Toward Distributed Optimal Movement Strategy for Data Harvesting in Wireless Sensor Networks

Chul-Ho Lee Do Young Eun

Department of Electrical and Computer Engineering
North Carolina State University, Raleigh, NC 27695
{clee4, dyeun}@ncsu.edu

***Abstract*—In this paper, we address how to design the distributed movement strategy for mobile collectors, which can be either physical mobile agents or query/collector packets periodically launched by the sink, to achieve successful data gathering in wireless sensor networks. Formulating the problem as general random walks on a graph composed of sensors, we analyze how many data can be successfully gathered in time under any Markovian movement strategies for mobile collectors moving over a graph (or network), while each sensor is equipped with limited buffer space and data arrival rate to each node is heterogeneous. In particular, from the analysis, we obtain the optimal movement strategy among a class of Markovian strategies so as to minimize the data loss rate over all sensors, and explain how such optimal movement strategy can be made to work in a distributed fashion. We demonstrate that our distributed optimal movement strategy leads to about 2 times smaller loss rate than the simple random walk strategy under diverse scenarios. In particular, our strategy can result in about 50% cost savings for the deployment of multiple collectors to achieve the target data loss rate than the simple random walk.**

I. Introduction

Data gathering or harvesting is a generic research problem in wireless sensor networks (WSNs) – how to collect the observed (or measured) data or information from sensor nodes, and so has been actively studied in the literature. For the data gathering process, a sink (or base station) periodically generates query packets or collector packets to gather certain information of interest from sensor nodes [21], [2], [3], [28], [27], or each sensor, instead, directly informs the sink node about its observed data or events [14], [10], [4], [6]. In addition, mobile sinks (agents), e.g., data mules, can move around the sensor field and collect information observed at each sensor [19], [13], [16], [29], [30], [22], [23]. On the other hand, a random walk has been widely used as a means of *randomized* routing or *probabilistic* packet forwarding in the data gathering process thanks to its inherent distributed nature and other preferable properties such as simplicity of implementation, scalability, robustness to topology changes, and avoiding critical points of failure (or hot-spot formation) [21], [2], [3], [28], [27], [14], [10], [4], [6]. In addition, the random walk has been also popularly used as the mobility pattern for the mobile agents [19], [13].

As to the random walk-based routing for data gathering, the existing works have mainly focused on the performance of the following metrics: delay – the time for a random walk (a data or query packet) to reach its destination (a sink node or a sensor having a certain information of interest) [21], [14], [10], [4], [6], and cover time or its partial cover time – the time for the random walk until to visit all or partial set of sensors [2], [3]. These metrics are suitable for *one-shot* information delivery or search/query. In contrast, in this paper, we look at the problem of data gathering using the random walk agent(s) from a different, but important perspective. Note that WSNs are composed of low-cost, low-power sensors, each of which is equipped with limited buffer/storage space for data. Also, the random walk-based data gathering is typically for delay-insensitive applications in which the collected data are mainly used for post-processing or other research works later. It is thus more important to measure how many data can be collected before they are lost due to limited buffer space, when the sink periodically generates query packets or collector packets moving over the network in a random walk fashion to gather measured data or its aggregated/compressed version from sensor nodes [21], [2], [3], [28], [27].

There is another set of research works on exploiting *physical* mobile agents (e.g., data mules) for data harvesting over the sensor field [19], [13], [16], [29], [30], [22], [23] rather than relying on multi-hop communication over sensor nodes. The reason of using the mobile agents is to save the limited battery power of each sensor and also to overcome possible network isolation. In this regard, many research works are based on traveling salesman problem, i.e., finding a Hamiltonian cycle (a NP-hard problem), or its variants, and their solutions are, in general, only globally solvable or require global network information such as sensor location/distance information [29], [30], [16], [22], [23]. In this paper, we focus on a class of Markovian random walks *without such global information*. A crucial reason of using random walks is their distributed nature, as it is desirable or sometimes imperative not to rely on global (or near-global) information. On the other hand, the previous works [19], [13], closely related to our work in this paper, analyzed how many data can be successfully collected by the agents in time under the setting of simple random walks on a grid (torus) [19] or by directly taking into account the time interval between successive visits of any of random walk agents to a sensor [13]. The performance analysis were done from a single node's point of view, as the data arrival rate is the same over all nodes. In contrast,

This work was supported in part by National Science Foundation under grants CNS-0831825, CCF-0830680, and CNS-0545893.

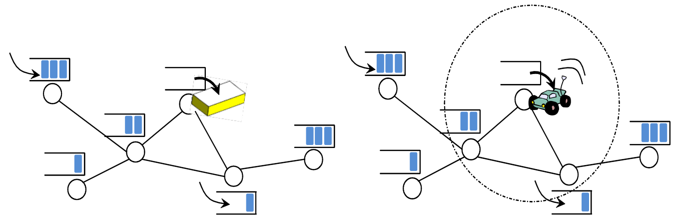
we investigate the problem under a general network (graph) setting, while allowing different data arrival rates to each node. More importantly, our focus is on the *design* of the distributed optimal movement strategy for such mobile agents among a class of Markovian random walks, as opposed to be on the analysis of resulting performance under a given simple random walk strategy.

Specifically, we formulate the data harvesting problem under the framework of a random walk on a graph composed of sensor nodes, as similarly used in [21], [2], [3], [28], [27], [14], [10], [4], [6]. This framework can also be a viable discrete-time/space approximation for the continuous trajectory of a mobile agent over the sensor field. Each sensor has a limited buffer space, while mobile collectors (the query packets periodically launched at the sink or the physical mobile agents) move over the graph (network) for data gathering. Our performance metric is the network loss probability – the ratio of the total number of lost data to the total number of generated data over all sensor nodes on a long time scale. We develop an analytical framework to evaluate the network loss probability under *any* Markovian random walks. In particular, we obtain the distributed optimal movement strategy for mobile collectors requiring only local information so as to minimize the network loss probability. We then demonstrate that our distributed optimal movement strategy leads to remarkable performance improvement over the simple random walk under various settings of buffer sizes and the number of deployed mobile agents in addition to various data arrival scenarios in a sensor field. In particular, our strategy results in about 2 times smaller network loss probability than the simple random walk, while at the same time also becoming more cost-effective in term of the required buffer size or number of mobile collectors to achieve the target loss probability under various scenarios.

II. System Model

There are a set of sensor nodes $\mathcal{N}=\{1,2,\dots,n\}$ and one mobile collector (or possibly, multiple mobile collectors) for data gathering (or harvesting) in the network. We consider a discrete-time system ($t=0,1,2,\dots$) with each slot of unit length. Each sensor node $i\in\mathcal{N}$ monitors its local area and observes some information or data (e.g., temperature) at each time slot with some probability $p_i\in(0,1)$, and then store the observed data in its buffer of size $B=O(1)$. We assume that this procedure at each node is done independently over i and t. When a new data arrives to find a full buffer at time t, the oldest data in the buffer is deleted (or lost) to make a room for the newest one. In other words, the buffer management is done in a FIFO fashion.

