# Evaluating Service Disciplines for On-Demand Mobile Data Collection in Sensor Networks

Liang He, *Member, IEEE* , Zhe Yang, *Student Member, IEEE* , Jianping Pan, *Senior Member, IEEE* , Lin Cai, *Senior Member, IEEE* , Jingdong Xu, and Yu (Jason) Gu, *Member, IEEE*

**Abstract—**Mobility-assisted data collection in sensor networks creates a new dimension to reduce and balance the energy consumption for sensor nodes. However, it also introduces extra latency in the data collection process due to the limited mobility of mobile elements. Therefore, how to schedule the movement of mobile elements throughout the field is of ultimate importance. In this paper, the on-demand scenario where data collection requests arrive at the mobile element progressively is investigated, and the data collection process is modelled as an $M/G/1/c$-$NJN$ queuing system with an intuitive service discipline of *nearest-job-next* (NJN). Based on this model, the performance of data collection is evaluated through both theoretical analysis and extensive simulation. NJN is further extended by considering the possible requests combination (NJNC). The simulation results validate our models and offer more insights when compared with the *first-come-first-serve* (FCFS) discipline. In contrary to the conventional wisdom of the starvation problem, we reveal that NJN and NJNC have better performance than FCFS, in both the average and more importantly the worst cases, which offers the much needed assurance to adopt NJN and NJNC in the design of more sophisticated data collection schemes, as well as other similar scheduling scenarios.

**Index Terms—**Wireless ad hoc sensor networks, mobile elements, on-demand data collection

## 1 Introduction

Many applications in wireless sensor networks are data oriented [1], [2]. Data collection in sensor networks typically relies on the wireless communications between sensor nodes and the sink, which may excessively consume the limited energy supply of sensor nodes due to the super-linear path loss exponents [3]. Furthermore, sensor nodes near the sink also tend to deplete their energy much faster than other nodes due to the data aggregation towards the sink, which imposes them a much heavier volume of data to forward and leads to a very unbalanced energy consumption in the entire network [4]. In addition, these approaches are based on a fully connected network, which requires the dense deployment of nodes and thus introduces extra costs.

Another data collection approach utilizes the often-available, controlled mobility of certain devices, referred to as *mobile elements (MEs)* in this paper [5], [6], [7]. By utilizing mobile elements, not only more energy can be conserved and balanced on sensor nodes, but also the communications and networking become possible in very sparse networks with the *store-carry-forward* approach. For example, the seabed crawler deployed in the observatory of the NorthEast Pacific Time-series undersea networked experiments (NEPTUNE) can cruise through several experimentation sites, "talk" to experiment devices, and bring the data back to the junction boxes [11]. Another mobile element example is the Seaeye Sabertooth [12], a battery-powered autonomous underwater vehicle, which travels in deep water environments to collect data from deployed equipments, and uploads the data to the control center at the docking station. Other examples of the mobility-assisted data collection include the smart buoy equipped with Seatext from WFS [13], the structural health monitoring with radio-controlled helicopters [14], and so on.

Although mobile elements create a new dimension for data collection, they also introduce extra challenges. First, the data collection latency may be large due to the relatively low travel speed of mobile elements [15], [18], especially when compared with that of electromagnetic or acoustic waves. This large latency must be addressed for applications with tight requirements on the timely data delivery. Second, with a large latency, data loss might occur due to the buffer overflow of sensor nodes, which is not desirable for data integrity sensitive applications [16]. Finally, mobile elements themselves are battery-powered as well in most cases, e.g., the travel distance of Sabertooth is about 20-50 Km with a fully charged battery, so the data collection must be accomplished before mobile elements deplete their own energy.

A lot of efforts have attempted to address these challenges by finding the optimal data collection tour for the mobile elements, under the *offline* scenario where the data

- *L. He and Y. Gu are with the SUTD-MIT International Design Center, Singapore University of Technology and Design, 20 Dover Drive, Singapore 138682. E-mail: he_liang@sutd.edu.sg.*
- *Z. Yang, J. Pan, and L. Cai are with the Department of Computer Science, University of Victoria, PO Box 3055, STN CSC, Victoria, BC V8W 3P6, Canada.*
- *J. Xu is with the Department of Computer Science, College of Information Technical Science, Nankai University, 93 Weijin Road, China.*

*Manuscript received 28 May 2012; revised 29 Dec. 2012; accepted 16 May 2013; date of publication 21 May 2013; date of current version 3 Mar. 2014.*
*For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.*
*Digital Object Identifier no. 10.1109/TMC.2013.62*

collection is carried out in a *periodic* way [17], [25], [26], [44]. However, in a more practical scenario where the data collection is carried out in an *on-demand* manner, we need to determine how the mobile element should carry out the data collection tasks without a priori information on the data collection demands. There are some existing efforts aiming to design data collection schemes for this *online* scenario [21], [49]. However, a critical and unaddressed issue is how to evaluate the efficiency and optimality of the proposed schemes, because without a priori knowledge on the request arrival, no optimal solution is available as the benchmark in this case.

In this paper, we focus on this *on-demand* data collection scenario and tackle this limitation by theoretically analyzing the data collection performance when certain classic disciplines are adopted through a queue-based modeling approach, which offers important guidelines in designing and evaluating more sophisticated on-demand data collection schemes for the mobile elements.

We first show that data collection requests in the online scenario can be captured by a Poisson arrival process, and with the travel distance (and time) distribution between any two sensor nodes in the sensing field, the collection process can be modelled as an $M/G/1/c\text{-}NJN$ queuing system, which accommodates at most $c$ requests at the same time. The NJN stands for *Nearest-Job-Next*, a simple and intuitive discipline adopted by the mobile element (server) to select the next to-be-served request (client). A challenge with the analysis of the NJN discipline is the *dynamic* and *state-dependent* service time, as will be explained later. Furthermore, observing that multiple requests can be combined and served together by the mobile element if a collection site within the communication ranges of all the corresponding sensor nodes can be found, we extend the NJN discipline to *NJN-with-combination* (NJNC), i.e., $M/G/1/c\text{-}NJNC$, to explore how much gain can be achieved by the possible requests combination. The resultant data collection efficiency is evaluated based on the analytical results on the system measures of these models.

To the best of our knowledge, this is the first time that the distributions of critical performance metrics with NJN and NJNC are obtained through a queue-based approach, and the approach can be extended to other dynamically-prioritized scenarios as well. Our results show that even though NJN may be unfair for farther-away requests temporarily, its average performance outperforms FCFS greatly and more importantly, even its worst-case performance is better than FCFS, especially in the case of NJNC. These results offer the much needed assurance to adopt NJN and NJNC in the design of more sophisticated on-demand data collection schemes.

The remainder of this paper is organized as follows. Section 2 briefly reviews the literature on the mobility-assisted data collection. In Section 3, we outline the problem setting and list the assumptions and definitions used in this paper. We present the analytical models of the NJN and NJNC disciplines in Section 4 and Section 5, respectively. The analytical and simulation results are presented and compared in Section 6 for model validation and further insights. In Section 7, we discuss the possible approaches to further extend and improve the work. Finally, we conclude this paper in Section 8.

## 2 Background and Related Work

Recently, a lot of efforts have been made on exploring the mobility-assisted data collection in wireless sensor networks [25], [27], [29], [44]. Many of them are scheme design oriented for the offline scenario where the mobile elements know all requests in advance and carry out the data collection periodically [17], [19], [28]. For example, Ryo et al. focused on the scenario where the ME has to accomplish the data collection task by visiting all the nodes within the shortest possible time in [17]. They formulated the problem as a *label-covering problem*, which was proved to be NP-hard. A tour selection algorithm for the ME was proposed in [20], which starts with a *connected dominating set* of the network, obtains a *minimum spanning tree* based on it, and finally generates a *Hamiltonian circuit* for the ME. The case where multiple MEs exist in the network was considered in [28]. Xing et al. tackled this problem by finding the rendezvous locations in [29], and extended the work to jointly optimize the data routing path and the tour of the ME in [30].

In contrast, the online mobile data collection, which is carried out in an on-demand manner, is much less explored, even though it is more practical in reality [21], [49]. The most intuitive service discipline is *first-come-first-serve* (FCFS), i.e., the order to serve requests is the same as their arrival order, whose performance is analytically evaluated in [10]. However, due to the randomness of service requests in both of the time and space domains, the ME may unnecessarily travel back-and-forth to serve requests, which is clearly undesirable since the travel time of the ME dominates the data collection latency. Another intuitive service discipline is to serve the geographically nearby requests first, or *nearest-job-next*, in order to reduce the distance that the ME has to travel and thus the time it has to spend on. However, in the literature and practical systems, NJN and its variants are much less explored, due to the concerns discussed below.

NJN is similar to the traditional *shortest-job-next* (SJN) service discipline [22]. However, two extra issues need to be considered. First, for SJN, the service time for each job has to be accurately estimated upon its arrival in the system and remain fixed before its service. But the service time with NJN for data collection in wireless sensor networks is jointly determined by the location of the requesting sensor node and that of the serving ME, which cannot be determined in advance until the job is *about* to be served. This dynamic service time makes not only the existing results on SJN not applicable to our problem [23], but also the analysis on NJN much more challenging due to the dynamic priority of a particular request. Second, SJN is known to lead to the starvation problem for large jobs, which limits its practical implementation. Thus whether NJN suffers from the similar unfairness problem for mobility-assisted data collection is another question we need to answer. Another discipline similar to NJN is the *shortest-seek-time-first* (SSTF) in disk scheduling [24]. The disk tracks are normally treated

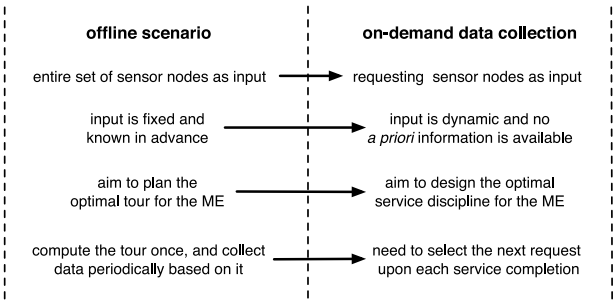



Fig. 1. Differences between the on-demand data collection and the offline data collection.

