Data Delivery in Fragmented Wireless Sensor Networks Using Mobile Agents

Hisham M. Almasaeid Ahmed E. Kamal
Electrical and Computer Engineering Department
Iowa State University, Ames, IA 50011
{hisham, kamal}@iastate.edu

ABSTRACT

Due to the wide range of applications in sensors and Wireless Sensor Networks (WSN), research in this area has recently received increasing attention. WSNs rely on network connectivity to deliver data to a base station through multihop communication. However, connectivity may not be always achievable for a number of reasons. In this paper, we study the problem of data delivery in disconnected WSNs. A special class of disconnected sensor networks called "Fragmented wireless sensor networks (FWSN)" is considered. A FWSN consists of several groups of connected sensors that we call "fragments". To achieve connectivity between these fragments, mobile agents move in the network and act as data relays between fragments, in order to eventually deliver data to the base station. The main contribution of this paper is the modeling of the movement of these mobile relay nodes as a closed queueing network to obtain steady state results of the distribution of the mobile relays in the network. Building on these results, we derive the distributions of the fragment-to-fragment, and fragment-to-sink delays. Comparing these analytical results to results from the TOSSIM simulator, it is shown that this model accurately captures the system behavior, and can be used to predict data delivery delays. The model is also used to study the effect of the movement policy, number and speed of mobile relays, and the service time at each fragment on the end-to-end delay.

1. INTRODUCTION

Recent advances in wireless communication technologies have enabled the development of small, low-cost and low-power multi-functional sensor nodes that are able to sense the environment, process data and communicate with each other in short range. A sensor network consists of a large number of tiny sensors that communicate in a multihop fashion in order to deliver the collected data to a central processing unit, usually called the base station or the sink node. Exploiting mobile agents to improve the performance of a sensor network has recently become an important area of research in sensor networks [1, 2]. *"Mobile agents"* is a very broad term that includes any mobile entity, like vehicles, humans, animals, and mobile robots, equipped with a radio transceiver. Mobile agents are used in applications that utilize mobility for communication-based operations, like data relay and collection, as well as physical operations, like replacement of defective sensor nodes. Mobility has been exploited by many researchers for data gathering in sensor networks. Some schemes rely on existing mobility in the environment, like vehicles or animals present in the network field [3, 4], whereas some others suggest supporting the system with mobile elements that have better buffer and energy capabilities than ordinary static sensor nodes and are able to communicate over longer distances [5, 6]. A mobile agent that is used for data delivery can take one of three different roles; data collector, data relay, or data sink.

The concept of mobile data collectors was first introduced in [3] to connect sparse sensor networks, where mobile data collectors, referred to as *data MULEs*, move randomly and collect data from sensor nodes. Then, data MULEs unload the carried data as they get close enough to a base station. The idea of sink mobility [6, 7] has been proposed as a method for data collection that prolongs the network lifetime by reducing the energy spent by static sensor nodes to relay traffic. Mobility of the base station poses several challenges regarding how data should be routed [8] and what the optimal movement strategy is [9].

The use of resource rich mobile nodes, referred to as mobile relays (MR) or routers, that keep moving in a network, in both sparse and dense deployments, to relay data between stationary sensor nodes has been recently utilized to prolong the network lifetime [10, 11] and to enhance the data delivery processMobile relays have been referred to as *Message Ferries (MF)* by some researchers. In reference [12], a single message ferry with deterministic movement has been exploited to efficiently deliver data in sparse mobile ad hoc networks.

Mobility has also been used to overcome network disconnection. Most of the current research attributes network disconnection to the sparse nature of the network. Sparseness has been assumed to be either due to the lack of sensors at the deployment phase or due to node failures. In this paper, we consider another form of network disconnection in which the network is fragmented into several subnetworks (fragments), where the *fragment* is a connected group of sensor nodes. We call such network a fragmented wireless sensor network (FWSN). Network fragmentation might be due to a number of reasons, including:

(1) *Node Failures*: which might cause network fragmenta-

.

tion in two different scenarios:

(i) In harsh and hostile environments nodes might fail at a mass scale in certain regions causing the network to be fragmented. For example, a bomb or land mine detonation in a battle field might break the deployed network into several fragments. In another scenario, natural phenomena like heavy rain or a mud slide can wash or move the sensor nodes away, or even bury them where they become useless.

(ii) Power depletion due to the unbalanced load distribution at individual sensors as a result of random deployments, i.e., hot spots.

(2) *Fragmented area of phenomenon*: in some applications, it might not be required to cover the whole field. Instead, specific regions in the field must be covered.

(3) *Environmental conditions*: the deployment of a connected wireless sensor network might not be feasible due to physical obstacles and restrictions.

Under the assumption that a FWSN faults, adding more static sensors between fragments to connect them might not be possible especially in hazardous scenarios, like battle fields, disaster areas, or areas contaminated with chemical materials. We propose a data-relay based scheme to deliver data in FWSNs that uses mobile agents to act as relay nodes between fragments. Agents move continuously in the network according to a certain policy in order to act as relay nodes between fragments. This way, the network will be connected over a time period instead of continuous connectivity. The major contribution of this paper is the modeling of the movement of these relay nodes as a closed queueing network to obtain steady state results of the distribution of the mobile relays in the network. Building on these results, we derive the distributions of the fragment-to-fragment and fragment-to-sink delays. The effect of the number of mobile relays and the underlying movement policy on the data latency is also studied. Moreover, we study the problem of engineering the service time, i.e., the time that a mobile relay spends in relaying data at each fragment.

The rest of this paper is organized as follows. In Section 2, we introduce a formal definition of the *"Data Delivery in Fragmented Wireless Sensor Networks"* problem and then present a Closed Queuing Network model to evaluate the performance of the system. Section 3 addresses the issue of fragment-to-fragment data latency. The fragment-to-sink data latency is studied in Section 4. Engineering the service time at a service center, which is the time spent by a mobile agent to relay data between two adjacent fragments, is studied in Section 5. We discuss some analytical and simulation results in Section 6. Section 7 concludes the paper and suggests some future research directions.

2. PROBLEM DEFINITION AND MODELING

In this section, we first define the problem of data delivery in FWSNs and make some operational assumptions. Then, we model the problem as a Gordon-Newell queueing network.

2.1 Problem Definition

The problem of *"Data Delivery in Fragmented Wireless Sensor Networks"* is defined as follows:

Given an FWSN that consists of n fragments, let K^* be the minimum number of static relay nodes required to connect the whole network[1]. We make the following assumptions:

- There are K mobile relays, $1 \leq K < K^*$.
- All mobile relays move at the same speed of $L\ m/s$.
- No more than one mobile relay is used to connect a pair of adjacent fragments. An MR relays data between a pair of adjacent fragment through direct communication if these fragments are close enough. If the two fragments are far apart, then the MR should carry the data from one fragment, move close enough to the second fragment, and relay the data.
- The connection between any pair of fragments i and j must persist for at least t^s_{ij} time units. During this period, data will be relayed from fragment i (source-fragment) to fragment j (destination-fragment) through the MR in between. This time period includes both the direct communication time and the movement time between the pair of fragments (if necessary). t^s_{ij} will be referred to as *service time* or *sojourn time* interchangeably. For the time being, t^s_{ij} is assumed to be long enough to relay all generated data, and we will revisit this issue of in Section 5.

Our objective in this study is the following:

- Find the distribution of the waiting time for a fragment before its data is relayed to the next fragment. We call this waiting time the *"idle time"*.
- Find the distribution of the end-to-end delay, i.e., the time to deliver data to the base station.

2.2 Problem Modeling

We model the movement of the mobile relays using a closed queueing network, also known as *Gordon-Newell* network. The locations between fragments where mobile relays stop at in order to relay data are modeled as infinite-buffer/infinite-server queues, and the mobile relays are modeled as customers. Figure 1 shows a case-study FWSN that will be used throughout this paper to illustrate our modeling approach. This network consists of six fragments labeled *(FRAG-1,...,FRAG-6)*. A Hexagon between a pair of fragments represent a potential location where an MR can stop to relay data from one fragment to another. For example, an MR at connection point 1 relays data from *FRAG-1* to *FRAG-2*, and an MR at connection point 3 relays data from *FRAG-3* to *FRAG-6* and so on. From now on, connection points will be referred to as *Service Centers* (s/c).

