Modeling Mobility-Assisted Data Collection in Wireless Sensor Networks

Hisham M. Almasaeid and Ahmed E. Kamal
Dept. of Electrical and Computer Eng., Iowa State University, Ames, IA 50011, USA
E-mail:{hisham,kamal}@iastate.edu

***Abstract*—Exploiting mobility to enhance the performance of wireless sensor networks (WSNs), in terms of connectivity, coverage, and lifetime elongation, has recently been the focus of several research studies. Mobility was adopted in two different ways; either using a network of mobile sensor nodes or deploying a few supplementary special mobile elements, usually referred to as *mobile agents* to help enhance connectivity and coverage. Different modes of operation (*roles*) were assigned to mobile agents including being a *data relay*, *data collector*, and *data sink*. In this paper we use a *closed queueing network* to model mobility and then evaluate data latency under all those roles. The proposed model provides powerful means to understand the effect of different parameters, like velocity and number of mobile agents as well as their movement strategy, on data latency.**

I. Introduction

A wireless sensor network might become disconnected (or even partitioned) into several islands for a variety of reasons. Node failures, random deployments, drift of nodes, channel quality degradations, and non-uniform energy consumption within the network are some of these reasons. To cope with coverage and connectivity gaps within the network, researchers followed two main approaches. First, deploying more static sensors either preventively to achieve certain level of fault-tolerance [1], or remedially to repair the network topology [2]. Second, exploiting supplementary mobile agents to increase the uniformity of energy consumption in the network and/or bridge connectivity gaps. Different roles have been assigned to those agents including being a *relay*, *collector*, or *sink*.

The concept of mobile collectors was introduced in [3], and were referred to as *Data MULEs*, to achieve connectivity in sparse WSNs. MULEs move randomly, collect data from nearby sensor nodes, and ultimately unload the carried data as they get close enough to a base station. On the other hand, by reducing the amount of traffic that static sensors have to relay, sink mobility was proposed by many researchers to remedy the *hot spot* problem and to prolong network lifetime [4].

The third role that a mobile agent can play is *data relay*. Mobile relays have been studied in three different modes; *message ferrying* [5], *fragment bridging* [6], and *supportive relaying* [7]. Both fragment bridging and message ferrying are used to support networks with different levels of disconnection. Message ferries forward packets between individual nodes in sparse mobile ad hoc networks, i.e., disconnection at the node level. However, fragment bridging aims at achieving connectivity between a number of disconnected fragments, where a fragment is a group of connected nodes, i.e., disconnection at the fragment level. In *supportive relaying*, the main objective of a mobile agent is to cooperate with original nodes in relaying traffic to save them power and prolong their lifetime without assuming any network disconnection.

The purpose of this paper is to present a modeling approach that can be used to evaluate the system performance, in terms of end-to-end delay, under all different roles of mobile agents. In our previous work, [6], we modeled the problem of fragment bridging using mobile relays as a *closed queueing network* to accurately evaluate the end-to-end delay and understand the effect of different system parameters. A similar modeling approach is used study the use of the two other roles: *mobile sinks* and *mobile collectors*.

The rest of this paper is organized as follows. In Section II, we describe the system model and how to build a closed queueing network to model the mobility of mobile agents in a clustered wireless sensor network. In Section III, we present the mathematical formulation and derive the distribution of the *loading time*, which is the time before the data is loaded from a cluster into a mobile agent, the trip time of a mobile collector from a certain cluster to the base station, and the end-to-end delay under both the use of mobile sinks and mobile collectors. Analytical and simulation results are presented in Section IV. We conclude our work in Section V.

II. System Model

We consider a large-scale wireless sensor network that consists of two types of nodes: static sensors and mobile agents. A mobile agent (MA) takes one of two different roles. First, a *mobile data collector (MC)* which collects information, carries it, and ultimately unloads it at a central base station. Second, a *mobile data sink (MS)* which is a final destination of data. In this case, a mobile agent could be a gateway to some *command nodes* or it could be a command node by itself [8].