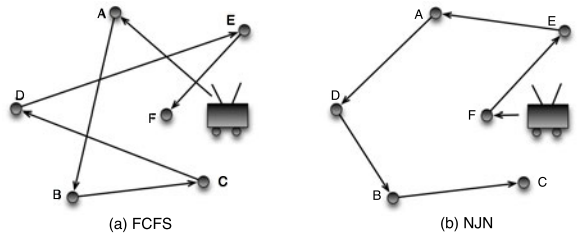



Fig. 2. Demonstration of the FCFS and NJN service disciplines. For clarity, only requesting sensor nodes are shown and no new requests are considered. The arrival order of requests is shown lexicographically.

in a one-dimensional space, while the sensing field of NJN is a two-dimensional space.

The design of dynamic scheduling schemes has been extensively explored in the scenarios such as the *dynamic vehicle routing* [45] and *dynamic traveling salesman problem* [46]. Greedy schemes similar to NJN have been explored by the research efforts on these problems [47], [48], and our queue-based analysis advances the investigation by obtaining the probability distributions of its critical performance metrics, such as the queue length, queuing time, and response time. The most relevant work to ours is [49], in which the communication range between the mobile elements and sensor nodes has been exploited to facilitate the data collection, as we did in NJNC. A *traveling salesman problem with neighbourhoods* (TSPN)-based schedule has been proposed, and the system stability under it has been proved. We treat the TSPN-based schedule as the state-of-the-art in the on-demand data collection problem in sensor networks and adopt it as an important benchmark in our evaluations.

## 3 On-Demand Data Collection

### 3.1 On-Demand Scenario versus Offline Scenario

Numerous research efforts on the mobile data collection in sensor networks exist, and most of them focus on the *offline* scenario, where the mobile elements carry out the data collection in a *periodic* manner. A potential issue with this periodic data collection is that certain sensor nodes may not have data to upload, and visiting these nodes just to find out no data to collect is clearly not good for a high-efficient data collection process.

Observing the limitation of the offline scenario, in this paper we consider the *online* scenarios, in which the data collection is carried out by the mobile elements based on the real-time demands from sensor nodes, i.e., the data collection is carried out in an *on-demand* manner.

More specifically, in the on-demand mobile data collection problem, we consider the scenario where mobile elements travel around in the sensing field to collect data from sensor nodes with wireless communication techniques. Sensor nodes gather sensory data according to application requirements, and when they have gathered enough data to report, or have captured certain events that are of particular interests, sensor nodes will send data collection requests to the MEs by adopting existing lightweight and efficient protocols that can track and communicate with the MEs [3], [31].

Note that usually these tracking protocols rely on the multi-hop forwarding of messages among sensor nodes. Thus adopting these protocols to report the sensory data, which are usually of much larger size, e.g., in camera sensor networks [32], is clearly not a good choice for the node energy efficiency. This is also the reason that we introduce the MEs to the network.

The ME maintains a buffer to store the received requests, and serve them according to its service discipline. As a baseline, in this paper we consider the case where only one ME is available in the network, and discussion on the case of multiple MEs is provided in Section 7.

Different from the offline scenario, the on-demand data collection shows great dynamics, which reside in both the temporal domain, i.e., when a new data collection request will arrive, and the spatial domain, i.e., where the new request will come from. This dynamic property means our objective will shift from the optimal path planning for the ME [17], [19], [29], [30], [44], to the design of efficient real-time service disciplines to select the request to serve next. The real-time character of the on-demand scenario also impose a low computation complexity requirement on the service disciplines. Fig. 1 summaries the differences between the on-demand data collection and that under the offline scenario.

### 3.2 Our Approaches

In the on-demand data collection, the ME needs to select request from its buffer as the next one to serve, which shows a clear queuing behavior. Inspired by this, our approach is to model the on-demand data collection process as a queuing system (an $M/G/1/c$ queuing system, as will be explained in Section 4), and theoretically analyze the performance of data collection with different service disciplines. These analytical results reveal insights on the data collection performance and guide the design of more sophisticated service schemes.

When talking about service disciplines, the first one comes to our mind would be the *first-come-first-serve* discipline, whose performance has been extensively explored by the queuing theory society [39]. However, FCFS discipline schedules requests based on their *temporal* features, which may make the ME unnecessarily move back-and-forth, as shown in Fig. 2a. This is clearly undesirable since usually the travel time of the ME dominates the data collection latency.

Noticing the inefficiency of FCFS, we explore two other disciplines that schedule the requests according to their *spatial* features. The first service discipline we explore is the *nearest-job-next* discipline (Fig. 2b), i.e., on finishing the service of the current data collection request, the ME selects the spatially nearest requesting sensor node in its service queue according to its current location, and travels to the node to collect data. Furthermore, with wireless communications, it is possible for the ME to collect data from multiple sensor nodes at a single collection site, provided that the collection site is within the communication ranges of all these nodes. Inspired by this, we extend the NJN discipline to *NJN-with-combination*, with which other requests in the queue can be combined with the nearest one and served together when possible. Analytical results on the data collection with NJNC are also derived to quantitatively evaluate the performance improvement by the possible requests combination.

To simplify the calculation and presentation, we assume a unit square sensing field in this paper. This assumption can be relaxed without violating the analysis and equivalent results can be obtained accordingly, although the computation may be more complicated. Also, we assume that the time from a sensor node sending out its request till the ME receiving the request is negligible when compared with the travel time the ME takes to serve the request. Discussion on how to remove these two simplifications can be found in Section 7.

### 3.3 Definitions and Symbols

For clarity, we list and briefly explain here the definitions used in this paper.

- $v$: the travel speed of the ME (normalized w.r.t. the side length of the field);
- $r$: the wireless communication range (normalized w.r.t. the side length of the field) between the ME and sensor nodes;
- $c$: the maximal number of requests that the ME can accommodate at the same time;
- $\mathcal{S}$: the service time of requests, or $\mathcal{S}_l$ for the to-be-served request selected when there are $l$ data collection requests in the service queue;
- $\mathcal{L}$: the size of the queuing system, i.e., the number of requests that are waiting for or under service;
- $\mathcal{I}$: the idle period of the queuing system;
- $\mathcal{B}$: the busy period of the queuing system;
- $\boldsymbol{\pi} = \{\pi_0, \pi_1, \ldots, \pi_{c-1}\}$: the system size probabilities at the departure time of requests;
- $\boldsymbol{w} = \{w_0, w_1, \ldots, w_c\}$: the system size probabilities at the arrival time of requests;
- $\boldsymbol{u} = \{u_0, u_1, \ldots, u_c\}$: the steady-state system size probabilities of the queuing system.

## 4 Data Collection with NJN

We explore the case where the ME serves data collection requests with the NJN discipline in this section. More specifically, by considering the arrival and departure processes of data collection requests, we first model the system as a non-preemptive $M/G/1/c\text{-}NJN$ queuing system, and then obtain the analytical results on the system measures, which offer important insights on evaluating the performance of the data collection.

### 4.1 $M/G/1/c\text{-}NJN$ Queuing Model

The obvious queuing behavior in the data collection process inspires us to construct a queuing model to capture its performance, in which the ME acts as the server and data collection requests are treated as clients. For any queuing model, the clients arrival and departure processes are the two most fundamental components, so we characterize them respectively in the following.

#### 4.1.1 Arrival Process

For a particular sensor node in a stable network, the probability for it to send out a data collection request at a given time instant is small, and the total number of sensor nodes in the network is usually expected to be relatively large. Theoretically, if the client population of a queuing system is relatively large and the probability by which clients arrive at the queue at a given time instant is relatively small, the arrival process can be adequately modeled as a Poisson process [34]. (Similar conclusions are also obtained in [35, Proposition 1.12], , pp. 11). Based on this, we adopt a Poisson process to capture the arrival of data collection requests to the ME, which is further verified by an event-driven simulator in Section 6.1.

#### 4.1.2 Departure Process

The data propagation speed in sensor networks is about several hundred meters per second [30], which is much faster than the travel speed of the ME. Thus we assume the communication time between the ME and the sensor node to accomplish the data uploading is negligible (further discussed in Section 7). In this way, the service time of each request can be captured by the time from the service completion of the current request to the time when the ME moves to the next selected requesting sensor node.

Due to the fact that the last collection site is also the starting point of the travel when the ME serves the next request, the service time of consecutively served requests seem not to be stochastically independent. However, denote the sequence of service times as $\{t_1, t_2, t_3, \ldots\}$, and if we examine only at every second element of the original process, it is clear that $\{t_1, t_3, t_5, \ldots\}$ are independent of each other, and the distribution-ergodic property of this sub-process can be easily observed [36]. The same is true for sub-process $\{t_2, t_4, t_6, \ldots\}$. The distribution-ergodic property still holds if we combine these two sub-processes since their asymptotic behaviors do not change after the combination. This means that if we can find the time distribution when the ME travels between consecutive to-be-served sensor nodes, we can adopt it as the service time distribution for the queuing system over a long time period, i.e.,

$$F_{\mathcal{S}}(x) = \lim_{h\to\infty} \sum_{i=0}^{h} \frac{1}{h} \cdot \Pr\{t_i \le x\}. \tag{1}$$

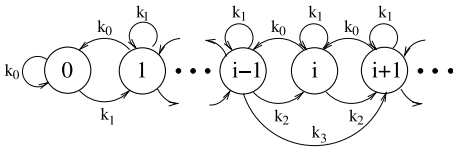



Fig. 3. State transition at departure time with FCFS.