Let s/c_k be the k^{th} service center that connects fragments i and j in a FWSN. Service center s/c_k is modeled as an infinite-buffer/infinite-server queue, which offers service according to an exponentially distributed service time, that corresponds to the sojourn time, with a rate $\mu_k = 1/t^s_{ij}$. We call such a queue a *relay queue*. For the time being we assume that the routing probabilities between service centers, i.e., the movement policy, are given. Table 1 shows a sample set of routing probabilities that are used for our case study, where q_{ij} represents the probability that a MR leaving s/c_i goes to s/c_j. We refer to the movement policy described in Table 1 as the *baseline* policy throughout this

[1]Note that there is always a way to connect the network over a sufficiently long time period, t, with K mobile relays, where $1 \leq K < K^*$. However, the fewer the number of mobile relays, the higher the delay.

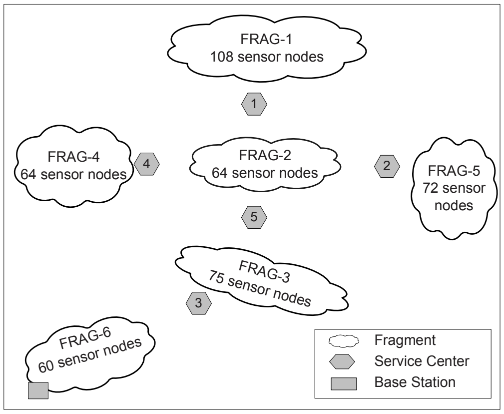


Figure 1: A case-study FWSN with six fragments and five service centers.

paper. To capture the trip time between service centers, we do the following:

- For every pair of *relay queues* (i, j), for which $q_{ij}{>}0$, we add an infinite-buffer/infinite-server queue, which we call a *movement queue*, with a service rate $\mu_i{=}L/d_{ij}$, where L is the MR speed and d_{ij} is the distance between s/c_i and s/c_j. This queue models the movement time from s/c_i to s/c_j. We assume in the case-study in Figure 1 that the route that an MR takes from s/c_i to s/c_j is the one it takes from j to i, therefore $d_{ij}{=}d_{ji}$.
- The probability that a customer leaving queue i goes to the new movement queue leading to s/c_j is the same as the probability that an MR leaving s/c_i goes to s/c_j, i.e., q_{ij}. Then, the probability that a customer leaving the movement queue goes to queue j is 1.

In the remainder of this paper, we will use the word *queue* to refer a queue of any type (i.e. movement or relay). To reference a particular type we use *"movement queue"* or *"relay queue"*. We model the number of mobile relays in the network as the number of customers that circulate in the closed queueing network. Tables 1 and 2 summarize all the parameters associated with the case-study in Figure 1 and they will be used in all analysis and simulations throughout this work unless mentioned otherwise.

Using the parameters in Tables 1 and 2 and the mapping procedure described above we construct the Gordon-Newell network model shown in Figure 2 for the case study shown in Figure 1. Gray nodes represent the *relay queues* while clear nodes represent the *movement queues*. μ_i represents the service rate of queue i. In Sections 3 and 4, we use the queueing network in Figure 2 to obtain steady state probabilities that we then use to evaluate the fragment-to-fragment and fragment-to-sink delays.

Service center	q_{i1}	q_{i2}	q_{i3}	q_{i4}	q_{i5}
s/c_1	0.0	0.0	0.0	0.4	0.6
s/c_2	0.0	0.0	0.6	0.0	0.4
s/c_3	0.0	0.2	0.0	0.8	0.0
s/c_4	0.5	0.0	0.0	0.0	0.5
s/c_5	0.35	0.25	0.2	0.2	0.0

Table 1: Sample routing probabilities between service centers for the case-study FWSN in Figure 1.

Parameter	Value	Parameter	Value
Num. of MRs	2	d_{25}	$401.6m$
Num. of s/c's	5	d_{23}	$680.34m$
Speed (L)	$1.2m/s$	d_{34}	$463.1m$
t^s_{ij}	$60sec$	d_{35}	$321.74m$
d_{14}	$391.76m$	d_{45}	$357.94m$
d_{15}	$368.76m$		

Table 2: System parameters of the case-study FWSN shown in Figure 1.

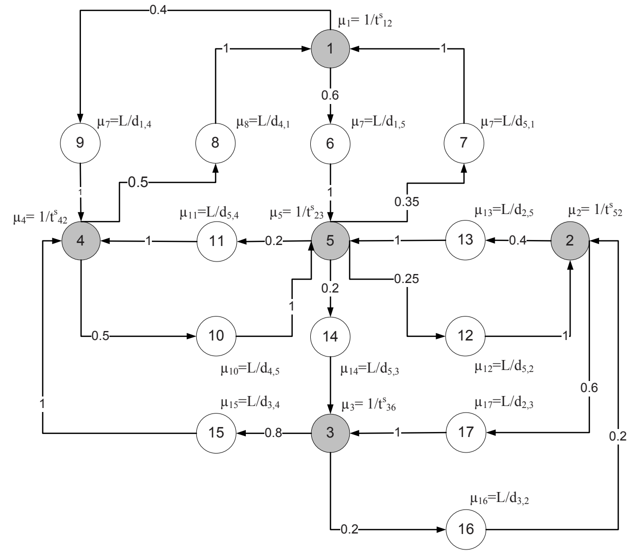


Figure 2: *Gordon-Newell Network* model for the case-study FWSN shown in Figure 1

3. IDLE TIME DISTRIBUTION

In this section we derive the idle time distribution at any queue. The idle time at a service center is the idle time at the *relay queue* that models this service center. We derive two forms of the idle time distribution:

(1) *Joint distribution*: in which the probability that *queue* i is idle for time $t \le \tau$ is taken jointly with the probability that *queue* i has been idle at $t{=}0$.

(2) *Conditional distribution*: in which the probability that *queue* i is idle for time $(t \le \tau)$ is conditioned on the fact that *queue* i was idle at $t{=}0$.

As will be seen in Section 4, the idle time distribution in these two different forms is required for the evaluation of the *end-to-end* delay. We start by computing the *end-to-end* delay along a certain path from the point where the first service center on that path becomes idle. Therefore, we use the *Conditional distribution* for the first service center, while the *Joint distribution* is used for the remaining service centers on the path.

3.1 Definitions

- M is the number of queues in the network (this includes both relay and movement queues).
- K is the number of customers (i.e., mobile relays).
- $\overrightarrow{N} = \{n_1(\overrightarrow{N}), \ldots, n_M(\overrightarrow{N})\}$ is the state of the system, i.e., queueing network, 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 $\overrightarrow{N}$.
- Ω is the set of all possible system states, $|\Omega| = \binom{M+K-1}{K}$.
- μ_j is the state-independent service rate of *queue* j.
- q_{ij} is the probability that a customer leaving *queue* i will move to *queue* j (we assume that there is one class of customers in the system). In our queueing model, $q_{ii} = 0$.
- $\pi(\overrightarrow{N})$ is the steady-state probability of state $\overrightarrow{N}$, such that $\sum_{\overrightarrow{N}\in\Omega} \pi(\overrightarrow{N}) = 1$.
- 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 probability that at time t, *queue* i is in state $\overrightarrow{N}$ in which $n_i(\overrightarrow{N})$=0, and n_i became zero at time t=0 and remained so in $[0, t]$ (i.e., joint probability). In other words, this is the probability that *queue* i is idle for a time that is greater than t. Note that, $p^i_{idle}(t, \overrightarrow{N})$ is defined only over network states in which n_i=0.
- $\overrightarrow{p}^i_{idle}(t)$=$[p^i_{idle}(t, \overrightarrow{N}_1),\ \ldots,\ p^i_{idle}(t, \overrightarrow{N}_{E_i})]^T$.
- U_i is a row vector of ones such that $|U_i|$=E_i.
- T_i is a random variable that represents the idle time at *queue* i.
- $F_{T_i}(t) = probability\,(T_i \le t)$, is the joint cumulative distribution function (CDF) of T_i (i.e., the joint distribution that queue i starts and idle period at the reference time (t^*=0) and stays idle for time $T_i \le t$).
- $f_{T_i}(t) = \frac{d}{dt} F_{T_i}(t)$, this is the joint *pdf* of T_i.
- $F_{T_i|I}(t) = probability\{\ T_i \le t \mid n_i = 0\ at\ t = 0\}$, i.e., the conditional distribution that queue i stays idle for time $T_i \le t$ given that it starts an idle period at the reference time t^*=0.
- $f_{T_i|I}(t) = \frac{d}{dt} F_{T_i\ |\ I}(t)$

