Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5913981, 13 pages
https://doi.org/10.1155/2018/5913981

Research Article

On-Demand Mobile Data Collection in Cyber-Physical Systems

Liang He,[1] Linghe Kong,[2] Jun Tao,[3] Jingdong Xu,[4] and Jianping Pan[5]

[1] *University of Colorado, Denver, CO, USA*
[2] *Shanghai Jiaotong University, Shanghai, China*
[3] *Southeast University, Nanjing, Jiangsu, China*
[4] *Nankai University, Tianjin, China*
[5] *University of Victoria, Victoria, BC, Canada*

Correspondence should be addressed to Linghe Kong; linghe.kong@sjtu.edu.cn

Received 10 October 2017; Revised 16 January 2018; Accepted 19 March 2018; Published 30 April 2018

Academic Editor: Yin Zhang

The collection of sensory data is crucial for cyber-physical systems. Employing mobile agents (MAs) to collect data from sensors offers a new dimension to reduce and balance their energy consumption but leads to large data collection latency due to MAs' limited velocity. Most existing research effort focuses on the offline *mobile data collection* (MDC), where the MAs collect data from sensors based on preoptimized tours. However, the efficiency of these offline MDC solutions degrades when the data generation of sensors varies. In this paper, we investigate the on-demand MDC; that is, MAs collect data based on the real-time data collection requests from sensors. Specifically, we construct queuing models to describe the *First-Come-First-Serve*-based MDC with a single MA and multiple MAs, respectively, laying a theoretical foundation. We also use three examples to show how such analysis guides online MDC in practice.

1. Introduction

Collecting data from sensors is a core function of large cyber-physical systems such as wind farm and smart grid [1–4]. Traditional data collection approaches rely on the wireless communications between sensor nodes and the sink, excessively consuming nodes' limited energy supply and leading to their unbalanced energy consumption. Adopting mobile agents (MAs) for data collection, that is, *mobility-assisted data collection* (MDC), reduces and balances the communication loads of nodes (and thus their energy consumption) [5–7]. Also, with MAs' controllable mobility, the communications and networking become possible even in sparse networks via the *store-carry-forward* approach. A real-life MDC example is the NEPTUNE project—a seabed crawler is deployed to collect sensory data from other underwater experiment nodes [8]. However, MDC leads to large data collection latency due to MAs' limited velocity, degrading the realtimeness of the collected data, and may cause data loss due to the buffer overflow at sensor nodes.

Much research effort on *offline* MDC exists in the literature, where the MAs *periodically* collect data from nodes with a preoptimized path [9]. On the other end of the spectrum, *on-demand* MDC—sensor nodes send data collection requests to the MAs when they have data to report and the MAs only visit (and collect data from) such requesting nodes—is a more efficient approach to exploit MAs' limited mobility resource, especially for event-driven systems with diverse data generation among sensors [10, 11]. The challenge in the on-demand MDC, however, is to determine how the MAs should collect data from nodes without a priori information on future data collection demands.

On-demand MDC shows clear queuing behavior. In this paper, we formulate two queuing models to capture the on-demand MDC with the *First-Come-First-Serve* (FCFS) discipline (FCFS is a simple and natural choice to maintain request fairness and is preferred in certain node-centric scenarios.), on the cases where a single MA and multiple MAs are deployed for data collection, respectively, and corresponding analytical results on the data collection performance are

derived. Furthermore, we use three examples to show how the analysis guides the on-demand MDC in practice: (i) how to use multiple MAs? (ii) When to request data collection? (iii) How likely the requests combination—that is, collect data from multiple sensor nodes at the same location—would happen via the wireless communication between the MAs and nodes? The contributions of this paper include the following:

(i) Formulation of an $M/G/1$ queuing model to capture and analytically evaluate the on-demand MDC when a single MA is deployed for data collection (Section 4)

(ii) An $M/G/c$ queuing model for the case when multiple MAs are deployed, based on which the data collection performance is explored via approximation (Section 5)

(iii) Three examples to show how the analysis guides the on-demand MDC in practice (Section 7)

The rest of this paper is organized as follows. The literature on MDC is briefed in Section 2. We formulate the problem in Section 3. The on-demand MDC with a single MA and multiple MAs is investigated in Sections 4 and 5 and evaluated in Section 6. The practical guidance is presented in Section 7, followed by further discussions in Section 8. The paper concludes in Section 9.

2. Related Work

Observing the advantages of MDC over traditional data collection approaches (e.g., via direct communication or multihop forwarding), much effort has been made to explore the *MDC with a single MA* [7, 11, 12]. For example, Sugihara and Gupta investigated the MDC with the objective of minimizing MA's travel distance in [9]. Zhao et al. proposed a three-layer framework for the offline MDC, including the sensor layer, cluster head layer, and MA layer. The MA collects data according to a preoptimized tour with dual antennas [13]. A unified framework for analyzing the MA's mobility and data collection latency was presented and solutions to the involved subproblems were proposed in [14]. An MA-tracking protocol was proposed in [15], in which the routing structure of data collection requests was additively updated with MA's movement. The joint energy replenishment and data collection was investigated in [16], with the consideration of various sources of energy consumption and time-varying nature of energy replenishment. The colocation of MA and wireless charger has been investigated in [17], with the objective of ensuring sustainable and lossless system operation.

Scalability is a critical bottleneck when only a single MA is used and a potential mitigation is to employ multiple MAs for data collection. An early investigation on the scenario of multiple MAs is [18], where the MAs travel along fixed tracks to collect data from nodes with the consideration of load balancing. A motion planning algorithm for the MAs was proposed in [19], which minimizes the number of MAs according to the constraints in distance and time. This work was extended for applications with strict distance/time constraints, and a data-gathering algorithm with multiple MAs was proposed in [20]. More detailed information on MDC is found in [21] and the references therein.

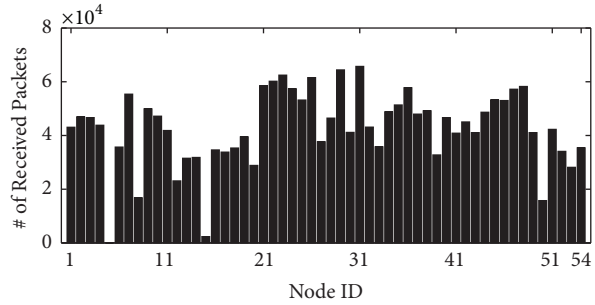

Figure 1: Diverse data generation among sensor nodes (original data from [24]).

Most of these existing efforts focus on the offline MDC, while we tackle the on-demand MDC in this paper. Although similar scenarios have been investigated in [12, 22, 23], our queue-based analytical framework provides detailed insights into the data collection process such as system size, queuing time, and response time, and these analytical insights guide the MDC in practice.

3. Preliminaries

3.1. On-Demand MDC. In many offline MDC solutions, the MAs periodically collect data from sensor nodes based on preoptimized tours [9]. These solutions perform well when nodes generate data at similar paces but degrade dramatically when the data generation at nodes varies—MAs may visit nodes with little or no data to report. Unfortunately, many event-driven systems demonstrate such diverse data generations [10]. For example, Figure 1 plots the number of packets received from 54 sensor nodes in a trace provided by Intel Berkeley Research Lab [24], showing clear diversity among nodes. Targeting on these scenarios with diverse data generations, we investigate the *on-demand* MDC, where the MAs collect data based on real-time demands from sensor nodes in this paper.

