DFT-MSN: The Delay/Fault-Tolerant Mobile Sensor Network for Pervasive Information Gathering

Yu Wang and Hongyi Wu
The Center for Advanced Computer Studies
University of Louisiana at Lafayette
P.O. Box 44330, Lafayette, LA 70504
Email:{yxw1516,wu}@cacs.louisiana.edu

***Abstract*— This paper focuses on the *Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN)* for pervasive information gathering. We develop simple and efficient data delivery schemes tailored for DFT-MSN, which has several unique characteristics such as sensor mobility, loose connectivity, fault tolerability, delay tolerability, and buffer limit. We first study two basic approaches, namely, direct transmission and flooding. We analyze their performance by using queuing theory and statistics. Based on the analytic results that show the tradeoff between data delivery delay/ratio and transmission overhead, we introduce an optimized flooding scheme that minimizes transmission overhead in flooding. Then, we propose a simple and effective DFT-MSN data delivery scheme, which 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 reflects the likelihood that a sensor can deliver data messages to the sink. The latter decides which messages to transmit or drop based on the *fault tolerance*, which indicates the importance of the messages. The system parameters are carefully tuned on the basis of thorough analyses to optimize network performance. Extensive simulations are carried out for performance evaluation. Our results show that the proposed DFT-MSN data delivery scheme achieves the highest message delivery ratio with acceptable delay and transmission overhead.**

I. Introduction

Pervasive information gathering plays a key role in many applications. One typical example is 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. Another example is air quality monitoring for tracking the average toxic gas taken by people everyday. The aforementioned 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. Note that, while concerns may be raised over personal privacy, it is a separate issue which is out the scope of this paper. 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 example, a person should not be asked to take special actions (e.g., to move to a specific location) to facilitate information acquisition and delivery.

Information gathering relies on sensors. The mainstream approach is to densely deploy a large number of small, highly portable, 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 the aforementioned application scenarios, because the connectivity between the *mobile* sensors is poor, and thus it is difficult to form a well connected mesh network for transmitting data through end-to-end connections from the sensor nodes to the sinks.

In this research, we study a *Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN)* for pervasive information gathering. A DFT-MSN consists of two types of nodes, 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_2 and S_3, S_4 and S_5, and S_6 and HES_2 can communicate with each other at this moment). Since the transmission range of a sensor is usually short, it cannot deliver the collected data to the destination (e.g., a data server) directly. As a result, 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). It is assumed that the high-end nodes that serve as sinks may connect to backbone access points all the time if necessary. With its self-organizing ability, DFT-MSN is established on an ad hoc basis without pre-configuration.

Although it is with similar hardware components, DFT-MSN distinguishes itself from conventional sensor networks by the following unique characteristics:

- **Nodal mobility:** The sensors and the sinks are attached to people with various types of mobility. Thus the network topology is dynamic (similar to the mobile ad hoc network).

1-4244-0222-0/06/$20.00 (c)2006 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the Proceedings IEEE Infocom.

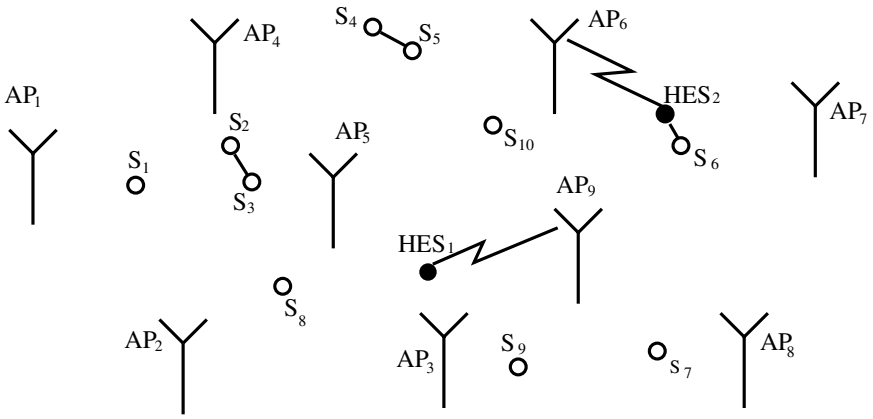


Fig. 1. An overview of DFT-MSN. S_1-S_{10}: mobile sensors; HES_1-HES_2: high end sensors (sinks); AP_1-AP_9: access points of the backbone network.