Let $G=(\mathcal{N},\mathcal{E})$ be a connected, undirected graph, where $\mathcal{N}$ is a set of vertexes (sensor nodes) and $\mathcal{E}$ is a set of edges. The mobile collector moving over G can be a collector packet or an actual physical mobile agent. So, the existence of an edge between nodes i,j can be different for each case. In the former, we say that if sensor node $i\in\mathcal{N}$ can reliably communicate with other sensor node $j\in\mathcal{N}$ ($i\neq j$), then there



(a) A query/collector packet (b) A physical mobile agent

Fig. 1. Illustrating our system model in which each sensor has limited buffer space and data arrival rate to each node is heterogeneous, while a mobile collector, which can be either a query/collector packet or a physical mobile agent, moves over the graph to collet the data.

exists an edge between nodes i,j, i.e., $(i,j)\in\mathcal{E}$. In this setup, the collector packet at current node i moves to one of its neighbors at next time slot. This model, for example, has been similarly used, but in the context of a data persistence problem in [28], [27] where a collector packet moves over the graph (or network) until to collect a enough number of data packets of interest from sensor nodes. Similar setting has been also adopted in [2], [3]. See Fig. 1(a) for illustration. On the other hand, when the mobile collector is a physical agent, we assume that the mobile agent can visit a single sensor node (or its proximity) at time t, and then move to one of its neighbors within its *detection* range r, at time t+1, as depicted in Fig. 1(b). Thus, if sensor nodes i and j are within distance r, then there is an edge between them, i.e., $(i,j)\in\mathcal{E}$. This model can be a reasonable discrete-time/space approximation for the continuous trajectory of a mobile agent (e.g., data mule) over the sensor field [19]. One critical reason to use the mobile agent(s) for data gathering is due to low energy budget at each sensor. Specifically, in the presence of mobile agent(s), each sensor can significantly reduce the power of its RF transceiver (for idle listening and data transmission/reception), while waiting for the mobile agent to come by for reliable communication. This way, each sensor needs not spend much power to reach distance r; instead, it just needs to wail until the mobile agent gets close enough (much shorter than r) to save its power. In either case, the mobile collector can visit only a single sensor and communicate with the sensor at a given time instant.

Given a graph G with n sensor nodes, we consider a class of strategies in which a mobile collector travels over G in a *random walk* fashion, and visits each sensor to collect all of the currently stored data (or its summarized or compressed version) at the buffer of node i. Let $X(t)$ be the location of mobile collector over G at time t. By definition, the random sequence $\{X(t)\}_{t\geq 0}$ is a discrete-time Markov chain on a finite state space $\mathcal{N}$ with its transition probability given by

$$P_{ij}\triangleq\mathbb{P}\{X(t+1)=j\mid X(t)=i\},\quad i,j\in\mathcal{N},\qquad(1)$$

for all $t\geq 0$. Let $\mathbf{P}\triangleq\{P_{ij}\}_{i,j\in\mathcal{N}}$ be its transition matrix. We assume that the Markov chain $\{X(t)\}$ is irreducible, i.e., every node j is reachable from any other node i in finite time with positive probability. In this case, the Markov chain $\{X(t)\}$ has a unique stationary distribution, denoted as $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n)$ [18]. In this set-up, we tackle the following problem: how can the mobile collector adjust its movement strategy over G, or equivalently find the transition matrix $\mathbf{P}$, in a *distributed* manner so as to *minimize* the data loss over all sensor nodes?

III. Performance Metric and Its Analysis

We first define a performance metric – the network loss probability, denoted as P_L, and then analyze this performance metric for any given movement strategy of the random walk (mobile collector), or its transition matrix $\mathbf{P}$. Let $\{A_i(t)\}_{t\geq 0}$ be the data arrival process into the buffer of node i. Since a unit of data arrives at node i with probability p_i, which is assumed to be independent over i and t, the arrival process $\{A_i(t)\}$ is an independent Bernoulli process with $\mathbb{P}\{A_i(t)=1\} = p_i = 1 - \mathbb{P}\{A_i(t)=0\}$. Let $S_i(t) = \sum_{k=1}^{t} A_i(t)$ be the total number of data arrivals at node i till time t, and $L_i(t)$ be the total number of data losses at sensor node i until time t. Then, we have

$$P_L \triangleq \lim_{T\to\infty} \frac{\sum_{i=1}^{n} L_i(T)}{\sum_{i=1}^{n} S_i(T)}. \tag{2}$$

Fix the transition matrix $\mathbf{P}$ of an irreducible Markov chain (or random walk) $\{X(t)\}$. Let $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n)$ be its stationary distribution. Observe that

$$\lim_{T\to\infty} \frac{\sum_{i=1}^{n} S_i(T)}{T} = \sum_{i=1}^{n} p_i = n\bar{p}, \tag{3}$$

where $\bar{p} \triangleq \sum_{i=1}^{n} p_i/n$. We define the following *return time* of the random walk to node i:

$$R_i \triangleq \min\{t \geq 1 : X(t) = i \mid X(0) = i\}. \tag{4}$$

Then, it is well known that $\mathbb{E}\{R_i\} = 1/\pi_i$. Further, by strong Markov property [5], this return time R_i forms a *regenerative cycle* for the random walk (mobile collector) between two successive visits to node i. Recall that once the mobile collector visits node i, it gathers all the data currently in the buffer of size B.

Let Y_i and Z_i be the number of data arrivals into the buffer at node i and the number of lost data at node i during a regenerative cycle R_i, respectively. Then, we have

$$Y_i = \sum_{t=0}^{R_i-1} A_i(t) \overset{d}{=} \sum_{t=1}^{R_i} A_i(t), \tag{5}$$

$$Z_i = [Y_i - B]^+, \tag{6}$$

where $[x]^+ = \max\{x, 0\}$. Since R_i (movement strategy of data collector) and $A_i(t)$ for each i (data arrivals) are independent, we get

$$\mathbb{E}\{Y_i\} = \mathbb{E}\{R_i\}\mathbb{E}\{A_i(1)\} = p_i/\pi_i. \tag{7}$$

Then, we can interpret Z_i as a *reward* during a regenerative cycle R_i, and thus from the standard renewal theory on the (delayed) renewal reward process [18], we have

$$\lim_{T\to\infty} \frac{L_i(T)}{T} = \frac{\mathbb{E}\{Z_i\}}{\mathbb{E}\{R_i\}} = \pi_i \mathbb{E}\{[Y_i - B]^+\}, \tag{8}$$

which is *independent* of the initial position $X(0)$ for all i. Therefore, from (2), (3), and (8), the network loss probability P_L can be written as

$$P_L = \lim_{T\to\infty} \frac{\sum_{i=1}^{n} L_i(T)}{\sum_{i=1}^{n} S_i(T)} = \frac{\sum_{i=1}^{n} \pi_i \mathbb{E}\{[Y_i - B]^+\}}{n\bar{p}}. \tag{9}$$

First, since $[y - B]^+ \leq y$ for all $y \geq 0$, we have

$$P_L \leq \frac{\sum_{i=1}^{n} \pi_i \mathbb{E}\{Y_i\}}{n\bar{p}} = \frac{\sum_{i=1}^{n} p_i}{n\bar{p}} = 1, \tag{10}$$

where the equalities are from (7) and $\bar{p} = \sum_{i=1}^{n} p_i/n$. Similarly, since $[y - B]^+$ is convex in $y \geq 0$, from Jensen's inequality, we have

$$\mathbb{E}\{[Y_i - B]^+\} \geq [\mathbb{E}\{Y_i\} - B]^+ = [p_i/\pi_i - B]^+ .$$

This gives

$$P_L \geq \frac{\sum_{i=1}^{n} \pi_i \left[\frac{p_i}{\pi_i} - B\right]^+}{n\bar{p}} \geq \frac{[n\bar{p} - B]^+}{n\bar{p}} = \left[1 - \frac{B}{n\bar{p}}\right]^+, \tag{11}$$

where the second inequality in (11) is again from the convexity of $[y - B]^+$ in $y \geq 0$ and $\sum_{i=1}^{n} \pi_i = 1$.

