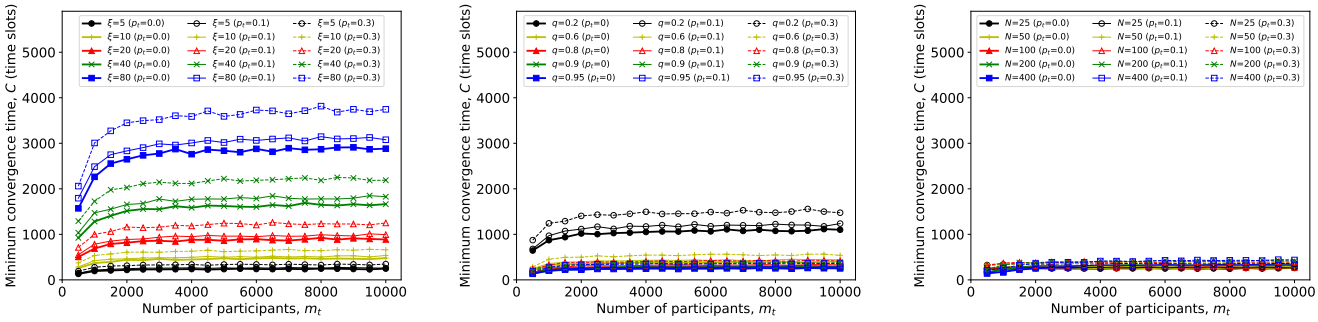
Fig. 2. Spatial density distribution of vehicles in the *TAPASCologne* dataset [59]. Each rectangle represents a network outage area as described in Section 6.3.3 (p_t: set value of the message loss ratio).

TABLE 2
Main parameter settings in Section 6.3. The underlined values are used in Sections 6.3.1 and 6.3.2.

S	125, 250, 500, 1000 (m)
T	300, 600, 1800, 3600 (s)
Δ	1, 2, 4, 8 (s)
ξ	10
θ_k	[1, 3]
q_k	[0.7, 0.9]
p_t	0, 0.1, 0.3, 0.5

System configuration: We set other parameters according to dataset analysis and preliminary experiments, as follows.

- In Eq. (1), for $\xi = 10$, the time slot length Δ was determined as 1, 2, 4, or 8 s, corresponding to the S values of 125, 250, 500, or 1000 m, respectively. The average vehicle velocity v was 12.5 m/s.
- In Eq. (10), the width of the moving time window, t_w, was set to 6 hrs.
- In Eq. (11), α was set to 1 and β was set to 3.
- In Eq. (13), the ratio of the number of disappearing participants to the total participants at each time slot, l, was 0.002Δ.
- In Eq. (50), ϵ was set to 0.001.
- In Eq. (56), the maximum period for which a worker could not access the server, η, was set to 300 s.
- In Line 16 of Algorithm 2, γ was set to 0.9.



(a) Time interval of a participant's sensing process, ξ (b) Probability of each worker becoming a non-worker, q (c) Required number of workers across the sensing area per time slot, N

Fig. 3. Effects of the parameters specifying participant behavior on the minimum convergence time C.

6.2 Parameters Settings for Controlling Participants

First, we evaluated how the parameters specifying a participant's behavior affected the stability of the number of workers in terms of the minimum convergence time C defined in Eq. (51). Then, we determined what values to use in the evaluation. Three parameters specify each participant's behavior: the time interval of a participant's sensing process, ξ; the threshold for non-worker k to become a worker, θ_k; and the probability of worker k becoming a non-worker, q_k. Among these parameters, the mean of θ_k, θ, has no effect on C, and θ_k was thus set within $[1, 3]$ to ensure that θ was nearly 2.

We examined the effects of ξ and the mean of q_k, q, in various vehicular environments by varying the number of participants, m_t, up to 10000 and the message loss ratio p_t up to 0.3. We also evaluated the influence of the required number of workers, N.

Time interval of participant's sensing process, ξ: We evaluated ξ in the range of 5 to 80 time slots. We set N to 200, which corresponded to a space interval $S = 125$ m and time interval $T = 600$ s, and we set q to a neutral value, 0.5. As shown in Fig. 3(a), for $\xi = 5$, C decreases and is hardly affected by m_t and p_t. As ξ becomes large, C also becomes large, as does the influence of p_t. A small ξ thus stabilizes the number of workers for any m_t and p_t. However, it also synchronizes all participants' sensing processes, which demands intensive computation in the server and high-bandwidth networking. We thus set ξ to 10, which is small enough for stabilization and large enough for desynchronization.

Probability of worker becoming non-worker, q: As shown in Fig. 3(b), we performed the same analysis for the mean of q_k, q, in the range of 0.2 to 0.95. The impact of m_t and p_t on worker stability is suppressed when q is larger than 0.8. Large q quickens the rate of role changes between non-workers and workers, which stabilizes temporal changes in the number of workers. Thus, we set the q_k so that their mean, q, was nearly 0.8, which is appropriate for stabilization.

Required number of workers, N: When S and T are changed to fulfill an application requirement, N changes in our model. Thus, as shown in Fig. 3(c), we evaluated the influence of N in the range of 25 to 400 in the same manner, with $\xi = 10$ and $q = 0.8$. We found that worker stability is hardly influenced by either the environmental conditions (m_t and p_t) or the application requirement (N) as long as the server sets such appropriate values of ξ and q_k for every participant.

6.3 Evaluation Results

The simulation using trace data was implemented in *Python* on a machine with a 10-core CPU (3.3 GHz) and 128 GB memory. The computation time through the 24-hr trace data was about 3 hrs, and the simulation was repeated five times for each set of parameters.

6.3.1 Details of Temporal Coverage

To evaluate our model in the time domain, especially the stability of the number of workers, we computed the mean absolute relative errors (MAREs) of the number of participants, m_t, the number of workers, $n(t)$, and the required number of workers, N. The MAREs were defined as follows:

$$\mathrm{MARE}(m_t) = \frac{1}{H}\sum_{t=1}^{H}\frac{|m_t - \hat{m}_t|}{m_t}, \tag{57}$$

$$\mathrm{MARE}(n(t)) = \frac{1}{H}\sum_{t=1}^{H}\frac{|n(t) - \hat{n}(t)|}{n(t)}, \tag{58}$$

$$\mathrm{MARE}(N) = \frac{1}{H}\sum_{t=1}^{H}\frac{|N - n(t)|}{N}, \tag{59}$$

where H is the total number of time slots, and $\hat{m}_t$ and $\hat{n}(t)$ are the estimated values of m_t and $n(t)$ by Eqs. (54) and (55).