As certain spots in the network might be inaccessible, we propose that a small number of easily accessible locations are chosen as potential *stop-stations*, or *stations* for short, for a mobile agent. In the case of an MS, data will be loaded at stop-stations therefore we call them *loading points*. On the other hand, for an MC, a stop-station could be the point where data is loaded or unloaded. Therefore, we call the locations of static base stations *unloading points*. The term *stop-station* will be used to refer to both loading and unloading points. The

This research was supported in part by the National Science Foundation under grants CNS-0626822 and ECS-0601570.

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

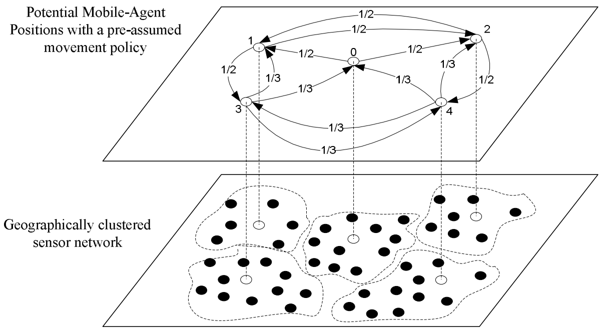

Fig. 1. A WSN clustered around predetermined stop-stations.

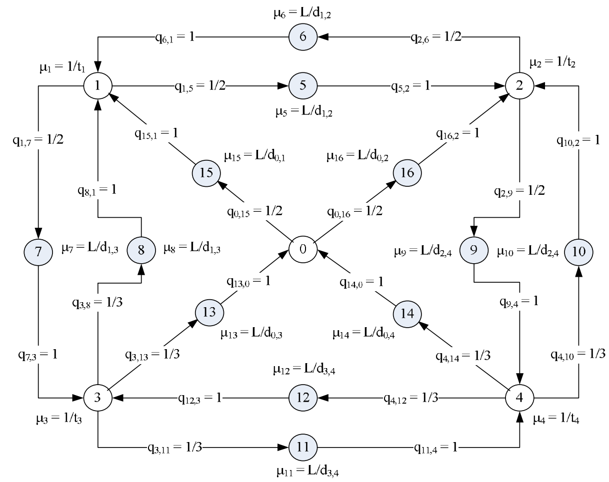

Fig. 2. The closed queueing network model of the WSN shown in Figure 1

network is clustered around stop-stations using any clustering technique [9] as shown in Figure 1. When a mobile agent stops at a loading point, data will be transferred from all over the intended cluster in multihops to the agent.

A. Assumptions and Definitions

Before we proceed with the modeling approach, we introduce some important assumptions:

- End-to-End delay is dominated by the movement time of mobile agents, therefore we neglect the communication delay between static sensors and mobile agents. For MCs, the end-to-end delay is composed of two parts:
  (1) *Loading Time*: is the time period elapsed before an MC arrives at an unoccupied station starting from the moment at which the station became unoccupied.
  (2) *Unloading Time*: the time it takes an MC to deliver the data collected at a certain loading point.

  However, for the case of mobile sinks, the end-to-end delay is only the *loading time*.
- No communication takes place between mobile agents.
- For simplicity, we assume that only one unloading point exists. Station (0) is always the unloading point in case of MCs, and is treated as a loading point in case of MSs.
- MAs are homogeneous in terms of communication and mobility capabilities and role of operation.
- The time t_i that an MA has to spend at station i is supposed to be known. We start with a sufficiently large value of t_i, then Algorithm-1 that we proposed in [6] can be used to reduce t_i to its minimum acceptable value.

B. Modeling Approach

We assume a discrete movement scheme in which MAs stop for certain time with a mean value of t_i at each *station* i to either load data from the intended cluster, or unload data at a static base station. An MA upon leaving certain *station* i chooses the next *station* to be j according to a preassumed distribution q^*_{ij}. We refer to q^*_{ij} as the *movement policy*.