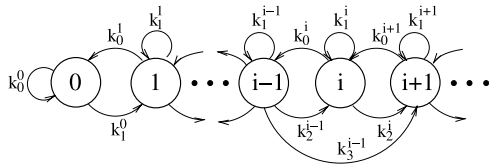



Fig. 4. State transition at departure time with NJN.

One challenge in the performance analysis of NJN is its *dynamic* service time, in the sense that it is determined by both the location of the requesting sensor node and that of the ME just before its service. This dynamic property of service time makes the existing results on the traditional *shortest-job-next* discipline not applicable in our problem, which requires the service times of clients are both known and fixed upon their arrival to the queue [23]. We tackle this problem by adopting existing results on geometrical probability.

By the results in [9], the distribution of the distance between two random locations in a unit square is

$$f_D(d) = \begin{cases} 2d(\pi - 4d + d^2) & 0 \le d \le 1, \\ 2d\left[2\sin^{-1}\left(\frac{1}{d}\right) - 2\sin^{-1}\sqrt{1-\frac{1}{d^2}}\right. & \\ \left. +4\sqrt{d^2-1} - d^2 - 2\right] & 1 \le d \le \sqrt{2}, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

and $F_D(d) = \int_0^d f_D(x)dx$.

Based on (2), and with the constant travel speed $v$ of the ME, the travel time distribution between two uniformly and randomly distributed sensor nodes is thus

$$F_T(t) = \Pr\{D \le vt\} (0 \le t \le \sqrt{2}/v), \tag{3}$$

and $f_T(t) = \partial F_T(t)/\partial t$.

The analysis of NJN becomes even more complicated due to its greedy nature: the service time of the to-be-served request tends to be shorter if more requests are waiting in the system, which makes the service time *state-dependent*. We begin our analysis by conditioning on the system size to tackle the conditional service time.

#### 4.1.3 Finite Capacity System

Note that for a specific application, certain constraints on the maximal acceptable data collection latency exist, either because of the requirement on the timely delivery of sensory data or the possible buffer overflow of sensor nodes. Thus a finite capacity queuing model is a better choice to capture the system behavior when compared with the regular $M/G/1$ queue. However, additional attention is needed for those requests who arrive and find a fully occupied system.

### 4.2 Service Time with a Given System Size

Denote $l$ as the number of requests that are waiting for service when the ME just accomplishes the service of the current request, and is about to select the next request, i.e., the one sent by the nearest requesting sensor node according to the current location of the ME, to serve. Observing the randomness of both the current location of the ME and the requesting sensor nodes, the distances from these $l$ requesting nodes to the ME can be approximately viewed as $l$ i.i.d. random variables with distribution $f_D(d)$ (further verified in Section 6.3.1). Thus based on the results on the *first-order statistic*, the probability distribution of the distance between the ME and the nearest requesting node is

$$F_{D,l}(d) = 1 - (1 - F_D(d))^l, \tag{4}$$

with probability density function (PDF) $f_{D,l}(d) = \partial F_{D,l}(d)/\partial d$. The conditional service time distribution of the nearest request with a given system size is thus

$$f_{\mathcal{S}_l}(t) = v \cdot f_{D,l}(vt), \tag{5}$$

where $\mathcal{S}_l$ represents the service time with system size $l$.

### 4.3 System Size at Departure Time

If we record the system size every time it changes, we can see the next system size only depends on the current one, which is exactly the *Markov property*. Also, the time the system stays at the current size is a random variable jointly determined by the arrival and the departure processes of requests. From these, we can see the evolution process of system size is a *semi-Markov process*. Then if we only examine the system size at the departure time of the requests, we can observe an *embedded discrete-time Markov chain*, which is similar to the case with FCFS [10] (Fig. 3). However, since the service time is now dependent on the current system size, the chain becomes heterogeneous in its transition probabilities (Fig. 4).

With the conditional service time distribution obtained by (5), we can define and calculate the probability of $i$ new arrivals during the time serving a request selected from $l$ available ones as

$$k_i^l = \int_0^{\sqrt{2}/v} \frac{e^{-\lambda t}(\lambda t)^i}{i!} f_{\mathcal{S}_l}(t)dt, \tag{6}$$

where $\lambda$ is the intensity of the arrival process. Note that for FCFS shown in Fig. 3,

$$k_i = \int_0^{\sqrt{2}/v} \frac{e^{-\lambda t}(\lambda t)^i}{i!} f_T(t)dt. \tag{7}$$

The state transition matrix of the finite capacity queue with NJN at the request departure time is then

$$\mathbf{P} = \{p_{ij}\} = \begin{bmatrix} k_0^0 & k_1^0 & \cdots & k_{c-1}^0 & 1-\sum_{i=0}^{c-1} k_i^0 \\ k_0^1 & k_1^1 & \cdots & k_{c-1}^1 & 1-\sum_{i=0}^{c-1} k_i^1 \\ 0 & k_0^2 & \cdots & k_{c-2}^2 & 1-\sum_{i=0}^{c-2} k_i^2 \\ 0 & 0 & \cdots & k_{c-3}^3 & 1-\sum_{i=0}^{c-3} k_i^3 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & k_0^c & 1-k_0^c \end{bmatrix}, \quad (8)$$

and $\pi\mathbf{P} = \pi$. Thus $\pi$ can be calculated by $\sum_{i=0}^{c-1} \pi_i = 1$.

### 4.4 General Service Time

We have derived the conditional service time distribution in Section 4.2, and have just obtained the system size probabilities at departure times in Section 4.3. Note that the service time of the current request is determined at the departure instant of the previous request in most cases, and the only exception is when the previous departure leaves behind an empty system, in which case the service time simply follows the distribution of $f_{\mathcal{S}_1}(t)$. Thus, we can obtain the general service time distribution of requests as

$$F_{\mathcal{S}}(t) = \pi_0 \cdot \int_0^t f_{\mathcal{S}_1}(x)dx + \sum_{l=1}^{c-1} \pi_l \cdot \int_0^t f_{\mathcal{S}_l}(x)dx, \quad (9)$$

and $f_{\mathcal{S}}(t) = \partial F_{\mathcal{S}}(t)/\partial t$. The expected service time of the system can be calculated by

$$E[\mathcal{S}] = \int_0^{\sqrt{2}/v} t \cdot f_{\mathcal{S}}(t)dt. \quad (10)$$

### 4.5 Steady-State System Size

It is proved in [39] that with an infinite system capacity, the departure time system size probabilities of the standard $M/G/1$ queue are the same as those in the steady state. However, this conclusion does not hold in the finite capacity case, since we only have $c$ states for the departure time system size $(0, 1, \ldots, c-1)$, while $c+1$ states $(0, 1, 2, \ldots, c)$ have to be considered for the steady state distribution. With the *level crossing methods*, we can observe the fact that the distribution of system sizes just prior to the arrival time is identical to the departure time probabilities as long as arrivals and departures occur individually, which clearly holds for the Poisson arrival and the NJN disciplines. Thus

$$\begin{aligned} \pi_i = & \Pr\{\text{new request finds } i \text{ in queue} \mid \text{request joins}\} \\ & = w_i/(1-w_c)(0 \le i \le c-1), \end{aligned}$$

so

$$w_i = (1-w_c)\pi_i \ (i = 0, 1, \ldots, c-1). \quad (11)$$

To obtain $w_c$, we can equate the arrival rate with the departure rate of the system,

$$\begin{aligned} \lambda(1-w_c) &= (1-w_0)/E[\mathcal{S}], \\ w_c &= 1-(1-w_0)/(\lambda \cdot E[\mathcal{S}]). \end{aligned} \quad (12)$$

Thus $\mathbf{w}$ can be obtained based on (11) and (12). Furthermore, with the PASTA property of the Poisson arrival process, $\boldsymbol{u} = \boldsymbol{w}$, therefore the steady state system size distribution is derived. The expectation of the steady state system size can be calculated as $E[\mathcal{L}] = \sum_{i=0}^{c} i \cdot u_i$, and a new request arrives to find a fully occupied system and thus gets dropped with probability $w_c$. Note that certain existing hybrid data collection protocols (i.e., using multiple homogeneous or heterogeneous MEs) could be good choices to address these dropped requests if the missing data are indeed needed [40].

### 4.6 Response Time

The ultimate metric to evaluate the data collection performance is the response time $\mathcal{R}$ of requests, i.e., from the time the request is sent to the time it is served. With the expected system size and by *Little's Law*, we know

$$E[\mathcal{R}] = E[\mathcal{L}]/(\lambda(1-w_c)). \quad (13)$$

However, similar to the case with the traditional SJN discipline, the possible starvation problem needs to be investigated when NJN is adopted due to its greedy nature.

We argue that although NJN may suffer similar problem, its severity would be much less significant for the following two reasons. First, the service time of requests with NJN cannot be arbitrarily large, which is bounded by the maximum travel time of the ME when serving two consecutive requests, i.e., $\sqrt{2}/v$ to cross the sensing field. Second, the service time of a specific request changes as the ME travels in the field, and the probability that it keeps at a large value during a long time period is small.

