Analytic, Simulation, and Empirical Evaluation of Delay/Fault-Tolerant Mobile Sensor Networks

Hongyi Wu, Yu Wang, Ha Dang, and Feng Lin

***Abstract*— The *Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN)* has been proposed recently for pervasive information gathering. DFT-MSN distinguishes itself from conventional sensor networks by several unique characteristics such as sensor mobility, loose connectivity, and delay/fault tolerability. This paper focuses on the performance evaluation of DFT-MSN. We first introduce a queuing model by using Jackson network theory. While the queuing model is based on a few simplification assumptions for analytic tractability, it provides insights into the queuing behavior of the mobile sensors in DFT-MSN. Extensive simulations are performed under realistic environment and assumptions. Our simulation results show that the dynamic DFT-MSN data delivery scheme achieves the highest message delivery ratio with acceptable delay and transmission overhead, compared with simple schemes such as flooding and direct transmission or other approaches in the literature such as Zebranet. We have also implemented a DFT-MSN testbed by deploying Crossbow motes for noise level monitoring in our university library. Though in a small scale, the testbed demonstrates the feasibility of DFT-MSN and provides guidance for future large scale deployment.**

***Index Terms*— Delay fault tolerant mobile sensor network, delivery delay, delivery probability, DFT-MSN, pervasive information gathering, analysis, queuing theory, transmission overhead.**

I. Introduction

The recent developments in sensor technology has made it possible to gather massive information through a low-cost distributed embedded sensor system. The mainstream approach is to densely deploy a large number of small and inexpensive sensor nodes with low power, short range radio to form a connected wireless mesh network. The sensors in the network collaborate together to acquire the target data and transmit them to the sink nodes [1]. This approach, however, may not work effectively in certain application scenarios, such as flu virus tracking, where the goal is to collect data of flu virus in the area with high human activities in order to monitor and prevent the explosion of devastating flu, or air quality monitoring for tracking the average toxic gas taken by people everyday. These applications share several unique characteristics. First, the data gathering is *human-oriented*. More specifically, while samples can be collected at strategic locations for flu virus tracking or air quality monitoring, the most accurate and effective measurement shall be taken at the people, making it a natural approach to deploy wearable sensing units that closely adapt to human activities. Second, we observe that delay and faults are usually tolerable in such applications, which aim at gathering massive information from a statistic perspective and to update the information base periodically. In addition, this information gathering should be transparent, without any interference on people's daily lives. For instance, a person should not be asked to take special actions (e.g., to move to a specific location) to facilitate information acquisition and delivery.

A *Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN)* has been proposed in [2], [3] for pervasive information gathering in the aforementioned applications. The DFT-MSN consists of two types of nodes, i.e., the wearable sensor nodes and the high-end sink nodes. The former are attached to people, gathering target information and forming a loosely connected mobile sensor network for information delivery (see Fig. 1 for mobile sensors S_1 to S_{10} scattered in the field, where only S_1 and S_2, S_7 and S_8, and S_5 and HES_2 can communicate with each other at this moment). In addition, a number of high-end nodes (e.g., mobile phones or personal digital assistants with sensor interfaces) are either deployed at strategic locations with high visiting probability or carried by a subset of people, serving as the *sinks* to receive data from wearable sensors and forward them to access points of the backbone network (see HES_1 and HES_2 in Fig. 1).

Since the connectivity between the *mobile* sensors in DFT-MSN is poor, it is difficult to form a well connected mesh network for data transmission. Consequently, the data delivery protocols for conventional sensor network relying on end-to-end connections simply fail in DFT-MSN. In an opportunistic network like DFT-MSN, replication is necessary for data delivery in order to achieve certain success ratio [3]. Clearly, replication also increases the transmission overhead. Thus an effective data delivery scheme for DFT-MSN must deal with the trade-off between data delivery ratio/delay and overhead. With this consideration in mind, a dynamic DFT-MSN data delivery scheme has been proposed in [3], whose basic idea is to dynamically tune the message redundance level with the objective of achieving the desired delivery ratio with minimum overhead. As to be discussed in Sec. II-A, it consists of two key components for data transmission and queue management, respectively. The former makes decision on when and where to transmit data messages based on the *delivery probability*, which signifies the likelihood that a sensor can deliver data messages to the sink. The latter decides which messages to transmit or drop based on *fault tolerance*, which indicates the importance of the messages. In this paper we carry out a thorough performance evaluation of this DFT-MSN data delivery

Manuscript received January 19, 2006; revised November 29, 2006; accepted January 14, 2007. The associate editor coordinating the review of this paper and approving it for publication was X. Shen.

The authors are with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, P.O. Box 44330, LA 70504 (e-mail: {yxw1516, wu, ha}@cacs.louisiana.edu).

Digital Object Identifier 10.1109/TWC.2007.06046.

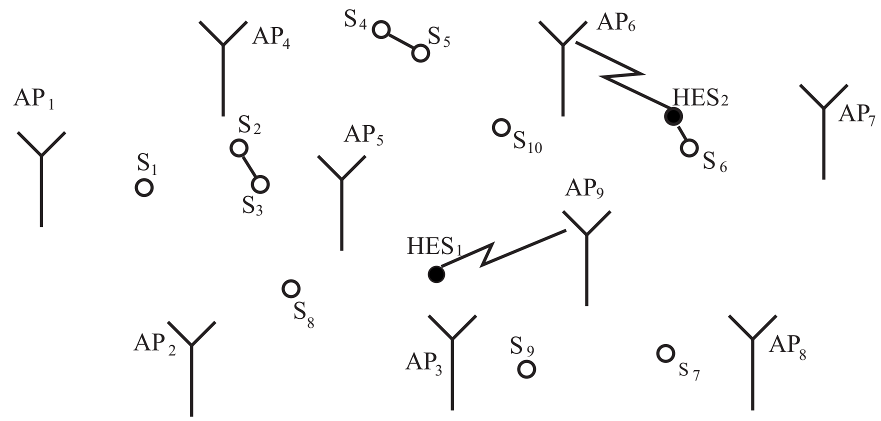


Fig. 1. An overview of the delay/fault-tolerant mobile sensor network. S_1-S_{10}: sensors; HES_1-HES_2: high end sensors (sinks); AP_1-AP_9: access points of backbone network.

scheme, via analysis, simulations, and testbed experiments. In Sec. II we give an overview of the DFT-MSN data delivery scheme and discuss related work in the literature. In Sec. III, we introduce a queuing model for DFT-MSN by using Jackson network theory. While the queuing model is based on a few simplification assumptions for analytic tractability, it provides insights into the queuing behavior of the mobile sensors in DFT-MSN. Extensive simulations are performed under realistic environment and accurate assumptions. The simulations are discussed in Sec. IV. Our simulation results show that the dynamic DFT-MSN data delivery scheme achieves the highest message delivery ratio with acceptable delay and transmission overhead, compared with simple schemes such as flooding and direct transmission or other approaches in the literature such as Zebranet. We have also implemented a DFT-MSN testbed by deploying Crossbow motes for noise level monitoring in our university library. We present our testbed implementation and experimental results in Sec. V. Though in a small scale, the testbed demonstrates the feasibility of DFT-MSN and provides guidance for future large scale deployment. Finally, Sec. VI concludes the paper.