Each station i is modeled as an *infinite-buffer/infinite-server* queue with a state-independent exponential service time of rate $\mu_i = \frac{1}{t_i}$. We refer to queues used to model *stations* as *service queues*. To capture the movement time between stations, we add another *infinite-buffer/infinite-server* queue, which we call *movement queue*, k, between every pair of service queues i and j that have ($q^*_{ij} > 0$). This queue has a state-independent exponential service time of rate $\mu_k = \frac{L}{d_{ij}}$, where d_{ij} is the Euclidean distance between stations i and j, and L is the speed of MAs. If q_{uv} is the probability of moving from queue u to queue v, then $q_{ik} = q^*_{ij}$ and $q_{kj} = 1$. Figure 2 shows a closed queueing network model for the network in Figure 1. In [6], we used a closed queueing network to model the problem of fragment bridging using mobile relays. This model is extended in this paper to include the roles of being a *sink* or a *collector* node. Therefore, we refer the reader to the work in [6] for a concrete understanding of the basic foundations of the model. Note that this modeling approach depends only on the number of *stations* and *MAs* which makes it highly scalable with respect to the network size.

III. Mathematical Model

We start this section by introducing some notations and definitions. Then, we mathematically derive the distribution of the *loading time*, *unloading time*, and the *end-to-end* delay.

A. Notations

- K is the number of MAs (MCs or MSs).
- M^* is the number of *stations* in the network field.
- M is the number of queues in the closed queueing network (this includes both service and movement queues).
- All MA's move at the same speed of $L\ m/s$.
- d_{uv} is the Euclidean distance between *stations* u and v, and it is assumed that $d_{uv}{=}d_{vu}$.
- t_u is the amount of time that an MA has to spend at station u. In practice, the time an MA spends at station u is fixed and is equal to t_u. However, we assume that this time is exponentially distributed with a mean of t_u in order to make our queueing network model tractable.
- $\overrightarrow{N}{=}\{n_1(\overrightarrow{N}), \ldots, n_M(\overrightarrow{N})\}$ is the state of the queueing system in which the K customers are distributed over the M queues such that queue i has $n_i(\overrightarrow{N})$ customers in this state. Note that $\sum_{i=1}^{M} n_i(\overrightarrow{N}){=}K$ for any state $\overrightarrow{N}$ and the total number of states is $\binom{M+K-1}{K}$.
- μ_j is the state-independent service rate of queue j.

- q^*_{uv} is the probability that an MA leaving *station* u goes to station v (this probability is assumed to be given).
- q_{uv} is the probability that an MA leaving *queue* u goes to queue v, where q_{ii}=0.
- $\pi(\overrightarrow{N})$ is the steady-state probability of state $\overrightarrow{N}$ obtained using the *Convolution Algorithm* [10].
- n_i is used to refer to the number of customers at *queue* i regardless of the system state.
- E_i is the number of system states in which n_i=0.
- $p^i_{idle}(t, \overrightarrow{N})$ is the joint probability that *queue* i became empty (i.e., $n_i = 0$) at time t=0 and remained so in $[0, t]$ ending up in state $\overrightarrow{N}$ at time t. In other words, this is the probability that *queue* i is idle for a time greater than t.
- $\overrightarrow{p}^i_{idle}(t)=[p^i_{idle}(t, \overrightarrow{N}_1),\ \ldots,\ p^i_{idle}(t, \overrightarrow{N}_{E_i})]^T$.
- U_i is a row vector of ones such that $|U_i|=E_i$.
- $\overrightarrow{1}_i$ is a row vector of length M in which the i^{th} element is 1, and all other elements are set to zero.
- X_i is a random variable that represents the *Loading time* of queue i, that is the time period during which queue i has no customers, i.e., MAs.
- Y_i is a random variable that represents the *Unloading time* for queue i, that is the time it takes a customer (an MC) that has just left queue i to reach queue (0).
- $F_{X_i}(t)=\Pr(X_i \le t)$ and $F_{Y_i}(t)=\Pr(Y_i \le t)$ are the cumulative distribution functions of X_i and Y_i respectively.
- $f_{X_i}(t)=\frac{d}{dt}F_{X_i}(t)$ and $f_{Y_i}(t)=\frac{d}{dt}F_{Y_i}(t)$ are the probability density functions of X_i and Y_i respectively.
- $\psi^s_i(t)$ and $\psi^c_i(t)$ are the cumulative distribution functions of the end-to-end delay experienced by data loaded from loading point i by MSs and MCs respectively.