- **Sparse connectivity:** The connectivity of DFT-MSN is very low, forming a sparse sensor network where a sensor is connected to other sensors only occasionally.
- **Delay tolerability:** Data delivery delay in DFT-MSN is high, due to the loose connectivity among sensors. Such delay, however, is usually tolerable by the applications that aim at pervasive information gathering from a statistic perspective.
- **Fault tolerability:** Redundancy (e.g., multiple copies of a data message) may exist in DFT-MSN during data acquisition and delivery. Thus, a data message may be dropped without degrading the performance of information gathering.
- **Limited buffer:** Similar to other sensor networks, DFT-MSN consists of sensor nodes with limited buffer space. This constraint, however, has a higher impact on DFT-MSN, because the sensor needs to store data messages in its queue for a much longer time before sending them to another sensor or the sink, exhibiting challenges in queue management.

In addition, DFT-MSN also shares characteristics of other sensor networks such as the short radio transmission range and the low computing capability.

DFT-MSN is fundamentally an opportunistic network, where communication links exist with certain probabilities. In such a network, replication is necessary for data delivery in order to achieve certain success ratio. Clearly, replication also increases the transmission overhead. Thus it is a key issue to deal with the trade-off between data delivery ratio/delay and overhead in DFT-MSN.

This paper focuses on the development of simple and efficient data delivery schemes, tailored for DFT-MSN with the above unique characteristics. Motivated by the Delay-Tolerant Network (DTN) [2] and pertinent work to be discussed in Sec. II, we first study two basic approaches, namely, direct transmission and flooding. We analyze their performance by using queuing theory and statistics. Based on the analytic results that show the tradeoff between data delivery delay/ratio and transmission overhead, we introduce an optimized flooding scheme that minimizes transmission overhead in flooding. Then, we propose an efficient DFT-MSN data delivery scheme, which 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. The system parameters are carefully tuned based on thorough analyses to optimize network performance. Extensive simulations are carried out for performance evaluation. Our results show that the proposed DFT-MSN data delivery scheme achieves the highest message delivery ratio with acceptable delay and transmission overhead.

The rest of the paper is organized as follows: Sec. II discusses related work. Sec. III presents our studies on two basic approaches. Sec. IV introduces the proposed DFT-MSN data delivery scheme and presents the simulation results and discussion. Finally, Sec. V concludes the paper.

II. 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 [2]. 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 [2], [3]. An overall architecture of DTN has been proposed in [3]. In [4], Burleigh *et al.* identify several fundamental principles and propose a new end-to-end overlay network protocol called Bundling. In [5], 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 [6]. In [7], 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 [8]. DTN/SN focuses on the deployment of sensor networks that are inter-operable with the Internet protocols [9]. Ref. [10] 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 [11] 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, ZebraNet [12] 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, where the routing decision is made according to the node's 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 [13], [14] 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. 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 [15], [16], in order to address the tradeoff between delivery ratio/delay and overhead.

DTN technology has also been employed in mobile ad hoc networks. A Context-Aware Routing (CAR) algorithm is proposed in [17] to provide asynchronous communication in partially-connected mobile ad hoc networks. In [18], 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. Ref. [19] proposes a Message Ferrying (MF) approach 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. In PROPHET [20], each node maintains a delivery predictability vector, which indicates its likelihood to meet other nodes. The messages can then be forwarded from the low-predictability nodes to the high-predictability nodes. This simple approach may result in high overhead due to the maintenance of delivery predictability vector and the excessive message copies generated during forwarding. Ref. [21] studies the human mobility patterns. It reveals that some nodes are more likely to meet with each other so that the network may be better described by a community model. Ref. [22] studies the sociological movement pattern of mobile users and proposes a series of sociological orbit based routing protocols.

III. Studies of Two Basic Approaches

We first study two basic approaches and analyze their performance. Without loss of generality, we consider a network that consists of N sensors and n sink nodes uniformly distributed in an area of 1×1. We assume that a sensor or a sink has a fixed radio transmission range, forming a radio coverage area denoted by a ($a \ll 1$). We define the *service area* of a sink node to be its radio coverage area (i.e., a). The total service area of all sink nodes in the network is denoted by A ($A < 1$). Clearly, $A = 1-(1-a)^n$. Given the very short radio transmission range and the small number of sinks, the probability that two or more sinks share an overlapped service area is low. Thus $A = 1-(1-a)^n \approx na$.