II. Background and Related Work

In this section we first give an overview of the dynamic DFT-MSN data delivery scheme. Then we discuss related work in the literature.

A. An Overview of The Dynamic DFT-MSN Data Delivery Scheme

The dynamic DFT-MSN data delivery scheme is elaborated below. We first discuss two important parameters, namely, the nodal delivery probability and the message fault tolerance, which are keys to enable our proposed approach for data delivery in DFT-MSN. Then, we introduce the queue management and data transmission schemes.

1) DFT-MSN Parameters.: The proposed data delivery scheme for DFT-MSN is based on the nodal delivery probability and the message fault tolerance, as discussed below separately.

(1) Nodal Delivery Probability. The decision on data transmission is made based on *delivery probability*, which indicates the likelihood that a sensor can deliver data messages to the sink. Let ξ_i denote the delivery probability of a sensor i. ξ_i is initialized with zero and updated upon an event of either message transmission or timer expiration. More specifically, the sensor maintains a timer. If there is no message transmission within an interval of Δ, the timer expires, generating a timeout event. The timer expiration indicates that the sensor couldn't transmit any data messages during Δ, and thus its delivery probability should be reduced. Whenever sensor i transmits a data message to another node k, ξ_i should be updated to reflect its current ability in delivering data messages to the sinks. Note that since end-to-end acknowledgement is not employed in DFT-MSN due to its low connectivity, sensor i doesn't know whether the message transmitted to node k will eventually reach the sink or not. Therefore, it estimates the probability of delivering the message to the sink by the delivery probability of node k, i.e., ξ_k. More specifically, ξ_i is updated as follows,

$$\xi_i = \begin{cases} (1-\alpha)[\xi_i] + \alpha\xi_k, & Transmission \\ \\ (1-\alpha)[\xi_i], & Timeout, \end{cases} \quad (1)$$

where $[\xi_i]$ is the delivery probability of sensor i before it is updated, and $0 \leq \alpha \leq 1$ is a constant employed to keep partial memory of historic status. If k is the sink, $\xi_k = 1$, because the message is already delivered to the sink successfully. Otherwise, $\xi_k < 1$. Clearly, ξ_i is always between 0 and 1.

(2) Message Fault Tolerance. DFT-MSN is a store-and-forward network. In a typical store-and-forward network, the packets are deleted from the buffer after they are transmitted to the next hop successfully. By using the proposed dynamic data delivery scheme, however, the sensor in DFT-MSN may still keep a copy of the message after its transmission to other sensors. Therefore, multiple copies of the message may be created and maintained by different sensors in the network, resulting in redundancy. The fault tolerance is introduced to represent the amount of redundancy and to indicate the importance of a given message. We assume that each message carries a field that keeps its fault tolerance. Let $\mathcal{F}_i^j$ denote the fault tolerance of message j in the queue of sensor i.

Here the fault tolerance of a message is defined to be the probability that at least one copy of the message is delivered to the sink by other sensors in the network. When a message is generated, its fault tolerance is initialized to be zero. Let's consider a sensor i, which is multicasting a data message j to Z nearby sensors, denoted by $\Xi = \{\psi_z \mid 1 \leq z \leq Z\}$. The multicast transmission essentially creates totally $Z+1$ copies. An appropriate fault tolerance value needs to be assigned to each of them. More specifically, the message transmitted to sensor ψ_z is associated with a fault tolerance of $\mathcal{F}^j_{\psi_z}$,

$$\mathcal{F}^j_{\psi_z} = 1 - (1 - [\mathcal{F}^j_i])(1 - \xi_i) \prod_{m=1,\ m \neq z}^{Z} (1 - \xi_{\psi_m}), \quad (2)$$

and the fault tolerance of the message at sensor i is updated as

$$\mathcal{F}^j_i = 1 - (1 - [\mathcal{F}^j_i]) \prod_{m=1}^{Z} (1 - \xi_{\psi_m}), \quad (3)$$

where $[\mathcal{F}^j_i]$ is the fault tolerance of message j at sensor i before multicasting. The above process repeats at each time when message j is transmitted to another sensor node. In general, the more times a message has been forwarded, the more copies of the message are created, thus increasing its delivery probability. As a result, it is associated with a larger fault tolerance.

2) *DFT-MSN Data Delivery.:* The proposed DFT-MSN data delivery scheme consists of two key components for queue management and data transmission, discussed below.

(1) Queue Management. Each sensor has a data queue that contains data messages ready for transmission. The data messages of a sensor come from three sources. (a) After the sensor acquires data from its sensing unit, it creates a data message, which is inserted into its data queue; (b) When the sensor receives a data message from other sensors, it inserts the message into its data queue; (c) After the sensor sends out a data message to a non-sink sensor node, it may also insert the message into its own data queue again, because the message is not guaranteed to be delivered to the sink. The queue management is to appropriately sort the data messages in the queue, to determine which data message to be sent when the sensor meets another sensor, and to determine which data message to be dropped when the queue is full.

Our proposed queue management scheme is based on the fault tolerance, which signifies how important the messages are. The message with smaller fault tolerance is more important and should be transmitted with a higher priority. This is done by sorting the messages in the queue with an increasing order of their fault tolerance. Message with the smallest fault tolerance is always at the top of the queue and transmitted first. A message is dropped at the following two occasions. First, if the queue is full when a message arrives, its fault tolerance is compared with the message at the end of the queue. If the new message has a larger fault tolerance, it is dropped. Otherwise, the message at the end of the queue is dropped, and the new message is inserted into the queue at appropriate position according to its fault tolerance. Second, if the fault tolerance of a message is larger than a threshold, the message is dropped, even if the queue is not full. This is to reduce transmission overhead, given that the message will be delivered to the sinks with a high probability by other sensors in the network. A special example is the message which has been transmitted to the sink. It will be dropped immediately because it has the highest fault tolerance of 1.

With the above queue management scheme, a sensor can determine the available buffer space in its queue for future arrival messages with a given fault tolerance. Assume a sensor has a total queue space for at most K messages. Let k^m_i denote the number of messages with a fault tolerance level of m in the queue of sensor i. Then, the available buffer space at sensor i for new messages with fault tolerance x is $B_i(x) = K - \int_0^x k^m_i dm$. If $B_i(x) = 0$, any arrival message with a fault tolerance of x or higher will be dropped. Note that, however, even when the queue is filled by K messages and becomes full, $B_i(x)$ may still be larger than 0, for a small x (i.e., for messages with a low fault tolerance). Buffer space information is important to make decision on data transmission, as discussed next.