3.2 The Joint Distribution

In an *infinite-server exponential service time queue* i, the probability of serving a customer within a very small time interval $\Delta t \to 0$, given that there are n_i customers in the queue, is given by:

$$n_i\mu_i\Delta t\ +\ o(\Delta t) \tag{1}$$

where $o(\Delta t)$ is a function that approaches zero faster than Δt. Therefore, the following *forward Chapman-Kolmogorov equation* holds;

$$\begin{aligned} p^i_{idle}(t{+}\Delta t, \overrightarrow{N}) &= p^i_{idle}(t, \overrightarrow{N}) \cdot prob(\overrightarrow{N}\ at\ t{+}\Delta t|\overrightarrow{N}\ at\ t) \\ &+ \sum_{\overrightarrow{N}^*\in\Omega} p^i_{idle}(t, \overrightarrow{N}^*) \cdot prob(\overrightarrow{N}\ at\ t{+}\Delta t|\overrightarrow{N}^*\ at\ t) \end{aligned} \tag{2}$$

In this Markovian system only one event (i.e., transition) is possible in a very short time period Δt. Therefore, the probability that the system will evolve from state $\overrightarrow{N^*}$ at t to state $\overrightarrow{N}$ at $t + \Delta t$ is given by:

$$prob(\overrightarrow{N}\ at\ t + \Delta t|\overrightarrow{N}^*\ at\ t) =$$

$$\begin{cases} n_j(\overrightarrow{N}^*)\mu_j q_{jk}\Delta t + o(\Delta t) & if\ \overrightarrow{N} = \overrightarrow{N}^* + \overrightarrow{1}_j - \overrightarrow{1}_k\ \forall \\ & j, k \in \{1, 2, \ldots, M\},\ j, k \ne i \\ 0 & otherwise \end{cases} \tag{3}$$

Note that this probability is zero when the difference between the two states is more than one customer, because it means that a single transition cannot move the system to state $\overrightarrow{N}$. On the other hand, the probability of no transition in the time period $[t, t + \Delta t]$ is given by:

$$prob(\overrightarrow{N}\ at\ t{+}\Delta t|\overrightarrow{N}\ at\ t) = 1 - \sum_{\substack{j=1\\ j\ne i}}^{M} n_j(\overrightarrow{N})\mu_j\Delta t + o(\Delta t) \tag{4}$$

Using equations (3) and (4) in equation (2) we get,

$$\begin{aligned} &p^i_{idle}(t + \Delta t, \overrightarrow{N}) = \\ &\sum_{\substack{j=1\\ j\ne i}}^{M} \sum_{\substack{k=1\\ k\ne i\\ n_k(\overrightarrow{N})>0}}^{M} n_j(\overrightarrow{N}{+}\overrightarrow{1}_j{-}\overrightarrow{1}_k)\mu_j\Delta t \cdot p^i_{idle}(t, \overrightarrow{N}{+}\overrightarrow{1}_j{-}\overrightarrow{1}_k)q_{jk} \\ &+ \left(1 - \sum_{\substack{j=1\\ j\ne i}}^{M} n_j(\overrightarrow{N})\mu_j\Delta t\right) \cdot p^i_{idle}(t, \overrightarrow{N}) + o(\Delta t) \end{aligned}$$

Rearranging the terms and then taking the limit as $\Delta t \to 0$, we obtain

$$\frac{d}{dt}\overrightarrow{p}^i_{idle}(t) = \mathbf{A}_i\overrightarrow{p}^i_{idle}(t) \tag{5}$$

Where $\mathbf{A}_i = [a^i_{xy}]$ is an $E_i \times E_i$ matrix such that:

$$a^i_{xy} = \begin{cases} -\sum_{\substack{j=1\\ j\ne i}}^{M} n_j(\overrightarrow{N}_x)\mu_j & \text{if } x = y \\ n_j(\overrightarrow{N}_y)\mu_j q_{jk} & \text{if } \overrightarrow{N}_y = \overrightarrow{N}_x + \overrightarrow{1}_j - \overrightarrow{1}_k \\ 0 & otherwise \end{cases} \tag{6}$$

The solution of the differential equation in (5) is,

$$\overrightarrow{p}^i_{idle}(t) = e^{\mathbf{A}_i t}\overrightarrow{p}^i_{idle}(0). \tag{7}$$

$p^i_{idle}(0, \overrightarrow{N})$ can be found using the steady state probabilities as well as the routing probabilities between service centers. Let ξ^{ik}_j be the transition of a customer from queue i to queue k (for $k \ne i$) that will make the system evolve from state $\overrightarrow{N}_j + \overrightarrow{1}_i - \overrightarrow{1}_k$ to state $\overrightarrow{N}_j$ such that $n_i(\overrightarrow{N}_j) = 0$ and $n_k(\overrightarrow{N}_j) > 0$, i.e., $n_i(\overrightarrow{N}_j{+}\overrightarrow{1}_i{-}\overrightarrow{1}_k)$=1. In other words, ξ^{ik}_j is the transition that will initiate an idle period at queue i. Let $prob(\xi^{ik}_j)$ be the probability that this transition takes place before any other transition. Then,

$$p^i_{idle}(0, \overrightarrow{N}_j) = \sum_{\substack{k=1\\ k\ne i, n_k(\overrightarrow{N})>0}}^{M} \pi(\overrightarrow{N}_j + \overrightarrow{1}_i - \overrightarrow{1}_k) \cdot prob(\xi^{ik}_j) \tag{8}$$

It has been shown in [13] that the steady state (equilibrium) distribution of the network state is given by,

$$\pi(\overrightarrow{N}) = \frac{1}{G(M, K)} \prod_{i=1}^{M} \left(\frac{\lambda_i}{\mu_i}\right)^{n_i(\overrightarrow{N})} \tag{9}$$

where λ_i is the effective arrival rate at queue i, which can be obtained by solving the following set of dependent equations:

$$\lambda_i = \sum_{j=1}^{M} \lambda_j q_{ji}, \quad 1 \le i \le M. \tag{10}$$

There is no unique solution for equation (10). However, the normalization constant $G(M,K)$ is used to make

$$\sum_{\overrightarrow{N}\in\Omega}\pi(\overrightarrow{N})=1.$$

This normalization constant can be obtained using the *convolution algorithm* [14].
Given $\pi(\overrightarrow{N}_j+\overrightarrow{1}_i-\overrightarrow{1}_k)$, and based of the Markovian property of the system, $prob(\xi_j^{ik})$ can be computed as:

$$\begin{aligned} prob(\xi_j^{ik}) &= \frac{n_i(\overrightarrow{N}_j+\overrightarrow{1}_i-\overrightarrow{1}_k)\mu_i}{\sum_{l=1}^{M}\mu_l n_l(\overrightarrow{N}_j+\overrightarrow{1}_i-\overrightarrow{1}_k)}q_{ik} \\ &= \frac{\mu_i}{\sum_{l=1}^{M}\mu_l n_l(\overrightarrow{N}_j+\overrightarrow{1}_i-\overrightarrow{1}_k)}q_{ik} \end{aligned} \tag{11}$$