A. Basic Approach I: Direct Transmission

The Basic Approach I is a direct transmission scheme, where a sensor transmits directly to the sink nodes only. More specifically, assume that the generated data message is inserted into a first come first serve (FCFS) queue. Whenever the sensor meets a sink, it transmits the data messages in its queue to the sink. A sensor does not receive or transmit any data messages of other sensors.

The sensors are usually activated and deactivated periodically. For analytic tractability, we assume the sensor's activation period to be an exponentially distributed random variable with a mean of T. The sensor performs sensing and generates one data message upon waking up in each period.

In addition, we assume the length of the message equals a constant of L. Since the activation period is exponentially distributed, the message arrival is a Poisson process with an average arrival rate of $\lambda = 1/T$. The service rate, μ, depends on the available bandwidth (w) between a sensor and a sink and the probability (p) that a sensor is able to communicate with the sink. To facilitate our illustration, we first assume the bandwidth to be a constant. Possible bandwidth variation due to channel contention will be considered later in this section. Since the sensors and the sink nodes are uniformly distributed, the probability that a sensor is within the coverage of at least one sink node is determined by the total service area of all sink nodes, i.e., $p = A = 1-(1-a)^n \approx na$. We now prove that the service time is a random variable with Pascal distribution.

Lemma 1: Given a constant message length of L, a fixed channel bandwidth of w, and a service probability of p, the service time of the message is a random variable with Pascal distribution.

Proof: Denote a random variable X to be the service time. Let s be the number of time slots required to transmit a message if the node is within the service area. With constant message length L and fixed bandwidth w, we have $s = \frac{L}{w}$. In each time slot, a node has the probability of p to be within the service area. Thus, the distribution function of X, i.e., the probability that the message can be transmitted within no more than x time slots, is

$$F_{\mathbf{X}}(x) = \sum_{i=0}^{x-s} \left(\begin{array}{c} s+i-1 \\ s-1 \end{array} \right) p^s (1-p)^i. \tag{1}$$

This is the Pascal distribution, with mean value of $\frac{s}{p}$ and variation of $\frac{s\times(1-p)}{p^2}$. ■

1) Infinite Buffer Space: We first assume that the sensor has infinite buffer space. With a Poisson arrival rate and a Pascal service time, data generation and transmission can be modelled as an M/G/1 queue, with $\lambda = \frac{1}{T}$ and $\mu = \frac{p}{s} = \frac{Aw}{L}$. In order to arrive at the steady state, we have $\lambda < \mu$, leading to the minimum service area,

$$A > \frac{L}{T \times w}. \tag{2}$$

In other words, the queue will be built up to infinite length if the service area is less than $\frac{L}{T\times w}$.

For given message arrival rate λ and service rate μ, we can derive the average number of messages (including the one currently being served) at a sensor,

$$q = \rho + \frac{\rho^2 + \lambda^2 \times \rho^2}{2 \times (1-\rho)}, \tag{3}$$

where $\rho = \frac{\lambda}{\mu}$, and the average message delivery delay of,

$$\omega = \frac{q}{\lambda}. \tag{4}$$

Assume each sensor consumes J Joule to transmit a message and ignore the data processing power. The average power consumption to deliver a message to the sink is,

$$E = J. \tag{5}$$

2) Finite Buffer Space: With finite buffer space (e.g., by assuming each sensor is able to keep maximum K messages in its queue), the data generation and transmission can be modelled as an M/G/1/K queue. The message arrival rate (λ) and the service rate (μ) are calculated in the same way as discussed in Sec. III-A.1. Now we derive the steady state probabilities of this M/G/1/K queue. Let k_n denote the probability of n arrivals during the period for serving a message. According to the Poisson distribution of message arrival, we have

$$k_n = \sum_{t=s}^{\infty} \frac{e^{-\lambda t}(\lambda t)^n}{n!} \times \binom{t-1}{s-1} p^s (1-p)^{t-s}. \tag{6}$$