(2) Data Transmission. Data transmission decision is made based on the delivery probability. Without loss of generality, we consider a sensor i, which has a message j at the top of its data queue ready for transmission and is moving into the communication range of a set of Z' sensors. Sensor i first learns their delivery probabilities and available buffer spaces via simple handshaking messages. Let $\Xi' = \{\psi_z \mid 1 \leq z \leq Z'\}$ designate the Z' sensors, sorted by a decreasing order of their delivery probabilities. Sensor i multicasts its message j to a subset of the Z' sensors, denoted by Φ, which have higher delivery probabilities (i.e., $\xi_i < \xi_{\psi_z}$), so that the total delivery probability of Message j (i.e., $1-(1-\mathcal{F}^j_i) \prod_{m \in \Phi}(1-\xi_m)$) is just enough to reach a given threshold γ. In order to avoid unnecessary message drops due to buffer overflow at the receiver, sensor i checks the available buffer space of its neighboring nodes for message j (i.e., $B_{\psi_z}(\mathcal{F}^j_i)$) before data transmission.

B. Related Work

The Delay-Tolerant Network (DTN) is an occasionally connected network that may suffer from frequent partitions and that may be composed of more than one divergent set of protocol families [4]. DTN originally aimed to provide communication for the *Interplanetary Internet*, which focused primarily on the deep space communication in high-delay environments and the inter-operability between different networks deployed in extreme environments lacking continuous connectivity [4], [5]. An overall architecture of DTN has been proposed in [5], and it operates as an overlay above the transport layer to provide services such as in-network data storage and retransmission, interoperable naming, authenticated forwarding, and a coarse-grained class of service. In [6], Burleigh *et al.* identify several fundamental principles that would underlie a DTN architecture and propose a new end-to-end overlay network protocol called Bundling. In [7], Fall *et al.* investigate the custody transfer mechanism to ensure reliable hop-by-hop data transmission, thus enhancing the reliability of DTN.

DTN technology has been recently introduced into wireless sensor networks. Its pertinent work can be classified into the following three categories, according to their differences in nodal mobility. *(1) Network with Static Sensors*. The first type of DTN-based sensor networks are static. Due to a limited transmission range and battery power, the sensors are loosely connected to each other and may be isolated from the network frequently. For example, the Ad hoc Seismic Array developed at the Center for Embedded Networked Sensing (CENS) employs seismic stations (i.e., sensors) with large storage space and enables store and forward of bundles with custody transfer between intermediate hops [8]. In [9], wireless sensor networks are deployed for habitat monitoring, where the sensor network is accessible and controllable by the users through the Internet. The SeNDT (Sensor Networking with Delay Tolerance) project targets at developing a proof-of-concept sensor network for lake water quality monitoring, where the radio connecting sensors are mostly turned off to save power, thus forming a loosely connected DTN network [10]. DTN/SN focuses on the deployment of sensor networks that are inter-operable with the Internet protocols [11]. [12] proposes to employ the DTN architecture to mitigate communication interruptions and provide reliable data communication across heterogeneous, failure-prone networks. *(2) Network with Managed Mobile Nodes*. In the second category, mobility is introduced to a few special nodes to improve network connectivity. For example, the Data Mule approach is proposed in [13] to collect sensor data in sparse sensor networks, where a mobile entity called data mule receives data from the nearby sensors, temporarily store them, and drops off the data to the access points. This approach can substantially save the energy consumption of the sensors as they only transmit over a short range, and at the same time enhance the serving range of the sensor network. *(3) Network with Mobile Sensors*. While all of the above delay-tolerant sensor networks center at static sensor nodes, there are two examples in the literature that are based on mobile sensors. ZebraNet [14] employs the mobile sensors to support wildlife tracking for biology research. The ZebraNet project targets at building a position-aware and power-aware wireless communication system. A history-based approach is proposed for routing, which is in fact a special case of the dynamic DFT-MSN data delivery scheme. More specifically, each node maintains its past success rate of transmitting data packets to the base station directly. When a sensor meets another sensor, the former transmits data packets to the latter if the latter has a higher success rate. This simple approach, however, doesn't guarantee any desired data delivery ratio. The Shared Wireless Info-Station (SWIM) system is proposed in [15], [16] for gathering biological information of radio-tagged whales. It is assumed in SWIM that the sensor nodes move randomly and thus every node has the same chance to meet the sink. A sensor node distributes a number of copies of a data packet to other nodes so as to reach the desired data delivery probability. This approach is equivalent to the optimal flooding scheme discussed in [3]. In many practical applications, however, different nodes may have different probabilities to reach the sink, and thus SWIM may not work efficiently. Worst yet, some nodes may never meet the sink, resulting in failure of data delivery in SWIM. The pioneering work of ZebraNet and SWIM has motivated our research on mobile sensor networks. At the same time, we observe that the data transmission schemes employed in ZebraNet and SWIM are based on direct contact probability between sensor and sink, and thus inefficient. In addition, several erasure coding based data forwarding schemes have been proposed in [17], [18], where the erasure coding parameters are tuned based on the nodal delivery probability.

DTN technology has also been employed in mobile ad hoc networks. A Context-Aware Routing (CAR) algorithm is proposed in [19] to provide asynchronous communication in partially-connected mobile ad hoc networks. In [20], the authors consider highly mobile nodes that are interconnected via wireless links. Such a network can be used as a transit network to connect other disjoint ad-hoc networks. Five opportunistic forwarding schemes are studied and compared therein. [21] proposes a routing protocol for intermittently-connected mobile ad hoc networks, based on direct contact probability between mobile nodes. A Message Ferrying (MF) approach is proposed in [22] for sparse mobile ad hoc networks, where network partitions can last for a significant period. The basic idea is to introduce deterministic nodal movement and exploit such non-randomness to help data delivery.

III. A Queuing Analytic Model

In this section we analyze the performance of DFT-MSN by using a queuing model based on Jackson network. While it is desirable to accurately analyze the data delivery scheme discussed in Sec. II-A, this is not practical, given its complexity in data transmission and queue management. Thus, for analytic tractability, we employ a simplified network model in this section. More specifically, we make two simplification assumptions in delivery probability and messages fault tolerance, respectively. First, the delivery probability (i.e., ξ_i of sensor i) is assumed to be the probability that sensor i meets the sink nodes. Second, after sensor i sends a message, the fault tolerance of the message copy at sensor i is set to be 1 (i.e., it will be removed from the data queue of sensor i).

We consider a cell partitioned network as described in [23]. Herein, the sensing area is divided into C non-overlapping cells and time is slotted so that a sensor remains in its current cell for a time slot and potentially moves to a new cell at the end of that slot. During a time slot, a sensor can transmit at most one data packet to another sensor in the same cell. If there are more than two sensors in a cell, multiple messages may be transmitted. But only the sensor with the highest delivery probability will receive data messages from other sensors. Each sensor has a data queue that can contain maximum K messages. A sensor i generates data and inserts data messages into the queue at a rate of r_i.