Using equations (8) and (11) we can write $p^i_{idle}(0,\overrightarrow{N}_j)$ as,

$$p^i_{idle}(0,\overrightarrow{N}_j)=\sum_{\substack{k=1\\k\neq i\\n_k(\overrightarrow{N}_j)>0}}^{M}\frac{\mu_i q_{ik}\pi(\overrightarrow{N}_j+\overrightarrow{1}_i-\overrightarrow{1}_k)}{\sum_{l=1}^{M}\mu_l n_l(\overrightarrow{N}_j+\overrightarrow{1}_i-\overrightarrow{1}_k)} \tag{12}$$

Using equation (7), $F_{T_i}(t)$ can be written as,

$$F_{T_i}(t)=1-U_i e^{A_i t}\overrightarrow{p}^{\,i}_{idle}(0) \tag{13}$$

where $\overrightarrow{p}^{\,i}_{idle}(0)$ is given by,

$$\overrightarrow{p}^{\,i}_{idle}(0)=[p^i_{idle}(0,\overrightarrow{N}_1),\ldots,p^i_{idle}(0,\overrightarrow{N}_{E_i})]^T$$

and we obtain the elements of $\overrightarrow{p}^{\,i}_{idle}(0)$ using equation (12).

3.3 Conditional Distribution

The next step is to evaluate $F_{T_i|I}$. Let $p^i_{idle}(t,\overrightarrow{N}|n_i=0\ at\ t=0)$ be the probability that queue i is idle for more than t given that it was idle at time $t=0$. Therefore,

$$p^i_{idle}(t,\overrightarrow{N}|n_i=0\ at\ t=0)=e^{A_i t}p^i_{idle}(0,\overrightarrow{N}|n_i=0\ at\ t=0) \tag{14}$$

such that,

$$p^i_{idle}(0,\overrightarrow{N}|n_i=0\ at\ t=0)=\frac{p^i_{idle}(0,\overrightarrow{N})}{\sum_{\substack{\overrightarrow{N}^*\in\Omega\\n_i(\overrightarrow{N}^*)=0}}p^i_{idle}(0,\overrightarrow{N}^*)} \tag{15}$$

Substituting equation (12) in (15) we get,
$p^i_{idle}(0,\overrightarrow{N}|n_i=0\ at\ t=0)=$

$$\frac{\sum_{\substack{k=1\\k\neq i\\n_k(\overrightarrow{N})>0}}^{M}\left(\frac{\mu_i q_{ik}\pi(\overrightarrow{N}+\overrightarrow{1}_i-\overrightarrow{1}_k)}{\sum_{l=1}^{M}\mu_l n_l(\overrightarrow{N}+\overrightarrow{1}_i-\overrightarrow{1}_k)}\right)}{\sum_{\substack{\overrightarrow{N}^*\in\Omega\\n_i(\overrightarrow{N}^*)=0}}\sum_{\substack{k=1\\k\neq i\\n_k(\overrightarrow{N}^*)>0}}^{M}\left(\frac{\mu_i q_{ik}\pi(\overrightarrow{N}^*+\overrightarrow{1}_i-\overrightarrow{1}_k)}{\sum_{l=1}^{M}\mu_l n_l(\overrightarrow{N}^*+\overrightarrow{1}_i-\overrightarrow{1}_k)}\right)} \tag{16}$$

Finally, the conditional distribution $F_{T_i|I}(\tau)$ is given by,

$$F_{T_i|I}(\tau)=1-U_i e^{A_i\tau}\overrightarrow{p}^{\,i}_{idle}(0|n_i{=}0\ at\ t{=}0), \tag{17}$$

where $\overrightarrow{p}^{\,i}_{idle}(0|n_i{=}0\ at\ t{=}0)$ is given by,

$$\overrightarrow{p}^{\,i}_{idle}(0|n_i{=}0\ at\ t{=}0)=$$
$$[p^i_{idle}(0,\overrightarrow{N}_1|n_i{=}0\ at\ t{=}0),\ldots,p^i_{idle}(0,\overrightarrow{N}_{E_i}|n_i{=}0\ at\ t{=}0)]^T \tag{18}$$

3.4 Average Idle Time

The average idle time for service center i, denoted by $E[T_i|I]$, is given by:

$$\begin{aligned} E[T_i|I] &= \int_{t=0}^{\infty}\left(1-F_{T_i|I}(t)\right)dt \\ &= \int_{t=0}^{\infty}U_i e^{A_i\tau}\overrightarrow{p}^{\,i}(0|n_i{=}0\ at\ t{=}0)dt \end{aligned} \tag{19}$$

Evaluating equation (19) involves the integration of a matrix exponential ($e^{A_i t}$) which can be very complicated depending on the size of the matrix A_i. Therefore, we evaluate this expression numerically by dividing t into a lesser number of small time steps, each of which is equal to Δ, as follows:

$$E[T_i|I]=\sum_{k=0}^{\infty}\left(1-F_{T_i|I}(k\Delta)\right)\Delta \tag{20}$$

By keeping Δ small, we can obtain very accurate results. For the rest of this paper, we will use this discrete method for evaluating average values and distributions.

4. END-TO-END DELAY DISTRIBUTION

Evaluating the distribution of the *end-to-end* delay possesses a trade-off between the accuracy of the solution and its complexity. To evaluate the exact end-to-end delay distribution along a path, we need to consider the dependency between the idle periods at all service centers that form the path. In the worst case, we need to consider the state of the whole path, and that gets worse as the length of the path gets longer since we might have to consider the state of the whole queuing system. This might make the evaluation intractable. Therefore, we propose two approximate approaches, namely, the *Convolution* approach (CONV) and the *Dynamic Programming-Like Approach* (DPA), to evaluate the end-to-end delay distribution based on the level of dependence to assume between service centers along the path. It is to be noted that we evaluate the end-to-end delay distribution as a probability mass function over a small time interval Δ which reduces the computational complexity while assuming that the time t consists of $\lceil\frac{t}{\Delta}\rceil$ such intervals. It is also worth pointing out that our analysis only applies to the case of one data path from a fragment to base station. The case of multiple data paths from a fragment to the base station is beyond the scope of this paper.

4.1 Convolution Approach

The convolution approach is based on the assumption that the distributions of the idle periods at different service centers along the path are independent. As the end-to-end delay is the sum of all the independent idle times along the path, then the distribution of the end-to-end delay is the convolution of all idle time distributions. Let $\mathcal{F}$ be a path of $|\digamma|$ service centers, and $\psi(t, \digamma)$ be the probability mass function that the end-to-end delay along the path $\digamma$ is t. Then, the end-to-end delay is given as,

$$\psi(\digamma, t) = F_{T_1|I} \circledast F_{T_2} \cdots \circledast F_{T_n}(t) \tag{21}$$

This approximation significantly reduces the complexity by ignoring the dependence between the idle periods. However, assuming independence underestimates the end-to-end delay since the presence of an MR at previous service centers is ignored, i.e., the probability of being *idle/busy* at any service center is evaluated without considering the status of previous service centers along the path.

4.2 Dynamic Programming Like Approach

We propose a dynamic programming-like approach (DPA) as a compromise between the computational complexity and solution accuracy. This compromise is based on the relaxation of the level of dependency between idle periods, i.e., by only considering the dependency between every pair of successive service centers. Since any service center will be in one of two different states (i.e., *busy* or *idle*) at any point in time, we have four different situations:

(1) s/c_i is *idle* for time t given that s/c_j was also *idle*; let the probability density function of this event be $p_{I|I}(i, j, t)$,

(2) The current service center (s/c_i) is *idle* for time t given that the previous service center (s/c_j) was *busy*; let the probability density function of this event be $p_{I|B}(i, j, t)$,

(3) s/c_i is *busy* given that s/c_j was *idle*; let the probability of this event be $p_{B|I}(i, j)$, and

(4) s/c_i is *busy* given that s/c_j was *busy*; let the probability of this event be $p_{B|B}(i, j)$.