From this lower bound, we note that $B/n\bar{p} = O(1)$ should be obeyed in order for the problem (i.e., finding $\mathbf{P}$ to minimize P_L) to be non-trivial. If $B/n\bar{p} = o(1)$ (i.e., $\bar{p}$ is orderwise larger than $O(1/n)$), then from (11), the lower bound of P_L gets close to one as n increases, *regardless of* any choice of transition matrix $\mathbf{P}$. In this case, the network is 'overloaded' and there are simply too many data arrivals to the network in that the buffer of size B and one mobile collector cannot handle such load. Even for a network of reasonable size, if $B/n\bar{p}$ is too small to satisfy $P_L \leq P_L^*$ for a given desired network loss probability P_L^*, we can consider a method of network partitioning, i.e., each mobile collector covers a partitioned sub-network area, or the use of multiple mobile collectors to drive the network loss probability below P_L^*. We will address this case later in Section V. From now on, for analytical tractability and to avoid triviality, we focus the case of $n\bar{p} = C$, where C is some positive constant.

IV. Distributed Optimal Movement Strategy

In this section, we formally define the optimization problem in order to find the movement strategy of the mobile collector (random walk) in minimizing the network loss probability P_L, and then explain how the mobile collector can adjust its movement strategy to achieve the optimal strategy in a distributed manner.

In our scenarios, sensors are low-cost, low-power devices, each of which is equipped with very small amount of storage space available for data. To properly capture such limited buffer space, we first consider the case of $B = 1$, where B is the buffer size at each sensor, and develop our analytical framework for a single mobile collector. We will later discuss more general settings with the case of $B > 1$ and multiple mobile collectors in Section V.

First, when $B = 1$, the term $\mathbb{E}\{[Y_i - B]^+\}$ in the RHS of (8) can be written as

$$\begin{aligned}\mathbb{E}\{[Y_i - 1]^+\} &= \sum_{k=1}^{\infty}(k-1)^+\mathbb{P}\{Y_i = k\} \\ &= \sum_{k=2}^{\infty} k\mathbb{P}\{Y_i = k\} - \sum_{k=2}^{\infty}\mathbb{P}\{Y_i = k\} \\ &= \mathbb{E}\{Y_i\} - 1 + \mathbb{P}\{Y_i = 0\}. \qquad (12)\end{aligned}$$

Also, from (5), we have

$$\begin{aligned}\mathbb{P}\{Y_i = 0\} &= \mathbb{E}\{\mathbb{P}\{Y_i = 0|R_i\}\} = \mathbb{E}\{\mathbb{P}\{\sum_{t=1}^{R_i} A_i(t) = 0 \mid R_i\}\} \\ &= \mathbb{E}\left\{(1-p_i)^{R_i}\right\}, \qquad (13)\end{aligned}$$

where the last equality is from $\mathbb{P}\{A_i(t) = 0\} = 1 - p_i$ and the independence of $A_i(t)$ and R_i. From (7), and (12)–(13), (9) becomes

$$P_L = \frac{n\bar{p} - 1 + \sum_{i=1}^n \pi_i \mathbb{E}\left\{(1-p_i)^{R_i}\right\}}{n\bar{p}}. \qquad (14)$$

Observe that P_L is governed by R_i and π_i that clearly depend on the movement strategy or the transition matrix $\mathbf{P}$. The return time R_i of any reversible Markov chain is known to have exponentially decaying tail, and when the mixing time* is not so large, it is known to be well approximated by an exponentially distributed random variable [1]. The exponential (or geometric) approximation for the return time or first passage time of a random walk has been adopted and shown to be well supported in many research problems such as data harvesting in WSNs [19], [13], detection of a mobile target in a mobile sensor network [7], peer-to-peer file search [31], [32], overlay topology construction [8], to name a few. We thus similarly assume that R_i is exponentially distributed with mean $\mathbb{E}\{R_i\} = 1/\pi_i$. Under this setting, we have $\mathbb{E}\{(1-p_i)^{R_i}\} = \mathbb{E}\{e^{-\theta_i R_i}\} = \pi_i/(\pi_i + \theta_i)$, where $\theta_i \triangleq -\log(1-p_i) > 0$. So, P_L in (14) can be written as

$$P_L = 1 - \frac{1}{n\bar{p}} + \frac{1}{n\bar{p}}\sum_{i=1}^n \frac{\pi_i^2}{\pi_i + \theta_i}. \qquad (15)$$

Now, P_L is solely dependent on π_i, and thus the problem of finding the optimal transition matrix $\mathbf{P}$ is equivalent to finding the transition matrix $\mathbf{P}$ achieving the optimal $\boldsymbol{\pi}^\star = (\pi_1^\star, \pi_2^\star, \ldots, \pi_n^\star)$ to minimize P_L in (15). Since $n\bar{p}$ is a global constant, optimizing P_L over $\boldsymbol{\pi}$ is equivalent to the following problem: for a function $f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$,

$$(\mathbf{P1}) \qquad \begin{aligned}&\textbf{minimize} \quad f(\boldsymbol{x}) = \sum_{i=1}^n \frac{x_i^2}{x_i + \theta_i} \\ &\textbf{subject to} \quad x_i > 0, \ \sum_i x_i = 1,\end{aligned}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$. Here, the optimal solution $\boldsymbol{x}^\star$ for the problem $(\mathbf{P1})$ is the optimal $\boldsymbol{\pi}^\star$ to minimize P_L in (15). We then have the following.

Theorem 1: The optimal $\pi_i^\star$ in minimizing P_L, or the optimal $x_i^\star$ for $(\mathbf{P1})$, is $\pi_i^\star = -\log(1-p_i)/K$ where $K = -\sum_{i=1}^n \log(1-p_i) > 0$ is a normalizing constant. □

Proof: The function $f(\boldsymbol{x})$ is clearly convex, as $f(\boldsymbol{x}) = \sum_{i=1}^n f_i(x_i)$ has a separable form and each term $f_i(x_i) = x_i^2/(x_i + \theta_i)$ is convex in x_i. The constraint set is also a convex set. Thus, the problem $(\mathbf{P1})$ is a well-defined convex optimization problem. Since $x_i > 0$, the lagrange multiplier for this constraint should be zero, so we set the Lagrangian for this problem as

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda(1 - \sum_i x_i),$$

and setting $\nabla L(\boldsymbol{x}, \lambda) = 0$ gives

$$\frac{x_i^2 + 2\theta_i x_i}{(x_i + \theta_i)^2} - \lambda = 0, \text{ for each } i. \qquad (16)$$

Rewriting (16) as $x_i^2 + 2\theta_i x_i = c\theta_i^2$ for some constant $c > 0$. Solving this gives $x_i = \theta_i(1 + \sqrt{1+c})$. Thus, it follows that the optimal solution x_i should be *proportional to* $\theta_i = -\log(1-p_i)$, and the normalizing constant K is from $\sum_{i=1}^n \pi_i^\star = 1$. ■