3.2. Network Model. We consider the scenario where controllable MAs collect data from stationary sensor nodes randomly deployed in a square sensing field [9] (our model formulation and analysis are also applicable to sensing fields of other shapes, as will be explained in Section 4). Sensor nodes monitor their surrounding environments, store the gathered data in their buffers, and send out data collection requests to MAs when their buffers are to be full [10]. The data collection requests can be delivered to the MAs via existing MA-tracking protocols [15]. Because the typical data relay speed is much faster than the MAs' travel, we assume that the time since a request is sent by a sensor node till it is received by MAs is short and negligible [12]. Note that, instead of using these MA-tracking protocols to upload the sensory data directly, which are normally of much larger volume, only data

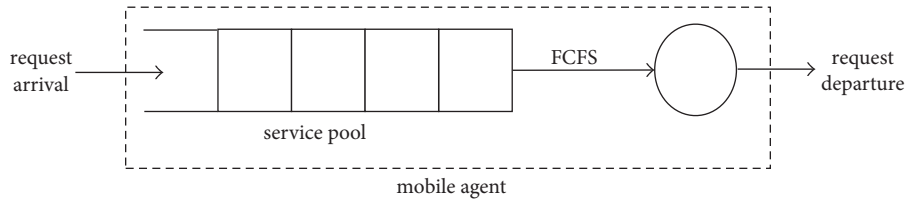

Figure 2: Queuing model when a single MA is deployed for data collection.

collection requests are forwarded to MAs to reduce nodes' communication loads.

MAs maintain a service pool to store the received data collection requests and serve them according to the *First-Come-First-Serve* (FCFS) discipline. By *serving* the request, we mean one of the MAs moves to the corresponding requesting node to collect its data via short-range wireless communications. FCFS, albeit not the optimal solution for on-demand MDC, is a classic scheduling discipline known to be fair for clients [25]. Moreover, the theoretically established data collection performance with FCFS serves as a good baseline for the evaluation of more sophisticated MDC solutions.

Also, it is not necessary for the MA to travel to the exact location of nodes to collect data because of the wireless communications between the MA and the nodes [12, 26]. This way, the MA can potentially collect data from all nodes within its communication range at a single site. The impact of the communication range on MDC relies on both the field size and node density. To establish a theoretic foundation, we do not directly incorporate the communication range into our modeling; however, we investigate and evaluate its impact in Section 7.3.

3.3. Problem Statement. The on-demand MDC is dynamic both temporally and spatially, that is, when a data collection request will be received and where (which sensor node) the request is from. This dynamic property not only shifts our objective from MAs' optimal path planning (as in the *offline* MDC) to the design of efficient real-time service disciplines to select the next request (i.e., the requesting node) to serve (i.e., collect data from) but also makes the MDC hard to capture and thus its performance challenging to evaluate. In this paper, we evaluate the on-demand MDC via a queue-based analytical approach.

4. MDC with a Single MA

We investigate the on-demand MDC in this section and the following ones. Specifically, in this section, an $M/G/1$ queuing model is constructed to capture the MDC when a single MA is deployed.

4.1. Construction of the M/G/1 Model. The on-demand MDC shows clear queuing behavior, inspiring us to capture it with a queuing model—the MA serves as the server and the data collection requests from sensors are treated as the clients (Figure 2). For any queuing system, two fundamental components to be characterized are the client arrival and departure.

4.1.1. Request Arrival. The aggregated request arrival process at the MA is the superposition of n requesting processes of individual sensors, where n is the number of sensors in the system. This way, the request arrival at the MA can be captured by a Poisson process according to *Palm-Khintchine theorem* [27]. This is because (i) for a stable data collection process the number of sensors in the system is large when compared with the number of to-be-served requests at any given time instance, indicating low dependency in their requesting of data collections; (ii) the probability for a sensor to initiate a data collection request at a specific time instance is small. Theoretically, if the client population of a queuing system is large and the probability by which clients arrive at the queue is low at a specific time, the arrival process can be adequately modeled as Poisson [28]. We will further statistically verify this Poisson arrival of requests in Section 6.

Assume that a memory buffer of size B is equipped for each sensor and its asymptotic data generation rates are f_i $(i = 1, 2, \ldots, n)$. The request arrival rate λ can be approximated as

$$\lambda \approx \frac{1}{B}\sum_{i=1}^{n} f_i. \tag{1}$$

Essentially, (1) is a lower bound on the aggregated requests arrival rate because the requesting node would not request again before its data has been collected. Denoting λ^* as the true request arrival rate, we have

$$(\lambda^* - \lambda) \longrightarrow 0^+ \quad \text{as } n \longrightarrow \infty. \tag{2}$$

4.1.2. Request Departure. The MA travels to the requesting node to collect the data therein. Because the data propagation speed is much faster than the MA's travel speed, we simplify our investigation by assuming a negligible data transmission latency. This way, the departure process, or the service time of clients, can be characterized by the time from the service completion of the current request to the time when the MA moves to the next requesting node.

As the previous data collection site is also the starting location when the MA serves the next request, the service time of consecutively served requests seems not to be independent. However, denoting the sequence of service times as $\{t_1, t_2, t_3, \ldots\}$, if we examine only at every second element of

the original process, it is clear that $\{t_1, t_3, \ldots\}$ are independent of each other, and the distribution-ergodic property of this subprocess can be observed [29]. The same is true for subprocess $\{t_2, t_4, \ldots\}$. The distribution-ergodic property still holds if we combine these two subprocesses because their asymptotic behaviors do not change after the combination. A demonstration on this distribution-ergodic property is shown in Figure 3. This means that if we identify the time distribution when the MA travels between consecutively served nodes, we can use it as the service time distribution for the queuing model over a long time period.

From existing results in geometrical probability [11], the distance distribution between two random locations in a unit square is

$$f_{\mathscr{D}}(x) = \begin{cases} 2x\left(\pi - 4x + x^2\right) & 0 \le x \le 1 \\ 2x\left[2\sin^{-1}\left(\frac{1}{x}\right) - 2\sin^{-1}\sqrt{1 - \frac{1}{x^2}} + 4\sqrt{x^2 - 1} - x^2 - 2\right] & 1 \le x \le \sqrt{2} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Thus, with MA travel speed v, the service time distribution can be derived as

$$\begin{aligned} F_{\mathscr{S}}(t) &= P\{\text{service time} < t\} = P\{\text{travel distance} < vt\} \\ &= \begin{cases} \int_0^{vt} 2x\left(\pi - 4x + x^2\right) dx & t \le \frac{1}{v} \\ \int_0^1 2x\left(\pi - 4x + x^2\right) dx + \int_1^{vt} 2x\left[2\sin^{-1}\left(\frac{1}{x}\right) - 2\sin^{-1}\sqrt{1 - \frac{1}{x^2}} - x^2 - 2\right] dx & \frac{1}{v} < t \le \frac{\sqrt{2}}{v} \\ 1 & t > \frac{\sqrt{2}}{v}. \end{cases} \end{aligned} \tag{4}$$

Its expectation, variance, and coefficient of variation are

$$\begin{aligned} \mathbb{E}[\mathscr{S}] &= \mathbb{E}\left[\frac{\mathscr{D}}{v}\right] = \frac{1}{v}\mathbb{E}[\mathscr{D}], \\ \mathbb{V}[\mathscr{S}] &= \mathbb{V}\left[\frac{\mathscr{D}}{v}\right] = \frac{1}{v^2}\mathbb{V}[\mathscr{D}], \\ \mathbb{C}\mathrm{ov}[\mathscr{S}] &= \frac{\sqrt{\mathbb{V}[\mathscr{S}]}}{\mathbb{E}[\mathscr{S}]} = \frac{\sqrt{\mathbb{V}[\mathscr{D}]}}{\mathbb{E}[\mathscr{D}]} = \mathbb{C}\mathrm{ov}[\mathscr{D}]. \end{aligned} \tag{5}$$

After characterizing the request arrival and departure, we can model the on-demand MDC with a single MA as an $M/G/1$ queuing system. Note that the distance distributions between the random locations in other field shapes are also available in the literature [30, 31], which can be used in our model accordingly (e.g., by substituting (3)).