Next we derive the formulas for all the four conditional probabilities listed above.

4.2.1 Conditional Probabilities

$\mathbf{p_{I|I}(i, j, t)}$. Let us start with the probability $p_{I|I}(i, j, t)$. Note that for the first service center on the path, $p_{I|I}(i, j, t) = f_{T_i|I}(t)$. For other service centers, it is the same as $f_{T_i|I}(t)$ except for the initial condition $\overrightarrow{p}^{\,i}_{idle}(0)$ where we have a different condition to be satisfied which is: *s/c_j has just ended its idle period, i.e., n_j changed from 0 to 1 and hence $n_j = 1$ at the beginning of t_i.* We refer to the probability of this condition for a certain network state $\overrightarrow{N}$ as $h^i_{idle}(0, \overrightarrow{N}|n_j{=}1 \; at \; t{=}0)$. This probability is given by:

$$h^i_{idle}(0, \overrightarrow{N}|n_j{=}1 \; at \; t{=}0) = \begin{cases} \dfrac{prob(\overrightarrow{N}, n_j(\overrightarrow{N}){=}1 \; at \; t{=}0)}{prob(n_j{=}1 \; at \; t{=}0)} & \text{if } n_j(\overrightarrow{N}){=}1, \; n_i(\overrightarrow{N}){=}0 \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

where $prob(\overrightarrow{N}, n_j(\overrightarrow{N}){=}1 \; at \; t{=}0)$ is given by,

$$prob(\overrightarrow{N}, n_j(\overrightarrow{N}){=}1, \; at \; t{=}0) = \sum_{\substack{k=1 \\ k \neq j}}^{M} \frac{n_k(\overrightarrow{N} + \overrightarrow{1}_k - \overrightarrow{1}_j)\mu_k}{\sum_{l=1}^{M} \mu_l n_l(\overrightarrow{N} + \overrightarrow{1}_k - \overrightarrow{1}_j)} q_{kj} \pi(\overrightarrow{N} + \overrightarrow{1}_k - \overrightarrow{1}_j) \tag{23}$$

We derived equation (23) following the same approach we used to derive equation (12). We used the probability that the transition from queue k to queue j, that will lead to state $\overrightarrow{N}$, takes place before any other transition. The probability $prob(n_j{=}1, \; at \; t{=}0)$ is given as,

$$prob(n_j{=}1, \; at \; t{=}0) = \sum_{\substack{\overrightarrow{N}^* \in \Omega \\ n_j(\overrightarrow{N}^*)=1}} prob(\overrightarrow{N}^*, n_j(\overrightarrow{N}^*){=}1 \; at \; t{=}0) \tag{24}$$

Let,

$$\overrightarrow{h}^{\,i}_{idle}(0|n_j{=}1) = [\overrightarrow{h}^{\,i}_{idle}(0, \overrightarrow{N}_1|n_j{=}1 \; at \; t{=}0), \ldots, \overrightarrow{h}^{\,i}_{idle}(0, \overrightarrow{N}_{E_i}|n_j{=}1 \; at \; t{=}0)]^T \tag{25}$$

Then, using equations (7) and (25) we get,

$$p_{I|I}(i, j, t) = \overrightarrow{U}_i(-\mathbf{A}_i)e^{\mathbf{A}_i t}\overrightarrow{h}^{\,i}_{idle}(0|n_j{=}1) \tag{26}$$

$\mathbf{p_{I|B}(i, j, t)}$. For $p_{I|B}(i, j, t)$, we do not know where the last transition was because we consider the dependency between consecutive queues only. But, we know that the initial state must have s/c_j busy, i.e., $n_j{>}0$, and $n_i{=}0$. Therefore, we approximate the probability of the initial state, which we refer to it as $h^i_{idle}(0, \overrightarrow{N}|n_j > 0 \; at \; t{=}0)$, using steady state probabilities. Equation (27) defines $h^i_{idle}(t, \overrightarrow{N}|n_j > 0 \; at \; t{=}0)$.

$$h^i_{idle}(0, \overrightarrow{N}|n_j > 0 \; at \; t{=}0) = \begin{cases} \dfrac{\pi(\overrightarrow{N})}{\sum_{\substack{\overrightarrow{N}^* \in \Omega \\ n_j(\overrightarrow{N}^*)>0}} \pi(\overrightarrow{N}^*)} & \text{if } n_j(\overrightarrow{N}) > 0, \; n_i(\overrightarrow{N}) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

Let,

$$\overrightarrow{h}^{\,i}_{idle}(0|n_j{>}0) = [\overrightarrow{h}^{\,i}_{idle}(0, \overrightarrow{N}_1|n_j{>}0 \; at \; t{=}0), \ldots, \overrightarrow{h}^{\,i}_{idle}(0, \overrightarrow{N}_{E_i}|n_j{>}0 \; at \; t{=}0)]^T \tag{28}$$

Then, using equations (7) and (28) we get,

$$p_{I|B}(i, j, t) = \overrightarrow{U}_i(-\mathbf{A}_i)e^{\mathbf{A}_i t}\overrightarrow{h}^{\,i}_{idle}(0|n_j > 0) \tag{29}$$

$\mathbf{p_{B|I}(i, j)}$. The third conditional probability is $p_{B|I}(i, j)$. The condition that must be satisfied at the initial state for $p_{B|I}(i, j)$ is the same as that for $p_{I|I}(i, j, t)$, which is that s/c_j has just ended its idle period. However, when this happens, s/c_i is busy, i.e., $n_j{=}1$ and $n_i{>}0$. We define $h^i_{busy}(0, \overrightarrow{N}|n_j{=}1 \; at \; t{=}0)$ as the probability of having the initial (at time 0) state $\overrightarrow{N}$ in which $n_i > 0$ given that s/c_j has just ended its idle period, i.e., $n_j{=}1$, and $n_i{>}0$. Therefore,

$$h^i_{busy}(0, \overrightarrow{N}|n_j{=}1 \; at \; t{=}0) = \begin{cases} \dfrac{prob(\overrightarrow{N}, n_j(\overrightarrow{N}){=}1 \; at \; t{=}0)}{prob(n_j{=}1 \; at \; t{=}0)} & \text{if } n_j(\overrightarrow{N}) = 1, \; n_i(\overrightarrow{N}) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

$prob(\overrightarrow{N}, n_j(\overrightarrow{N}){=}1 \; at \; t{=}0)$ and $prob(n_j{=}1 \; at \; t{=}0)$ are as given in equations (23) and (24) respectively. Equation (31) gives the final expression for $p_{B|I}(i,j)$.

$$p_{B|I}(i,j) = \sum_{\substack{\overrightarrow{N}\in\Omega \\ n_i(\overrightarrow{N})>0}} h^i_{busy}(0,\overrightarrow{N}|n_j{=}1 \ at \ t{=}0) \quad (31)$$

$\mathbf{p_{B|B}(i,j)}$. The last conditional probability to find is $p_{B|B}(i,j)$. Let $h^i_{busy}(0,\overrightarrow{N}|n_j{>}0 \ at \ t{=}0)$ be the probability of having the initial (at time 0) state $\overrightarrow{N}$ in which $n_i{>}0$ given that s/c_j is busy (i.e., $n_j{>}0$). Then, using the same approximation we made to derive equation (27) we get,

$h^i_{busy}(0,\overrightarrow{N}|n_j{>}0 \ at \ t{=}0) =$

$$\begin{cases} \dfrac{\pi(\overrightarrow{N})}{\sum_{\substack{\overrightarrow{N}^*\in\Omega \\ n_j(\overrightarrow{N}^*)>0}} \pi(\overrightarrow{N}^*)} & \text{if } n_j(\overrightarrow{N})>0, \ n_i(\overrightarrow{N})>0 \\ 0 & \text{otherwise} \end{cases} \quad (32)$$

Equation (33) gives the final expression for $p_{B|B}(i,j)$.

$$p_{B|B}(i,j) = \sum_{\substack{\overrightarrow{N}\in\Omega \\ n_i(\overrightarrow{N})>0}} h^i_{busy}(0,\overrightarrow{N}|n_j{>}0 \ at \ t{=}0) \quad (33)$$

4.2.2 Recursive approach for calculating the end-to-end delay distribution

Before we get into the details of our dynamic program, we start with some definitions.