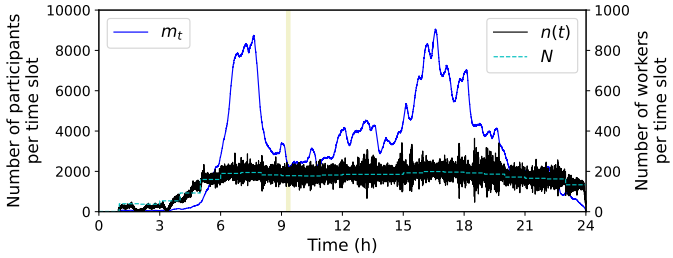
Here, we describe the workers' behavior in detail, for the underlined parameter values in Table 2. First, Fig. 4 shows the temporal changes in the numbers of workers. As seen in Fig. 4(a), while the total number of participants, m_t, has two peaks of about 9,000 at 7:37 and 16:35, the total number of workers, $n(t)$, varies closely with the required number of workers, N, which is about 180 vehicles and equals the number of workers in a steady state, as indicated in Eq. (20).

The MARE of m_t is 0.002, that of $n(t)$ is 0.10, and that of N is 0.11 (see the values at $p_t = 0$ in the left plot of Fig. 8(e)). These errors arise because each participant randomly starts a sensing process, probabilistically becomes and quits being a worker, and sometimes moves into or out of the sensing area. Nevertheless, our mechanism ensures the stability of $n(t)$, as follows.

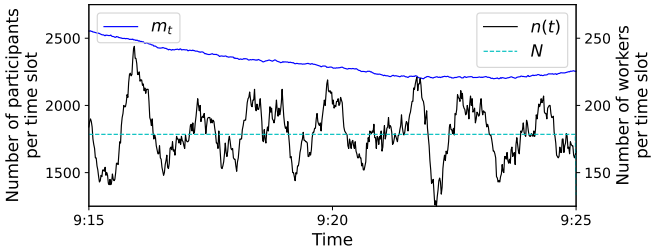
Fig. 4(b) magnifies the changes between 9:15 and 9:25 in Fig. 4(a), and it indicates that $n(t)$ oscillates around N regardless of the decrease in m_t. This oscillation corresponds to the case of $D < 0$ in Section 4.2 and demonstrates that $n(t)$ is in a state of stabilizing and approaching N at the highest rate (i.e., Conditions 1 and 2 in Section 5.1 are satisfied). This happens as the server adaptively controls each non-worker becoming a worker by continuously giving the optimal incentives shown in Fig. 4(c). Here, the amount of the weighted base incentive, $\omega_t s(t)$, also oscillates around the amount of the weighted incentive in a steady state, $\omega_t \bar{s}$, as calculated by Eq. (19). In this figure, $\omega_t \bar{s}$ increases slightly in response to the decrease in m_t shown in Fig. 4(b).

These results show that the server exactly allocates N workers on average over a cycle of changes in $n(t)$. In our evaluation setting, the dominant cycle of these changes is about 75 time slots, which does not depend on S, T, or N. Meanwhile, there is an apparent shortage of participants from 0:00 to 5:00, as shown in Fig. 4(a). Hereafter, we omit this period from our analysis.

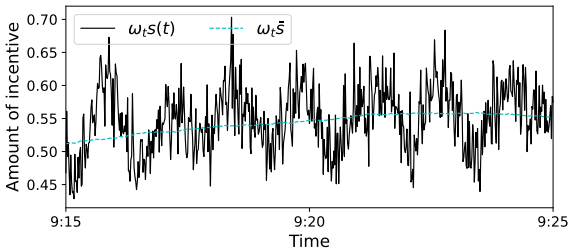
In addition, we present a comparison with an existing probabilistic control mechanism, the *Asymptotic Opportunistic algorithm for Satisfaction Index (AO-S)* [54]. The appendix describes how to apply AO-S to control $n(t)$. We evaluated AO-S throughout a day, but the changes between 9:15 and 9:25 are magnified in Fig. 5 for comparison with Fig. 4(b). In the case of AO-S, while the average of $n(t)/N$ over the day is 1.04, $n(t)$ continues to oscillate between 0 and m_t. As a result, the MARE of N is 1.7. AO-S is based on the *Asymptotically Optimal Backoff* algorithm [62], which aims to reduce network congestion in wireless LANs without knowledge of network contention. The result shows that AO-S might not be appropriate for our application scenario. In contrast, the proposed mechanism shows that the average $n(t)/N$ is 1.02 and the MARE of N is 0.11. The magnitude of the oscillation in $n(t)$ is relatively small.



(a) Total numbers of participants and workers, m_t and $n(t)$ (24 hrs)



(b) Total numbers of participants and workers, m_t and $n(t)$ (10 min)



(c) Amount of the weighted base incentive, $\omega_t s(t)$ (10 min)

Fig. 4. Temporal changes in the total numbers of participants and workers and the amount of the weighted base incentive, for the case of $S = 125$ m, $T = 600$ s, and $p_t = 0$.

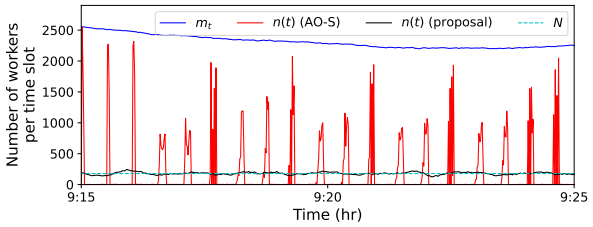


Fig. 5. Comparison of changes in the number of workers, $n(t)$, with an existing mechanism, the *Asymptotic Opportunistic algorithm for Satisfaction Index (AO-S)* [54].

6.3.2 Details of Spatial-Temporal Coverage

Spatial coverage: For the temporal view shown in Fig. 4(a), the corresponding spatial view is shown in Fig. 6. This figure plots the average sensing frequency, i.e., the average number of sensing actions per time interval T, between 5:00 and 24:00 at each location, as well as the average arrival frequency of active participants. We calculated the average sensing frequency by inverting the average time difference between consecutive sensing actions, and the average arrival frequency in a similar way. For comparison, Fig. 6 also shows the case when the participants were given non-weighted incentives, i.e., the weight coefficient $w_{j,t}$ in Eq. (8) was set to 1.