Total N sensors and S sink nodes are uniformly distributed in the C cells initially. Each sensor is associated with a home cell and moves randomly according to power law distribution [24]. More specifically, the probability that a sensor visits a cell x is

$$P_i(x) = k_i(\frac{1}{d_i(x)})^{\beta}, \tag{4}$$

where k_i and β are the constant and the exponent of the power-law distribution, respectively. $d_i(x)$ denotes the distance from cell x to the home cell of sensor i. Noting that, the probability to find sensor i in the entire sensing area is 1, i.e.,

$$\sum_{x=1}^{C} P_i(x) = 1. \tag{5}$$

Thus we have

$$k_i = \frac{1}{\sum_{x=1}^{C}(\frac{1}{d_i(x)})^{\beta}}. \tag{6}$$

We first discuss the queueing behavior of each individual sensor. Assume that the data processing time is negligible. Then the service time of a sensor equals the time to transmit a data message. A message transmission occurs in two situations. First, the sensor sends a data message directly to a sink if they collocate in the same cell. Denote Φ_s to be the set of cells, where each cell contains at least one sink node. The probability that sensor i will meet a sink is:

$$P_i(\Phi_s) = \sum_{S\in\Phi_s} k_i(\frac{1}{d_i(S)})^{\beta}. \tag{7}$$

Under the assumptions given above, $\xi_i = P_i(\Phi_s)$. Second, the sensor will forward a data message to another sensor if they are in the same cell and the latter has the highest delivery probability in the cell. Consider sensor i in cell x, the probability that there is at least another sensor with higher delivery probability in that cell is $1 - \prod_{\forall j}^{P_j(\Phi_s)>P_i(\Phi_s)}(1 - P_j(x))$. Thus the average probability that sensor i will transmit a data message in a cell is

$$p_i = P_i(\Phi_s) + \sum_{\substack{x=1\\ x\notin\Phi_s}}^{C} [1 - \prod_{\forall j}^{P_j(\Phi_s)>P_i(\Phi_s)} (1 - P_j(x))] P_i(x). \tag{8}$$

Let X denote a random variable representing the number of moves required for sensor i to successfully send out a data message. X thus is geometrically distributed with the parameter p_i. Its mass distribution function is

$$f_X\{x = n\} = (1 - p_i)^{n-1} p_i, \tag{9}$$

and its cumulative probability distributed function is given by

$$F_X\{x \le n\} = \sum_{i=1}^{n} (1 - p_i)^{n-1} p_i. \tag{10}$$

Note that geometric distribution is the discrete case of exponential distribution. Hence, the service time of sensor i can be approximated as an exponential random variable with parameter $\mu_i = -ln(1 - p_i)$, i.e.,

$$F_X\{x \le n\} = 1 - e^{-\mu_i n}. \tag{11}$$

We now discuss how to obtain the arrival rate for each individual queue. As mentioned earlier, the arrival rate is the combination of the self-generated message rate and the data rate coming from other sensors. Suppose that the self-generated messages at sensor i follow Poisson distribution with parameter r_i. When sensor i meets k other sensors in a particular cell, sensor i will receive data message(s) if it has the highest delivery probability among all sensors in the cell and there is no sink in the cell. Suppose that sensor i is currently in cell x. Let Ω_i denote the set of sensors in the network, which have lower delivery probabilities than sensor i, and $|\Omega_i|$ denote the total number of sensors in Ω_i. Let $\Omega_i^k(j)$ denote a subset of Ω_i with total k sensors, where j is an index ranging from 1 to $\binom{|\Omega_i|}{k}$. $\Omega_i^k(j) = \emptyset$ if $k > |\Omega_i|$. The probability that there are exactly k other sensors with lower delivery probability than that of sensor i and no additional sensors in cell x is

$$A_i^k(x) = \sum_{j=1}^{\binom{|\Omega_i|}{k}} [\prod_{m\in\Omega_i^k(j)} P_m(x)][\prod_{n\neq i, n\notin\Omega_i^k(j)} (1 - P_n(x))]. \tag{12}$$

Given sensor i is in cell x and there is no sink in cell x, the average number of data messages that sensor i receives from other sensors in cell x is

$$B_i(x) = \sum_{k=1}^{|\Omega_i|} k \times A_i^k(x). \tag{13}$$

Since a sensor will not receive any message from other sensors if it is in the range of a sink, the average number of messages that sensor i will receive in a cell is

$$l_i = \sum_{\substack{x=1\\ x\notin\Phi_s}}^{C} B_i(x) P_i(x). \tag{14}$$

Applying the Kleinrock's approximation [25], the combined message arrival at sensor i is Poisson with an average arrival rate of

$$\lambda_i = l_i + r_i. \tag{15}$$

Therefore, each individual queue in the network is an M/M/1/K queue. Assume a message will be dropped if the queue of the receiver is already full. In the equilibrium state, the probability that there are exactly $q \le K$ data messages in queue i is

$$\wp_i^q = (\frac{\lambda_i}{\mu_i})^q \frac{1 - (\frac{\lambda_i}{\mu_i})}{1 - (\frac{\lambda_i}{\mu_i})^{q+1}}, \tag{16}$$

and the average number of data messages in the queue of sensor i is given by:

$$\mathcal{L}_i = \frac{\lambda_i[1 + K(\frac{\lambda_i}{\mu_i})^{K+1} - (K+1)(\frac{\lambda_i}{\mu_i})^K]}{(\mu_i - \lambda_i)(1 - (\frac{\lambda_i}{\mu_i})^{K+1})}. \tag{17}$$

By seeing our system as a network of queues and applying the theorem of Jackson network, we arrive at the average number of data messages in the system

$$\mathcal{L} = \sum_{i=1}^{N} \mathcal{L}_i. \tag{18}$$

To estimate the average time a message spending in the system, we need to calculate the self-generating rate that actually enter the queues. Note that when queue i is already full, it no longer accepts any data messages. Hence, the effective self-generating rate (of the messages that actually enter queue i) is $r_i(1 - \wp_i^K)$. Therefore, the average delay that a data message stays in the system is

$$\mathcal{T} = \frac{L}{\sum_{i=1}^{N} r_i(1 - \wp_i^K)}. \tag{19}$$

Since a sensor sends one message only in a time slot and a message is delivered successfully only when it is received by the sink, the network-wide message delivery ratio is

$$\mathcal{D} = \frac{\sum_{i=1}^{N} P_i(\Phi_s)}{\sum_{i=1}^{N} r_i}. \quad (20)$$