The next question is how to construct the corresponding transition matrix $\mathbf{P}$ for the random walk (the movement strategy of the mobile collector) in a distributed manner to achieve the optimal stationary distribution $\pi_i^\star = -\log(1-p_i)/K$. For this purpose, we employ the famous Metropolis-Hastings (MH) algorithm [15], [12], which is the most popular Markov Chain Monte Carlo (MCMC) method for sampling a given probability distribution $\boldsymbol{\pi}$ by constructing a Markov chain with $\boldsymbol{\pi}$ as its unique stationary distribution. Specifically, start with *any* arbitrary irreducible Markov chain with its transition matrix $\mathbf{Q} = \{q_{ij}\}_{i,j \in \mathcal{N}}$ defined on the same state space $\mathcal{N}$. From the transition matrix $\mathbf{Q}$, the MH algorithm then gives the transition matrix $\mathbf{P}$ so as to achieve the desired stationary distribution $\boldsymbol{\pi}^\star$. Following the most popular version of the MH algorithm used in the networking literature [28], [27], [31], [32], [24], we consider the transition matrix of the simple random walk as $\mathbf{Q}$ in which $q_{ij} = 1/d_i$ if $(i,j) \in \mathcal{E}$, and $q_{ij} = 0$, otherwise. Here, d_i denotes the degree of node i, i.e., $d_i = |\{j \in \mathcal{N} : (i,j) \in \mathcal{E}\}|$, the number of neighbors of node i. The transition matrix $\mathbf{P}$ by the MH algorithm is then given by

$$P_{ij} = \begin{cases} \min\left\{\frac{1}{d_i}, \frac{1}{d_j}\frac{\pi_j^\star}{\pi_i^\star}\right\} & \text{if } (i,j) \in \mathcal{E}, \\ 0 & \text{if } (i,j) \notin \mathcal{E}, i \neq j, \end{cases} \qquad (17)$$

with $P_{ii} = 1 - \sum_{j \neq i} P_{ij}$. One can easily check that the transition matrix $\mathbf{P}$ is reversible with respect to $\boldsymbol{\pi}^\star$ or satisfies the reversibility condition (detailed balance equation), i.e., $P_{ij}\pi_i^\star = P_{ji}\pi_j^\star$ for all i, j.

*Roughly speaking, the mixing time is the amount of time it takes for the Markov chain to converge to its stationary distribution starting from any initial distribution.

As can be seen in (17), the transition probability from node i to its neighbor j depends only on the local information of node i, i.e., d_i, d_j and $\pi_j^\star/\pi_i^\star$. Thanks to this preferable distributed nature of the MH algorithm, the mobile collector can easily adjust its movement strategy or the transition matrix $\mathbf{P}$ in a distributed fashion. In particular, the degree information can be easily obtained through a neighbor discovery process [25]. If the mobile collector is a physical mobile agent, the mobile agent can collect the degree information at its current position nearby a sensor node. Further, it is worth noting that the MH algorithm only requires the *ratio* of steady-state probabilities $\pi_j^\star/\pi_i^\star$ instead of the actual values of $\pi_j^\star$ and $\pi_i^\star$. Thus, we never have to compute the exact normalizing constant K.

Our solution based on the MH algorithm has also an interesting implication on the estimate $\hat{p}_i$ – the estimated probability with which a unit of data arrives to node i during one time slot. Note that the mobile collector takes one step per time slot. In reality, such information on the length of a time slot may not be available for each sensor, although each sensor can maintain its own clock. Observe that if p_i is small enough (a unit of data rarely arrives to node i as is often the case), $\pi_i^\star \propto -\log(1-p_i) \approx p_i$, and so $\pi_j^\star/\pi_i^\star \approx p_j/p_i$. Therefore, each sensor only needs to estimate the number of data arrivals during some pre-defined time duration T, without having to synchronize this clock with that of the mobile collector.

V. Extension To More General Setting

In this section, we turn our attention to more general settings in which the buffer size of each sensor is $B > 1$ and/or there are multiple mobile collectors ($M > 1$) in the network.

A. $B > 1$ with Single Mobile Collector

Fix $M = 1$. When $B > 1$, we no longer have the simple form of the loss probability P_L as in (14) and (15). We here instead derive an upper bound of P_L, which turns out to be in a similar form as (15), and then find the optimal $\boldsymbol{\pi}^\star$ to minimize the upper bound. To proceed, we need the following.

Definition 1: [20] For random variables U and V, we define a *convex* order, written as $U \leq_{cx} V$, if $\mathbb{E}\{\phi(U)\} \leq \mathbb{E}\{\phi(V)\}$ holds for any convex function ϕ for which the expectation exists. □

Clearly, if $U \leq_{cx} V$, then $\mathbb{E}\{U\} = \mathbb{E}\{V\}$ and $\mathrm{Var}\{U\} \leq \mathrm{Var}\{V\}$ by setting $\phi(\cdot) = (\cdot)^2$.

Theorem 2: [20] Let $U_1, U_2, \ldots$ and $V_1, V_2, \ldots$ each be a sequence of $i.i.d.$ random variables. Let N be an integer-valued non-negative random variable that is independent of $\{U_i\}$ and $\{V_i\}$. If $U_i \leq_{cx} V_i$ for all i, then $\sum_{i=1}^N U_i \leq_{cx} \sum_{i=1}^N V_i$. □

We then obtain the following upper bound of P_L for $B>1$.

Proposition 1: For $B > 1$, we have

$$P_L \leq B\left(\frac{1}{B} - \frac{1}{n\bar{p}} + \frac{1}{n\bar{p}}\sum_{i=1}^{n}\frac{\pi_i^2}{\pi_i + \tilde{\theta}_i}\right), \tag{18}$$

where $\tilde{\theta}_i \triangleq -\log(1-p_i/B)$. □

Proof: From (9), we can write

$$P_L = \frac{B\sum_{i=1}^n \pi_i \mathbb{E}\left\{\left[\frac{Y_i}{B} - 1\right]^+\right\}}{n\bar{p}}. \tag{19}$$

First, fix $i \in \mathcal{N}$. Let $I_i(1), I_i(2), \ldots$ be a sequence of $i.i.d.$ Bernoulli random variables with $\mathbb{P}\{I_i(t)=1\} = p_i/B = 1 - \mathbb{P}\{I_i(t)=0\}$. Observe that for any convex function ϕ, and for each t,

$$\begin{aligned}\mathbb{E}\left\{\phi\left(\frac{A_i(t)}{B}\right)\right\} &= \phi\left(\frac{1}{B}\right)p_i + \phi(0)(1-p_i)\\ &= \phi\left(0\cdot\left(1-\frac{1}{B}\right) + 1\cdot\frac{1}{B}\right)p_i + \phi(0)(1-p_i)\\ &\leq \phi(1)\frac{p_i}{B} + \phi(0)\left(1-\frac{p_i}{B}\right) = \mathbb{E}\{\phi(I_i(t))\},\end{aligned}$$

where the inequality is from Jensen's inequality. This means $A_i(t)/B \leq_{cx} I_i(t)$ for all t from Definition 1. Since R_i is independent of these Bernoulli random variables, from Theorem 2, we have

$$\frac{Y_i}{B} = \sum_{t=1}^{R_i}\frac{A_i(t)}{B} \leq_{cx} \sum_{t=1}^{R_i} I_i(t) \triangleq \tilde{Y}_i. \tag{20}$$