Let π_i denote the probability that the system size (i.e., the remaining number of messages right after the current message being served) is i. Then, the stationary equations are

$$\pi_i = \begin{cases} \pi_0 k_i + \sum_{j=1}^{i+1} \pi_j k_{i-j+1}, & (i = 0, 1, \cdots, K-2) \\ 1 - \sum_{j=0}^{K-2} \pi_j, & (i = K-1). \end{cases} \tag{7}$$

Plugging Equation (6) into Equation (7), we obtain K equations with K unknowns. Solving them, we arrive at $\{\pi_i \mid 0 \le i \le K-1\}$. Thus, the average number of messages (including the one currently being served) at a sensor is

$$q = \sum_{i=0}^{K-1} i\pi_i. \tag{8}$$

Note that since the buffer space is limited, a fraction of messages are dropped upon arrival. Denote $q_i^{'}$ to be the probability that an arriving message finds a system with i messages. Then $q_K^{'}$ is the message dropping probability,

$$q_K^{'} = \frac{\rho - 1 + \frac{\pi_0}{\pi_0 + \rho}}{\rho}, \tag{9}$$

where $\rho = \frac{\lambda}{\mu}$. Since dropped messages do not join the queue, the effective message arrival rate is

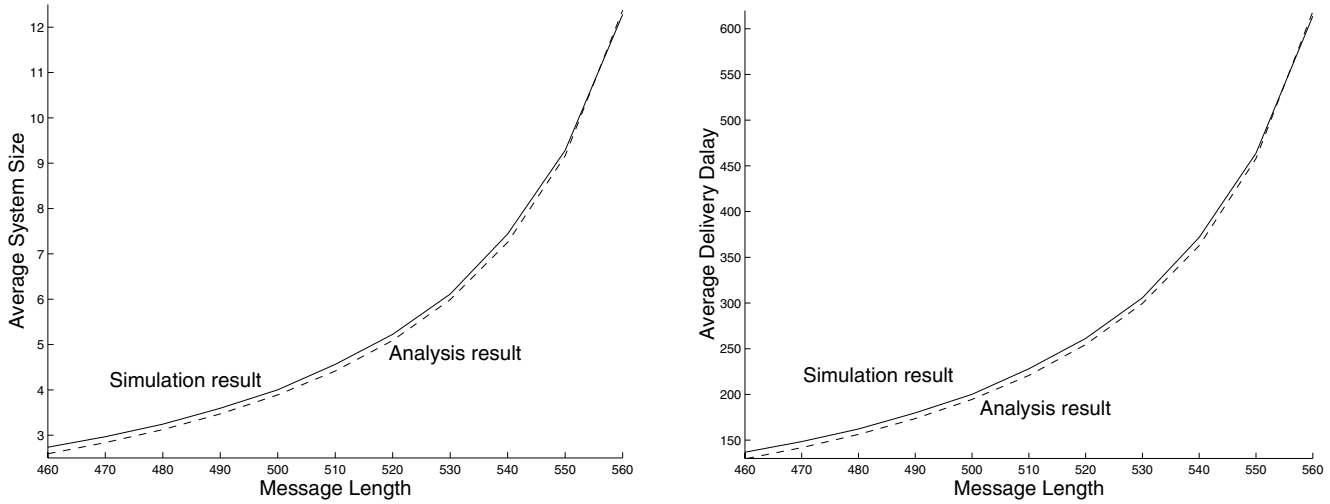
$$\lambda_e = \lambda(1 - q_K{}'). \tag{10}$$

Thus, the average message delivery delay equals

$$\omega = \frac{q}{\lambda_e}. \tag{11}$$

3) Further Discussion: If the service area of a sink (i.e., a) is large, multiple nearby sensors may transmit at the same time. Thus the channel bandwidth, w, is not a constant. As a result, this is no longer a Markov process. If we consider the average service time only, however, we may still use the queuing models discussed in Secs. III-A.1 and III-A.2 to obtain approximate results.

Assume total available bandwidth W is shared by all sensors that are in the service area of a sink. The average data transmission rate of a sensor is $w = \frac{W}{L} \times \frac{1}{1+(N-1)a\frac{\lambda}{\mu}}$, where $\frac{\lambda}{\mu}$ is the probability that a sensor has data messages in its



(a) System size.