As we sorted location identifiers (LIDs) by the average arrival frequency of active participants in descending order, the arrival frequency reaches a maximum of 532 at the first location, decreases with the LIDs, and becomes less than 1 when the LID is more than 10120, indicating a location at which not enough participants arrive and the required sensing frequency cannot be satisfied. Nevertheless, the average sensing frequency is at most 1.8 at the first location, it is nearly the required value (i.e., 1) when the LID is less than 10120, and it is close to the maximum (i.e., the average arrival frequency of active participants) when the LID is more than 10120. In contrast, for the case of non-weighted incentives, the

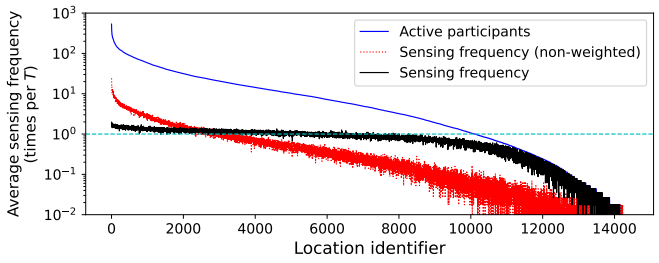
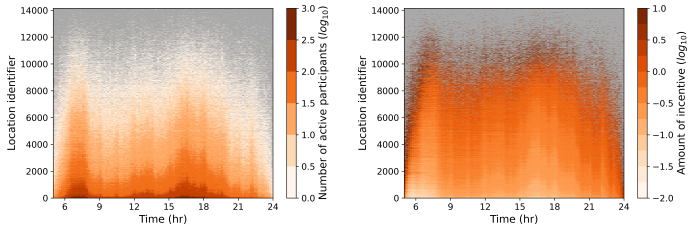
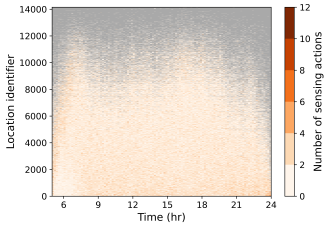


Fig. 6. Average arrival frequency of active participants and average sensing frequency at each location (in times per T), for the case of $S = 125$ m, $T = 600$ s, and $p_t = 0$.



(a) Number of active participants (b) Amount of weighted incentive



(c) Number of sensing actions

Fig. 7. Number of active participants, amount of weighted incentive, and number of sensing actions at each time interval and each location, for the case of $S = 125$ m, $T = 600$ s, and $p_t = 0$. The gray areas indicate spatial-temporal locations at which no active participants arrive.

average sensing frequency decreases in proportion to the average arrival frequency of active participants.

Spatial-temporal coverage: Lastly, we show the spatial-temporal coverage of the sensed data. The spatial-temporal views in Fig. 7 correspond to the temporal view in Fig. 4(a) and the spatial view in Fig. 6. Figure 7(a) shows the spatial-temporal distribution of the number of active participants. The gray areas indicate spatial-temporal locations at which no active participants arrive. There are two peaks of 1400 active participants during 7:30 - 7:40 and 1543 active participants during 16:30 - 16:40 at the first location. In contrast, Fig. 7(b) shows that the distribution of the amount of weighted incentive given to active participants, $\sigma(j,t)$, has two valleys at the corresponding spatial-temporal areas. Note that, for visibility, Fig. 7(b) shows the average amount of incentive over T. As a result, as shown in Fig. 7(c), the number of sensing actions is roughly 1 during each 600-s interval at every location and is little influenced by changes in the number of participants in both the space and time domains. However, there are somewhat densely or sparsely sensed locations, such as those around 6:00 in Fig. 7(c). Around that time, locations with an LID of more than 2,000 are sensed more frequently than those with an LID of less than 2,000. The reason is that the former locations have little sensed data, and thus, participants arriving at those locations are given larger incentives and become workers with higher probability than those arriving at the latter locations.

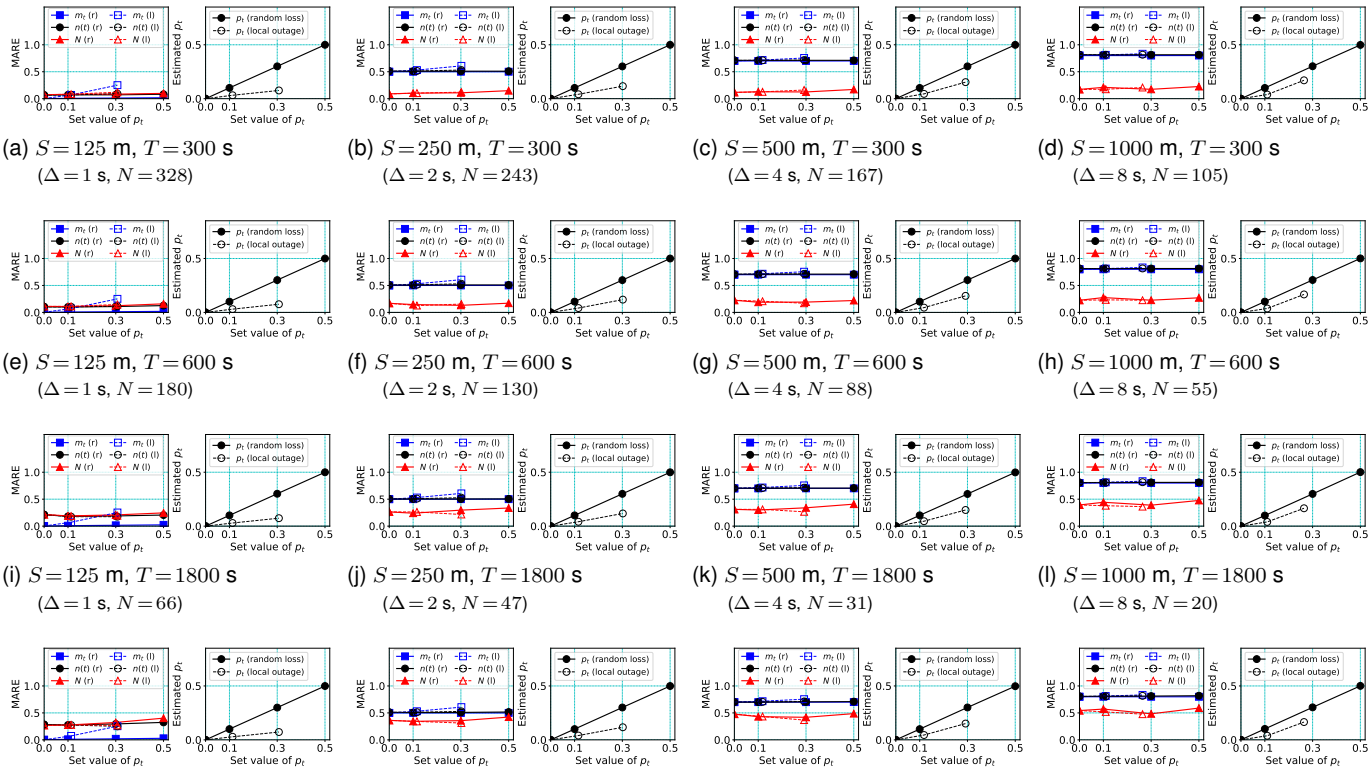
We can express the sensing distributions more precisely by analyzing the sensing frequencies (in times per T), each of which is calculated by inverting the corresponding time between two consecutive sensing actions at a location. We can also analyze the arrival frequencies of active participants in the same way. The distribution in Fig. 7(a) (excluding the gray areas) has a harmonic mean of 25 and first, second, and third quartiles of 35, 100, and 300, respectively. In contrast, the distribution in Fig. 7(c) has a harmonic mean of 1.2 and first, second, and third quartiles of 0.9, 1.8, and 4.8, respectively.