4.2. Analysis Based on the M/G/1 Queuing Model. With the $M/G/1$ queuing model, the data collection latency is equivalently the client response time in the queuing model. We next derive analytical results on the latter to shed light on the former.

4.2.1. System Size Distribution. Denote X_n as the number of requests in the service pool immediately after the departure of a request at time t_n; then

$$X_{n+1} = \begin{cases} X_n - 1 + A_{n+1} & (X_n \ge 1) \\ A_{n+1} & (X_n = 0), \end{cases} \tag{6}$$

where A_{n+1} is the number of new arrivals when serving the $(n+1)$th request. It is clear that A_{n+1} depends only on the service time of the $(n+1)$th request rather than any events that occurred earlier (i.e., the system size at earlier departure points, $X_{n-1}, X_{n-2}, \ldots$). Thus, the embedded discrete-time process $\{X_1, X_2, \ldots\}$ observed at departure times is a *Discrete-Time Markov Chain* (Figure 4) with transition probabilities.

$$p_{ij} = P\{X_{n+1} = j \mid X_n = i\}. \tag{7}$$

Define the probability that i new requests are received when serving a request as

$$\begin{aligned} k_i &= P\{A = i\} = \int_0^\infty P\{A = i \mid S = t\} \times f_{\mathscr{S}}(t)\, dt \\ &= \int_0^\infty \frac{e^{-\lambda t}(\lambda t)^i}{i!} f_{\mathscr{S}}(t)\, dt. \end{aligned} \tag{8}$$

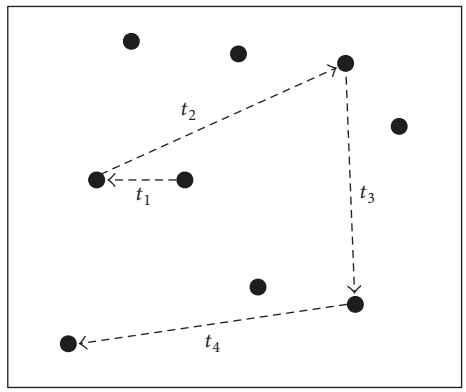

Figure 3: Distribution-ergodic property of the service time.

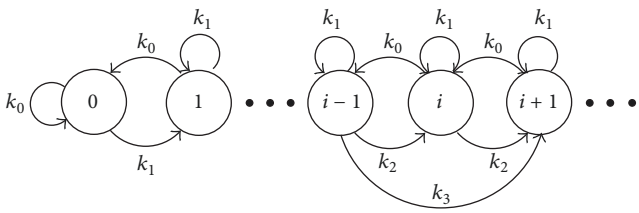

Figure 4: Transition diagram of the Markov Chain.

With k_i, we have the following state transition matrix:

$$\mathbf{P} = \left\{p_{ij}\right\} = \begin{bmatrix} k_0 & k_1 & k_2 & k_3 & \cdots \\ k_0 & k_1 & k_2 & k_3 & \cdots \\ 0 & k_0 & k_1 & k_2 & \cdots \\ 0 & 0 & k_0 & k_1 & \cdots \\ 0 & 0 & 0 & k_0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{bmatrix}. \tag{9}$$

Denote $\boldsymbol{\pi}$ as the steady-state system size probabilities at the departure times.

$$\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi}, \tag{10}$$

from which we know that

$$\pi_i = \pi_0 k_i + \sum_{j=1}^{i+1} \pi_j k_{i-j+1}, \quad \text{for } i = 0, 1, 2, \ldots. \tag{11}$$

Define the following generating functions:

$$\begin{aligned} \Pi(z) &= \sum_{i=0}^{\infty} \pi_i z^i, \quad |z| \le 1 \\ K(z) &= \sum_{i=0}^{\infty} k_i z^i, \quad |z| \le 1. \end{aligned} \tag{12}$$

Because $\pi_0 = 1 - \rho$, we have

$$\Pi(z) = \frac{(1-\rho)(1-z)K(z)}{K(z) - z}, \tag{13}$$

from which $\boldsymbol{\pi}$ can be derived. Note that $\boldsymbol{\pi}$ are not the same as the steady-state system size probabilities $\{p_n\}$ in general. However, for the $M/G/1$ queue, it has been proven that these two quantities are asymptotically identical [25].

4.2.2. Response Time Distribution. Next we derive the response time distribution based on $\boldsymbol{\pi}$, which consists of two parts: (i) the queuing time since its arrival at the system to its service start and (ii) its service time. With FCFS, for a new request arriving at the queue with n' existing requests, its queuing time is the sum of the service times of these n' requests. By *convolution theorem*, we have

$$q\left(t, n'\right) = \begin{cases} 0 & n' = 0 \\ s(t) & n' = 1 \\ q\left(t, n'-1\right) * s(t) & n' > 1, \end{cases} \tag{14}$$

where $*$ is the convolution operator. This way, the probability distribution of the queuing time can be derived as

$$\begin{aligned} \mathcal{Q}(t) &= \sum_{i=0}^{\infty} \Pr\{i \text{ existing requests}\} \\ &\quad \cdot \Pr\{\text{queuing time} \le t \mid i \text{ existing requests}\} \\ &= \sum_{i=0}^{\infty} \pi_i \int_0^t q(x, i)\, dx, \end{aligned} \tag{15}$$

and its density function is $q(t) = \partial\mathcal{Q}(t)/\partial t$.

Similarly, the response time distribution for a request being received by the MA with a system size n' is

$$r\left(t, n'\right) = \begin{cases} s(t) & n' = 0 \\ q\left(t, n'\right) * s(t) & n' > 1. \end{cases} \tag{16}$$

Its probability distribution can be calculated as

$$\begin{aligned} \mathcal{R}(t) &= \sum_{i=0}^{\infty} \Pr\{i \text{ existing requests}\} \\ &\quad \cdot \Pr\{\text{response time} \le t \mid i \text{ existing requests}\} \\ &= \sum_{i=0}^{\infty} \pi_i \int_0^t r(x, i)\, dx, \end{aligned} \tag{17}$$

and $r(t) = \partial\mathcal{R}(t)/\partial t$.

5. MDC with Multiple MAs

Scalability is a critical bottleneck when only a single MA is deployed for data collection, especially with a large sensing field or with a high node density. Employing multiple MAs to collect data collaboratively is a straightforward mitigation, which we investigate next. Specifically, we consider the scenario where each MA has the full knowledge on the received data collection requests, which can be achieved by the communications among the MAs and the sink, for

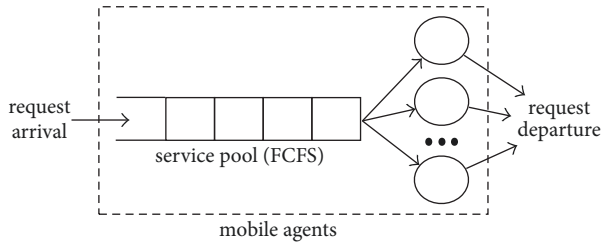

Figure 5: Queuing model when multiple MAs are deployed for data collection.

example, via satellite or cellular communications. Whenever an MA accomplishes its current data collection task, it selects the next-to-be-served request with FCFS.

5.1. Construction of the M/G/c Model. Our approach is to extend the previously constructed $M/G/1$ queuing model to $M/G/c$, where c is the number of MAs (Figure 5). The employment of multiple MAs does not affect sensors' data generation, and thus the request arrival process is the same as in the single MA case. For the request departure, the service time for individual data collection requests is still the same as that with a single MA (i.e., as in (4)), but the aggregated system departure rate will be $c/\mathbb{E}[\mathcal{S}]$.