To validate the analytic model, we have also performed simple stand-alone simulations based on the simplified assumptions discussed above. In the simulation, we divide the sensing area into 10×10 cells with the number of mobile sensors ranging from 2 to 20. Initially the sensors are randomly and uniformly distributed over the sensing area. Each sensor is assigned a home cell that is the first cell it lands on. These sensors roam around the sensing area within the simulation time of 50000 time units. The movement of the sensors follows power law distribution with the exponent $\alpha = 1.5$. We assume all sensors have the same queue size $K = 200$ and the same self-generating message rate $r = 0.03$. For each simulation setup, we run the simulation 40 times and average the collected results. The results of average queue size, queuing delay, and message delivery ratio are shown in Figs. 2. As can be seen from the figures, the simulation and analytic results are close and show the same trend. With more sensors being deployed, a node has better chance to meet other sensors for data transmission, resulting in smaller average queue size and faster delivery, as depicted in Fig. 2(a) and Fig. 2(b), respectively. Meanwhile, both analytic and simulation results show that delivery ratio keeps constant with the increase of node density, as shown in Fig. 2(c). This is reasonable because the bottleneck is now at the sink nodes. More specifically, although a higher node density increases the chance that a node transmits to other nodes, it also fills the queue of the receiver faster, incurring higher message dropping rate at the receiver, especially at the nodes near the sinks. This observation can be also explained by Equation (20), where delivery ratio is approximated as the total sink service rate divided by the total message generating rate. Apparently, changing sensor node density only will not affect this ratio from a statistic perspective. The difference between simulation and analysis is mainly due to the approximation of message arrival and service rates for analytic tractability.

IV. Simulation Results

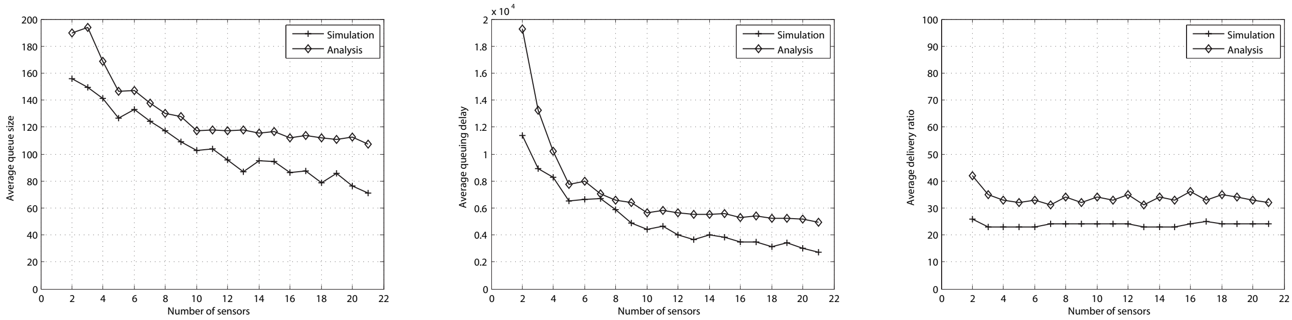
The above analytic model is based on a few simplification assumptions for analytic tractability. To evaluate DFT-MSN under realistic environment, we have nullified those simplification assumptions and carried out extensive simulations. The simulation environment and default parameters are described below. Three sink nodes and 100 sensor nodes are randomly deployed in an area of 200×200 m^2. The whole area is divided into 25 non-overlapped cells, each with an area of 40×40 m^2. A sensor node initially resides in its home cell, which is random chosen among the 25 cells in our simulation. It moves within the simulated area according to the following pattern: it continuously moves for a random period of time t_m (between 0 and 60 seconds), and then stays there for another random period of time t_s (between 0 and 120 seconds) before moving again. The nodal speed is randomly chosen between 0 and 5 m/s. Whenever a node reaches the boundary of its cell, it moves out with a probability of 20%, and bounces back with a probability of 80%. After entering a new cell, the sensor repeats the above process. However, if it reaches the boundary to its home cell, it returns to its home cell with a probability of 100%. Each sensor has a maximum transmission range of 10 m and a maximum queue size of 200 messages. The data generation of each sensor follows a Poisson process with an average arrival interval of 100 s. Each data message has 200 bits. The channel bandwidth is 10 $kbps$. The fault tolerance threshold is set to be $\gamma = 0.8$.

TABLE I

Simulation Results with Default Parameters.

	DFT-MSN	ZebraNet	Direct Transmission	Flooding
Delivery ratio (%)	89	55.6	42.6	20.6
Average copies for each message	9.8	2.7	1	1292
Average delay (s)	2811	1700	3307	1180

We have implemented four schemes, i.e., the direct transmission (where the sensor only transmits its own generated data to the sink directly), the simple flooding, the ZebraNet approach (based on historic information) [14], and the proposed DFT-MSN dynamic data delivery approach. Note that results of SWIM are not included here for comparison, because it is under the assumption that all mobile nodes have the same probability to meet the sink nodes. As a result, its performance is much lower than other schemes in our simulated environment. A simple acknowledgement is incorporated in the ZebraNet and DFT-MSN, according to the delete-list scheme discussed in [14]. We first compare the performance of the four schemes under the default parameters. The results are presented in Table I. As we can see, the DFT-MSN approach has the highest delivery ratio of 89%. The flooding approach performs worst in terms of delivery ratio, because it generates too many message copies and thus results in frequent buffer overflow. We also observe that the ZebraNet approach can only deliver messages generated by the nodes close to the sinks, while the messages generated by the nodes far away from the sinks are likely to be dropped, as shown in Fig. 3. This problem stems from its hierarchy level updating policy where a node's hierarchy level is calculated according to the probability that it can reach the sinks directly. For those sensors never reachable to the sink nodes, their data messages are just randomly transmitted around, resulting in very low delivery probability. In contrast, the DFT-MSN approach can deliver most of the messages even if they are far away from the sink nodes. In addition to the delivery ratio, we are also interested in data delivery delay and overhead. As shown in Table I, the direct transmission approach has the lowest overhead (i.e., the average number of transmissions for each message) but longest delay, since a sensor always transmits data messages to the sink directly. On the other hand, the



(a) Average queue size. (b) Average queuing delay. (c) Average message delivery ratio.

Fig. 2. Comparison between analytic and simulation results.