Summary: As our adaptive mechanism adjusts only the probability of each non-worker becoming a worker, it does not strictly control sensing actions over space and time, yet the resulting data coverage statistically satisfies the required spatial-temporal intervals. The server implements this mechanism by counting only the messages sent by active participants and workers; thus, the mechanism is not affected by random movements of participants. However, each participant's random start of a sensing process and probabilistic change to/from being a worker cause estimation errors in the numbers of participants and workers, m_t and $n(t)$. Nevertheless, our mechanism can maintain the stability of $n(t)$—that is, $n(t)$ always approaches the required number N and never approaches 0 or m_t, even when m_t varies greatly; this stability results in statistically uniform data coverage. This derives from the mathematical analysis in Section 4.

6.3.3 Coverage in Various Network Environments

Finally, we examine how the sensing frequencies over space and time are affected by the application requirements (i.e., the space interval S and time interval T) and the mobile network quality (i.e., the message loss ratio p_t). We especially evaluate two network situations with regard to p_t, in which messages are randomly lost over the entire area and all messages are lost in a local area.

Figure 8 shows the errors in the time domain. The left plot of each sub-figure shows the MARE of the number of participants, m_t, the MARE of the number of workers, $n(t)$, and the MARE of the required number of workers, N. The right plot of each sub-figure shows the mean estimated value of the message loss ratio p_t. Furthermore, Fig. 9 shows the sensing frequency distributions, with each sub-figure showing the cumulative distribution functions (CDFs) for varying p_t with a pair of S and T settings. The distribution shown in Fig. 7(c) corresponds to the CDF with $p_t = 0$ in Fig. 9(e). In each CDF, the ○



(a) $S=125$ m, $T=300$ s ($\Delta=1$ s, $N=328$)
(b) $S=250$ m, $T=300$ s ($\Delta=2$ s, $N=243$)
(c) $S=500$ m, $T=300$ s ($\Delta=4$ s, $N=167$)
(d) $S=1000$ m, $T=300$ s ($\Delta=8$ s, $N=105$)
(e) $S=125$ m, $T=600$ s ($\Delta=1$ s, $N=180$)
(f) $S=250$ m, $T=600$ s ($\Delta=2$ s, $N=130$)
(g) $S=500$ m, $T=600$ s ($\Delta=4$ s, $N=88$)
(h) $S=1000$ m, $T=600$ s ($\Delta=8$ s, $N=55$)
(i) $S=125$ m, $T=1800$ s ($\Delta=1$ s, $N=66$)
(j) $S=250$ m, $T=1800$ s ($\Delta=2$ s, $N=47$)
(k) $S=500$ m, $T=1800$ s ($\Delta=4$ s, $N=31$)
(l) $S=1000$ m, $T=1800$ s ($\Delta=8$ s, $N=20$)
(m) $S=125$ m, $T=3600$ s ($\Delta=1$ s, $N=34$)
(n) $S=250$ m, $T=3600$ s ($\Delta=2$ s, $N=24$)
(o) $S=500$ m, $T=3600$ s ($\Delta=4$ s, $N=16$)
(p) $S=1000$ m, $T=3600$ s ($\Delta=8$ s, $N=10$)

Fig. 8. Errors in the time domain for set values of the message loss rate p_t. The left plot of each sub-figure shows the mean absolute relative error (MARE) of the number of participants, m_t, the MARE of the number of workers, $n(t)$, and the MARE of the required number of workers, N (r: random loss, l: local outage). The right plot of each sub-figure shows the mean estimated value of the message loss ratio p_t.