5.2. Analysis Based on the M/G/c Queuing Model. Although extending the queuing model to the multiple MAs case is straightforward, evaluating an $M/G/c$ queue is analytically intractable. Even when closed-form solutions can be obtained, often they are complicated and require particular probability distributions [32, 33]. Thus, instead of pursuing the exact analytical results on system measures, we use an approximation approach. The basic idea is to combine the analytical results on simple queuing systems such as $M/M/c$ and $M/D/c$ to approximate the measures of the $M/G/c$ queue [34].

5.2.1. Expected Response Time. We first explore the expected data collection latency or, equivalently, the expected response time of the $M/G/c$ queue. A simple two-moment approximation formula with verified accuracy for the mean queuing time in an $M/G/c$ queue can be derived from [35]

$$\mathbb{E}\left[\mathcal{Q}_{M/G/c}\right] \approx \frac{1+\gamma}{2\gamma/\mathbb{E}\left[\mathcal{Q}_{M/M/c}\right] + (1-\gamma)/\mathbb{E}\left[\mathcal{Q}_{M/D/c}\right]}, \quad (18)$$

where $\gamma = \mathbb{C}\mathrm{ov}[\mathcal{S}]^2$.

The above approximation is essentially a weighted combination of $\mathbb{E}[\mathcal{Q}_{M/M/c}]$ and $\mathbb{E}[\mathcal{Q}_{M/D/c}]$. The former can be calculated by

$$\mathbb{E}\left[\mathcal{Q}_{M/M/c}\right] = \frac{(c\rho)^c}{c!c\mu(1-\rho)^2} \cdot \left[\sum_{i=0}^{c-1}\frac{(c\rho)^i}{i!} + \frac{(c\rho)^c}{c!(1-\rho)}\right]^{-1}, \quad (19)$$

and the latter can be obtained by *Crommelin's formula* [36].

$$\mathbb{E}\left[\mathcal{Q}_{M/D/c}\right] = \frac{1}{\mu}\sum_{i=1}^{\infty}\sum_{j=ic+1}^{\infty}\left[\frac{(ic\rho)^{j-1}}{(j-1)!} - \frac{(ic\rho)^j}{\rho j!}\right]e^{-ic\rho}. \quad (20)$$

However, the series in (20) converges slowly especially with high traffic intensity [37]. Again, approximations are adopted to speed up its convergence. An approximation on $\mathbb{E}[\mathcal{Q}_{M/D/c}]$ with simple computation complexity and promising accuracy is presented in [34].

$$\mathbb{E}\left[\mathcal{Q}_{M/D/c}\right] \approx \left[1 + h(\theta)\, g(\rho)\left(1 - e^{-\theta/h(\theta)g(\rho)}\right)\right] \cdot \mathbb{E}\left[\mathcal{Q}_{M/M/c}\right], \quad (21)$$

where

$$\theta = 1 - \frac{2}{c+1}, \quad c \geq 1,$$
$$h(\theta) = \frac{\theta\left[((9+\theta)(1-\theta))^{1/2} - 2\right]}{8(1+\theta)}, \quad (22)$$
$$g(\rho) = \frac{1}{\rho} - 1.$$

Substituting (19) and (21) into (18), we can calculate $\mathbb{E}[\mathcal{Q}_{M/G/c}]$, with which the requests' expected response time in the $M/G/c$ queue can be derived as

$$\mathbb{E}\left[\mathcal{R}_{M/G/c}\right] = \mathbb{E}\left[\mathcal{Q}_{M/G/c}\right] + \mathbb{E}\left[\mathcal{S}\right]. \quad (23)$$

The expected response time is crucial because it not only offers us insights into the asymptotic data collection latency but also helps to obtain the measures on the size of the $M/G/c$ queue, based on which more insights into the MDC can be obtained. Again, denote X as the number of requests either waiting or being served at arbitrary time in the $M/G/c$ queue. Let $\mathcal{P}_i = \mathbb{P}\{X = i\}$ $(i \geq 0)$, and define

$$\alpha = \frac{\mathbb{E}\left[\mathcal{Q}_{M/G/c}\right]}{\mathbb{E}\left[\mathcal{Q}_{M/M/c}\right]}. \quad (24)$$

A geometric-form approximation for the system size probability is proposed in [38].

$$\mathcal{P}_{i,M/G/c} = \begin{cases}\left[\dfrac{(c\rho)^i}{i!}\right]\mathcal{P}_{0,M/M/c} & i = 0,\ldots,c-1\\ (1-\zeta)\zeta^{i-c}\mathcal{U}_{M/M/c} & i \geq c,\end{cases} \quad (25)$$

where

$$\zeta = \frac{\rho\alpha}{1-\rho+\rho\alpha},$$
$$\mathcal{P}_{0,M/M/c} = \left[\sum_{i=0}^{c-1}\frac{(c\rho)^c}{i!} + \frac{(c\rho)^c}{c!(1-\rho)}\right]^{-1}, \quad (26)$$
$$\mathcal{U}_{M/M/c} = \frac{(c\rho_c)^c P_{0,M/M/c}}{c!(1-\rho_c)}$$

is the probability that a newly arriving request has to wait before being served in an $M/M/c$ queue. Note that $\zeta < 1$ if $\rho < 1$ and $\zeta = \rho$ if the service time is exponentially distributed.

We calculate the expected system size based on its approximated distribution as

$$E[\mathscr{L}] = \sum_{i=0}^{\infty} i \cdot \mathscr{P}_{i,M/G/c}, \tag{27}$$

and the probability that a newly arrived request has to wait before being served, that is, the equivalent of $\mathscr{U}_{M/M/c}$ in $M/G/c$ queue, can be calculated as

$$\mathscr{U}_{M/G/c} = 1 - \sum_{i=0}^{c-1} \mathscr{P}_{i,M/G/c}. \tag{28}$$

By *distributional Little's law* [39], the number of customers in the queue has the same distribution as the number of arrivals during the waiting time. Based on this and the above approximation results on the system size distribution, an approximation for the queuing time distribution in the $M/G/c$ queue is proposed in [40].

$$\begin{aligned} \mathscr{Q}_{M/G/c}(t) &\approx 1 - e^{(-c\mu(1-\rho)t)/\alpha} \mathscr{U}_{M/M/c}, \\ q_{M/G/c}(t) &= \frac{\partial Q_{M/G/c}(t)}{\partial t}. \end{aligned} \tag{29}$$

Furthermore, because the response time of a request is the sum of its queuing time and service time, which are independent of each other, by *convolution theorem*, we have

$$r_{M/G/c}(t) = q_{M/G/c}(t) * s(t). \tag{30}$$

6. Performance Evaluations

We verify the model soundness and the analysis accuracy in this section. We consider a system deployed in a square field of size $100 \times 100\,\mathrm{m}^2$. A total number of 100 sensors are randomly deployed unless otherwise specified. The MA velocity is set to 1 m/s based on Power Bot [41]. The simulation is implemented with Matlab. A total number of 10,000 requests are generated and served during each run of the simulation, which is repeated for 50 times.