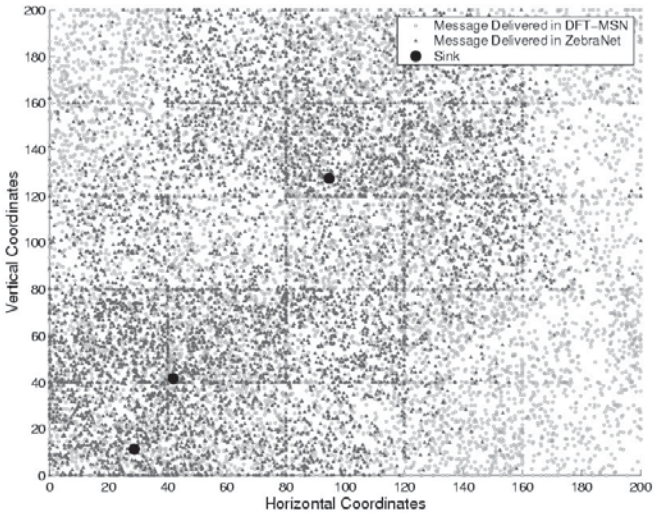


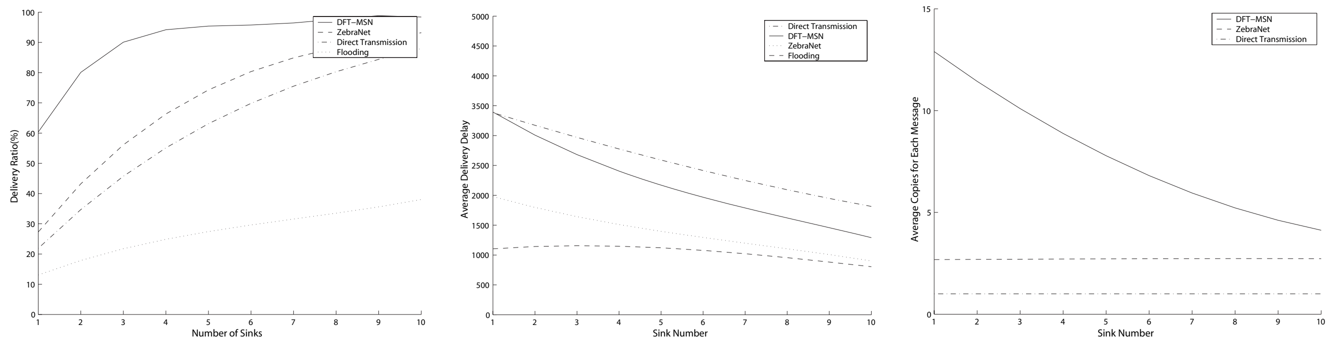
Fig. 3. The comparison of generating possitions of the successfully delivered messages.

flooding approach has highest overhead but shortest delay. In DFT-MSN, the messages generated far away from the sinks need to be relayed many times before they reach the sink nodes, and thus resulting in higher average overhead and longer average delay than the ZebraNet, which can only deliver the messages generated near the sink nodes as discussed above.

We also vary several parameters to observe their impacts on the performance. Fig. 4 compares the performance of different approaches by varying the number of sink nodes. As expected, the proposed DFT-MSN approach always has a higher delivery ratio than other approaches, especially when a small number of sinks are deployed (see Fig. 4(a)). With a large number of sinks, the DFT-MSN approach, the ZebraNet approach, and the direct transmission approach all yield good results. This is reasonable because the sensors then have high probabilities to reach the sink nodes, thus resulting in high delivery ratio close to 100%. Fig. 4(b) demonstrates that the average delay of every approach decreases with more sink nodes deployed in the network. Although the flooding approach has the smallest message delivery delay, its delivery ratio is very low. In addition, as discussed above, since the ZebraNet scheme can only deliver the messages generated by nodes near the sinks, it has a shorter average delivery delay than DFT-MSN approach. Clearly, the direct transmission approach suffers from the longest delay since messages can be delivered only when the source node meets the sink. Energy consumption of the sensor is due mainly to data transmission. Thus, the more duplicated copies generated, the higher the energy consumption. As depicted in Fig. 4(c), the number of message copies in direct transmission is always 1, since a sensor always transmits data messages to the sink directly. The results of flooding are not shown here, because it generates excessive copies which are several orders of magnitude higher than those of other approaches. The duplicated message number decreases in DFT-MSN schemes with the increase of sink node number, largely due to its effective overhead control. Meanwhile, because DFT-MSN delivers many messages generated by nodes far away from the sink nodes, it has a longer average delivery delay than ZebraNet.

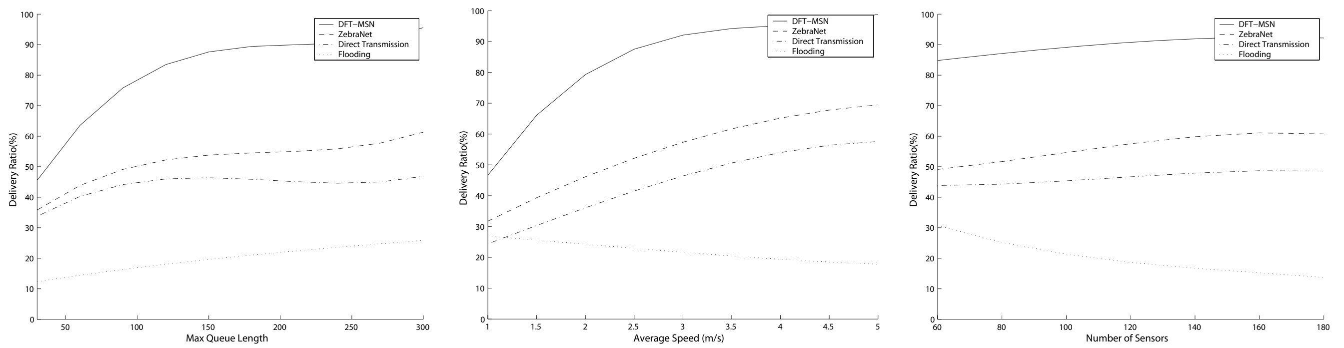
The impact of the maximum queue length is shown in Fig. 5(a). With an increase in maximum queue length, the delivery ratio increases for all approaches because the messages can then stay in the memory for a longer time before they are dropped. It is also noticed that the DFT-MSN approach achieves higher gain than other approaches with the increase of queue size. Fig. 5(b) depicts the impact of nodal moving speed. As the speed increases, the delivery ratios of all approaches rise, except the flooding approach. This is reasonable since the node with a higher speed has a better opportunity to meet other nodes and also has higher probabilities to reach the sink nodes. In the flooding approach, however, the nodes may make more copies for each message, resulting in much more buffer overflows and, consequently, lower delivery ratio. Fig. 5(c) illustrates the impact of node density by varying the total number of sensor nodes in the network. As we can see, the delivery ratio increases slightly in DFT-MSN and ZebraNet because more nodes can help relaying the messages. The node density doesn't have significant impact on direct transmission, because the probability that a sensor meets the sinks remains the same. The delivery ratio of flooding decreases because much more copies are generated for each message and thus more overflows may happen, especially for those nodes close to the sinks.

Though the results are not shown here, we have also analyzed the impacts of the aforementioned parameters on average delay and overhead. With the increase of maximum queue length, the average overhead and delay in the proposed DFT-



(a) Average delivery ratio. (b) Average delay. (c) Average overhead.