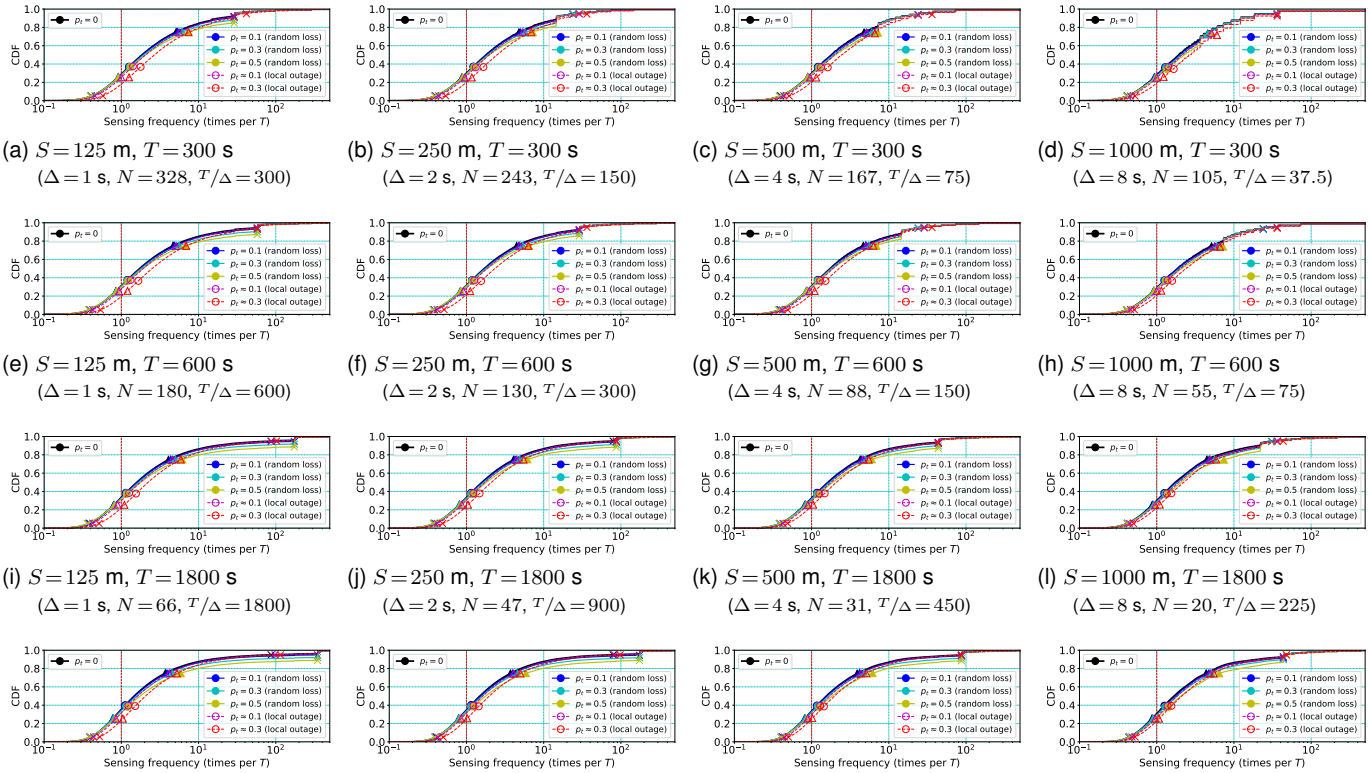
mark indicates the distribution's harmonic mean, the two $\triangle$ marks indicate the first and third quartiles (denoted by Q_1 and Q_3, respectively), and the two $\times$ marks indicate the 5th and 95th percentiles (denoted by P_5 and P_{95}, respectively).

No message loss: In each sub-figure of Fig. 8, the values at $p_t=0$ represent the errors for the given S and T when there is no message loss. When S is set to either 125, 250, 500, or 1000 m, the MARE of m_t increases to 0.002, 0.5, 0.7, or 0.8, respectively, and the MARE of $n(t)$ increases to 0.06 - 0.3, 0.5, 0.7, or 0.8, respectively. These errors have little dependence on T, because Δ becomes larger when S becomes larger, and a larger Δ more strongly increases the changes in m_t and $n(t)$. The MARE of N takes a minimum of 0.07 at $S=125$ m and $T=300$ s and a maximum of 0.54 at $S=1000$ m and $T=3600$. This error becomes larger when N becomes smaller. Overall, these errors are sufficiently small to maintain the stability of the total number of workers.

In Fig. 9, the CDFs with $p_t=0$ represent the sensing frequency distributions when there is no message loss. Although the influence of S and T on the CDFs is not so large, each CDF becomes roughly more uniform—that is, the mean gets close to 1 and the IQR shrinks—when T increases from 300 to 3600 s and S decreases from 1000 to 125 m. This is mainly due to the *law of large numbers* [63] with respect to the number of time slots per T, T/Δ. This is the number of time slots during which ξ workers are expected to be allocated at a location and one of them is expected to be active and perform a single sensing action there. When T/Δ becomes larger, a single active worker is probabilistically allocated at a location per T with higher accuracy, which makes the mean closer to 1. Accordingly, the mean of each CDF decreases from 1.3 at $S=1000$ m and $T=300$ s to 1.1 at $S=125$ m and $T=3600$ s, while T/Δ increases from 37.5 to 3600 at that time. The IQR also decreases from 3.8 at $S=1000$ m and $T=300$ s to 3.0 at $S=125$ m and $T=3600$ s. This decrease is obvious with fixed S and increased T, but it is slight or nonexistent with decreased S and fixed T. The main reason is that the uniformization effect of T/Δ becomes stronger when N is smaller (i.e, fewer workers are allocated at once).

Random message loss: As shown in Fig. 8, when messages are randomly lost over the area, the MAREs of m_t, $n(t)$, and N hardly change when p_t increases from 0 to 0.5. The server can accurately estimate p_t for each setting of S and T, even when p_t is set to a large value. Then, the overall stability of the total number of workers is not affected by the random message loss.

In Fig. 9, the CDFs with random loss show the same tendencies with respect to S and T as those when no messages are lost. An increase in p_t causes increases in Q_3 and P_{95}. For example, while the average Q_3 is 4.6 when p_t is 0, it increases to 4.9 (106%), 5.5 (119%) and 6.8 (148%)



(a) $S = 125$ m, $T = 300$ s ($\Delta = 1$ s, $N = 328$, $T/\Delta = 300$)
(b) $S = 250$ m, $T = 300$ s ($\Delta = 2$ s, $N = 243$, $T/\Delta = 150$)
(c) $S = 500$ m, $T = 300$ s ($\Delta = 4$ s, $N = 167$, $T/\Delta = 75$)
(d) $S = 1000$ m, $T = 300$ s ($\Delta = 8$ s, $N = 105$, $T/\Delta = 37.5$)
(e) $S = 125$ m, $T = 600$ s ($\Delta = 1$ s, $N = 180$, $T/\Delta = 600$)
(f) $S = 250$ m, $T = 600$ s ($\Delta = 2$ s, $N = 130$, $T/\Delta = 300$)
(g) $S = 500$ m, $T = 600$ s ($\Delta = 4$ s, $N = 88$, $T/\Delta = 150$)
(h) $S = 1000$ m, $T = 600$ s ($\Delta = 8$ s, $N = 55$, $T/\Delta = 75$)
(i) $S = 125$ m, $T = 1800$ s ($\Delta = 1$ s, $N = 66$, $T/\Delta = 1800$)
(j) $S = 250$ m, $T = 1800$ s ($\Delta = 2$ s, $N = 47$, $T/\Delta = 900$)
(k) $S = 500$ m, $T = 1800$ s ($\Delta = 4$ s, $N = 31$, $T/\Delta = 450$)
(l) $S = 1000$ m, $T = 1800$ s ($\Delta = 8$ s, $N = 20$, $T/\Delta = 225$)
(m) $S = 125$ m, $T = 3600$ s ($\Delta = 1$ s, $N = 34$, $T/\Delta = 3600$)
(n) $S = 250$ m, $T = 3600$ s ($\Delta = 2$ s, $N = 24$, $T/\Delta = 1800$)
(o) $S = 500$ m, $T = 3600$ s ($\Delta = 4$ s, $N = 16$, $T/\Delta = 900$)
(p) $S = 1000$ m, $T = 3600$ s ($\Delta = 8$ s, $N = 10$, $T/\Delta = 450$)