B. Loading Time Distribution

The first step towards evaluating the end-to-end delay is to evaluate the *loading time* distribution. Note that the *loading time* represents the end-to-end delay in the case of MSs, but just part of it in the case of MCs.

For queue i, let $\mathbf{A}_i = [a^i_{xy}]$ be an $E_i \times E_i$ matrix such that:

$$a^i_{xy} = \begin{cases} -\sum_{j=1, j\neq i}^{M} n_j(\overrightarrow{N}_x)\mu_j & \text{if } x=y \\ n_j(\overrightarrow{N}_y)\mu_j q_{jk} & \text{if } \overrightarrow{N}_y=\overrightarrow{N}_x+\overrightarrow{1}_j-\overrightarrow{1}_k \\ 0 & otherwise \end{cases} \tag{1}$$

and let $\overrightarrow{p}^i_{idle}(0)$ be, as defined earlier, a column vector with each element $p^i_{idle}(0, \overrightarrow{N}_j)$ given as,

$$p^i_{idle}(0, \overrightarrow{N}_j) = \begin{cases} \displaystyle\sum_{\substack{k=1, k\neq i \\ n_k(\overrightarrow{N}_j)>0}} \frac{\mu_i q_{ik}\pi\left(\overrightarrow{N}_j + \overrightarrow{1}_i - \overrightarrow{1}_k\right)}{\sum_{l=1}^{M}\mu_l n_l\left(\overrightarrow{N}_j + \overrightarrow{1}_i - \overrightarrow{1}_k\right)} & \text{if } n_i(\overrightarrow{N}_j) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Then *loading time* distribution $F_{X_i}(t)$ is the same as that derived for the idle time in [6], and is given as,

$$F_{X_i}(t) = 1 - U_i e^{A_i t}\overrightarrow{p}^i_{idle}(0). \tag{3}$$

Note that $p^i_{idle}(0, \overrightarrow{N}_j)$ is the probability that queue i became empty at reference time $t = 0$ in state $\overrightarrow{N}_j$. Therefore, the term

$$\sum_{k=1, k\neq i, n_k(\overrightarrow{N}_j)>0} \frac{\mu_i q_{ik}\pi\left(\overrightarrow{N}_j + \overrightarrow{1}_i - \overrightarrow{1}_k\right)}{\sum_{l=1}^{M}\mu_l n_l\left(\overrightarrow{N}_j + \overrightarrow{1}_i - \overrightarrow{1}_k\right)}$$

sums up all the probabilities of moving from state $\overrightarrow{N} = \overrightarrow{N}_j + \overrightarrow{1}_i - \overrightarrow{1}_k$ in which $n_i(\overrightarrow{N}) > 0$ to state $\overrightarrow{N}_j$ in which $n_i(\overrightarrow{N}_j) = 0$, and of course $p^i_{idle}(0, \overrightarrow{N}_j) = 0$ if $n_i(\overrightarrow{N}_j) \neq 0$.

C. Unloading Time Distribution

For a loading queue i, the *unloading time* distribution is the distribution of the time it takes a customer to reach an unloading queue after leaving queue i, i.e., the time it takes an MC to deliver the data it collected from loading station i. As we mentioned earlier, we assume that only one unloading stations exists, and that is station (0).

To evaluate this distribution, let us first assume that the queueing network has only one customer, i.e., $K = 1$. Then, the *unloading time* of queue (i) is the time it takes this customer to reach queue (0) after leaving queue (i). But this is equivalent to *the time that passes before queue (0) receives its first customer starting from the moment at which the customer left queue (i)*. Therefore, the distribution of this time is the same as the *loading time* distribution of queue (0) taking into account that the beginning of the time period is when the customer had just left queue (i). To sum up, the *unloading time* distribution of queue (i) is the same as the distribution of the *loading time* of queue (0) given the following two conditions:

(1) The queueing network has only one customer, i.e. K=1.
(2) The system state $\overrightarrow{N}^*$ at the reference time $t^* = 0$ is $\overrightarrow{1}_i$.