To deal with the inconvenience of the piecewise distance probability density function in (3), we approximate it by a 10-order polynomial with least squares fitting.

$$\begin{aligned} \tilde{f}(d) = {} & 0.2802d^{10} - 2.0964d^{9} + 2.2349d^{8} \\ & + 24.3629d^{7} - 106.8231d^{6} + 194.4928d^{5} \\ & - 182.8093d^{4} + 91.8223d^{3} - 29.3663d^{2} \\ & + 8.2843d - 0.0402. \end{aligned} \tag{31}$$

6.1. Verifying the Queuing Models. To verify the soundness of the queue-based modeling, we examine the request arrival with an event-driven simulator, where stochastic events occur randomly in the sensing field (note that when events happen in a clustered manner, this actually improves the MDC performance as the MAs' travel distance is reduced. This way, our models capture the worst cases of on-demand MDC, which are important to provide performance guarantees). Sensors within a certain distance (i.e., the sensing range) can detect the event, and corresponding sensory data are generated. The data size for recording each event varies from 10 to 100 B. Events happen independently in both the spatial and temporal domains, and sensor nodes initiate the data collection requests when their buffers become full. We explore the cases where the sensor node buffer size is 4 KB and 8 KB, respectively, and record the interarrival time of data collection requests for comparison with an exponential distribution with the same mean value (the Poisson arrival process indicates an exponentially distributed request interarrival time). Figure 6 indicates that the simulation results match the exponential distribution well, verifying the assumption on Poisson arrival. Furthermore, a larger node buffer results in a smaller request arrival rate, because the sensor nodes can hold the on-board data longer.

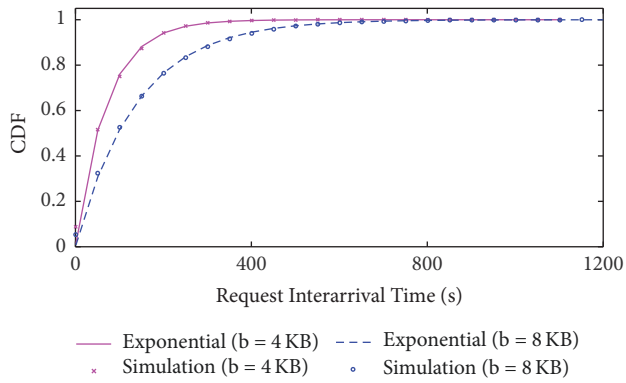

Figure 6: Verify the request arrival.

We further statistically verify the queuing models by validating the Poisson arrival of requests, the independence of request arrivals, and the service time independence. Kolmogorov-Smirnov (K-S) test with a significance level of 5% is used to verify the Poisson arrival. We perform the tests with a different number of sensor nodes (20–100), each with 50 trials. We record the number of trials that reject the Poisson arrival hypothesis. The verification results are listed in the first row of Table 1. The low rejection ratio indicates that the Poisson arrival in our modeling is sound. To evaluate the independence of the request arrival and service time, we record the request interarrival time and service time and calculate their 1-lag autocorrelations. Again, the simulation is repeated for 50 times with 20 to 100 sensor nodes, respectively, and the average absolute values of the autocorrelations are shown in the second and third rows of Table 1. The small correlations of both the request arrival and their service time support our queue-based modeling.

6.2. Single MA. Next, we evaluate our analytical results on the single MA case. Service time distribution is the core

Table 1: Verify the queuing model (with 50 trials).

# of nodes	20	40	60	80	100
# of rejections	0	1	1	3	3
Arrival corr.	0.0463	0.0371	0.0236	0.0183	0.0167
Service corr.	0.0942	0.1023	0.0985	0.0967	0.1000

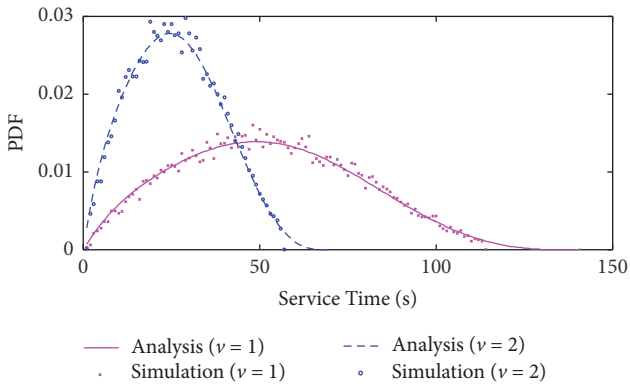

Figure 7: Service time distribution.

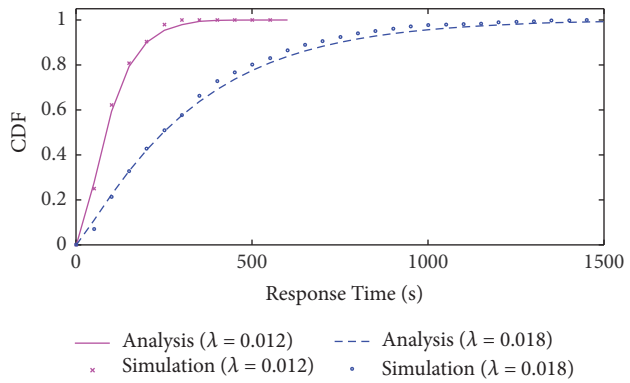

Figure 8: Response time distribution.

component in the queue-based analysis, which is obtained based on results from geometrical probability. We evaluate our analysis on the service time distribution with an MA velocity of 1 m/s and 2 m/s, respectively, and the results are shown in Figure 7. We can see that the analytical results and the simulation match greatly. The service time is significantly reduced after increasing the MA velocity from 1 m/s to 2 m/s, agreeing with (4).

The response time distributions with request arrival rates of 0.012 and 0.018 are shown in Figure 8. Besides the accuracy of the analysis, we can see that the response time of requests, or the data collection latency in our focus, is significantly increased when increasing the request arrival rate. This verifies the potential scalability issue when only one MA is used for data collection.

6.3. Multiple MAs. We evaluate our modeling and analysis results on the multiple MAs case in the following. We explore

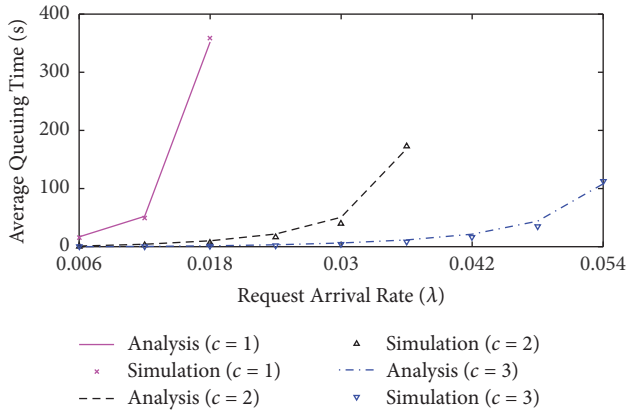

Figure 9: The average queuing time.

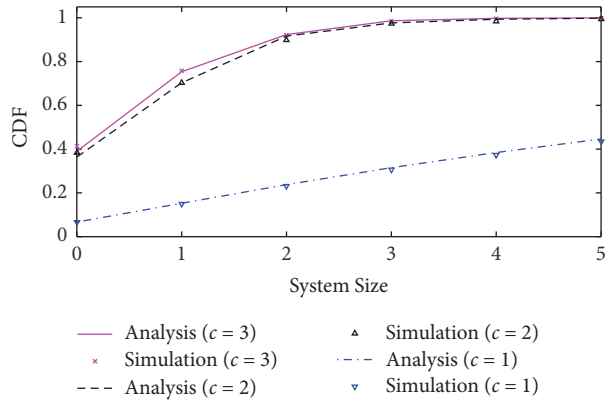

Figure 10: The system size distribution.

the cases with $c = 2$ and $c = 3$, respectively, and also present the results with $c = 1$ for comparison.

The approximation results on the expected queuing time are verified in Figure 9. The effect of deploying more MAs is obvious, especially when the request arrival rate is high. Note that no results for $c = 1$ or $c = 2$ are shown when λ is larger than 0.018 or 0.036, because the further increase of λ will result in a ρ greater than 1, and no steady-state measures can be obtained.