Fig. 9. Sensing frequency distributions in low-quality mobile network environments. The p_t values are set values of the message loss ratio. In each distribution, the ○ mark indicates the harmonic mean, the two △ marks indicate the first and third quartiles, and the two × marks indicate the 5th and 95th percentiles.

when p_t increases to 0.1, 0.3 and 0.5, respectively. This is because, when workers cannot communicate with the server, they cannot become a non-worker and continue to perform sensing actions until they can access the server again. Such consecutive sensing actions give the maximum sensing frequency, $T/\xi\Delta$; this value often appears at P_{95}. When p_t becomes larger and more messages are randomly lost, the maximum sensing frequency is reached more often, which increases Q_3 and P_{95}. Then, the mean also increases; nevertheless, this increase is relatively restricted. While the average of the mean is 1.21 when p_t is 0, it increases to 1.23 (101%), 1.24 (102%) and 1.28 (105%) when p_t increases to 0.1, 0.3 and 0.5, respectively.

Local network outages: Lastly, we evaluate the case when a network outage occurs in a local area. As shown in Fig. 8, the MAREs of $n(t)$ and N are little affected by p_t. However, the MARE of m_t increases when S (as well as Δ) becomes smaller, because p_t is more underestimated. For example, in the case of $S = 125$ m, p_t is estimated as 0.07 when it is set to 0.3. At that time, the MARE of m_t increases by 0.25 from when p_t is set to 0. This is because, while participants are dispersed over the area, almost all workers stay outside the outage area and fewer workers enter it, especially when Δ is small. As the message loss rate p_t is calculated by counting the messages sent by workers, errors arise in the estimated total numbers of participants. In this case, Eq. (4) does not hold.

In Fig. 9, each of the CDFs with local outages shows the sensing frequency distribution outside the corresponding outage area shown in Fig. 2. The CDFs vary with S and T similarly to the case of no message loss. However, unlike the case of random message loss, when p_t becomes large, not only Q_3 but also the mean of each CDF increases. While the average of the mean is 1.2 for $p_t = 0$, it increases to 1.3 (109%) for $p_t \approx 0.1$ and 1.6 (130%) for $p_t \approx 0.3$. Moreover, while the average Q_3 is 4.6 when $p_t = 0$, it increases to 5.1 (111%) for $p_t \approx 0.1$ and 6.3 (137%) for $p_t \approx 0.3$. These differences are enlarged when the outage area is larger or Δ is set to a smaller value. We attribute this to the proposed mechanism: although the server adds up the numbers of sensed data instances at each location and time, it counts the current numbers of participants and workers but does not monitor their locations. Nevertheless, the proposed mechanism ensures the stability of $n(t)$, as demonstrated by the result that the MARE of N does not increase during network outages. Note that the amount of sensed data varies greatly among the locations inside each network outage area. However, the scope of this paper does not include how to control the participants without exchanging messages.

Summary: When the given S and T change, as the number of time slots during which a single sensing action is performed increases, the sensing frequency distribu-

tion becomes more uniform. This effect becomes more obvious as fewer workers are simultaneously allocated at that time. This result arises from our probabilistic mechanism. Furthermore, the server ensures the stability of the number of workers when messages are frequently lost. When messages are randomly lost, the Q_3 and P_{95} of the sensing frequency distribution increase, but the mean remains nearly 1. This means that the resulting data coverage statistically fulfills the required spatial-temporal intervals. However, when all messages are lost in a local area and fewer workers stay in that area, the sensing frequencies outside the area increase. This is because the proposed mechanism considers the numbers of participants and workers but not their locations.

7 Conclusion

The key challenge in this paper is to achieve uniform spatial-temporal coverage of sensed data in vehicular MCS without any prior knowledge about participants. Thus, we have proposed an adaptive mechanism for worker allocation by applying the response-threshold model. This mechanism successively provides optimal incentives to participants to ensure the stability of the current number of workers on the basis of mathematical analysis. Through simulations using city-wide vehicle trace data over a day, we have demonstrated that our mechanism operates in a probabilistic manner but statistically satisfies the required spatial and temporal intervals. Even when there are locations whose number of participants is up to 1500 times higher than the required number of workers, the spatial-temporal coverage of sensing frequencies results in a distribution with a mean of 1.2 and an IQR around 4 for the required interval. Moreover, although our mechanism only counts messages from participants and data from workers, randomly lost messages hardly affect the mean of the resulting sensing frequencies. Messages lost in a local area also have little influence on the sensing frequency distribution outside the area as long as the area is relatively small.

To clarify the effect on the spatial-temporal data coverage, this paper has focused on a simple sensing scenario in which homogeneous tasks are uniformly distributed across the entire area at regular intervals. Hence, for our future work, we will need to consider a more complex scenario. We will also need to get deeper insight from the mathematical analysis, such as the impact of oscillation in the number of workers on the data coverage.

8 Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP21K11861.

References

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.

[2] E. Massaro, C. Ahn, C. Ratti, P. Santi, R. Stahlmann, A. Lamprecht, M. Roehder, and M. Huber, "The car as an ambient sensing platform [point of view]," *Proc. of the IEEE*, vol. 105, no. 1, pp. 3–7, Jan. 2017.

[3] R. Hussain and S. Zeadally, "Autonomous cars: Research results, issues, and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1275–1313, Secondquarter 2019.

[4] H. Qiu, J. Chen, S. Jain, Y. Jiang, M. McCartney, G. Kar, F. Bai, D. K. Grimm, M. Gruteser, and R. Govindan, "Towards robust vehicular context sensing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 1909–1922, Mar. 2018.

[5] R. Du, P. Santi, M. Xiao, A. V. Vasilakos, and C. Fischione, "The sensable city: A survey on the deployment and management for smart city monitoring," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1533–1560, Secondquarter 2019.

[6] P. H. Rettore, G. Maia, L. A. Villas, and A. A. F. Loureiro, "Vehicular data space: The data point of view," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2392–2418, thirdquarter 2019.