Let $g^{(i,0)}_{idle}(0, \overrightarrow{N}_j)$ be the joint probability that *queue* (i) became empty at the reference time $t^* = 0$ and *queue* (0) remained empty in $[0, t]$ ending up in state $\overrightarrow{N}_j$ at time t. In a queueing network with one customer, this is the probability that this customer will reach *queue* (0) after leaving *queue* i in a time that is greater than t. Similar to $p^i_{idle}(0, \overrightarrow{N}_j)$ but with the two conditions mentioned earlier (K=1, and $\overrightarrow{N}^*$=$\overrightarrow{1}_i$) taken into account, $g^{(i,0)}_{idle}(0, \overrightarrow{N}_j)$ is given by,

$$g^{(i,0)}_{idle}(0, \overrightarrow{N}_j) = \begin{cases} \displaystyle\sum_{\substack{k=1,\ k\neq i \\ n_k(\overrightarrow{N}_j)>0}} q_{ik}\pi\left(\overrightarrow{N}_j + \overrightarrow{1}_i - \overrightarrow{1}_k\right) & \text{if } \substack{n_i(\overrightarrow{N}_j)=0, \\ n_0(\overrightarrow{N}_j)=0} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Let $\overrightarrow{g}^{(i,0)}_{idle}(0)$ be a column vector over $g^{(i,0)}_{idle}(0, \overrightarrow{N}_j)$ for all N_j's in which $n_i = 0$ and $n_0 = 0$, i.e.,

$$\overrightarrow{g}^{(i,0)}_{idle}(0)=[g^{(i,0)}_{idle}(0, \overrightarrow{N}_1),\ \ldots,\ g^{(i,0)}_{idle}(0, \overrightarrow{N}_{E_0})]^T, \tag{5}$$

then, the *unloading time* distribution is given by,

$$F_{Y_i}(t) = 1 - U_0 e^{A_0 t}\overrightarrow{g}^{(i,0)}_{idle}(0). \tag{6}$$

D. End-to-End Delay Distribution

After deriving the distributions of both *loading time* and *unloading time*, we can evaluate the end-to-end delay distribution for both modes of operation; *Mobile sinks* and *Mobile* *collectors*. As mentioned earlier, the end-to-end delay using *mobile sinks* is the same as the *loading time*, thus:

$$\psi_i^s(t) = F_{X_i}(t). \tag{7}$$

Then, the average end-to-end delay of the data collected at loading point i by mobile sinks is given by,

$$\overline{\psi_i^s} = E[X_i] = \int_{t=0}^{\infty} (1 - F_{X_i}(t))\, dt \tag{8}$$

On the other hand, the end-to-end delay using mobile collectors is the sum of X_i (the *loading time*), Y_i (the *unloading time*), and t_i which represents the period of time between the arrival and departure of an MC at station i. As the two random variables are independent and t_i is fixed, the distribution of the end-to-end delay is given as,

$$\psi_i^c(t) = F_{X_i}(t - t_i) \circledast F_{Y_i}(t - t_i). \tag{9}$$

Similar to (8), the average end-to-end delay of the data collected at loading point i by mobile collectors until it is unloaded at station (0) is given by,

$$\overline{\psi_i^c} \;=\; \int_{t=0}^{\infty} (1 - \psi_i^c(t))\, dt \;=\; E[X_i] + E[Y_i] + t_i. \tag{10}$$

Both equations (8) and (10) can be evaluated numerically as summations over a sufficiently small interval Δ. Therefore,

$$\overline{\psi_i^s} = \sum_{k=0}^{\infty} (1 - F_{X_i}(k\Delta)) \cdot \Delta, \tag{11}$$

and,

$$\overline{\psi_i^c} = \sum_{k=0}^{\infty} (1 - F_{X_i}(k\Delta - t_i) \circledast F_{Y_i}(k\Delta - t_i)) \cdot \Delta \tag{12}$$

IV. Results and Discussion

We evaluate the following performance metrics for the case-study network in Figure 1 through analysis and simulation and under both roles *Collectors (C)* and *Sinks (S)*:

- The accuracy of the closed queueing network model.
- The effect of the speed of MA's on the end-to-end delay, using two values $1.2m/s$ and $3.89m/s$.
- The effect of the movement policy, i.e., q^*_{uv}, on the end-to-end delay. We propose two policies; *Policy (A)*, *completely probabilistic*, shown in Figure 1, and *Policy (B)*, *partially deterministic*, which is defined as follows: $q^*_{0,1} = q^*_{0,2} = 0.5,\ q^*_{1,3} = q^*_{3,0} = 1,\ q^*_{2,4} = q^*_{4,0} = 1.$
- The effect of the number of MAs on the end-to-end delay, and for this case we use either 1 or 2 MAs.

All possible combinations of the suggested values mentioned above are summarized in the scenarios shown in Table I.

a) Model Accuracy: Let $\mathcal{L}$ be the set of all loading queues, and let $\overline{\psi_i^c}^*$ and $\overline{\psi_i^s}^*$ be the average end-to-end delay of the data loaded from i using MCs and MSs respectively obtained by simulation. Then, the minimum, average, and maximum error of a certain policy using MCs is defined as,

$$\text{Minimum error} = \min_{i \in \mathcal{L}} \frac{|\overline{\psi_i^c}^* - \overline{\psi_i^c}|}{\overline{\psi_i^c}^*} \tag{13}$$

$$\text{Average error} = \frac{1}{|\mathcal{L}|} \left. \sum_{i \in \mathcal{L}} \frac{|\overline{\psi_i^c}^* - \overline{\psi_i^c}|}{\overline{\psi_i^c}^*} \right) \tag{14}$$

TABLE I

The scenarios studied in this paper based on different number, speed, role, and movement policy of mobile agents.

Scenario	M^*	K	L	*(Policy, Role)*
1	4	1	$1.2\ m/s$	(A,S)
2	4	1	$1.2\ m/s$	(B,S)
3	4	1	$3.89\ m/s$	(A,S)
4	4	1	$3.89\ m/s$	(B,S)
5	4	2	$1.2\ m/s$	(A,S)
6	4	2	$1.2\ m/s$	(B,S)
7	4	2	$3.89\ m/s$	(A,S)
8	4	2	$3.89\ m/s$	(B,S)
9	4	1	$1.2\ m/s$	(A,C)
10	4	1	$1.2\ m/s$	(B,C)
11	4	1	$3.89\ m/s$	(A,C)
12	4	1	$3.89\ m/s$	(B,C)
13	4	2	$1.2\ m/s$	(A,C)
14	4	2	$1.2\ m/s$	(B,C)
15	4	2	$3.89\ m/s$	(A,C)
16	4	2	$3.89\ m/s$	(B,C)

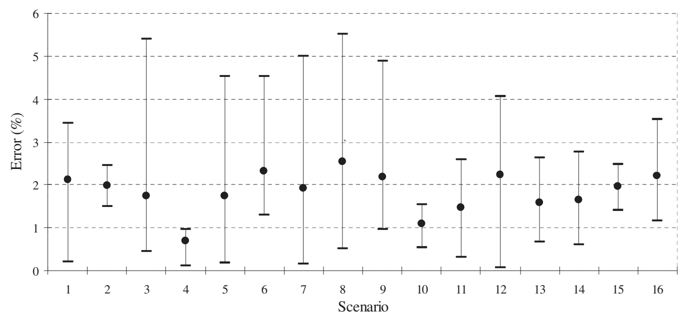

Fig. 3. The accuracy of the model in evaluating the end-to-end delay.

$$\text{Maximum error} = \max_{i \in \mathcal{L}} \frac{|\overline{\psi_i^c}^* - \overline{\psi_i^c}|}{\overline{\psi_i^c}^*} \tag{15}$$

Similarly, the error using MSs can be obtained using equations (13), (14), and (15) by replacing $\overline{\psi_i^c}^*$ and $\overline{\psi_i^c}$ with $\overline{\psi_i^s}^*$ and $\overline{\psi_i^s}$, respectively. Figure 3 shows the *maximum*, *minimum*, and *average* error for all the policies in Table I. The model achieves a high level of accuracy with an error that is always less than 6%, and an average error that does not exceed 3%.