However, the expected response time shown in (13) is not enough to verify the above reasoning, and more insights on the distribution of response time are needed. For a given request, its response time $\mathcal{R}$ consists of three parts, the residual service time $\mathcal{S}_r$ of the request under service upon its arrival, the waiting time from the first departure after its arrival to the time it enters service $\mathcal{W}$, and its service time $\mathcal{S}$, i.e.,

$$\mathcal{R} = \mathcal{S}_r + \mathcal{W} + \mathcal{S}. \quad (14)$$

Investigating one step further, we can see that the waiting time $\mathcal{W}$ is actually the sum of the service time of those requests served before the given request. Based on the results on the general service time of the queuing system (Section 4.4) and by *convolution theorem*, the distribution of the waiting time when conditioning on the fact that $i$ requests are served ahead can be obtained as

$$f_{\mathcal{W}}(t, i) \sim f_{\mathcal{S}}^{(i)}(t), \quad (15)$$

where $f^{(x)}(\cdot)$ is the $x$-fold convolution of $f(\cdot)$.

Again, consider the case that $l$ requests are waiting in the queue for service when the ME selects the next one to serve. As mentioned above, these $l$ distances from the ME to each of the requesting sensor nodes can be viewed as $l$ i.i.d random variables, and thus the probability for any one of them to be selected as the next to serve is approximately $1/l$.

Note that it is clear that given a specific requesting node, the probability for its request to be selected as the next one is dependent on its location in the field, and this probability varies for different requesting nodes (consider two requesting nodes located at the corner and the center of the field, respectively). However, if we focus on a random request in the queue and look at its asymptotic behavior, the selection probability of $1/l$ clearly follows.

Summing over all possible $l$, the probability for a request to be selected would be

$$p_s = \sum_{l=1}^{c-1} \pi_l/(l(1-\pi_0)). \tag{16}$$

Combining (15) and (16) , the distribution of the waiting time is thus

$$f_{\mathcal{W}}(t) = \sum_{i=1}^{N} f_{\mathcal{W}}(t,i)(1-p_s)^{i-1}p_s. \tag{17}$$

where $N$ is the number of sensor nodes in the network. Furthermore, it is well known that the distribution of the residual service time is [39]

$$f_{\mathcal{S}_r}(t) = (1-f_{\mathcal{S}}(t))/E[\mathcal{S}]. \tag{18}$$

From (14), (17), and (18), we know the response time of requests follows a distribution of

$$f_{\mathcal{R}}(t) \sim f_{\mathcal{S}_r}(t) * f_{\mathcal{W}}(t) * f_{\mathcal{S}}(t), \tag{19}$$

where $*$ represents the convolution operation.

With the response time distribution of requests, we can directly obtain observations on the possible starvation problem from the tail of the distribution, i.e., the longer the tail, the more serious is the starvation problem. We will examine this again in Section 6.3.5.

The above results on the response time are obtained based on the convolution theorem. Although theoretically sound, one potential limitation with convolution theorem is that depending on the specific form of the functions, sometimes it may not have closed-form analytical solutions. To address this limitation, next we investigate the busy period of the ME, which can serve as a stochastic upper bound of the response time when the analytical results on its distribution cannot be obtained.

### 4.7 Busy Period

Our approach to tackling the busy period of the ME is to approximate its distribution based on the analytical results of its statistical moments with verification. Note that with a Poisson arrival, the idle period distribution of the system is $F_{\mathcal{I}}(t) = 1 - e^{-\lambda t}$. Denote $\mathcal{T}_{i,j}$ as the time the system takes from entering state $i$ till it entering state $j$, and thus $\mathcal{T}_{0,0}$ is the busy cycle of the system. By conditioning on the number of new arrivals when serving a request, we know

$$\begin{aligned}
E[\mathcal{T}_{0,0}] &= E[\mathcal{I}+\mathcal{S}_1]k_0^1 + E[\mathcal{I}+\mathcal{S}_1+\mathcal{T}_{1,0}]k_1^1 \\
&\quad + E[\mathcal{I}+\mathcal{S}_1+\mathcal{T}_{2,0}]k_2^1 + \cdots \\
E[\mathcal{T}_{1,0}] &= E[\mathcal{S}_1]k_0^1 + E[\mathcal{S}_1+\mathcal{T}_{1,0}]k_1^1 \\
&\quad + E[\mathcal{S}_1+\mathcal{T}_{2,0}]k_2^1 + \cdots \\
E[\mathcal{T}_{2,0}] &= E[\mathcal{S}_2+\mathcal{T}_{1,0}]k_0^2 + E[\mathcal{S}_2+\mathcal{T}_{2,0}]k_1^2 \\
&\quad + E[\mathcal{S}_2+\mathcal{T}_{3,0}]k_2^2 + \cdots \\
&\cdots \quad \cdots
\end{aligned}$$

with some simple arrangement, we have

$$E[\mathcal{T}_{i,0}] = \begin{cases} E[\mathcal{I}] + E[\mathcal{S}_1] + \sum_{j=1}^{c} E[\mathcal{T}_{j,0}]k_j^1 & i=0, \\ E[\mathcal{S}_i] + \sum_{j=1}^{c} E[\mathcal{T}_{j,0}]k_{j-i+1}^i & i>0. \end{cases} \tag{20}$$

Thus $E[\mathcal{T}_{0,0}]$ can be calculated, and the second-order moment of $\mathcal{T}_{0,0}$ can be obtained with a similar approach

$$\begin{aligned}
E\left[\mathcal{T}_{0,0}^2\right] &= E[(\mathcal{I}+\mathcal{S}_1)^2]k_0^1 + E[(\mathcal{I}+\mathcal{S}_1+\mathcal{T}_{1,0})^2]k_1^1 \\
&\quad + E[(\mathcal{I}+\mathcal{S}_1+\mathcal{T}_{2,0})^2]k_2^1 + \cdots \\
E\left[\mathcal{T}_{1,0}^2\right] &= E[\mathcal{S}_1^2]k_0^1 + E[(\mathcal{S}_1+\mathcal{T}_{1,0})^2]k_1^1 \\
&\quad + E[(\mathcal{S}_1+\mathcal{T}_{2,0})^2]k_2^1 + \cdots \\
E\left[\mathcal{T}_{2,0}^2\right] &= E[(\mathcal{S}_2+\mathcal{T}_{1,0})^2]k_0^2 + E[(\mathcal{S}_2+\mathcal{T}_{2,0})^2]k_1^2 \\
&\quad + E[(\mathcal{S}_2+\mathcal{T}_{3,0})^2]k_2^2 + \cdots \\
&\cdots \quad \cdots,
\end{aligned}$$

and after some arrangement, we have

$$E\left[\mathcal{T}_{i,0}^2\right] = \begin{cases} E[(\mathcal{I}+\mathcal{S}_1)^2]k_0^1 \\ \quad + \sum_{j=1}^{c} E[(\mathcal{I}+\mathcal{S}_1+\mathcal{T}_{j,0})^2]k_j^1 & i=0, \\ E\left[\mathcal{S}_1^2\right]k_0^1 + \sum_{j=1}^{c} E[(\mathcal{S}_1+\mathcal{T}_{j,0})^2]k_j^1 & i=1, \\ \sum_{j=1}^{c} E[(\mathcal{S}_i+\mathcal{T}_{j,0})^2]k_{j-i+1}^i & i>1. \end{cases} \tag{21}$$

Therefore, the first and the second-order moments of the ME's busy period $\mathcal{B}$ can be calculated as

$$E[\mathcal{B}] = E[\mathcal{T}_{0,0}] - E[\mathcal{I}], \tag{22}$$

$$E[\mathcal{B}^2] = E[\mathcal{T}_{0,0}^2] - E[\mathcal{I}^2] - 2E[\mathcal{B}]E[\mathcal{I}]. \tag{23}$$

Observing the fact that the busy period of the system is actually the sum of the service times of several consecutively

served requests, we adopt the Gamma distribution to approximate that of the busy period as

$$f_{\mathcal{B}}(t) = t^{\eta-1}e^{-t/\theta}/(\theta^{\eta}\Gamma(\eta)) \ (t>0), \tag{24}$$

where $\eta = 1/(E[\mathcal{B}^2]/E[\mathcal{B}]^2 - 1)$ and $\theta = E[\mathcal{B}]/\eta$. The accuracy of this approximation is verified in Section 6.3.6.

## 5 Data Collection with NJNC

We have theoretically analyzed the system measures when the NJN discipline is adopted in the previous section. In this section, we extend NJN by taking the wireless communication properties into account: with wireless communications, the ME can collect data from several requesting nodes at the same collection site, provided that the site is within the communication ranges of these nodes. We consider this NJNC discipline in the following, with which the ME still selects the spatially nearest requesting node as the next one to serve, except that if there are other requesting nodes that are within the communication range of the nearest one, the ME will combine these requests and serve them together.

The data collection tends to fail due to the poor communication quality when the distance between the ME and the corresponding sensor node approaching the communication range of adopted devices. However, it is shown through measurement studies that the packet reception rates are uniformly high up to a certain transceiver distance [8]. This means the combination of requests is still reasonable since we can find a proper value for the communication range as the input of the NJNC discipline based on empirical results or deployment measurements.

The combination, if happens, can effectively reduce the system size, and thus shorten the response time of requests. It can also alleviate the possible starvation problem, since intuitively, starvation is less likely to happen with a smaller system size. We mainly deal with two questions in this section: how likely the combination can happen, and if it happens, how many requests can be combined; to what degree the combination can improve the data collection performance.