Since $[x-1]^+$ is a convex function of $x \geq 0$, from Definition 1 and (19)–(20), we have

$$P_L = \frac{B\sum_{i=1}^n \pi_i \mathbb{E}\left\{\left[\frac{Y_i}{B} - 1\right]^+\right\}}{n\bar{p}} \leq \frac{B\sum_{i=1}^n \pi_i \mathbb{E}\{[\tilde{Y}_i - 1]^+\}}{n\bar{p}}.$$

Since R_i is exponentially distributed with mean $1/\pi_i$, by repeating the aforementioned procedure to obtain (15), but now with $\tilde{Y}_i$, we obtain the upper bound of P_L in (18) for $B > 1$. ■

Similar to (**P1**), we can consider another optimization problem to find the optimal $\pi_i^\star$ in minimizing the upper bound of P_L in (18). Note that the upper bound of P_L in (18) has a similar form as (15), but with $\tilde{\theta}_i$. Thus, the optimal $\boldsymbol{\pi}^\star$ in minimizing the upper bound of P_L should be again proportional to $\tilde{\theta}_i = -\log(1-p_i/B)$ and following the same lines in the previous section using the MH algorithm, we can similarly construct an optimal, distributed movement strategy for the mobile collector.

B. Multiple Mobile Collectors $M > 1$

In this section, we first consider the case of $B=1$, while allowing the number of mobile collectors $M > 1$. We will then extend our analysis to $B > 1$ as well as $M > 1$.

Fix $B = 1$ and $M > 1$. From Section III, recall that for a single mobile collector, the time instants at which the mobile collector visits node i form a renewal sequence whose interarrival times are simply $i.i.d.$ copies of R_i. Similarly, when there are $M > 1$ mobile collectors, we can consider the time interval between successive visits by *any one* of M random walks (mobile collectors) to node i, denoted by $\tilde{R}_i$, as the interarrival time of M superposed renewal sequences (one sequence for each of M mobile collectors). Since the superposition of M $i.i.d.$ renewal processes, after stretching out the time axis by M times, converges to a Poisson process as M increases [1], [9], we naturally expect that $\tilde{R}_i$ behaves more likely an exponentially distributed random variable with its mean M times smaller than $\mathbb{E}\{R_i\}$, even when R_i (for a given mobile collector) is not exponentially distributed. This suggests that our distributed optimal solution to $(\mathbf{P1})$ will be very close to the true optimal strategy without such exponential R_i assumption when there are multiple mobile collectors in the network.

In this setup, we observe that the analytical framework in Sections III and IV holds for $M > 1$ expect that one cycle period is now $\tilde{R}_i$ instead of R_i, where $\tilde{R}_i$ is readily an exponential random variable with $\mathbb{E}\{\tilde{R}_i\} = 1/(M\pi_i)$ (even when R_i is not). Specifically, from (14)–(15), the network loss probability P_L now becomes, for $M > 1$,

$$P_L = \frac{n\bar{p} - 1 + \sum_{i=1}^{n} \pi_i \mathbb{E}\left\{(1-p_i)^{\tilde{R}_i}\right\}}{n\bar{p}} = 1 - \frac{1}{n\bar{p}} + \frac{1}{n\bar{p}} \sum_{i=1}^{n} \frac{\pi_i^2}{\pi_i + \eta_i}, \quad (21)$$

where $\eta_i \triangleq \theta_i/M = -\log(1-p_i)/M$. Note that (21) has the same form as (15). Therefore, from the optimization problem $(\mathbf{P1})$ and Theorem 1, we can show that the optimal $\pi_i^\star$ to minimize P_L in (21) is also proportional to η_i, i.e., $\pi_i^\star \propto \eta_i = -\log(1-p_i)/M$. Here, the constant M is the same for all i, and thus $\pi_i^\star \propto -\log(1-p_i)$. In other words, the optimal stationary distribution $\pi_i^\star$ remains the same, regardless of any choice of $M \geq 1$.

Finally, consider the case of $B > 1$ and $M > 1$. As done in Proposition 1, we can similarly derive the upper bound of P_L, which is given by

$$P_L \leq B\left(\frac{1}{B} - \frac{1}{n\bar{p}} + \frac{1}{n\bar{p}} \sum_{i=1}^{n} \frac{\pi_i^2}{\pi_i + \tilde{\eta}_i}\right), \quad (22)$$

where $\tilde{\eta}_i \triangleq -\log(1-p_i/B)/M$. After repeating the same argument as earlier, we obtain the optimal steady-state probability $\pi_i^\star \propto -\log(1-p_i/B)$ in minimizing the upper bound of P_L in (22). In summary, for any $B \geq 1$ and $M \geq 1$, the proposed optimal stationary distribution $\pi_i^\star$ should be proportional to $-\log(1-p_i/B)$, which then feeds the MH algorithm to obtain the transition probability for each mobile collector in a distributed manner.

VI. Numerical Simulations

In this section, we present simulation results on the network loss probabilities of our distributed optimal movement strategy under various setting of buffer sizes and the number of mobile agents as well as different data arrival patterns to the network. In particular, we here demonstrate its considerable performance improvement over the simple random walk strategy in which a random walk at the current node moves to any one of its neighbors uniformly at random at the next time slot. As mentioned earlier, the simple random walk strategy has been widely used as a randomized routing for query/collector or data packets as well as the movement strategy for physical mobile agents in the data gathering process [21], [2], [14], [10], [4], [6], [19].

For the performance evaluation, we implement a custom event-driven simulator using C language. The common simulation setups are the following. We consider random geometric graphs that have been widely used in the literature to model diverse wireless sensor or ad-hoc network topologies [2], [3], [4], [6], [28], [27]. The random geometric graph, denoted as $RGG(n,r)$, is a graph where n nodes are uniformly and independently located in the unit square, and two nodes are connected (i.e., they become a neighbor of each other) if they are within distance of r. We here consider $n = 100, 200$ nodes and $r = 0.3$. Fig 2(a) shows one sample topology of $RGG(100, 0.3)$. Throughout the simulations, the connectivity of each sample topology of $RGG(n,r)$ is ensured for our performance evaluations.

We consider geometrically correlated data arrival patterns. This is more realistic setting than purely random data arrivals, as geographically neighboring nodes are likely to observe/detect similar data/events from sensor field [26], [17], [11]. To this end, let $(x_i, y_i) \in [0,1]^2 \triangleq \Omega$ be the location of node $i \in \mathcal{N}$ over the unit square area. The data arrival rate at node i is now a function of its location (x_i, y_i), denoted as $p_i(x_i, y_i)$. To capture heterogeneous data arrivals to different sensor nodes, we consider the following three test-cases using different arrival patterns for $p_i(x_i, y_i)$:

Rectangle: $p_i(x_i, y_i) = \alpha_1$ for $(x_i, y_i) \in [0, 0.5]^2$, and $p_i(x_i, y_i) = \alpha_2$ for $(x_i, y_i) \in \Omega \setminus [0, 0.5]^2$.