b) Influence of the speed of MAs: Intuitively, increasing the speed of MA's should result in an improvement in the end-to-end delay. We plot the gain, in terms of end-to-end delay reduction, achieved by increasing the speed of MSs, Figure 4, and MCs, Figure 5, from $1.2m/s$ to $3.89m/s$. As it is evident from the examples, a high gain of up to around 58% was achieved by increasing the speed. The interesting thing, however, is the difference of the effect under the two roles of mobile agents. In Figure 4, approximately the same gain was achieved under the same policy using different numbers of MSs. On the other hand, the effect of speed differs depending on the number of MCs for the same policy, about 1.5% for policy B, as Figure 5 shows. The explanation for this is that the end-to-end delay in the case of MSs is just the *loading time* which is affected by roughly the same ratio, at the same loading point, when the speed is increased regardless of the number of MSs. However, the end-to-end delay in the case of MCs consists of the *loading time* and the *unloading time*, and the latter is independent of the number of MCs and is different for different loading points.

c) Influence of the number of MAs: As Figures 6 and 7 show, the gain achieved by increasing the number of mobile

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

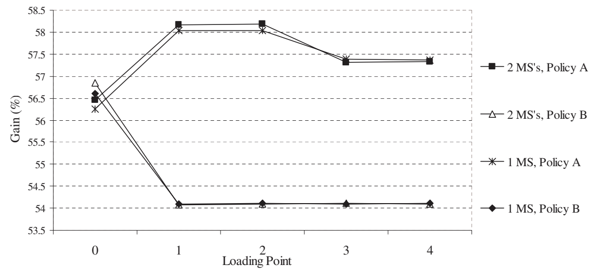

Fig. 4. The gain of increasing the speed of MSs to $3.89m/s$.

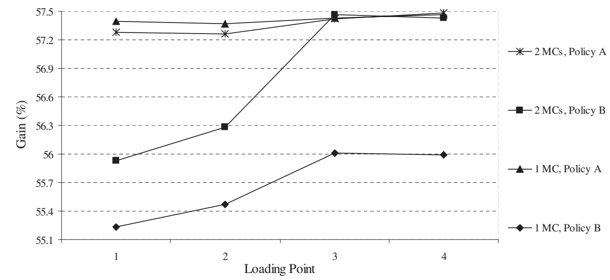

Fig. 5. The gain of increasing the speed of MCs to $3.89m/s$.

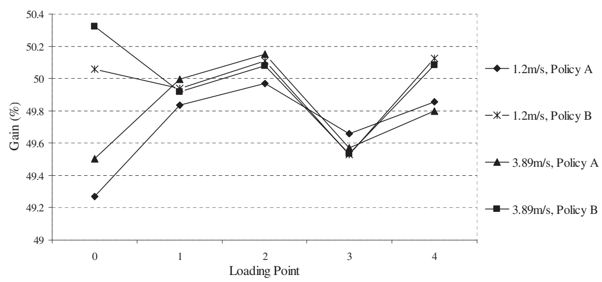

Fig. 6. The gain of increasing the number of MSs from 1 to 2.

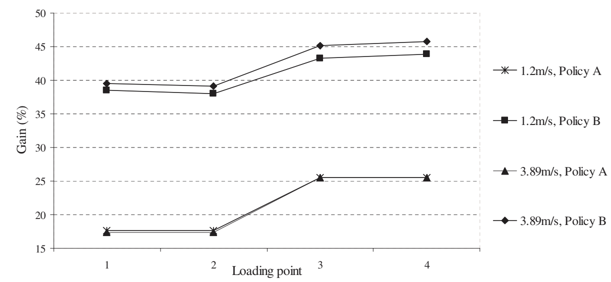

Fig. 7. The gain of increasing the number of MCs from 1 to 2.