### 5.1 Combination Probability

For the requests combination to happen, the collection site must be covered by the communication ranges of at least another requesting node, besides the nearest one. Same as before, assume $l$ requests are currently available when the ME selects the next request. The probability that $X_l$ requesting nodes, including the nearest one, can be combined together and served from one collection site can be captured by a binomial distribution

$$\Pr\{X_l = x\} = \binom{l-1}{x-1} F_D(r)^{x-1}(1-F_D(r))^{l-x}. \tag{25}$$

Thus the expected number of combined requests is

$$E[X_l] = \sum_{i=1}^{l} i \cdot \Pr\{X_l = x\}, \tag{26}$$

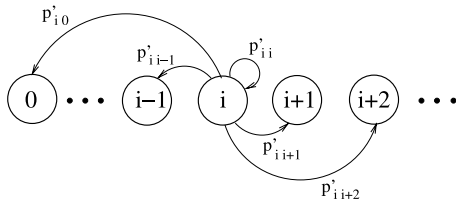



Fig. 5. State transition at departure time with NJNC (only those for state $i$ are shown for clarity).

and the probability for the combination to happen is

$$\Pr\{X_l > 0\} = 1 - (1 - F_D(r))^{l-1}. \tag{27}$$

Requests combination improves the system performance by effectively reducing the system size. Based on a similar $M/G/1/c$ queuing model as that for the NJN discipline, we present quantitative analysis on its impact on the system performance in the following.

### 5.2 $M/G/1/c$-$NJNC$ Queuing Model

The ME still selects the nearest request to serve with NJNC, so given the system size, the conditional service time distribution is identical to that of NJN. Following a similar idea of the embedded Markov chain, we can derive its departure time system size probabilities, and the results shown in (4), (5), and (6) still hold.

The difference between the heterogeneous Markov chains of NJNC and NJN is that, it is now possible to have multiple departures after one service period, as shown in Fig. 5. With the current system size $l$, denote $a_{n,l}$ as the probability of $n$ arrivals during the service of the nearest requests selected from $l$ requests with possible requests combination. Since the combination does not affect the arrival process, we know

$$a_{n,l} = k_n^l. \tag{28}$$

Denote $d_{m,l}$ as the probability of $m$ departures after serving the nearest requests selected from $l$ requests with possible requests combination, which means $m-1$ requests have been served together with the nearest one,

$$d_{m,l} = \binom{l-1}{m-1} F_D(r)^{m-1}(1-F_D(r))^{l-m}. \tag{29}$$

Thus the state transition probabilities of the embedded Markov chain for NJNC, i.e., the counterparts of (8) under NJNC, are

$$\mathbf{P}' = [p'_{ij}] = \left[\sum_{m=1}^{i} d_{m,i} \cdot a_{j-i+m,i}\right], \tag{30}$$

and $\pi'\mathbf{P}' = \pi'$, where $\pi'$ is the departure time system size distribution with NJNC, and $\sum_{i=0}^{c-1}\pi'_i = 1$. Hence, the general service time distribution can be derived with the same approach as in (9) and (10).

Based on (26) and $\pi'$, we have the following result on the system throughput with NJN and NJNC, i.e., the number of

served requests during a unit time, denoted as $\psi_{njnc}$ and $\psi_{njn}$, respectively,

$$\frac{\psi_{njnc}}{\psi_{njn}} = 1 + \left( \sum_{i=1}^{c-1} \pi'_i \cdot E[X_i] - 1 \right)^+. \quad (31)$$

When the requests intensity in the network is heavy, i.e., $\lambda \to \infty$, (31) can be simplified as

$$\frac{\psi_{njnc}}{\psi_{njn}} = E[X_c] \; as \; \lambda \to \infty. \quad (32)$$

The analysis on the request response time and the busy period of ME is much more complicated with NJNC, due to the possible multiple departures after one service, but the idea of obtaining these results still holds. However, we cannot use (11) and (12) to *precisely* calculate the steady state system size in the NJNC case anymore, because the possible multiple departures after one service invalidates (11). We can see in this case the results obtained by (12) are essentially a stochastic upper bound of the arrival time system size probabilities, which makes the results returned by (13) also an upper bound of the expected response time.

## 6 Performance Evaluations

We evaluate our modeling and analytical results on the performance of data collection with NJN and NJNC disciplines in this section, and we also show the corresponding results with FCFS in certain cases for comparison. Note that we have already explored the case with *FCFS-with-combination* (FCFSC) in [10], whose results are not shown here due to the space limit. We consider a $100 \times 100$ m$^2$ square sensing field with $100$ randomly distributed sensor nodes, unless otherwise specified. Based on the parameters from real systems [30], the travel speed of ME is $1$m/s. The communication range $r$ is $20$ m and system capacity $c$ is $8$ by default unless otherwise specified.

To deal with the inconvenience of the piecewise distance density function in (2), we approximate it by a polynomial function of order $10$ using *least squares fitting* with a verified accuracy [9]:

$$\begin{aligned}\widetilde{f}_D(d) = {} & 0.2802d^{10} - 2.0964d^9 + 2.2349d^8 + 24.3629d^7 \\ & - 106.8231d^6 + 194.4928d^5 - 182.8093d^4 \\ & + 91.8223d^3 - 29.3663d^2 + 8.2843d - 0.04.\end{aligned} \quad (33)$$

### 6.1 Validation of the $M/G/1/c$ Modeling

To validate the $M/G/1/c$ model, we need to examine the assumptions of both the Poisson arrival of requests and the distribution-ergodic property of their service.

We adopt a hot-spot model [37] to capture the data generation in our event-driven simulator. Specifically, several hot spots ($10$ in our simulation) exist in the field, and the probability for an event to occur at a certain location is inverse proportional to its distance to the closest hot spot. When an event occurs, sensor nodes whose sensing range covers it can detect the event and generate sensory data of size $\alpha e^{-\alpha d}$ bits to record it, where $d$ is the distance between the node and the event, and $\alpha$ is set to $0.5$ in our simulation. Sensor node sends out a data collection request when a total volume of 1 KB data are accumulated in its buffer. A total number of $100$ data collection requests are generated and served during each simulation, which is repeated for $100$ times.

For each simulation run, we record the inter-arrival time of the requests, and use an exponential distribution with the same mean value to approximate them. We adopt the *Kolmogorov-Smirnov* (K-S) test with a significance level of $0.05$ to verify the goodness-of-fit of the approximation, and record the percentage of runs that pass it. We repeat the whole process for $40$ times. To verify the distribution-ergodic property of the service time, we also record the service time of each request, and calculate their one-lag autocorrelation (as in [38]). Fig. 6a gives the results of the K-S test and the autocorrelation on the $M/G/1/c$ model, where each point corresponds to one of the $100 \times 100$ simulation runs. The $x$-value of the point is the percentage of simulation runs that pass the K-S test, and the $y$-value is the one-lag autocorrelation. Thus we expect that, if these points are clustered around the right bottom corner, as being observed from the validation results in Fig. 6a, the $M/G/1/c$ queuing model adopted in this paper is confirmed sound and acceptable.

### 6.2 Combination Probability

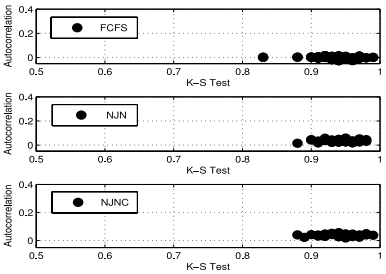
Next we evaluate our analytical results on the combination probability for NJNC. We explore the cases where five, $10$, and $15$ requests are in the system when the ME selects the next request to serve, respectively, and Fig. 6b shows the results obtained with $10$ requests as a representative. Three cases with a communication range of $10$, $20$, and $30$ m respectively are investigated. Besides the accuracy of the analytical results, we can see that the number of combined requests increases with a larger $r$, because it is more likely to collect data from more sensor nodes at the same collection site with a longer communication range. Furthermore, the probability for the combination to happen, even with the shortest explored communication range of $10$ m, is about $25$ percents, which increases to around $90$ percents when $r$ is $30$ m. The high probability for requests combination indicates that NJNC can improve the data collection performance most of the time, when compared with NJN.

### 6.3 System Measures

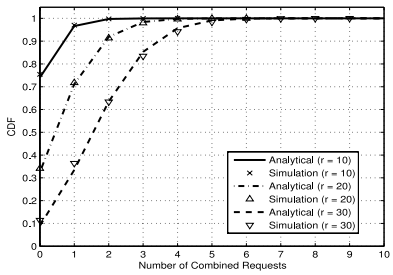
We evaluate the analytical results on the system measures with NJN and NJNC in the following.

#### 6.3.1 Service Time with a Given System Size

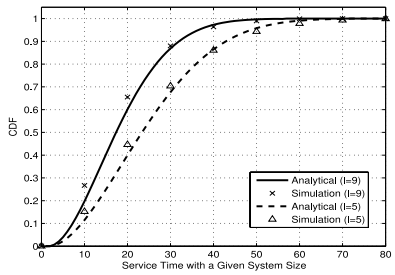
We first examine the results on the service time distribution with a given system size. The conditional service time is derived based on the first-order statistic, which assumes the distances from requesting nodes to the current location of the ME are independent. We start the verification by investigating the independence among these distances. In the simulation, after the deployment of sensor nodes, we randomly select one as the current location of the ME. Then we select