Figure 10 shows the evaluation results of the approximation on the system size distribution with λ of 0.018. Besides the accuracy of the approximation, we can see that increasing c from 1 to 2 can greatly shorten the system size, which in turn reduces the data collection latency. However, the benefit of increasing c further from 2 to 3 is quite limited. This is because the system utilization factor is already small when $c = 2$, and thus further increasing c cannot significantly improve the data collection performance anymore.

The results on the probability for requests to wait before being served are shown in Figure 11. Intuitively, the wait probability increases when the system becomes more heavily occupied, which results when (1) fewer MAs are adopted (c decreases) and (2) the data intensity in the network is higher (λ increases). The verification of this reasoning can be clearly observed from Figure 11.

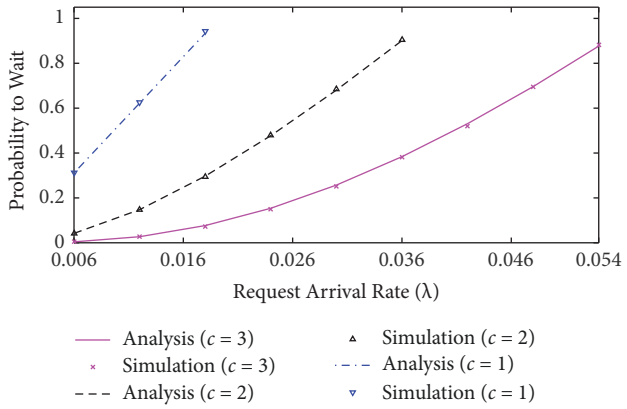

Figure 11: Waiting probability.

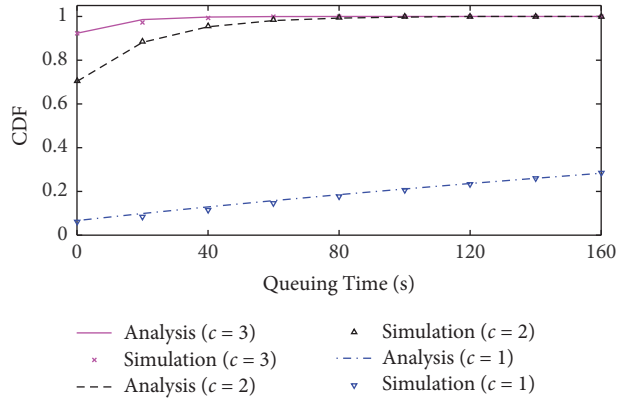

Figure 12: Queuing time distribution.

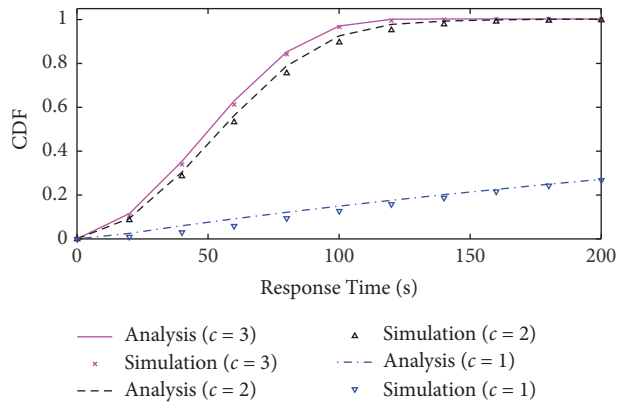

Figure 13: Response time distribution.

The queuing time and response time distributions with λ of 0.018 are shown in Figures 12 and 13, respectively. Besides the analysis accuracy, again we observe that further increasing c from 2 to 3 when $\lambda = 0.018$ cannot significantly reduce the queuing time (response time), agreeing with Figure 10.

7. Practical Guidance

The constructed queuing models not only reveal insights into the on-demand MDC but also guide its practical implementation. We use three examples to show how these models can assist the system implementation in this section.

7.1. How to Adopt Multiple MAs? In the first example, we explore the problem of how to employ multiple MAs for collaborative data collection. In general, two strategies can be used—the MAs can collaboratively collect data from the entire system, referred to as Strategy-I, or the system can be divided into subareas, and each MA is responsible to collect data from one subarea, which is referred to as Strategy-II. These two strategies can be captured by multiserver systems with shared (i.e., all MAs share the knowledge of data collection requests as with Strategy-I) and separate (i.e., each MA is only responsible for a subset of requests that fall in its service queue as with Strategy-II) queues, respectively.

Conventional wisdom says that, all things being equal, a shared queue outperforms separate queues most times [25]. To the best of our knowledge, however, no results on the comparison of the two strategies have been reported yet. Here, we close this gap based on the constructed $M/G/1$ and $M/G/c$ queuing models. Our results reveal that Strategy-II outperforms Strategy-I in both MAs' workloads and requests' response time, contradicting with the conventional wisdom.

Let us consider the case where c MAs are deployed in an $L \times L$ sensing field. For the ease of description, we assume that $c = i^2$ ($i = 1, 2, \ldots$), which can be relaxed as will be explained later. When Strategy-I is adopted, the data collection performance can be evaluated based on the results in Section 5. Specifically, the utilization factor of individual MAs is

$$\rho_1 = \frac{\lambda}{c}\mathbb{E}[\mathcal{S}] \tag{32}$$

and the expected data collection latency and its distribution can be obtained according to (23) and (30), respectively.

When Strategy-II is adopted, that is, the field is divided into c subareas of size $L/\sqrt{c} \times L/\sqrt{c}$ each, the data collection performance can be evaluated based on the results in Section 4 but in a smaller sensing field. Denoting the requests arrival rate at individual MA and the service time in this case as λ' and $\mathcal{S}'$, the utilization factor of individual MAs is

$$\rho_2 = \lambda'\mathbb{E}\left[\mathcal{S}'\right]. \tag{33}$$

With randomly distributed sensor nodes, it is clear that

$$\begin{aligned} \lambda' &= \frac{\lambda}{c}, \\ \mathbb{E}\left[\mathcal{S}'\right] &= \frac{\mathbb{E}[\mathcal{S}]}{\sqrt{c}}. \end{aligned} \tag{34}$$

Thus, from (32) and (33), we have

$$\frac{\rho_1}{\rho_2} = \frac{\mathbb{E}[\mathcal{S}]}{\mathbb{E}\left[\mathcal{S}'\right]} = \sqrt{c} > 1. \tag{35}$$

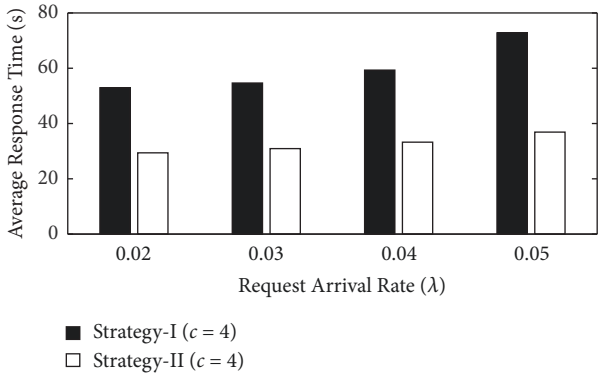

Figure 14: Average response time obtained with the two strategies.

(1) $\theta = 0$, $\Delta = 0.01$;
(2) **while** $p_{\text{overflow}} > 0.01$ and $\theta < 1$ **do**
(3) $\theta = \theta + \Delta; \lambda \dfrac{1}{(1-\theta)B} \sum_{i=1}^{n} f_i$
(4) calculate $R(t)$ with λ;
(5) $p_{\text{overflow}} = 1 - \int_0^{\theta B/f_i} r(t)dt$;
(6) **end while**

Algorithm 1: Find the optimal remaining buffer ratio θ.