Gaussian: Consider a 2-D Gaussian function $G_{a,b}(x,y) : \Omega \rightarrow [0,1]$ centered at $(a,b) \in \Omega$, defined as

$$G_{a,b}(x,y) = e^{-\left(\frac{(x-a)^2}{2\sigma_x^2} + \frac{(y-b)^2}{2\sigma_y^2}\right)}, \quad (23)$$

where we set $\sigma_x = \sigma_y = 0.15$ throughout our simulations. Then, we set

$$p_i(x_i, y_i) = \beta \cdot G_{x_c, y_c}(x_i, y_i), \quad (24)$$

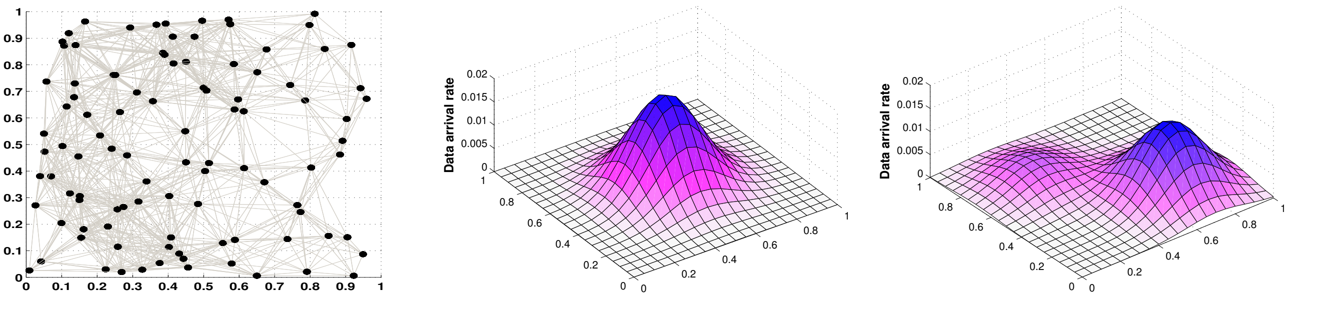
where $\beta \in (0,1)$ is the peak value that will be specified later. Here, x_c and y_c are uniformly and independently chosen over $[0,1]$. Fig. 2(b) depicts an example pattern of data arrival rates based on (24).

2-Gaussians: This is based on a superposition of two different 2-D Gaussian functions as follows:

$$p_i(x_i, y_i) = \gamma_1 \cdot G_{x_{c1}, y_{c1}}(x_i, y_i) + \gamma_2 \cdot G_{x_{c2}, y_{c2}}(x_i, y_i), \quad (25)$$

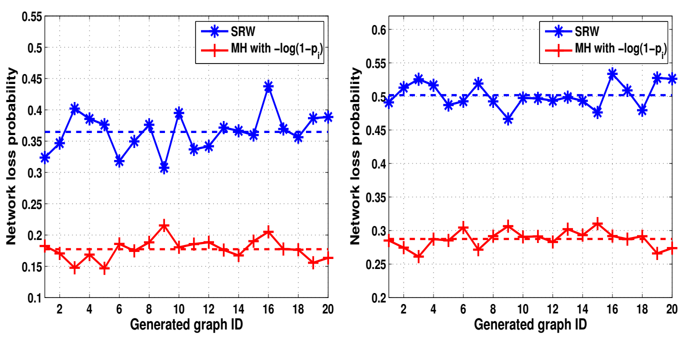
where γ_1 and γ_2 are two different peak values with $\gamma_1 + \gamma_2 \in (0,1)$. Similar to 'Gaussian', x_{c1}, x_{c2}, y_{c1}, and y_{c2} are uniformly and independently chosen over $[0,1]$ with $\sigma_x = \sigma_y = 0.15$. Fig. 2(c) shows an example pattern of data arrival rates following (25).

For each sample topology of $RGG(n,r)$ and a given arrival pattern with fixed B and M, we repeat 30 independent simulations. Each simulation runs for 10^5 time slots to measure the



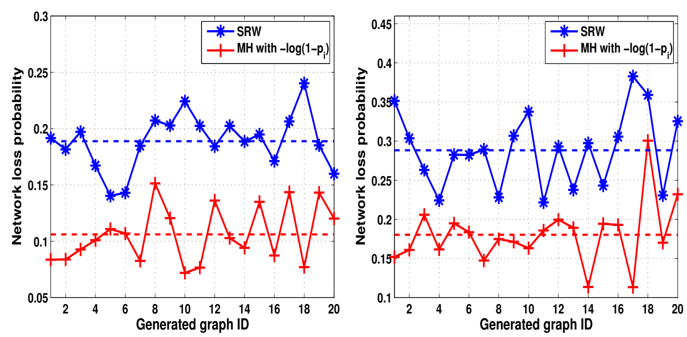
(a) (b) (c)

Fig. 2. (a) a sample topology of $RGG(100, 0.3)$; (b) an example pattern of 'Gaussian' data arrival rates in (24) with $(x_c, y_c) = (0.5, 0.5)$ and $\beta = 0.02$; (c) an example pattern of '2-Gaussians' data arrival rates in (25) with $(x_{c1}, y_{c1}) = (0.3, 0.7)$, $\gamma_1 = 0.015$, $(x_{c2}, y_{c2}) = (0.7, 0.3)$, and $\gamma_2 = 0.005$



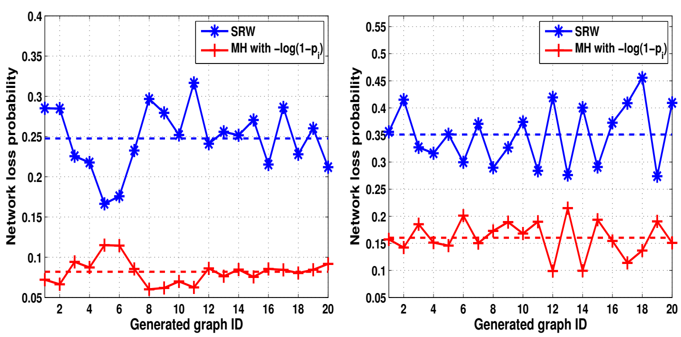
(a) $n = 100$ (b) $n = 200$

Fig. 3. The network loss probability P_L for 'Rectangle' data arrival pattern with $\alpha_1 = 0.005$ and $\alpha_2 = 0.0001$.



(a) $n = 100$ (b) $n = 200$

Fig. 5. The network loss probability P_L for '2-Gaussians' data arrival pattern in (25) with $\gamma_1 = 0.004$ and $\gamma_2 = 0.001$.



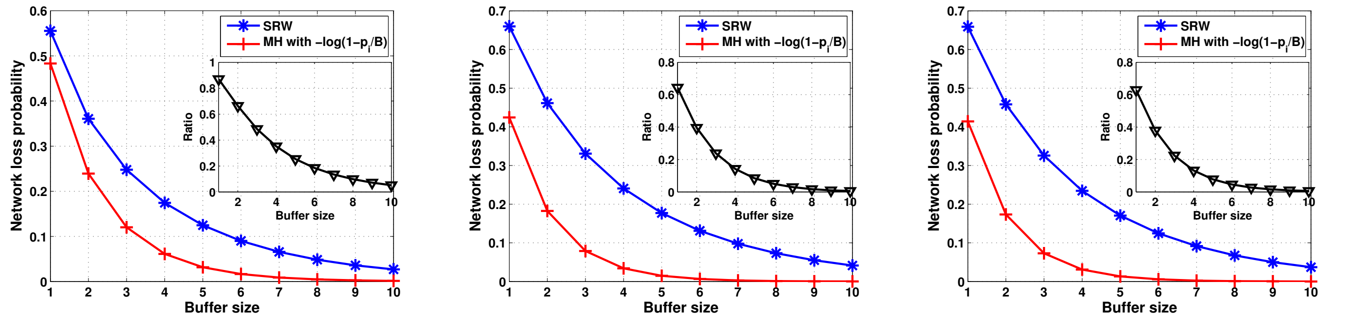
(a) $n = 100$ (b) $n = 200$

Fig. 4. The network loss probability P_L for 'Gaussian' data arrival pattern in (24) with $\beta = 0.005$.

network loss probability P_L – ratio of total number of lost data to the total number of generated data over all nodes during the simulation time. We then report the average value for each data point in every simulation figure. Also, in each figure, 'SRW' labeled plot stands for the network loss probability P_L when mobile collectors move in a simple random walk fashion, while 'MH with $-\log(1-p_i/B)$' labeled plot corresponds to the result of our distributed optimal movement strategy which employs the MH algorithm with the optimal stationary distribution $\pi_i^\star = -\log(1-p_i/B)$.