Fig. 4. Impact of the number of sink nodes.



(a) Impact of maximum queue length. (b) Impact of the nodal speed. (c) Impact of node density.

Fig. 5. Performance comparison by varying other parameters.

MSN scheme increase, since more messages can reside in the queue before being dropped. Similarly, the average delivery delay in ZebraNet also increases due to the same reason. When increasing the nodal speed, we notice that the transmission overhead decreases in DFT-MSN, but increases in ZebraNet. This is reasonable since in the DFT-MSN approach, the messages can be delivered with fewer hops and duplications when nodal speed is high. However, in the ZebraNet approach, more copies are generated because the source node meets more neighbors before the message is delivered to the sink. With higher nodal density, the ZebraNet approach delivers the messages more aggressively, thus producing more overhead. In contrast, the DFT-MSN scheme has very steady performance, exhibiting desirable scalability.

V. TESTBED IMPLEMENTATION AND EXPERIMENTS

In order to demonstrate DFT-MSN, we have built a testbed by using Xbow MICA2 sensors [26] for monitoring the noise level in the library.

A. Testbed Implementation

A MICA2 sensor node has a 4-MHz, 8-bit Atmel microprocessor and 512 KB of non-volatile flash memory that can be used for data logging. Its radio bandwidth is 38.4 Kbuad. The testbed runs TinyOS 1.1.0 operating system. There are two types of nodes in our testbed, namely mobile sensors and sink node. Their functions and implementations are discussed below.

```
/** Define the message header. **/
typedef struct{
  unit16_t source_address;
  unit16_t seqence_number;
  unit16_t counter;
  float    fault_tolerance;
}Message_Header;

/** Define the message. **/
typedef struct{
  Message_Header message_header;
  unit8_t data[MAX_DATA_SIZE];
}Message;
```

Fig. 6. Message Format.

a) Mobile Sensor.: A mobile sensor has three functions: collecting information and generating data messages, transmitting or relaying data messages, and recording necessary information for performance evaluation. In order to reduce energy consumption, a sensor only wakes up periodically for data acquisition and transmission. We employ two timers, namely sensing-timer and transmission-timer, to control sensor's activities. The former controls the sensor's sensing activity. When sensing-timer expires, the mobile sensor samples the current noise level and generates a data message. The format of the data message is illustrated in Fig. 6. When a new message is generated, it is assigned a sequence number, which is used, along with source address, to uniquely identify the message. Each message also contains a counter field to record its delivery delay. Once a data message is generated, it is inserted into the data queue, sorted by the fault tolerance of the messages.

The transmission timer controls the data transmission of the sensor. When the transmission timer expires, the sensor contacts its neighboring nodes if there are any. More specifically, it initiates a beacon message, which contains its address, its current delivery probability, and the minimal fault tolerance of its messages, to its neighbor nodes. Upon receiving a beacon message, the neighboring node decides whether to reply or not, based on deliver probability and buffer status. Then a message may be transmitted (depending on the reply) as we have discussed in Sec. II-A. After transmission, the data queues of the sender and receiver are updated accordingly.

For performance evaluation, the sensors also record some parameters, such as the total number of generated data messages, the total number of transmitted messages, and the number of dropped messages due to buffer overflow. These parameters are stored in the EEPROM and refreshed periodically. After experiment is finished, we collect all mobile sensors and place them in the transmission range of the sink node, which will in turn retrieve the logged data for calculating the delivery ratio and overhead.

b) Sink Node.: The sink node consists of one MICA2 sensor connected to a laptop via UART. The MICA2 at the sink is very similar to other sensors. The only difference is that it has no data queue. Upon receiving a data message, it will store the message in a text file on the laptop.

B. Experiment Setup

We have carried out a small scale experiment with six MICA2 nodes attached to students who move in the Dupré Library of our university. As shown in Fig. 7, the mobile sensor nodes are initially scattered in three different areas, i.e., the reading area, the bookshelf area, and the computer service area. Each area has two boundaries, namely movement boundary and communication boundary. The former limits the nodal mobility in each area. The latter indicates the maximum radio transmission range of sensors in each area. The communication boundaries of any two areas partially overlap with each other. Note that, the nodes within transmission range may not always be able to communicate with each other because of the lack of line-of-sight (due to the bookshelves, computers, walls, etc.). Generally, a node only moves within the movement boundary of the area where it is currently located, while periodically it may move out to another area with certain probability. The moving speed of the mobile nodes is around 0-3 m/s. Each sensor has a maximum data queue size of 50 messages. The sensing-timer is set to be 60,000 binary milliseconds (1 binary millisecond equals 1/1024 second), while the transmission-timer is set as 10,000 binary milliseconds. In addition, we set $\alpha = 0.02$ and $\gamma = 0.9$ in this experiment.

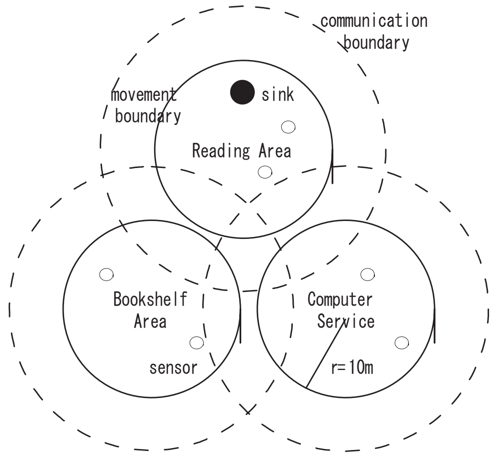


Fig. 7. Experimental Scenario. (The circular boundary is for illustration only. Actual boundary is irregular.)

TABLE II

EXPERIMENTAL RESULTS.

	During T	During t	Total
Generated messages	120	0	120
Transmitted messages	347	288	635
Received messages	87	29	116
Delivery ratio (%)	/	/	96.67
Average delay ($minutes$)	/	/	5.8

C. Experimental Results

We run the experiment for a period of T. Note that, due to the long delay of data delivery, many newly generated data messages are still stored at the intermediate nodes by the end of T. Since the sink node doesn't receive these data messages, we may falsely assume they are lost, resulting in a low delivery ratio. This problem will become negligible when $T \to \infty$. But a large T is not practical for our experiment that relies on the student volunteers to carry sensors. Alternatively, we choose a small T (e.g., $T = 20\ min$). After T, we do not terminate the experiment immediately, but instead, continue it for another period t (e.g., $t = 10\ min$), during which the sensors do not generate new data messages.