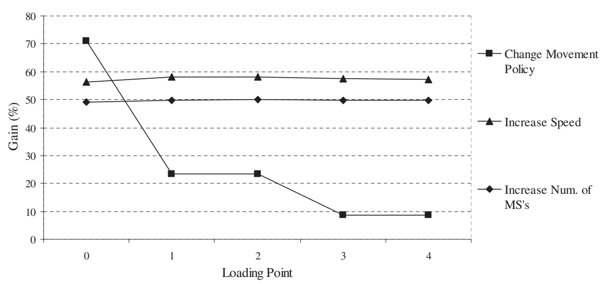

Fig. 8. The gain achieved in the network $\{(K, L, policy, role) = (1, 1.2m/s, A, S)\}$ by: *changing the policy to* B, *increasing the speed to* $3.89m/s$, or *increasing* K *by* 1.

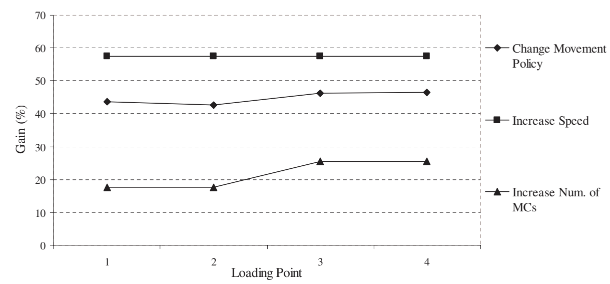

Fig. 9. The gain achieved in the network $\{(K, L, policy, role) = (1, 1.2m/s, A, C)\}$ by: *changing the policy to* B, *increasing the speed to* $3.89m/s$, or *increasing* K *by* 1.

agents from 1 to 2 is the same for the same policy regardless of the speed of those agents. However, the gain for the same policy is different depending on the role of mobile agents. The gain is higher when using mobile sinks than that achieved when using mobile collectors, because the unloading time in the case of MCs does not depend on the number of agents.

d) Influence of the movement policy: So far, we saw that a significant reduction of the end-to-end delay can be achieved by increasing the number of mobile agents or by using faster ones. But, this is the last thing a system designer should do as it entails extra cost. In fact, it turns out that having a better movement policy might achieve a higher gain than just increasing the number of mobile agents, as Figures 8 and 9 show. Using policy B instead of policy A for the same number and speed of MAs results in an average gain of 44.8% for MCs and 16.0% for MSs as shown in Figures 9 and 8 respectively, however, it could not override the gain of increasing the speed from $1.2m/s$ to $3.89m/s$.

V. Conclusions

In this paper, we modeled the movement of a limited number of mobile agents in a clustered WSN as a closed queueing network. The cumulative distribution functions of the *loading time*, the *unloading time*, and the *end-to-end* delay were derived for both *mobile sinks* and *mobile collectors*. The model can also be used to understand the effect of different system parameters on the end-to-end delay. As the results imply, changing the movement policy or increasing the speed of MAs might be more gainful than just adding more of them.

References

[1] J. Bredin et al. Deploying sensor networks with guaranteed capacity and fault tolerance. *MobiHoc*, May 2005.

[2] Yongguo Mei et al. Sensor replacement using mobile robots. *Computer Communications*, 30(13):2615–2626, 2007.

[3] R. Shah et al. Data mules: Modeling a three-tier architecture for sparse sensor networks. *SNPA*, May 2003.

[4] J. Luo and J. Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. *INFOCOM*, 2005.

[5] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. *MobiHoc*, 2004.

[6] H. Almasaeid and A. Kamal. Data delivery in fragmented wireless sensor networks using mobile agents. In *ACM MSWiM*, 2007.

[7] W. Wang, V. Srinivasan, and K. Chua. Using mobile relays to prolong the lifetime of wireless sensor networks. In *MobiCom*, 2005.

[8] M. Younis, M. Bangad, and K. Akkaya. Base station repositioning for optimized performance of sensor networks. *IEEE VTC*, 2003.

[9] A. Abbasi and M. Younis. A survey on clustering algorithms for wireless sensor networks. *Computer Comm.*, 30(14-15):2826–2841, 2007.

[10] J. Buzen. Computational algorithms for closed queueing networks with exponential servers. *Communications of the ACM*, 16(9), 1973.