This indicates that Strategy-II achieves a lower workload for the MAs with reduced MA travel distance, which dominates the service time in MDC.

Because the comparison on the response time achieved with the two strategies is not so obvious, numerical comparison is performed with a network scale of 100 sensor nodes and $c = 4$. The results are shown in Figure 14, where the aggregated request arrival rate at the MAs varies from 0.02 to 0.05. Again, Strategy-II reduces the response time by 50% when compared with Strategy-I, due to the fact that the service time is dominated by the MAs' travel time.

Although we simplify the above description by assuming that $c = i^2$ ($i = 1, 2, \ldots$), the conclusion on the advantage of Strategy-II holds in more general cases. However, dividing the field into c identical subareas of square shape may not be always feasible, in which cases the distance distributions between random locations in other field shapes can be used [42].

7.2. When to Request Data Collection? The time for sensors to request data collection plays a critical role in the on-demand MDC. Sending the request too early unnecessarily increases the workload of the MAs, which also increases the data collection latency of other requests. On the other hand, a belated request leaves little time for the MAs to complete the data collection before the buffer of the requesting node overflows.

The response time distribution, derived based on the constructed queuing models, helps identify the proper time instant to send out the data collection requests. With a memory buffer B for each sensor node and its respective data generation rate f_i, denote θ ($0 \le \theta \le 1$) as the remaining memory buffer ratio when node sends the data collection request. This way, the aggregated request arrival rate is

$$\lambda \approx \frac{1}{(1-\theta)B} \sum_{i=1}^{n} f_i. \tag{36}$$

Buffer overflow would occur if the response time is larger than $\theta B / f_i$. With the derived response time distribution (i.e., (17) and (30)), the probability for buffer overflow to occur can be calculated with given θ, which in turn allows us to identify the smallest θ (and thus the smallest workloads on MAs) that guarantees a small enough buffer overflow probability (e.g., $p_{\text{overflow}} < 0.01$), as illustrated in Algorithm 1.

7.3. Requests Combination and Preemption. The MAs can potentially collect data from multiple sensor nodes at the same location because of the wireless communication capabilities of both the MAs and sensor nodes, which corresponds to the scenario of batch service in queuing theory and is referred to as *requests combination* here.

Clearly, the probability for requests combination to occur is jointly determined by (i) the communication range R between the MAs and sensors (normalized to the field size) and (ii) the number of requests in the service pool when the new request arrives, that is, the queue length. Specifically, for a new request arriving when L requests are in the queue, it can be combined with at least one of these existing requests with probability

$$P_{\text{combine}}(R, L) = 1 - \left(1 - \int_0^R f_{\mathscr{D}}(x)\, dx\right)^L, \tag{37}$$

where $f_{\mathscr{D}}(x)$ is the distance distribution between two random locations as in (3). This indicates that the effect of requests combination on improving the on-demand MDC will be profound when the communication range is large or when the service queue is long.

Figure 15 shows the effect of requests combination with varying communication ranges. As expected, a larger communication range has a greater effect in reducing the data collection latency, when compared with the noncombination cases. Also, the advantage of requests combination is less significant when the number of MAs increases. This is because the service pool size is reduced when more MAs are adopted for the data collection tasks, and thus the probability for combination to happen is reduced as well.

Requests preemption is another potential way to improve the on-demand MDC—a new request arrival may preempt the service of existing requests if its requesting node is close to the current locations of the MAs. We have analytically explored the possibility and advantage of requests preemption in another work [43].

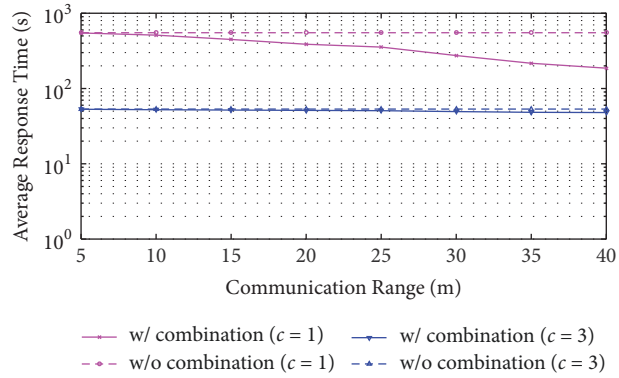

FIGURE 15: Effect of requests combination with communication range.

8. Further Discussions

8.1. System Stability Condition. A necessary and sufficient condition for the data collection process to be stable is $\rho/c < 1$ [12, 22]. This way, from (1), we know that

$$\frac{\rho}{c} = \frac{\lambda \mathbb{E}[\mathcal{S}]}{c} < \frac{\mathbb{E}[\mathcal{D}]}{vcB} \sum_{i=1}^{n} f_i < 1, \tag{38}$$

which implies that the minimum requirement on the travel speed of the MAs is

$$v > \frac{\mathbb{E}[\mathcal{D}]}{cB} \sum_{i=1}^{n} f_i. \tag{39}$$

From (39), we can see that, to provide a stable data collection performance, there is a clear trade-off between the number of required MAs and their capabilities such as travel speed v and memory size B, assisting us in determining the number of needed MAs in practice.

8.2. Insights for Sophisticated Discipline Design. We establish a theoretical foundation on the on-demand MDC when FCFS is adopted, which reveals insights into the design of more sophisticated service disciplines. Through the queue-based analysis, it is clear that the MAs' travel distance between two consecutively served requests is the dominant factor that determines the data collection latency, which should be minimized to achieve a better performance. Inspired by this, in our recent work [11], we have extended these queuing models to investigate the MDC with a greedy service discipline that minimizes the travel distance between two consecutively served requests, and significant asymptotic improvement can be observed. However, the greedy discipline may cause some unfairness among sensor nodes, which has to be addressed to guarantee the worst-case performance for every node. Also note that Petri nets could be another analytical tool to capture such data collection process [44], which we will explore more in the future.

9. Conclusions

In this paper, we have analytically investigated the on-demand MDC in cyber-physical systems. Two queuing models, namely, an $M/G/1$ and an $M/G/c$ model, have been constructed to capture the MDC with a single MA and multiple MAs, respectively. System measures of the queues, for example, the expected values and distributions of queue length, queuing time, and response time have been explored. These queuing models shed light on the impact of different parameters on MDC, and the corresponding analytical results serve as guidelines in the design of more sophisticated data collection solutions. The soundness of the models and the accuracy of the analysis have been verified via extensive simulations.

Through the queue-based analysis, it is clear that the MAs' travel distance between two consecutively served requests is the dominant factor that determines the data collection latency, which should be minimized to achieve a better performance. Inspired by this, in our recent work [11], we have extended these queuing models to investigate the MDC with a greedy service discipline that minimizes the travel distance between two consecutively served requests, and a significant improvement can be observed asymptotically.

Disclosure

A preliminary version of this work was published at IEEE GLOBECOM'11 [45].

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The work reported in this paper was supported in part by CNS-1739577, National Key Research and Development Program (Grant 2016YFE0100600), NSFC (61672349), the Natural Science Foundation of Jiangsu Province (no. BK20151416), and NSERC.

References

[1] B. Chai, J. Chen, Z. Yang, and Y. Zhang, "Demand response management with multiple utility companies: A two-level game approach," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 722–731, 2014.

[2] M. Dong, K. Ota, L. T. Yang, S. Chang, H. Zhu, and Z. Zhou, "Mobile agent-based energy-aware and user-centric data collection in wireless sensor networks," *Computer Networks*, vol. 74, pp. 58–70, 2014.