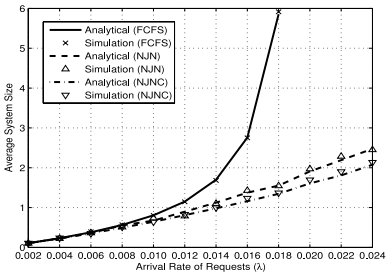

(a) Validation of the Queuing Model



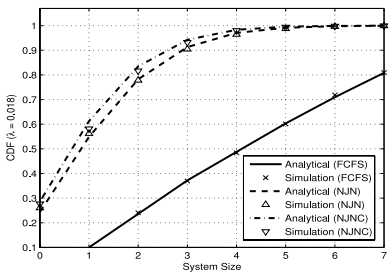

(b) Requests combination probability



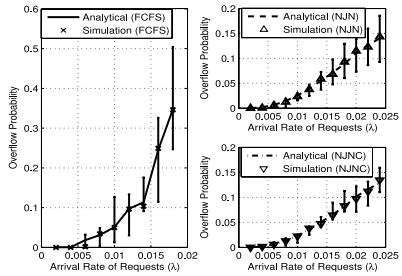

(c) Service time with a given system size



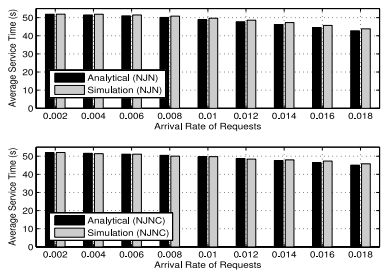

(d) Average system size



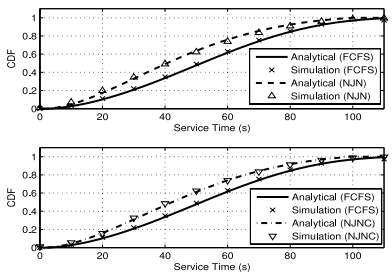

(e) System size distribution



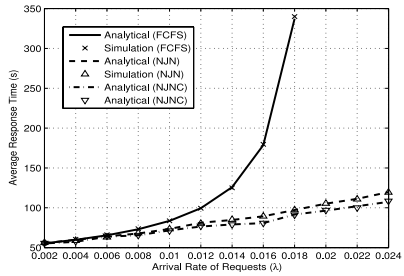

(f) System overflow probability



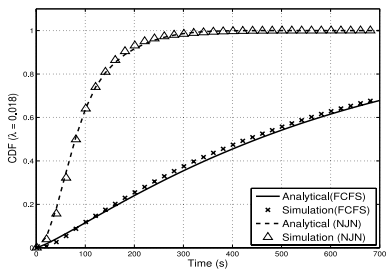

(g) Average service time with NJN and NJNC



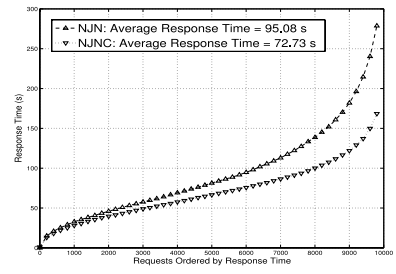

(h) General service time distribution



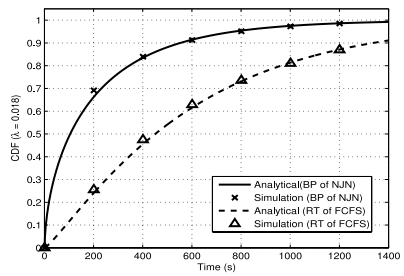

(i) Average response time




(j) Response time distribution with NJN




(k) Response time with NJN and NJNC




(l) Gamma approx. of busy period

Fig. 6. Evaluation results on the queue-based modeling and system measures.

two other sensor nodes as the requesting sensor nodes, calculate their distance to the ME, and record it. We repeat the calculation for $10{,}000$ times, and obtain two distances sequences of size $10{,}000$ each. Then we calculate the correlation coefficient of the two sequences. We repeat the whole process for $20$ times, and the mean square error (MSE) of the resultant correlation coefficients is $6.8010 \times 10^{-5}$, with the estimation of a strict independence (a correlation coefficient of $0$). This near-zero correlation supports our approximation based on their independence.

For the conditional service time distribution, two cases with a small and a large system size of $5$ and $9$ are explored respectively, and the results are shown in Fig. 6c. We can see that the analysis and simulation agree with each other, and the conditional service time becomes shorter with a larger system size, which is consistent with their greedy nature: more queued requests in the system bring more opportunities to have nearer ones.

### 6.3.2 System Size

The ultimate performance metric of data collection is the response time of requests, which is determined by both the service time and system size, and a shorter service time does not necessarily lead to a shorter response time. Thus, we move on to evaluate the system size.

The average system sizes with different request arrival rates $\lambda$ for FCFS, NJN, and NJNC are shown in Fig. 6d. The

system sizes under all the three disciplines are small and comparable to each other when the traffic is light, since the potential for both NJN and NJNC to take effect is quite limited in this case. The system size for FCFS increases very quickly when $\lambda$ increases, and cannot keep stable anymore when $\lambda$ exceeds a certain threshold ($0.018$ in Fig. 6d. In fact, the case of $\lambda = 0.018$ in our simulation roughly corresponds to the case that $\rho = \lambda \cdot E[\mathcal{S}] = 1$ for FCFS, where $\rho$ is the system utilization factor, and increasing $\lambda$ further will result in an unstable system where the steady state does not exist. When compared with FCFS, NJN can reduce the system size greatly because it tries to serve and finish nearby requests in a shorter time, and NJNC can further decrease the system size as a result of possible requests combination. Note that the capability of NJNC to further reduce the system size becomes more obvious when $\lambda$ increases, since a larger system size leads to a larger potential to combine already queued requests. (One thing to mention is that, not surprisingly, the resultant system size with FCFSC falls between those with FCFS and NJN, e.g., an average system size of $1.9$ is resulted with the FCFSC discipline when $\lambda$ is $0.018$ [10].)

We then explore the system size probabilities for FCFS, NJN and NJNC with $\lambda = 0.018$ (Fig. 6e. The maximal system size resultant by NJN and NJNC is around $5$ to $6$, but that for FCFS is too large to be shown clearly in the same figure (given enough system capacity, the maximal system size with FCFS observed in our simulation is 33). Furthermore, we can see that the requests combination shortens the queue consistently.

#### 6.3.3 Overflow Probability

Data collection requests may be lost due to the limited system capacity. To gain insights on this possible data loss problem, we explore the overflow probability of the finite capacity system in Fig. 6f, in which we consider a very small system capacity ($c = 3$) to amplify the problem and obtain a clear observation. These three disciplines result in similar overflow probabilities when $\lambda$ is small. However, the overflow probability for FCFS increases dramatically and becomes much less stable as $\lambda$ increases, and is much larger than those achieved by NJN and NJNC. On the other hand, even with a small capacity, the overflow probabilities for NJN and NJNC are very small and more stable (less than $0.2$ in the worst case by our simulation settings), and NJNC can further reduce the overflow probability by reducing the system size through possible requests combination.

#### 6.3.4 General Service Time

We then examine the general service time for both NJN and NJNC. The average service time of NJN and NJNC with different arrival rate of requests are shown in Fig. 6g. The decreasing trend of the service time as $\lambda$ increases agrees with the greedy nature of both NJN and NJNC with more queued requests. When comparing these two disciplines, the service time of NJNC is slightly larger than that of NJN because of the possible combination, since NJNC reduces the system size more aggressively and thus also reduces the possibility of finding a closer requesting sensor node in the future. Also note that the difference between the two becomes more obvious with a larger $\lambda$.

TABLE 1
Verification of Requests' Selection Probability

| $\lambda$ | 0.03 | 0.04 | 0.05 | 0.06 |
|---|---|---|---|---|
| # of Selections | 3.7166 | 5.0175 | 5.6479 | 6.1289 |
| # of Waiting Reqs. | 3.8036 | 5.0325 | 5.6533 | 6.1312 |

To gain more insights on the service time, Fig. 6h shows the general service time distribution with $\lambda = 0.018$, where the service time distribution for FCFS is also shown for comparison as well. We can see that when compared with FCFS, NJN and NJNC can shorten the service time noticeably because of their nature to serve nearby requests with less time, and similar insights can be observed as in Fig. 6g.

#### 6.3.5 Response Time

As mentioned above, both service time and system size affect the response time. Since NJN has a shorter service time and NJNC has a smaller system size, next we need to examine their response time.

The average response time for FCFS, NJN and NJNC is shown in Fig. 6i, from which we can see that when compared with FCFS, NJN and NJNC can greatly shorten the response time of requests, especially when $\lambda$ is large. NJNC can further reduce the response time as a result of possible requests combination, which also becomes more obvious as $\lambda$ increases. It shows that between a shorter service time for NJN and a smaller system size for NJNC, the system size reduction is more dominating for the resultant response time. A smaller system size also indicates a lower overflow probability for a given system capacity limit. Actually, the above observations can be inferred based on Fig. 6d and the linear relationship between the response time and system size (by Little's Law). We still include Fig. 6i here for a clear and straightforward comparison.

We have derived the distribution of the response time of requests to investigate the potential starvation problem when the NJN (NJNC) discipline is adopted. One fundamental part to obtain the response time distribution is the probability for a request to be selected for service, and we verify this before moving to the response time. For each request, after its arrival, we record the number of selections made by the ME until the request is chosen, and compare this to the average number of requests that are waiting for service in the queue. If (16) holds, then we expect these two results to be close. We intentionally set higher traffic intensities (and thus larger system sizes) for this verification to make the observation more clear. The comparison results are shown in Table 1, which verify the accuracy of (16).

Next we evaluate the results on the response time distribution, which are shown in Fig. 6j ($\lambda = 0.018$). The response time by the FCFS discipline (obtained according to [39]) is also shown for comparison. Besides the accuracy of the analytical results, we can see even the worst case of NJN, in terms of the longest response time experienced by requests, is much smaller than that by the FCFS discipline. This observation is a little unexpected since FCFS is known to be able to achieve a good fairness among clients, but it also alleviates our concerns on the possible starvation problem with NJN (NJNC).

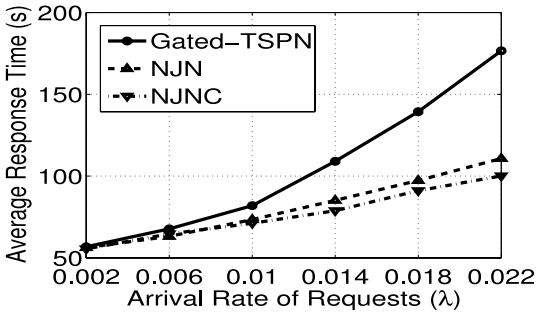



Fig. 7. Response time with request arrival rate $\lambda$.