[7] Japan Meteorological Agency, "AMeDAS," https://www.jma.go.jp/jma/en/Activities/amedas/amedas.html, accessed Sep. 20, 2022.

[8] MLIT Japan, "Radar observation of precipitation for river management in Japan," https://www.mlit.go.jp/river/pamphlet_jirei/pdf/xrain_en.pdf?0930, Sep. 2013, accessed Sep. 20, 2022.

[9] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2419–2465, thirdquarter 2019.

[10] Y. Liu, L. Kong, and G. Chen, "Data-oriented mobile crowdsensing: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2849–2885, thirdquarter 2019.

[11] J. Liu, H. Shen, H. S. Narman, W. Chung, and Z. Lin, "A survey of mobile crowdsensing techniques: A critical component for the Internet of Things," *ACM Trans. Cyber-Phys. Syst.*, vol. 2, no. 3, pp. 1–26, Jun. 2018.

[12] A. Longo, M. Zappatore, M. Bochicchio, and S. B. Navathe, "Crowd-sourced data collection for urban monitoring via mobile sensors," *ACM Trans. Internet Technol.*, vol. 18, no. 1, Oct. 2017.

[13] F. Kalim, J. P. Jeong, and M. U. Ilyas, "CRATER: A crowd sensing application to estimate road conditions," *IEEE Access*, vol. 4, pp. 8317–8326, 2016.

[14] C. Xiang, P. Yang, C. Tian, L. Zhang, H. Lin, F. Xiao, M. Zhang, and Y. Liu, "CARM: Crowd-sensing accurate outdoor RSS maps with error-prone smartphone measurements," *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2669–2681, Nov. 2016.

[15] C. T. Calafate, K. Cicenia, O. Alvear, J. C. Cano, and P. Manzoni, "Estimating rainfall intensity by using vehicles as sensors," in *2017 Wireless Days*, Mar. 2017, pp. 21–26.

[16] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 2254–2262.

[17] H. Xiong, D. Zhang, L. Wang, and H. Chaouchi, "EMC3: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint," *IEEE Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1355–1368, Jul. 2015.

[18] M. Xiao, J. Wu, He Huang, L. Huang, and Chang Hu, "Deadline-sensitive user recruitment for mobile crowdsensing with probabilistic collaboration," in *Proc. IEEE 24th Int. Conf. Netw. Protocols (ICNP)*, Nov. 2016, pp. 1–10.

[19] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 17, no. 1, pp. 16–28, Jan. 2018.

[20] X. Zhu, Y. Luo, A. Liu, W. Tang, and M. Z. A. Bhuiyan, "A deep learning-based mobile crowdsensing scheme by predicting vehicle mobility," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–12, Jul. 2020.

[21] A. Seetharam and P. Walker, "An empirical characterization of cellular network performance," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, Feb. 2016, pp. 1–5.

[22] I. Bocharova, B. Kudryashov, M. Rabi, N. Lyamin, W. Dankers, E. Frick, and A. Vinel, "Characterizing packet losses in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8347–8358, Sep. 2019.

[23] H. Berghel, "Vehicle telematics: The good, bad and ugly," *Computer*, vol. 52, no. 1, pp. 66–70, Jan. 2019.

[24] E. Bonabeau, G. Theraulaz, and J.-L. Deneubourg, "Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies," *Proc. the Royal Society of London B: Biological Sciences*, vol. 263, no. 1376, pp. 1565–1569, Nov. 1996.

[25] C. Zheng and D. C. Sicker, "A survey on biologically inspired algorithms for computer networking," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1160–1191, Thirdquarter 2013.

[26] Z. Zhang, K. Long, J. Wang, and F. Dressler, "On swarm intelligence inspired self-organized networking: Its bionic mechanisms, designing principles and optimization approaches," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 513–537, Firstquarter 2014.

[27] Y. Taniguchi, N. Wakamiya, M. Murata, and T. Fukushima, "An autonomous data gathering scheme adaptive to sensing requirements for industrial environment monitoring," in *Porc. 2008 New Technol., Mobility and Security*, Nov. 2008, pp. 1–5.

[28] P. Janacik, T. Heimfarth, and F. Rammig, "Emergent topology control based on division of labour in ants," in *Proc. 20th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, vol. 1, Apr. 2006, pp. 733–740.

[29] E. Rapti, C. Houstis, E. Houstis, and A. Karageorgos, "A bio-inspired service discovery and selection approach for iot applications," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jun. 2016, pp. 868–871.

[30] W. Lee, N. Vaughan, and D. Kim, "Task allocation into a foraging task with a series of subtasks in swarm robotic system," *IEEE Access*, vol. 8, pp. 107 549–107 561, 2020.

[31] T. Iwai, N. Wakamiya, and M. Murata, "Characteristic analysis of response threshold model and its application for self-organizing network control," in *Revised Selected Papers of 7th IFIP TC 6 Int. Workshop Self-Organizing Syst. (IWSOS)*, May 2013, pp. 27–38.

[32] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell, "Urban sensing systems: Opportunistic or participatory?" in *Proc. 9th Workshop Mobile Comput. Syst. Appl. (HotMobile)*, Feb. 2008, pp. 11–16.

[33] S. Xu, X. Chen, X. Pi, C. Joe-Wong, P. Zhang, and H. Y. Noh, "iLOCuS: Incentivizing vehicle mobility to optimize sensing distribution in crowd sensing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 8, pp. 1831–1847, Aug. 2020.

[34] G. Fan, H. Jin, Q. Liu, W. Qin, X. Gan, H. Long, L. Fu, and X. Wang, "Joint scheduling and incentive mechanism for spatio-temporal vehicular crowd sensing," *IEEE Trans. Mobile Comput.*, vol. 20, no. 4, pp. 1449–1464, Apr. 2021.

[35] X. Tao and W. Song, "Location-dependent task allocation for mobile crowdsensing with clustering effect," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1029–1045, Feb. 2019.

[36] Z. He, J. Cao, and X. Liu, "High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 2542–2550.

[37] G. Sun, Y. Wang, X. Ding, and R. Hu, "Cost-fair task allocation in mobile crowd sensing with probabilistic users," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 403–415, Feb. 2021.

[38] E. Wang, Y. Yang, J. Wu, D. Luan, and H. Wang, "Worker recruitment strategy for self-organized mobile social crowdsensing," in *Proc. 27th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2018, pp. 1–9.