[3] G. Li, M. Dong, K. Ota, J. Wu, J. Li, and T. Ye, "Towards QoE named content-centric wireless multimedia sensor networks with mobile sinks," in *Proceedings of the ICC 2017 - 2017 IEEE International Conference on Communications*, pp. 1–6, Paris, France, May 2017.

[4] G. Xie, K. Ota, M. Dong, F. Pan, and A. Liu, "Energy-efficient routing for mobile data collectors in wireless sensor networks with obstacles," *Peer-to-Peer Networking and Applications*, vol. 10, no. 3, pp. 472–483, 2017.

[5] M. Zhao and Y. Yang, "Optimization-based distributed algorithms for mobile data gathering in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 10, pp. 1464–1477, 2012.

[6] M. Zhao and Y. Yang, "Bounded relay hop mobile data gathering in wireless sensor networks," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 61, no. 2, pp. 265–277, 2012.

[7] Y. Gu, Y. S. Ji, J. Li, F. Ren, and B. Zhao, "EMS: efficient mobile sink scheduling in wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 5, pp. 1556–1570, 2013.

[8] "NEPTUNE Canada," http://www.neptunecanada.ca.

[9] R. Sugihara and R. K. Gupta, "Optimal speed control of mobile node for data collection in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 127–139, 2010.

[10] X. Xu, J. Luo, and Q. Zhang, "Delay tolerant event collection in sensor networks with mobile sink," in *Proceedings of the IEEE INFOCOM*, March 2010.

[11] L. He, . Zhe Yang, J. Pan, L. Cai, and J. Xu, "Evaluating service disciplines for mobile elements in wireless ad hoc sensor networks," in *Proceedings of the IEEE INFOCOM 2012 - IEEE Conference on Computer Communications*, pp. 576–584, Orlando, FL, USA, March 2012.

[12] G. D. Celik and E. H. Modiano, "Controlled mobility in stochastic and dynamic wireless networks," *Queueing Systems*, vol. 72, no. 3-4, pp. 251–277, 2012.

[13] M. Zhao, Y. Yang, and C. Wang, "Mobile data gathering with load balanced clustering and dual data uploading in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 770–785, 2015.

[14] Y. Gu, Y. Ji, J. Li, and B. Zhao, "ESWC: efficient scheduling for the mobile sink in wireless sensor networks with delay constraint," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1310–1320, 2013.

[15] Z. Li, Y. Liu, M. Li, J. Wang, and Z. Cao, "Ubiquitous data collection for mobile users in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 2, pp. 312–326, 2013.

[16] S. Guo, C. Wang, and Y. Yang, "Joint mobile data gathering and energy provisioning in wireless rechargeable sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2836–2852, 2014.

[17] L. Xie, Y. Shi, Y. T. Hou et al., "A Mobile Platform for Wireless Charging and Data Collection in Sensor Networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 8, pp. 1521–1533, 2015.

[18] D. Jea, A. Somasundara, and M. Srivastava, "Multiple controlled mobile elements (data mules) for data collection in sensor networks," in *Proceedings of the 1st IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '05)*, pp. 244–257, July 2005.

[19] M. Ma and Y. Yang, "Data gathering in wireless sensor networks with mobile collectors," in *Proceedings of the Proceeding of the 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS '08)*, pp. 1–9, Miami, Fla, USA, April 2008.

[20] M. Ma, Y. Yang, and M. Zhao, "Tour planning for mobile data-gathering mechanisms in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 4, pp. 1472–1483, 2013.

[21] Y. Gu, F. Ren, Y. Ji, and J. Li, "The evolution of sink mobility management in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 507–524, 2016.

[22] E. Altman and H. Levy, "Queueing in space," *Advances in Applied Probability*, vol. 26, no. 4, pp. 1095–1116, 1994.

[23] D. J. Bertsimas and G. V. Ryzin, "A stochastic and dynamic vehicle routing problem in the Euclidean plane," *Operations Research*, vol. 39, no. 4, pp. 601–615, 1991.

[24] "Intel Lab Data," http://www.select.cs.cmu.edu/data/labapp3/index.html.

[25] D. Gross, *Fundamentals of Queueing Theory*, John Wiley & Sons, New Jersey, 4th edition, 2008.

[26] L. He, J. P. Pan, and J. D. Xu, "A progressive approach to reducing data collection latency in wireless sensor networks with mobile elements," *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, pp. 1308–1320, 2013.

[27] D. R. Cox and H. D. Miller, *The Theory of Stochastic Processes*, Chapman and Hall, London, UK, 1965.

[28] G. Grimmett and D. Stirzaker, *Probability and random processes*, Oxford Science Publications, The Clarendon Press, Oxford University Press, New York, 1982.

[29] C. Bettstetter, H. Hartenstein, and X. Perez-Costa, "Stochastic properties of the random waypoint mobility model," *Wireless Networks*, vol. 10, no. 5, pp. 555–567, 2004.

[30] D. Moltchanov, "Distance distributions in random networks," *Ad Hoc Networks*, vol. 10, no. 6, pp. 1146–1166, 2012.

[31] L. E. Miller, "Distribution of link distances in a wireless network," *Journal of research of the National Institute of Standards and Technology*, vol. 106, no. 2, pp. 401–412, 2001.

[32] B. N. Ma and J. W. Mark, "Approximation of the mean queue length of an *M/G/c* queueing system," *Operations Research*, vol. 43, no. 1, pp. 158–165, 1995.

[33] M. J. Sobel, "Simple inequalities for multiserver queues," *Management Science*, vol. 26, no. 9, pp. 951–956, 1980.

[34] T. Kimura, "Approximations for multi-server queues: system interpolations," *Queueing Systems*, vol. 17, no. 3-4, pp. 347–382, 1994.

[35] T. Kimura, "A two-moment approximation for the mean waiting time in the *GI/G/s* queue," *Management Science*, vol. 32, no. 6, pp. 751–763, 1986.

[36] C. Crommelin, "Delay probability formulae," *Post Office Electrical Engineers Journal*, vol. 26, pp. 266–274, 1934.

[37] G. P. Cosmetatos, "On the Implementation of Page's Approximation for Waiting Times in General Multi-Server Queues," *Journal of the Operational Research Society*, vol. 33, no. 12, pp. 1158-1159, 1982.

[38] T. Kimura, *A transform-free approximation for the queue-length distribution in the finite capacity* M/G/s *queue*, vol. 18 of *Discussion Paper Series A*, Faculty of Economics, Hokkaido University, 1993.

[39] D. Bertsimas and D. Nakazato, "The distributional Little's law and its applications," *Operations Research*, vol. 43, no. 2, pp. 298–310, 1995.

[40] M. H. van Hoorn and H. C. Tijms, "Approximations for the waiting time distribution of the *M/G/c* queue," *Performance Evaluation*, vol. 2, no. 1, pp. 22–28, 1982.

[41] A. Whitbrook, http://robots.mobilerobots.com.

[42] F. Tong, M. Ahmadi, and J. Pan, *Random Distances Associated with Arbitrary Triangles: A Systematic Approach between Two Random Points*, University of Victoria, Victoria, Canada, 2013.

[43] L. He, L. Kong, Y. Gu, J. Pan, and T. Zhu, "Evaluating the On-Demand Mobile Charging in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 9, pp. 1861–1875, 2015.

[44] G. Liu, "Complexity of the deadlock problem for Petri nets modeling resource allocation systems," *Information Sciences*, vol. 363, pp. 190–197, 2016.

[45] . Liang He, . Jianping Pan, and . Jingdong Xu, "Analysis on Data Collection with Multiple Mobile Elements in Wireless Sensor Networks," in *Proceedings of the 2011 IEEE Global Communications Conference (GLOBECOM 2011)*, pp. 1–5, Houston, TX, USA, December 2011.