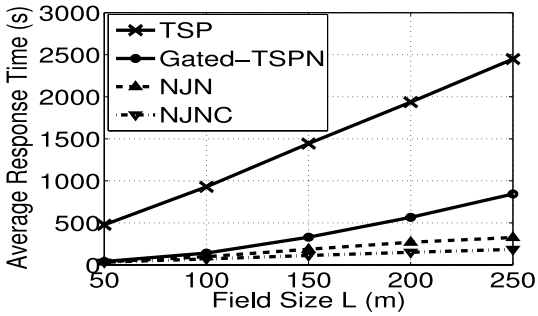



Fig. 8. Response time with field size $L$.

To present a clear comparison between the performance of NJN and NJNC, we presents the response time obtained by a particular run of simulation ($\lambda = 0.018$) in Fig. 6k, where a total number of $10,000$ requests are served (or dropped due to system overflow). We sort the response time in an ascending order, and only plot the response time every $200$ requests for clarity. Not surprisingly, NJNC further reduces the response time in terms of both the average and worst cases.

#### 6.3.6 Busy Period

Fig. 6l shows the evaluation results on the distribution of the busy period of the ME, where $\lambda$ is set to $0.018$. We can see that although the results are obtained through approximation methods, the accuracy is still good. For comparison, the response time distribution obtained with the FCFS discipline is shown here again. We can see that with NJN, even its busy period, which is a stochastic upper bound of the response time, is smaller than the response time with FCFS.

### 6.4 Comparison with Other Scheduling Schemes

In the following, we compare the performance of NJN and NJNC with two classic scheduling schemes:

- *TSP*: the offline optimized TSP-based scheduling scheme, with which the mobile element carries out the data collection according to the optimal solution of the TSP instance formulated based on the node deployment. This is involved in many offline data collection scheme designs [17], [19], [44].
- *Gated-TSPN*: the dynamic schedule scheme based on a TSPN instance according to the available data collection requests [49], which is treated as the state-of-the-art for the on-demand data collection.

The average data collection latency resultant with the Gated-TSPN, NJN, and NJNC are shown in Fig. 7, with the request arrival rate varies from $0.002$ to $0.022$. We can see that the NJN (NJNC) outperforms the Gated-TSPN scheme noticeably, especially when the request arrival rate is large. However, the system stability of the Gated-TSPN scheme has been theoretically guaranteed, while our analysis on NJN and NJNC has not proved this property yet, which is the direction of our future work. The results with the TSP scheme are not shown in the figure because the latency resultant is significantly longer, specifically, around $940\ s$ when $\lambda = 0.006$. However, the latency with the TSP scheme is relatively insensitive to the request arrival rate, e.g., around $960\ \mathrm{s}$ when $\lambda = 0.022$. This infers that the TSP scheme may be a good choice when the intensity of the data collection tasks in the network is quite heavy.

The response time with various field size are shown in Fig. 8, where the $x$-axis is the side length of the square field. The response time with the TSP scheme increases almost linearly, which agrees with the nature of the optimal TSP tour. The NJN and NJNC outperform the Gated-TSPN scheme, and the advantage becomes more obvious with a larger field. Again, we would like to emphasize that although verified to be efficient through simulation, unlike Gated-TSPN, the system stability with NJN and NJNC has not be theoretically proved yet.

## 7 Discussion and Ongoing Work

Our modeling and analysis are shown accurate but some issues need to be further explored. Here we discuss some of these issues and the directions of ongoing work.

Clearly, the assumption of a square sensing field may not hold in practice. However, our queue-based modeling and analysis approaches are still feasible even for general sensing fields, provided that the distance distribution between arbitrary locations in the field can be obtained, e.g., we have also analytically derived the random distance distributions associated with rhombuses and hexagons [41]. Furthermore, even if the sensing field is of irregular shape, which may be true in certain cases, we can adopt the polygon-approximation approach to approximate the field by the combination of several regular shapes [42], and derive the random distance distributions within and between them. Of course, more computation efforts will be needed.

Another simplification we made is that the time for transmitting the data from sensors to the ME is negligible, which may not hold if the data volume is large. However, note that given any sensor network deployment, the knowledge on the data transmission rate and communication range is available, or at least can be estimated through experimental measurements. Based on such knowledge, we can estimate the data volumes that can be collected if the ME travels without any stops, and then the time that the ME has to pause or slow down to collect the remaining data can be calculated as well. The statistics of this additional time can be incorporated when formulating the service time distribution.

The request response time in this work does not include the time since the request is sent by the sensor node to its

reception at the ME, which we assumed to be negligible. Observing the fact that these two times are independent to each other, thus by the convolution theorem, we can easily take the latter into account provided that its distribution $g(t)$ can be estimated,[1] i.e.,

$$f'_{\mathcal{R}}(t) = g(t) * f_{\mathcal{R}}(t). \quad (34)$$

When multiple MEs are available, a straightforward approach is to extend the model to the multi-server case. However, in addition to the queue length and response time, we also need to consider the load balance among the MEs as another metric to evaluate the system performance. Some preliminary results on the on-demand data collection with multiple MEs can be found in [9].

## 8 Conclusions

In this paper, we have analytically evaluated an intuitive service discipline, NJN, for data collection with mobile elements in wireless sensor networks, and also explored the case where the ME follows NJN and combines requests whenever possible, i.e., NJNC. We have modeled the system as an $M/G/1/c$ queue, and then with different service disciplines (NJN and NJNC), critical system metrics have been derived. We have verified our analytical results through extensive simulation, and gained more insights on the starvation problem that NJN and NJNC may suffer from. Our results have showed that not only the average performance of NJN is much better than that of FCFS, but also the worst-case performance of NJN is still better than that of FCFS, even though according to the conventional wisdom, NJN may suffer from the starvation problem. A possible reason is that the service time is not arbitrary for data collection applications in wireless sensor networks. Moreover, NJNC can further improve the performance as a result of possible requests combination. We have also discussed several possible extensions as the ongoing and future work.

## Acknowledgments

An early short version of this work was published at IEEE INFOCOM'12, which was done when L. He was a PhD candidate at Nankai University and a visiting research student at University of Victoria. The work is supported in part by Singapore-MIT International Design Center IDG31000101, grant SUTD SRG ISTD 2010 002, Project GREaT IPMD13012, and grant K93-9-2010-01 from Ministry of Educations Key Lab for Computer Network and Information Integration at Southeast University, China.

## References

[1] U. Park and J. Heidemann, "Data Muling with Mobile Phones for Sensornets," *Proc. ACM Ninth ACM Conf. Embedded Networked Sensor Systems (Sensys'11)*, 2011.

[2] C. Wang, C. Jiang, Y. Liu, S. Tang, X. Li, and H. Ma, "Aggregation Capacity of Wireless Sensor Networks: Extended Network Case," *Proc. IEEE INFOCOM '11*, 2011.

1. We have taken this delivery time into account by proposing a partition-based data collection scheme in a recent work [43].

[3] Z. Li, M. Li, J. Wang, and Z. Cao, "Ubiquitous Data Collection for Mobile Users in Wireless Sensor Networks," *Proc. IEEE INFOCOM '11*, 2011.

[4] R. Shah et al., "Data MULEs: Modeling a Three-Tier Architecture for Sparse Sensor Networks," *Proc. IEEE First Int'l Workshop Sensor Network Protocols and Applications (SNPA)*, 2003.

[5] W. Wang, V. Srinivasan, and K. Chua, "Extending the Lifetime of Wireless Sensor Networks through Mobile Relays," *IEEE/ACM Trans. Networking*, vol. 16, no. 5, pp. 1108-1120, Oct. 2008.

[6] J. Luo and J. Huabux, "Joint Sink Mobility and Routing to Maximize the Lifetime of Wireless Sensor Networks: The Case of Constrained Mobility," *IEEE/ACM Trans. Networking*, vol. 18, no. 3, pp. 871-884, June 2010.

[7] M. Francesco, S. Das, and G. Anastasi, "Data Collection in Wireless Sensor Networks with Mobile Elements: A Survey," *ACM Trans. Sensor Networks*, vol. 8, no. 1, article 7, 2011.

[8] J. Zhao and R. Govindan, "Understanding Packet Delivery Performance in Dense Wireless Sensor Networks," *Proc. ACM First Int'l Conf. Embedded Networked Sensor Systems (Sensys '03)*, 2003.

[9] L. He, J. Pan, and J. Xu, "Analysis on Data Collection with Multiple Mobile Elements in Wireless Sensor Networks," *Proc. IEEE GLOBECOM '11*, 2011.

[10] L. He, Y. Zhuang, J. Pan, and J. Xu, "Evaluating On-Demand Data Collection with Mobile Elements in Wireless Sensor Networks," *Proc. IEEE 72nd Vehicular Technology Conf. Fall (VTC '10-Fall)*, 2010.

[11] http://www.neptunecanada.ca, NEPTUNE Canada, 2011.

[12] http://www.rov-online.com, Sabertooth, 2011.

[13] http://www.wirelessfibre.co.uk, WFS, 2011.

[14] M. Todd et al., "A Different Approach to Sensor Networking for SHM: Remote Powering and Interrogation with Unmanned Aerial Vehicles," *Proc. Sixth Int'l Workshop Structural Health Monitoring (IWSHM '07)*, 2007.

[15] R. Sugihara and R. Gupta, "Optimal Speed Control of Mobile Node for Data Collection in Sensor Networks," *IEEE Trans. Mobile Computing*, vol. 9, no. 1, pp. 127-139, Jan. 2010.

[16] A. Somasundara, A. Ramamoorthy, and M. Srivastava, "Mobile Element Scheduling with Dynamic Deadlines," *IEEE Trans. Mobile Computing*, vol. 6, no. 4, pp. 395-410, Apr. 2007.