We first present the simulation results for the case of $B = 1$ and $M = 1$. Fig. 3 shows the network loss probabilities of our distributed optimal movement strategy and simple random walk for 20 different sample topologies of $RGG(n, 0.3)$ with $n = 100, 200$ and 'Rectangle' data arrival pattern with $\alpha_1 = 0.005$ and $\alpha_2 = 0.0001$. Here, dotted plots represent the statistical average of the network loss probabilities obtained over 20 different sample topologies. Figs. 4 and 5 show the results for 'Gaussian' and '2-Gaussians' arrival patterns, respectively, under $n = 100$ and $n = 200$ each. In all these simulations, we observe that the network loss probability under our distributed optimal (MH) strategy is about 2 times smaller than that of the simple random walk strategy. In addition, the amount of reduction does not change much for different network sizes ($n = 100, 200$).

We now turn our attention to the case of general buffer size $B > 1$ while we fix $n = 200$ and $M = 1$. To properly capture the effect of relatively large buffer size, we here increase the overall data arrival intensity in each arrival pattern. Fig. 6 shows the network loss probability results for the three arrival patterns while we vary B from 1 to 10 for a given sample topology of $RGG(200, 0.3)$. In each inset figure, we plot the ratio $P_L^{\mathsf{MH}}/P_L^{\mathsf{SRW}}$ as the buffer size B increases at each sensor node, where P_L^{SRW} and P_L^{MH} be the network loss probabilities of the simple random walk and our distributed optimal movement strategy based on the MH algorithm, respectively. In all cases, our movement strategy is consistently better than the simple random walk, and the ratio keeps decreasing, implying that our strategy is increasingly

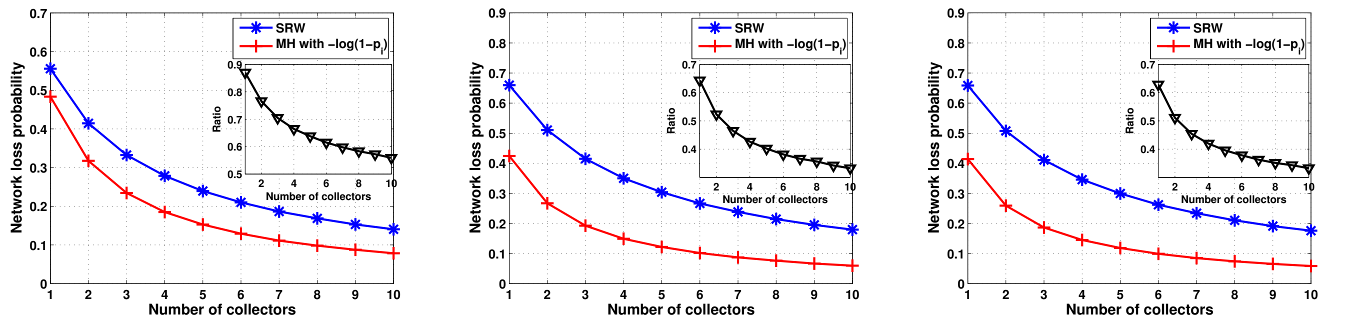


(a) ‘Rectangle’ data arrival pattern with $\alpha_1 = 0.01$ and $\alpha_2 = 0.001$

(b) ‘Gaussian’ data arrival pattern in (24) with $\beta = 0.02$

(c) ‘2-Gaussians’ data arrival pattern in (25) with $\gamma_1 = 0.015$ and $\gamma_2 = 0.005$

Fig. 6. The network loss probability P_L for $n = 200$ and $M = 1$, while varying the buffer size B for each node from 1 to 10; the inset figures show the ratio of the network loss probability of our distributed optimal movement strategy to that of the simple random walk, i.e., $P_L^{\mathsf{MH}}/P_L^{\mathsf{SRW}}$.



(a) ‘Rectangle’ data arrival pattern with $\alpha_1 = 0.01$ and $\alpha_2 = 0.001$

(b) ‘Gaussian’ data arrival pattern in (24) with $\beta = 0.02$

(c) ‘2-Gaussians’ data arrival pattern in (25) with $\gamma_1 = 0.015$ and $\gamma_2 = 0.005$

Fig. 7. The network loss probability P_L for $n = 200$ and $B = 1$, while varying the number of mobile collectors M from 1 to 10; the inset figures represents the ratio $P_L^{\mathsf{MH}}/P_L^{\mathsf{SRW}}$.

more advantageous as the buffer size increases. It is also worth noting that our strategy leads to *almost zero* network loss probability even under a small amount of buffer size, while the simple random walk strategy would require much larger buffer space to achieve such a very small loss probability. For instance, if the target loss probability one would like to achieve is no larger than 0.1 (i.e., $P_L \leq 0.1$ is the requirement), then $B = 3$ or 4 is enough under our strategy, while $B = 6$ or 7 is required for the simple random walk.

We next evaluate the effect of multiple collectors $M > 1$ on the network loss probability while we fix $n = 200$ and $B = 1$. We here again consider a relatively high data-arrival intensity, and use the same arrival patterns as above. Similar to the above case of larger buffer size, Fig. 7, obtained under a given sample topology of $RGG(200, 0.3)$, also shows that our movement strategy performs steadily better than the simple random walk. As before, the ratio $P_L^{\mathsf{MH}}/P_L^{\mathsf{SRW}}$ decreases as the number of mobile collectors M increases for all considered data arrival patterns. Fig. 7 also reveals how many mobile collectors (e.g., physical mobile agents) we can save in order to achieve a given target network loss probability by switching to our strategy from the simple random walk for each collector. For instance, for ‘Rectangle’ arrival pattern, if the requirement is $P_L \leq 0.2$, we would need about 7 collectors following simple random walks, while about 4 collectors would suffice

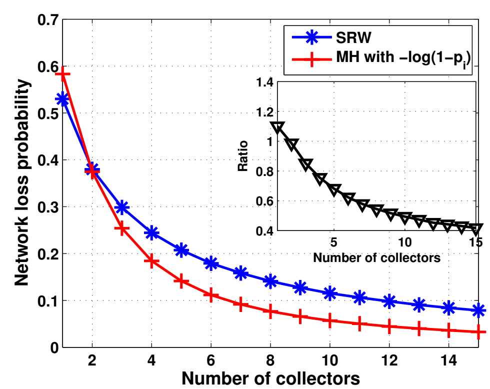


Fig. 8. The network loss probability P_L for $n = 100$ and $B = 1$, while varying M from 1 to 15. The data arrival pattern is ‘2-Gaussians’ data arrival pattern in (25) with $\gamma_1 = \gamma_2 = 0.02$.

if they follow our strategy. Similarly, for ‘Gaussian’ and ‘2-Gaussians’ arrival patterns, the saving in the number of mobile collectors is more striking. For $P_L \leq 0.2$ requirement in these cases, about 3 collectors are enough under our strategy, while one would need 8∼9 collectors under the simple random walk strategy, as shown in Fig. 7(b)–(c).