- The time t is divided into an integer multiple of a small time step Δ. This is done to simplify computations. Therefore, once the probability density functions in (26) and (29) are multiplied by Δ, they becomes probability mass functions.
- As defined before, F is a path of service centers.
- v is used to index the service center over the path, i.e., $F(v)$ is the v^{th} service center along the path.
- φ is the status of a service center: $\varphi{=}I^+$ means that the service center has just ended its idle period, $\varphi{=}B$ means a busy service center, and $\varphi{=}X$ means unknown status and it is used for the first service center along the path as there is no previous service center for this one.
- $\alpha(t,F,v,\varphi)$ is the probability that the v^{th} service center along F is idle for time t given that its predecessor (i.e. $(v{-}1)^{th}$ service center) was in a status φ.
- $\beta(F,v,\varphi)$ is the probability that the v^{th} service center along the path F is busy given that its predecessor (i.e. $(v-1)^{th}$ service center) was in a status φ.
- $\gamma(t,F,v,\varphi)$ is the probability mass distribution function of the end-to-end delay along the path F conditioned on the fact that service center s/c_v was in a status φ. Therefore, $\psi(F,t) = \gamma(t,F,F(1),X)$.

Using the conditional probabilities we obtained earlier (viz., equations (26), (29), (31), and (33)) we express $\alpha(t,F,v,\varphi)$ and $\beta(F,v,\varphi)$ as follows:

$$\alpha(t,F,v,\varphi) = \begin{cases} p_{I|I}(F(v),F(v-1),t) & \text{if } \varphi{=}I^+, \ v{>}1 \\ p_{I|B}(F(v),F(v-1),t) & \text{if } \varphi{=}B, \ v{>}1 \\ f_{T_{F(v)}|I}(t) & \text{if } \varphi{=}X, \ v{=}1 \end{cases} \quad (34)$$

$$\beta(F,v,\varphi) = \begin{cases} p_{B|I}(F(v),F(v-1)) & \text{if } \varphi=I^+, \ v>1 \\ p_{B|B}(F(v),F(v-1)) & \text{if } \varphi=B, \ v>1 \\ 0 & \text{if } v=1 \end{cases} \quad (35)$$

The basic idea of the dynamic programming-like approach is that each service center has two possible states, either it is busy (i.e., has at least one MR) or idle. The recursive step of the dynamic programming-like approach for the evaluation of $\gamma(t,F,\varphi,v)$ is given by equation (36).

$$\begin{aligned} \gamma(t,F,\varphi,v) &= \beta(F,v,\varphi)\gamma(t,F,B,v{+}1) \\ &+ \sum_{k=1}^{t/\Delta} \alpha(k\Delta,F,v,\varphi)\gamma(t{-}k\Delta,F,I^+,v{+}1)\Delta \end{aligned} \quad (36)$$

The term, $\beta(F,v,\varphi)\gamma(t,F,B,v+1)$, is the probability that the current service center, viz.,s/c_v, is busy and the rest of the path is idle for time t, and this is why the entire idle time t should be incurred over all downstream service centers. Moreover, we pass $\varphi = B$ in the recursion so that in the following step in the recursion, the previous service center will be known as a busy center. The second term, $\sum_{k=1}^{t/\Delta} \alpha(k\Delta,F,v,\varphi)\gamma(t-k\Delta,F,I,v+1)$, is the probability that the current service center, viz., v, is idle for time $k\Delta$ and an idle time of $t-k\Delta$ is incurred over the rest of the path, summed over all possible values of K. Equation (37) shows the boundary conditions for the recursive formula above. For the case of $v = |F|$, the equation returns one of two different values, depending on the remaining time.

$$\gamma(t,F,\varphi,v) = \begin{cases} \beta(F,v,\varphi) & \text{if } v=|F| \text{ and } t=0 \\ \alpha(t,F,v,\varphi) & \text{if } v=|F| \text{ and } t>0 \end{cases} \quad (37)$$

5. ENGINEERING THE SERVICE TIME

The assumption that t^s_{ij} is the same for all service centers is not a practical one, since different fragments with different sizes and different locations in the network require different service times. This time depends on three parameters; the number of sensors within the fragment, the average data generation rate at each fragment, and the average amount of data relayed through the fragment. Before we get into details, we first introduce some definitions. For a fragment i, let:

- S_i be the number of sensor nodes in fragment i.
- ρ_i be the data generation rate in $bits/sec$ of a sensor in fragment i.
- R be the sensor's data transmission rate in $bits/sec$.
- ς_i be the average amount of relayed data, in bits, that will be temporarily buffered in fragment i.
- σ_i be the average total amount of data (generated and relayed) that is temporarily buffered in fragment i.
- H_i is a set of all fragments that $FRAG{-}i$ lies on their data paths toward the sink. For example, in our case study in Figure 1, $H_6{=}\{1,2,3,4,5\}$, $H_3{=}\{1,2,4,5\}$, $H_2{=}\{1,4,5\}$, and $H_1{=}H_4{=}H_5{=}\{\}$.

For a service center s/c_k, let:

- $s(k)$ be the source fragment that an MR at s/c_k relays data from. For example, in Figure 1, $s(1) = 1$, $s(2) = 5$, $s(3) = 3$, $s(4) = 4$, and $s(5) = 2$.
- $d(k)$ be the destination fragment that an MR at s/c_k relays data to. For example, in Figure 1, $d(1) = 2$, $d(2) = 2$, $d(3) = 6$, $d(4) = 2$, and $d(5) = 3$.

By considering the idle time and the following sojourn time at s/c_k to form an alternating renewal process, equation (38) evaluates the average of the total amount of data relayed into fragment $s(k)$ by summing the data generation rates at all the fragments whose data is relayed through our $s(k)$ during the average idle time at s/c_k as well as the sojourn time.

$$\varsigma_{s(k)} = \begin{cases} \left(t^s_{s(k),d(k)} + E[T_k|I]\right) \sum\limits_{i \in H_{s(k)}} S_i \rho_i & \text{if } |H_{s(k)}| > 0 \\ 0 & \text{otherwise} \end{cases} \tag{38}$$

Then, equation (39) is used to estimate the average amount of data that might be buffered at a fragment, relayed into and generated within the fragment.

$$\sigma_{s(k)} = \left(t^s_{s(k),d(k)} + E[T_k|I]\right) S_{s(k)} \rho_{s(k)} + \varsigma_{s(k)} \tag{39}$$

We propose an algorithm that rely on this approximation to calculate t^s_{ij}. Algorithm 1 first initializes t^s_{ij}, and then solves the closed queueing network and calculates the average idle time at all service centers using the equation (20). The next step is to calculate new t^s_{ij}'s based on the average amount of buffered data evaluated using equation (39). This process is repeated until all the differences between the new values of t^s_{ij} and the old ones are less than a predetermined threshold.

Algorithm 1: Calculate t^s_{ij}

Initialize t^s_{ij} $\forall\ i, j$
repeat
  Solve the queueing network to find the average idle time $E[T_k|I]$ for every service center s/c_k.
  for all service center s/c_k **do**
    Set $i = s(k)$ and $j = d(k)$
    Evaluate σ_i using Equation (39)
    $\epsilon_k = |t^s_{ij} - \frac{2\sigma_i}{R}|$
    $t^s_{ij} = \frac{2\sigma_i}{R}$
  end for
until $\epsilon_k < threshold \ \ \forall k$

Algorithm 1 does not consider the time that an MR might need to move between two fragments to relay data. For a fixed MR speed and movement route, this movement time is fixed and cannot be engineered. Therefore, we just engineer the communication time and then, the total sojourn time will be the sum of the engineered communication time and the fixed movement time. For the case-study of this paper, we assume that all pairs of adjacent fragments are close enough so that MRs can deliver data through direct communication without any movement.