[17] R. Sugihara and R. Gupta, "Optimizing Energy-Latency Trade-off in Sensor Networks with Controlled Mobility," *Proc. IEEE INFOCOM '09*, 2009.

[18] A. Kansal et al., "Intelligent Fluid Infrastructure for Embedded Networks," *Proc. ACM Second Int'l Conf. Mobile Systems, Applications, and Services (MobiSys '04)*, 2004.

[19] B. Yuan, M. Orlowska, and S. Sadiq, "On the Optimal Robot Routing Problem in Wireless Sensor Networks," *IEEE Trans. Knowledge and Data Eng.*, vol. 19, no. 9, pp. 1252-1261, Sept. 2007.

[20] A. Srinivasan and J. Wu, "Track: A Novel Connected Dominating Set Based Sink Mobility Model for WSNs," *Proc. IEEE Seventh Int'l Conf. Computer Comm. and Networks (ICCCN '08)*, 2008.

[21] X. Xu, J. Luo, and Q. Zhang, "Delay Tolerant Event Collection in Sensor Networks with Mobile Sink," *Proc. IEEE INFOCOM '10*, 2010.

[22] P. Bhattacharya and A. Ephermides, "Optimal Scheduling with Strict Deadlines," *IEEE Trans. Automatic Control*, vol. 34, no. 7, pp. 721-728, July 1989.

[23] N. Bansal and M. Balter, "Analysis of SRPT Scheduling: Investigating Unfairness," *Proc. ACM Int'l Conf. Measurement and Modeling of Computer Systems (SIGMETRICS '01)*, 2001.

[24] M.M. Salem, A.O. El-Gwad, H.M. Harb, and O.A. Zahran, "Computers Operating Systems: V-SSTF Dynamic Disk Scheduling Technique," *Proc. Third Int'l Conf. Software Eng. for Real Time Systems*, 1991.

[25] M. Ma and Y. Yang, "SenCar: An Energy-Efficient Data Gathering Mechanism for Large-Scale Multihop Sensor Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 10, pp. 1476-1488, Oct. 2007.

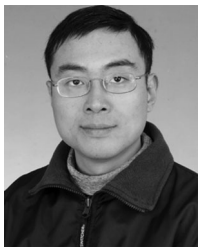
[26] D. Bhadauria, V. Isler, and O. Tekdas, "Efficient Data Collection from Wireless Nodes Under Two-Ring Communication Model," Technical Report UM-CS-11-015, Dept. of Computer Science and Eng., Univ. of Minnesota, 2011.

[27] A. Somasundara, A. Ramamoorthy, and M. Srivastava, "Mobile Element Scheduling for Efficient Data Collection in Wireless Sensor Networks with Dynamic Deadlines," *Proc. IEEE 25th Int'l Real-Time Systems Symp. (RTSS '04)*, 2004.

[28] O. Tekdas, V. Isler, J. Lim, and A. Terzis, "Using Mobile Robots to Harvest Data from Sensor Fields," *Wireless Comm.*, vol. 16, no. 1, pp. 22-28, 2009.
[29] G. Xing et al., "Rendezvous Planning in Mobility-Assisted Wireless Sensor Networks," *Proc. IEEE 28th Int'l Real-Time Systems Symp. (RTSS '07)*, 2007.
[30] G. Xing, T. Wang, W. Jia, and M. Li, "Rendezvous Design Algorithms for Wireless Sensor Networks with a Mobile Base Station," *Proc. ACM MobiHoc '08*, 2008.
[31] X. Liu, H. Zhao, X. Yang, X. Li, and N. Wang, "Trailing Mobile Sinks: A Proactive Data Reporting Protocol for Wireless Sensor Networks," *Proc. IEEE Seventh Int'l Conf. Mobile Adhoc and Sensor Systems (MASS '10)*, 2010.
[32] M. Rahimi et al., "Cyclops: In Situ Image Sensing and Interpretation in Wireless Sensor Networks," *Proc. ACM Third Int'l Conf. Embedded Networked Sensor Systems (SenSys '05)*, 2005.
[33] D. Jea, A. Somasundara, and M. Srivastava, "Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Network," *Proc. IEEE First IEEE Int'l Conf. Distributed Computing in Sensor Systems (DCOSS '05)*, 2005.
[34] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, third ed., Oxford Univ. Press, July 2010.
[35] R. Mazumdar, *Performance Modelling, Loss Networks, and Statistical Multiplexing*. Synthesis Lectures on Communication Networks. Margan & Calypool Publishers, 2009.
[36] 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.
[37] J. Zhao, R. Govindan, and D. Estrin, "Residual Energy Scans for Monitoring Wireless Sensor Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '02)*, 2002.
[38] J. Huang, G. Xing, G. Zhou, and R. Zhou, "Beyond Co-Existence: Exploiting WiFi White Space for ZigBee Performance Assurance," *Proc. IEEE Int'l Conf. Network Protocols (ICNP '10)*, 2010.
[39] D. Gross, *Fundamentals of Queueing Theory*, fourth ed., p. 232. John Wiley & Sons, 2008.
[40] Y. Gu, D. Bozdag, and E. Ekici, "Mobile Element Based Differentiated Message Delivery in Wireless Sensor Networks," *Proc. IEEE WOWMOM '06*, 2006.
[41] Y. Zhuang and J. Pan, "*Random Distances Associated with Hexagons*," arXiv: 1106. 2200, 2011.
[42] H. Sanchez-Cruz and E. Bribiesca, "Polygonal Approximation of Contour Shapes Using Corner Detectors," *J. Applied Research and Technology*, vol. 7, no. 3, pp. 275-291, 2009.
[43] M. Ahmadi et al., "A Partition-Based Data Collection Scheme for Wireless Sensor Networks with a Mobile Sink," *Proc. IEEE Int'l Conf. Comm. (ICC '12)*, 2012.
[44] L. He, J. Pan, and J. Xu, "A Progressive Approach to Reducing Data Collection Latency in Wireless Sensor Networks with Mobile Elements," *IEEE Trans. Mobile Computing*, vol. 12, no. 7, pp. 1308-1320, July 2013.
[45] S.L. Smith, M. Pavone, F. Bullo, and E. Frazzoli, "Dynamic Vehicle Routing with Heterogeneous Demands," *Proc. IEEE 47th Conf. Design and Control (CDC '08)*, 2008.
[46] G. Ghiani, A. Quaranta, and C. Triki, "New Policies for the Dynamic Traveling Salesman Problem," *Optimization Methods and Software*, vol. 22, no. 6, pp. 971-983, 2007.
[47] E. Altman and H. Levy, "Queueing in Space," *Advances in Applied Probability*, vol. 26, no. 4, pp. 1095-1116, 1994.
[48] 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.
[49] G.D. Celik and E.H. Modiano, "Controlled Mobility in Stochastic and Dynamic Wireless Networks," *Queueing Systems*, vol. 72, nos. 3/4, pp. 251-277, 2012.

**Liang He** (S'09-M'12) received the BEng degree in 2006 and the PhD degree in 2011 from Tianjin University, China, and Nankai University, China, respectively. He is currently a postdoc research fellow at Singapore University of Technology and Design. During October 2009 to October 2011, he was at Panlab at University of Victoria as a visiting research student. He received the best paper awards of IEEE WCSP 2011 and IEEE GLOBECOM 2011. He is a member of the IEEE.

**Zhe Yang** (S'09) received the BS degree in information engineering in 2005 and the MS degree in control theory and engineering in 2008, both from Xian Jiaotong University, Xian, China. He is currently working toward the PhD degree at the Department of Electrical and Computer Engineering, University of Victoria, British Columbia, Canada. His current research interests include cross-layer design for cooperative networks, scheduling and resources allocation for wireless networks and synchronization. He is a student member of the IEEE.



**Jianping Pan** (S'96-M'98-SM'08) received the bachelor's and PhD degrees in computer science from Southeast University, China, and he did his postdoctoral research at the University of Waterloo, Canada. He is currently an associate professor of computer science at the University of Victoria, Canada. He was also at Fujitsu Labs and NTT Labs. He received the IEICE Best Paper Award in 2009, the Telecommunications Advancement Foundation's Telesys Award in 2010, the WCSP 2011 Best Paper Award, and the IEEE Globecom 2011 Best Paper Award, and has been serving on the technical program committees of major computer communications and networking conferences including IEEE INFOCOM, ICC, Globecom, WCNC, and CCNC. He is a senior member of the IEEE.

**Lin Cai** (S'00-M'06-SM'10) received the MASc and PhD degrees in electrical and computer engineering from the University of Waterloo, Canada, in 2002 and 2005, respectively. Since 2005, she has been currently an associate professor with the Department of Electrical and Computer Engineering, University of Victoria, Canada. She has been an associate editor for *IEEE Transactions on Wireless Communications*, *IEEE Transactions on Vehicular Technology*, *EURASIP Journal on Wireless Communications and Networking*, *the International Journal of Sensor Networks*, and the *Journal of Communications and Networks*. She is a senior member of the IEEE.

**Jingdong Xu** received the BS, MS, and PhD degrees from the Department of computer science from Nankai University, China, in 1988, 1991, and 2002, respectively. She is currently a professor and the department chair of computer science, Nankai University. His research focuses on wireless ad hoc network,wireless sensor network, and mobile computing.

**Yu (Jason) Gu** received the PhD degree from the Department of Computer Science and Engineering at the University of Minnesota in 2010. He is an assistant professor at the Singapore University of Technology and Design. He is the author and coauthor of more than 21 papers in premier journals and conferences. His publications have been selected as graduate course materials by more than 10 universities in the United States and other countries. He has received several prestigious awards from the University of Minnesota. He is a member of the IEEE.