Fig. 8 shows the network loss probability under ‘2-Gaussians’ arrival pattern for $n = 100$ and $B = 1$, while varying the number of mobile collectors M. In this figure, when there is a single collector ($M = 1$), note that the performance of the simple random walk is slightly better than our strategy. This is largely due to the error associated with our approximation of R_i (return time) as a exponential random variable with mean $1/\pi_i$, in which case the amount of deviation of R_i from being exponential dominates, thus our strategy that optimizes the 'exponentialized' (first-order approximated) term may not be true optimal. Nonetheless, as mentioned before, when multiple collectors are deployed, such approximation becomes almost exact, i.e., the time interval between successive visits by any one of M random walks (mobile collectors) to node i is very close to an exponential random variable with mean $1/(M\pi_i)$. Accordingly, as Fig. 8 shows, when M gets larger, our movement strategy becomes true optimal and leads to superior performance over the simple random walk. For example, in order to achieve $P_L < 0.1$, our solution requires 6$\sim$7 mobile agents, while the simple random walk strategy necessitates 11$\sim$12 agents. In other words, our strategy gives about 50% cost savings for the deployment of multiple collectors. Note that for larger network with $n = 200$, such 'crossover' didn't take place (see Fig. 7(c) for instance), since the benefit of optimizing the first-order ('exponentialized') term significantly outweighes any negative effect from the higher-order terms caused by such approximation, which is also in agreement with the results in Figures 3–5.

VII. Conclusion

We have developed an analytical framework to evaluate the network loss probability as a performance measure for different Markovian movement strategies of mobile collectors moving over a graph (or network) for data harvesting in WSNs. Under this framework, we are able to find the optimal movement strategy for the mobile collectors under mild conditions so as to minimize the network loss probability. Our optimal strategy can be made distributed using only local information via the Metropolis-Hastings algorithm. We have demonstrated through extensive numerical simulations that such distributed optimal movement strategy remarkably outperforms the simple random walk strategy under various settings of buffer size and the number of mobile collectors, as well as heterogeneous and spatially-correlated data arrival patterns. We expect that our reasoning behind the distributed optimal movement strategy can be also applicable for the design of Markovian random walk-based applications in general networks beyond WSNs.

References

[1] D. Aldous and J. Fill, *Reversible Markov Chains and Random Walks on Graphs*. monograph in preparation.

[2] C. Avin and C. Brito, "Efficient and robust query processing in dynamic environments using random walk techniques," in *ACM/IEEE IPSN*, Berkeley, CA, April 2004.

[3] C. Avin and B. Krishnamachari, "The power of choice in random walks: an emprical study," in *ACM MSWiM*, Torremolinos, Spain, Oct. 2006.

[4] P. Basu and C.-K. Chau, "Opportunistic forwarding in wirelss networks with duty cycling," in *CHANTS*, San Francisco, CA, Sep. 2008.

[5] P. Brémaud, *Markov chains: gibbs fields, monte carlo simulation, and queues*. Springer-Verlag, 1999.

[6] C.-K. Chau and P. Basu, "Exact analysis of latency of stateless opportunistic forwarding," in *IEEE INFOCOM*, Rio de Janeiro, Brazil, April 2009.

[7] J.-C. Chin, Y. Dong, W.-K. Hon, C. Y. T. Ma, and D. K. Y. Yau, "Detection of intelligent mobile target in a mobile sensor network," *IEEE/ACM Trans. on Networking*, vol. 18, no. 1, pp. 41–52, 2010.

[8] B. F. Cooper, "An optimal overlay topology for routing peer-to-peer searches," in *ACM/IFIP/USENIX Middleware*, Grenoble, France, Nov. 2005.

[9] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes*. Springer-Verlag, 1988.

[10] G. Froc, I. Mabrouki, and X. Lagrange, "Design and performance of wireless data gathering networks based on unicast random walk routing," *IEEE/ACM Trans. on Networking*, vol. 17, no. 4, pp. 1214–1227, 2008.

[11] H. Gupta, V. Navda, S. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 1, pp. 1–31, 2008.

[12] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[13] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting mobility for energy efficient data collection in wireless sensor networks," *Mobile Networks and Applications*, vol. 11, no. 3, pp. 327–339, 2006.

[14] I. Mabrouki, X. Lagrange, and G. Froc, "Random walk based routing protocol for wireless sensor networks," in *InterPerf*, Nantes, France, Oct. 2007.

[15] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[16] R. Moazzez-Estanjini and I. C. Paschalidis, "Improved delay-minimized data harvesting with mobile elements in wireless sensor networks," in *WiOpt*, Princeton, NJ, May 2011.

[17] S. Pattem, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 4, pp. 1–31, 2008.

[18] S. M. Ross, *Stochastic processes*, 2nd ed. John Wiley & Son, 1996.

[19] R. C. Shah, S. R. adn S. Jain, and W. Brunette, "Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 215–233, 2003.

[20] M. Shaked and J. G. Shanthikumar, *Stochastic orders and their applications*. Academic Press, 1994.

[21] S. Shakkottai, "Asymptotics of query strategies over a sensor network," in *IEEE INFOCOM*, Hong Kong SAR, China, March 2004.

[22] Y. Shi and Y. Hou, "Theoretical results on base station movement problem for sensor network," in *IEEE INFOCOM*, Phoenix, AZ, Apr. 2008.

[23] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines," in *IEEE RTSS*, Lisbon, Portugal, Dec. 2004.

[24] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, and W. Willinger, "On unbiased sampling for unstructured peer-to-peer networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 377–390, 2009.

[25] S. Vasudevan, D. Towsley, D. Goeckel, and R. Khalili, "Neighbor discovery in wireless networks and the coupon collector's problem," in *ACM MobiCom*, Beijing, China, Sep. 2009.

[26] P. von Rickenbach and R. Wattenhofer, "Gathering correlated data in sensor networks," in *DIALM-POMC*, Philadelphia, PA, Sep. 2004.

[27] D. Vukobratović, Č. Stefanović, V. Crnojević, F. Chiti, and R. Fantacci, "Rateless packet approach for data gathering in wireless sensor networks," *IEEE JSAC*, vol. 28, no. 7, pp. 1169–1179, 2010.

[28] D. Vukobratović, M. S. Č. Stefanović, and V. Stanković, "Raptor packets: a packet-centric approach to distributed raptor code design," in *IEEE ISIT*, Seoul, South Korea, June 2009.

[29] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *ACM MobiHoc*, Roppongi, Japan, May 2004.

[30] ——, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *IEEE INFOCOM*, Miami, FL, Mar. 2005.

[31] M. Zhong and K. Shen, "Popularity-biased random walks for peer-to-peer search under the square-root principle," in *IPTPS*, Santa Barbara, CA, Feb. 2006.

[32] M. Zhong, K. Shen, and J. Seiferas, "The convergence-guaranteed random walk and its applications in peer-to-peer networks," *IEEE Trans. on Computers*, vol. 57, pp. 619–633, 2008.