The experimental results are summarized in Table II, where "generated messages" refer to the number of new messages generated by the mobile sensors; "transmitted messages" refer to the total number of message copies transmitted (i.e., including duplicate messages); "received messages" refer to the number of unique message copies received by the sink (i.e., excluding duplicate messages). As we have observed from our experiments, the proposed DFT-MSN data delivery scheme is efficient, with a total delivery ratio higher than 96% and average delay around 5.8 minutes. These results have been verified by multiple similar experiments with running time from 30 minutes to 2 hours. The high overhead is reasonable, given the very low network connectivity and nodal delivery probability. In addition, we observe a few messages (including duplicate message copies) dropped during the experiments due to buffer overflow (e.g., about 10 messages dropped during a 30-minute period). Based on the small-scale testbed, a university-wide large-scale experiment will be carried out in the future.

VI. Conclusion

This paper focuses on the performance evaluation of the *Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN)* proposed for pervasive information gathering. DFT-MSN has several unique characteristics such as sensor mobility, loose connectivity, fault tolerability, delay tolerability, and buffer limit. We have established a queuing model for DFT-MSN by using Jackson network theory. While the queuing model is based on a few simplification assumptions for analytic tractability, it provides insights into the queuing behavior of the mobile sensors in DFT-MSN. Under realistic environment and assumptions, we have carried out extensive simulations. Our simulation results show that DFT-MSN achieves the higher message delivery ratio with acceptable delay and transmission overhead, compared with simple schemes such as flooding and direct transmission or other approaches in the literature such as Zebranet. We have also implemented a DFT-MSN testbed by deploying Crossbow motes for noise level monitoring in our university library. Though in a small scale, the testbed demonstrates the feasibility of DFT-MSN and provides guidance for future large scale deployment.

References

[1] I. Akyildiz, W. Su, and Y. Sankarasubramaniam, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, 2002.

[2] Y. Wang, F. Lin, and H. Wu, "Poster: efficient data transmission in delay fault tolerant mobile sensor networks (DFT-MSN)," in *Proc. IEEE International Conference on Network Protocols (ICNP 2005).*

[3] Y. Wang and H. Wu, "DFT-MSN: the delay fault tolerant mobile sensor network for pervasive information gathering," in *Proc. Twenty-Fifth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2006)*, to appear.

[4] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay tolerant network architecture," draft-irtf-dtnrg-arch-02.txt, 2004.

[5] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proc. 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, pp. 27–34, 2003.

[6] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking–an approach to interplanetary Internet," *IEEE Commun. Mag.*, vol. 41, no. 6, pp. 128–136, 2003.

[7] K. Fall, W. Hong, and S. Madden, "Custody transfer for reliable delivery in delay tolerant networks." Technical Report, Intel Research, Berkeley, IRB-TR-03-030, 2003.

[8] http://www.cens.ucla.edu/.

[9] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pp. 88–97, 2002.

[10] http://down.dsg.cs.tcd.ie/sendt/.

[11] http://www.sics.se/cna/dtnsn/index.html.

[12] M. Ho and K. Fall, "Poster: delay tolerant networking for sensor networks," in *Proc. IEEE Conference on Sensor and Ad Hoc Communications and Networks*, 2004.

[13] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: modeling a three-tier architecture for sparse sensor networks," in *Proc. First International Workshop on Sensor Network Protocols and Applications*, pp. 30–41, 2003.

[14] http://www.princeton.edu/ mrm/zebranet.html.

[15] T. Small and Z. J. Haas, "The shared wireless infostation model–a new ad hoc networking paradigm (or where there is a whale, there is a way)," in *Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC 2003).*

[16] T. Small and Z. J. Haas, "Resource and performance tradeoffs in delay-tolerant wireless networks," in *Proc. ACM SIGCOMM'05 Workshop on Delay Tolerant Networking and Related Topics (WDTN-05).*

[17] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in *Proc. ACM SIGCOMM Workshop on Delay Tolerant Networking*, 2005.

[18] Y. Wang and H. Wu, "Replication-based efficient data delivery scheme (RED) for DFT-MSN," in *Technical Report*, 2005.

[19] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive routing for intermittently connected mobile ad hoc networks," in *Proc. IEEE 6th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*, pp. 1–7, 2005.

[20] J. LeBrun, C.-N. Chuah, and D. Ghosal, "Knowledge based opportunistic forwarding in vehicular wireless ad hoc networks," in *Proc. IEEE Vehicular Technology Conference (VTC) Spring*, pp. 1–5, 2005.

[21] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," in *Proc. First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR 2004).*

[22] W. Zhao, M. Ammar, and E. Zegura, "A Message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proc. 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc)*, pp. 187–198, 2004.

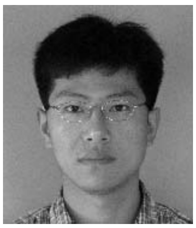
[23] M. J. Neely and E. Modiano, "Capacity and delay tradeoffs for ad-hoc mobile networks," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1917–1937, 2005.

[24] J. Leguay, T. Friedman, and V. Conan, "DTN routing in a mobility pattern space," in *Proc. ACM SIGCOMM'05 Workshop on Delay Tolerant Networking and Related Topics (WDTN 2005).*

[25] L. Kleinrock, *Queueing Systems, Volume 1, 2*. John Wiley and Sons, Inc., 1976.

[26] http://www.xbow.com/.

Yu Wang received his BS and MS degrees from the University of Electronic Science and Technology of China (UESTC) in 2000 and 2003, respectively. He is currently a Ph.D. candidate at the Center for Advanced Computer Studies (CACS), University of Louisiana (UL) at Lafayette. His research interests include wireless mobile ad hoc networks, wireless sensor networks, and distributed systems.



Hongyi Wu (M'02) received his Ph.D. degree in Computer Science and M.S. degree in Electrical Engineering from State University of New York (SUNY) at Buffalo in 2002 and 2000, respectively. He received his B.S. degree in Scientific Instruments from Zhejiang University in 1996. He is currently an Assistant Professor at the Center for Advanced Computer Studies (CACS), University of Louisiana (UL) at Lafayette. His research interests include wireless mobile ad hoc networks, wireless sensor networks, next generation cellular systems, and integrated heterogeneous wireless systems. He has served as chair and technical committee member of several IEEE conferences, and the guest editor of two special issues of ACM MONET. He has published more than fifty technical papers in leading journals and conference proceedings. He received NSF CAREER Award in 2004.

Ha Dang received his BS and MS degrees from Hanoi University of Technology in 1999 and University of Louisiana at Lafayette in 2005, respectively. He is currently a PhD student at the Center for Advanced Computer Studies (CACS), University of Louisiana (UL) at Lafayette. His research interests include wireless mobile ad hoc networks, wireless sensor networks, and distributed systems.

Feng Lin received his M.S. degree in computer science from Sichuan University, China, in 2002. Since then, he is a lecturer in Computer Science School, Sichuan University, China. During 2005-2006, he worked at the Center for Advanced Computer Studies (CACS), University of Louisiana at Lafayette, as a visiting scholar. His research interests include protocol design and network performance analysis for wireless ad hoc networks, wireless sensor networks and other communication systems.