[39] L. Wang, R. Guan, Z. Yu, E. Wang, B. Guo, and Q. Han, "Failure-aware mobile crowd sensing: A social relationship-based transfer approach," *IEEE Access*, vol. 7, pp. 186 615–186 625, 2019.

[40] J. Wang, F. Wang, Y. Wang, D. Zhang, L. Wang, and Z. Qiu, "Social-network-assisted worker recruitment in mobile crowd sensing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 7, pp. 1661–1673, Jul. 2019.

[41] Y. Yang, W. Liu, E. Wang, and J. Wu, "A prediction-based user selection framework for heterogeneous mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 11, pp. 2460–2473, Nov. 2019.

[42] B. Guo, Y. Liu, L. Wang, V. O. K. Li, J. C. K. Lam, and Z. Yu, "Task allocation in spatial crowdsourcing: Current state and future directions," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1749–1764, Jun. 2018.

[43] W. Liu, Y. Yang, E. Wang, and J. Wu, "User recruitment for enhancing data inference accuracy in sparse mobile crowdsensing," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 1802–1814, Mar. 2020.

[44] X. Cao, Y. Li, J. Han, P. Yang, F. Lyu, D. Guo, and X. Shen, "Online worker selection towards high quality map collection for autonomous driving," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[45] L. Wang, Z. Yu, D. Zhang, B. Guo, and C. H. Liu, "Heterogeneous multi-task assignment in mobile crowdsensing using spatiotemporal correlation," *IEEE Trans. Mobile Comput.*, vol. 18, no. 1, pp. 84–97, Jan. 2019.

[46] X. Wang, R. Jia, X. Tian, X. Gan, L. Fu, and X. Wang, "Location-aware crowdsensing: Dynamic task assignment and truth inference," *IEEE Trans. Mobile Comput.*, vol. 19, no. 2, pp. 362–375, Feb. 2020.

[47] J. Wang, F. Wang, Y. Wang, L. Wang, Z. Qiu, D. Zhang, B. Guo, and Q. Lv, "HyTasker: Hybrid task allocation in mobile crowd sensing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 3, pp. 598–611, Mar. 2020.

[48] X. Wang, W. Wu, and D. Qi, "Mobility-aware participant recruitment for vehicle-based mobile crowdsensing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4415–4426, May 2018.

[49] X. Chen, S. Xu, J. Han, H. Fu, X. Pi, C. Joe-Wong, Y. Li, L. Zhang, H. Y. Noh, and P. Zhang, "PAS: Prediction-based actuation system for city-scale ridesharing vehicular mobile crowdsensing," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3719–3734, May 2020.

[50] X. Zhang, Z. Yang, Y. Gong, Y. Liu, and S. Tang, "SpatialRecruiter: Maximizing sensing coverage in selecting workers for spatial crowdsourcing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5229–5240, Jun. 2017.

[51] C. Fiandrino, F. Anjomshoa, B. Kantarci, D. Kliazovich, P. Bouvry, and J. N. Matthews, "Sociability-driven framework for data acquisition in mobile crowdsensing over fog computing platforms for smart cities," *IEEE Trans. Sustain. Comput.*, vol. 2, no. 4, pp. 345–358, Oct. 2017.

[52] D. Zhao, H. Ma, L. Liu, and J. Zhao, "On opportunistic coverage for urban sensing," in *Poc. IEEE 10th Int. Conf. Mobile Ad-Hoc Sensor Syst.*, Oct. 2013, pp. 231–239.

[53] M. Fiore, A. Nordio, and C. Chiasserini, "Driving factors toward accurate mobile opportunistic sensing in urban environments," *IEEE Trans. Mobile Comput.*, vol. 15, no. 10, pp. 2480–2493, Oct. 2016.

[54] F. Montori, L. Bedogni, and L. Bononi, "Distributed data collection control in opportunistic mobile crowdsensing," in *Proc. 3rd Workshop Experiences with the Design and Implementation of Smart Objects (SMARTOBJECTS)*, Oct. 2017, pp. 19–24.

[55] W. Alasmary and S. Valaee, "Crowd sensing in vehicular networks using uncertain mobility information," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11 227–11 238, Nov. 2019.

[56] A. Musa, J. Biagioni, and J. Eriksson, "Trading off accuracy, timeliness, and uplink usage in online GPS tracking," *IEEE Trans. Mobile Comput.*, vol. 15, no. 8, pp. 2124–2136, Aug 2016.

[57] M. W. Hirsch, S. Smale, and R. L. Devaney, *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Academic Press, 2012.

[58] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.

[59] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," *IEEE Trans. Mobile Comput.*, vol. 13, no. 5, pp. 1061–1075, May 2014.

[60] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. WieBner, "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582.

[61] "TAPASCologne," https://sumo.dlr.de/docs/Data/Scenarios/TAPASCologne.html, accessed Jul. 19, 2021.

[62] L. Bononi, M. Conti, and E. Gregori, "Runtime optimization of IEEE 802.11 wireless LANs performance," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 1, pp. 66–80, Jan 2004.

[63] M. DeGroot and M. Schervish, *Probability and Statistics*. Addison-Wesley, Jan. 2011.

Yukio Ogawa received an M.S. degree in science from Nagoya University, Japan, in 1994, and a Ph.D. degree in information science and technology from Osaka University, Japan, in 2012. He joined the Hitachi Central Research Laboratory in 1994. He is currently an associate professor at the Center for ICT Education, Muroran Institute of Technology, Japan. His research interests include distributed architectures for cloud systems and the Internet of Things. He is a member of the IEEE, ACM, and IEICE.

Go Hasegawa received M.E. and D.E. degrees in information and computer sciences from Osaka University, Japan, in 1997 and 2000, respectively. From July 1997 to June 2000, he was a research assistant at the Graduate School of Economics, Osaka University. From 2000 to 2018, he was an associate professor at the Cybermedia Center, Osaka University. He is now a professor at the Research Institute of Electrical Communication, Tohoku University. His research is in information network architecture. He is a member of the IEEE and IEICE.

Masayuki Murata received M.E. and D.E. degrees in information and computer science from Osaka University, Japan, in 1984 and 1988, respectively. In April 1984, he joined the Tokyo Research Laboratory, IBM Japan, as a researcher. From September 1987 to January 1989, he was an assistant professor at the Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University, where he became a professor in the Graduate School of Engineering Science in April 1999. He has been with the Graduate School of Information Science and Technology since April 2004. His research interests include information network architecture and performance modeling and evaluation. He is a member of the ACM and IEICE.