6. RESULTS AND DISCUSSIONS

In this section, we present some analytical and simulation results that validate the accuracy of our modeling approach and provide some conclusions about the performance of the system versus several parameters including, number and speed of MRs, movement policy, and sojourn time.
We applied the expressions in equations (13) and (17) to evaluate $F_{T_i|I}(t)$ and $F_{T_i}(t)$ for all the service centers in our case study (Figure 1 and Tables 1 and 2). The results are shown in Figure 3. Notice that $F_{T_i|I}(0)$=0 because at t=0 s/c_i has just become idle and it will stay idle for non-zero time. However, $F_{T_i}(0)$ can be non-zero, which is the probability that s/c_i is idle at time t=0.

Policy	s/c	q_{i1}	q_{i2}	q_{i3}	q_{i4}	q_{i5}
Uniform	s/c_1	0.0	0.0	0.0	0.5	0.5
	s/c_2	0.0	0.0	0.5	0.0	0.5
	s/c_3	0.0	0.5	0.0	0.5	0.0
	s/c_4	0.5	0.0	0.0	0.0	0.5
	s/c_5	0.25	0.25	0.25	0.25	0.0
Distance-Based	s/c_1	0.0	0.0	0.0	0.43	0.57
	s/c_2	0.0	0.0	0.69	0.0	0.31
	s/c_3	0.0	0.39	0.0	0.61	0.0
	s/c_4	0.46	0.0	0.0	0.0	0.54
	s/c_5	0.28	0.20	0.27	0.25	0.0
Deterministic	s/c_1	0.0	0.0	0.0	0.0	1.0
	s/c_2	0.0	0.0	1.0	0.0	0.0
	s/c_3	0.0	0.0	0.0	1.0	0.0
	s/c_4	1.0	0.0	0.0	0.0	0.0
	s/c_5	0.0	1.0	0.0	0.0	0.0

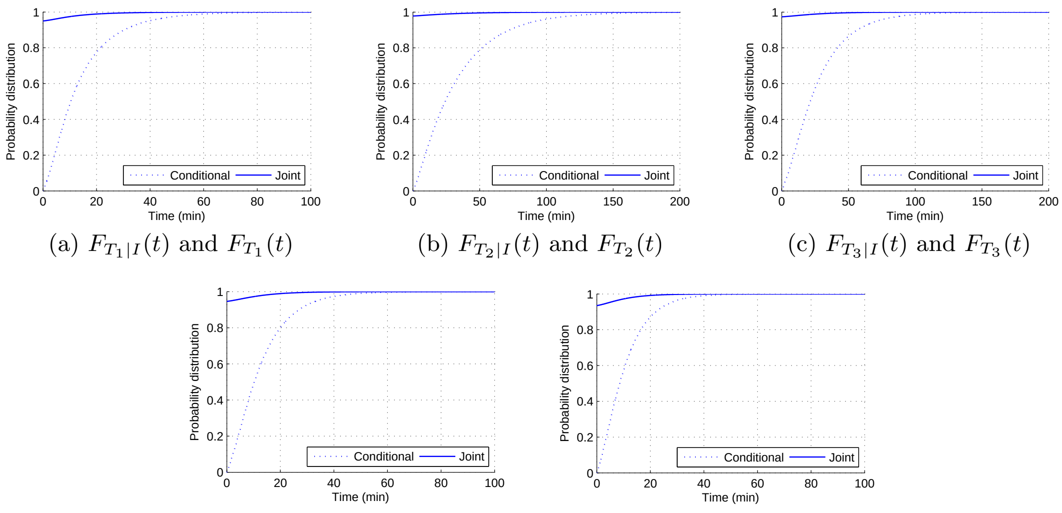
Table 3: Three policies for case-study in Figure 1.

To validate the accuracy of our analysis and to understand the effect of the movement policy, we evaluated, through analysis and simulation, the average idle time at all service centers in the case study shown in Figure 1 using the parameters shown in Table 2 under four different movement policies. In addition to the *baseline* policy in Table 1, we propose three different movement policies: *uniform*, *distance based*, and *deterministic*. In a uniform policy, an MR leaving a certain service center goes to any of the reachable neighboring service centers with the same probability. The distance based policy suggests that an MR goes to closer reachable neighboring service centers with higher probability than farther ones. The third policy assumes that an MR follows a deterministic route (i.e., a cycle) through all the service centers. Table 3 shows all the three policies.

For our simulation model, we used the TOSSIM simulator [15]. TOSSIM simulates TinyOs sensor networks at the bit level which guarantees a high level of fidelity. We also used Tython scripting language [16] to implement mobility.

Figure 4 shows the average idle time at all service centers using all four policies obtained by simulation using a speed of $1.2m/s$ and Δ=$0.01min$. Note that s/c_2 and s/c_3 suffer more than others in all policies, except under the deterministic policy, and that s/c_3 lies on the data paths of $FRAG-1$ though $FRAG-5$. Therefore, the *end-to-end* delay of the data generated at those fragments will be negatively affected at s/c_3. On the other hand, s/c_3 receives good service under the deterministic policy compared to other policies. It is evident that the deterministic policy is the fairest among all policies. Every service center is idle for the time required to travel over the cycle. In our particular case study, a deterministic policy is optimal in terms of fairness because all sojourn times (i.e. t^s_{ij}) are equal. The average idle times obtained using our queueing model are shown in Figure 5. It is evident that our analysis is very accurate as it shows an average error of about 5%.

As a matter of fact, it turns out that enhancing the movement policy might be better than increasing the number of mobile relays to minimize the maximum idle time in a FWSN. To verify this, we simulated our case study with different numbers of mobile relays ranging from 1 to 4 under the *baseline* and the *deterministic* policies. For the same speed, as shown in Figure 6, the maximum idle time under



(a) $F_{T_1|I}(t)$ and $F_{T_1}(t)$
(b) $F_{T_2|I}(t)$ and $F_{T_2}(t)$
(c) $F_{T_3|I}(t)$ and $F_{T_3}(t)$
(d) $F_{T_4|I}(t)$ and $F_{T_4}(t)$
(e) $F_{T_5|I}(t)$ and $F_{T_5}(t)$

Figure 3: The conditional, $F_{T_i|I}(t)$, and joint, $F_{T_i}(t)$, distributions at all service centers at a speed of 1.2m/s.

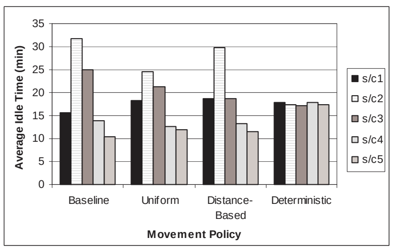


Figure 4: Average idle time at a speeds of $1.2m/s$ (Simulation).

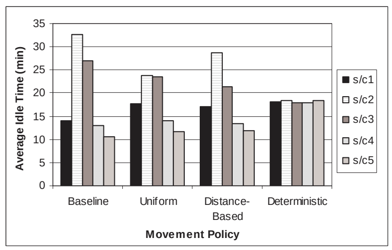


Figure 5: Average idle time at a speeds of $1.2m/s$ (Analysis).

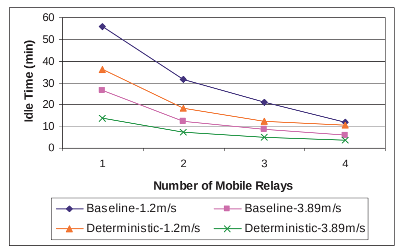


Figure 6: Maximum idle time among all service centers.

the *deterministic* policy with two and three MRs is better than that under the *baseline* policy with three and four MRs respectively. Moreover, the maximum idle time under the *deterministic* policy with two MRs is very close to that under the *baseline* policy with four MRs. It can be therefore concluded that the movement policy plays an important role in influencing the performance of the system. Another thing that we conclude from Figure 6 is that the effect of movement policy and the speed of movement degrades as the number of MRs increases.

We evaluated the end-to-end delay using the two approximations proposed in Section 4, namely, CONV and DPA. Figure 7 shows, as we expected in Section 4, that the CONV approach overestimates the probability distribution function and therefore underestimates the average end-to-end delay. Figure 8 show the average end-to-end delay along all the data paths in our case-study (Figure 1, Table 1, and Table 2) obtained using DPA, CONV, and simulation at a speed of $1.2m/s$. The results shown in Figure 8 indicate that the DPA approach accurately predicts the end-to-end delay, but the CONV approach underestimates the average end-to-end delay by up to 66%.

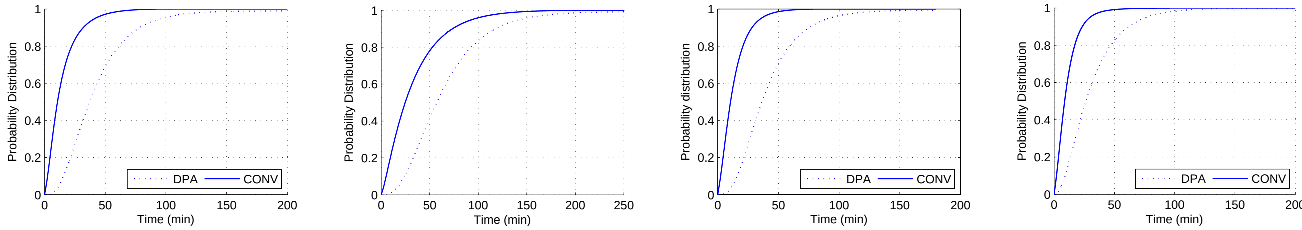
Figure 9 shows σ_i for fragments 1 through 5 obtained using equation (39) as well as simulation for our case study in Figure 1, using the parameters in Table 1, the distances in Table 2, and the following additional parameters:

- $\rho_i = 1pkt/min = 4.8bit/sec\ \forall\ i$, i.e., based on a packet size of 36 bytes.
- $R = 38.4kbps$.
- Two MRs that move at a speed of $3.89m/s$.
- S_i's are as shown in Figure 1.

Figure 9 proves the accuracy of the proposed algorithm as it shows a maximum error of only 10%, compared to simulation, in estimating the average amount of buffered data in a fragment (the average load). Figure 10 shows the new calculated average sojourns times. Note that the $1min$ that was assumed turns our to be much longer than the required time. We evaluated the end-to-end delay using the new sojourn times shown in Figure 10. Therefore, by properly engineering the sojourn time, the average end-to-end delay was reduced, as shown in Figure 11, by about 28%.

7. CONCLUSIONS

A new form of network disconnection called *Fragmented Wireless Sensor Network* (*FWSN*) was addressed in this paper. We proposed the use of resource rich mobile agents that move in the field and operate as data relays between fragments to eventually deliver data to the base station. A mathematical model based on modeling the network and the mobility of mobile relays as a closed queueing network was presented and used to evaluate the performance of a FWSN. The closed queueing network model was developed to capture a number of parameters including number and speed of MRs as well as the movement policy. Using steady state



(a) Path {1,5,3} at $1.2m/s$ (b) Path {2,5,3} at $1.2m/s$ (c) Path {4,5,3} at $1.2m/s$ (d) Path {5,3} at $1.2m/s$

Figure 7: End-to-end delay distribution for all paths in the case-study of Figure 1 and Tables 1 and 2.

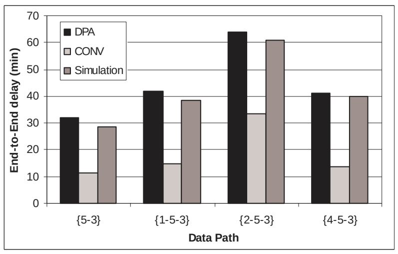


Figure 8: The average *end-to-end* delay at $1.2m/s$

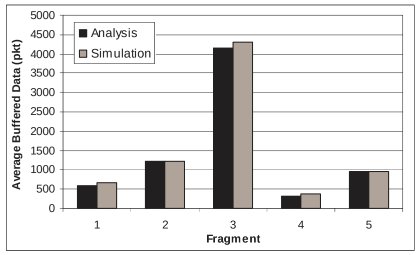


Figure 9: Average buffered data at fragments $1-5$.

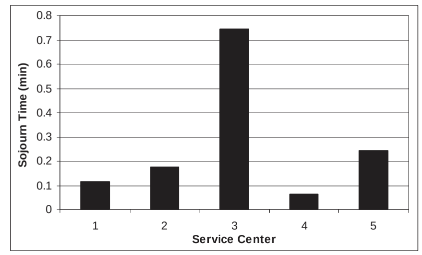


Figure 10: New calculated sojourn times.

probabilities from the model, we then evaluated the distribution of the delay to deliver data between two fragments as well as the end-to-end delay. Moreover, we studied the issue of engineering the sojourn time, i.e., the amount of time that an MR needs to spend at a service center to relay data. The results show that our model accurately evaluates the *fragment-to-fragment* and *fragment-to-sink* delays. It also suggests that enhancing the movement policy might lead to a better performance than just adding more MRs, which implies that the movement policy has a significant influence on the system performance. We can also conclude from the results that engineering the sojourn times is so important as it significantly affects the end-to-end delay.

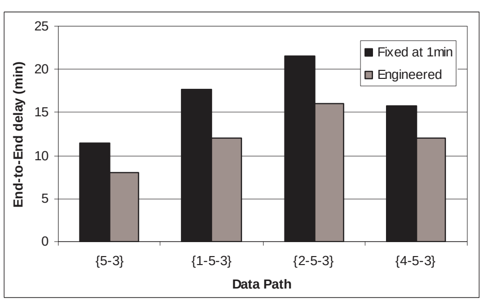


Figure 11: End-to-End delay before and after engineering the sojourn times.

8. REFERENCES

[1] E. Ekici, Y. Gu, and D. Bozdag. Mobility based communication in wireless sensor networks. *IEEE Communications Magazine*, July 2006.

[2] D. Chae et al. Power saving mobility protocol for sensor network. *WSTFEUS*, 2004.

[3] R. Shah et al. Data mules: Modeling a three-tier architecture for sparse sensor networks. *SNPA'03*.

[4] S. Jain et al. Exploiting mobility for energy efficient data collection in wireless sensor networks. *Mobile Networks and Applications*, 2006.

[5] A. Kansal, A. Somasundara, et al. Intelligent fluid infrastructure for embedded networks. *MobiSys*, 2004.

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

[7] I. Chatzigiannakis et al. Sink mobility protocols for data collection in wireless sensor networks. *MobiWac*, 2006.

[8] J. Luo, J. Panchard, et al. Mobiroute: Routing towards a mobile sink for improving lifetime in sensor networks. *DCOSS*, June 2006.

[9] K. Akkaya et al. Sink repositioning for enhanced performance in wireless sensor networks. *Computer Networks*, pages 512–534, 2005.

[10] W. Wang, V. Srinivasan, and K. Chua. Using mobile relays to prolong the lifetime of wireless sensor networks. In *MobiCom'05*.

[11] H. Jun et al. Trading latency for energy in densely deployed wireless ad hoc networks using message ferrying. *Ad Hoc Networks*, 2007.

[12] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. *MobiHoc '04*, pages 187–198, 2004.

[13] W. Gordon and G. Newell. Closed queuing systems with exponential servers. *Operations Research*, 15(2).

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

[15] P. Levis. Tossim: Accurate and scalable simulation of entire tinyos applications. *SenSys*, 2003.

[16] M. Demmer et al. Tython: a dynamic simulation environment for sensor networks. Technical report, EECS Dept., University of California, Berkeley, 2005.