(b) Delivery delay.

Fig. 2. Performance of direct transmission with *infinite* buffer space under $N = 100$, $n = 10$, $T = 50$, $w = 150$, $a = 0.0314$.

queue and accordingly $1+(N-1)a\frac{\lambda}{\mu}$ is the average number of active sensors that transmit to the sink. Therefore,

$$\mu = \frac{wp}{L} = \frac{p}{L} \times \frac{W}{1+(N-1)a\frac{\lambda}{\mu}}, \tag{12}$$

i.e.,

$$\mu = \frac{pW}{L} - (N-1)a\lambda. \tag{13}$$

The validity of above analytic models will be discussed next in Sec. III-A.4.

4) Numeric Results: We have carried out simulations to validate our analytic models. The network is deployed in an area of 100×100 m^2, and the transmission range of each node is 9 m. For simplicity, the sink nodes are placed far away from each other so that there is no overlap among their service areas. The sensor nodes and the sink nodes are all moving randomly. Other simulation parameters are shown in the captions of Figs. 2 and 3.

Fig. 2 depicts the results with infinite buffer space. As can be seen, the analytic results match the simulation results very well. With an increase in message length, the traffic load increases, resulting in a longer average system size (i.e., the total number of messages that are in the queue or being served) and longer message delivery delay.

For the network with finite buffer space, we also observe a good match between simulation and analytic results (see Fig. 3). Since the buffer size is limited, a faction of traffic is dropped when the queue is full. As a result, the average system size is smaller compared with the case of infinite buffer space. The message dropping rate increases with the message length.

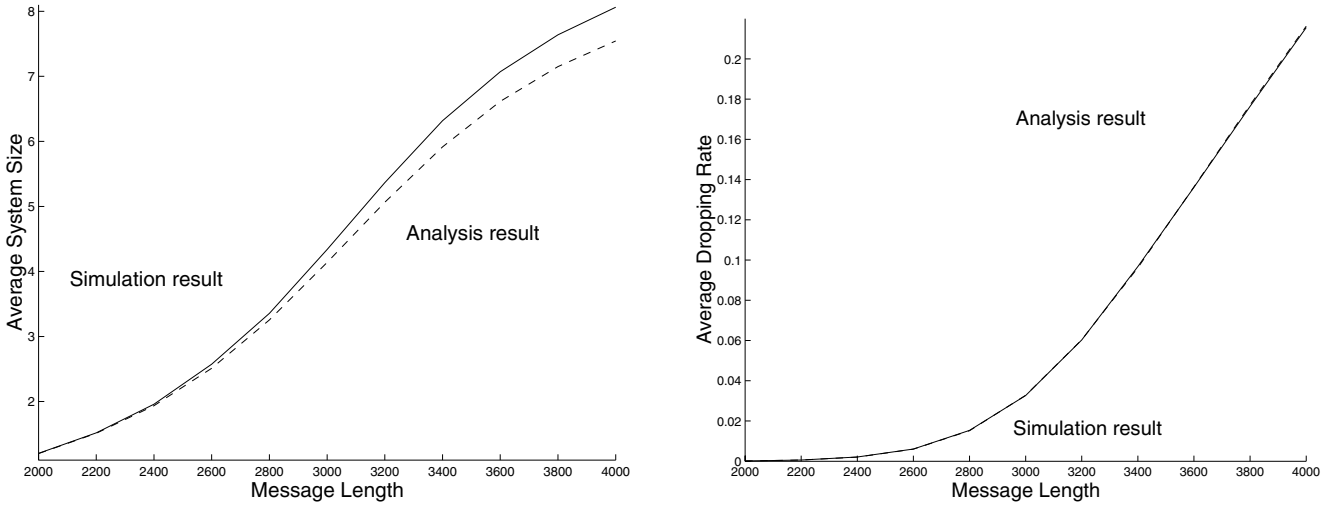
B. Basic Approach II: Flooding

The second basic approach is flooding. We first discuss the simple flooding scheme and then introduce an optimized flooding scheme.

1) Simple Flooding: In the simple flooding scheme, a sensor always broadcasts the data messages in its queue to nearby sensors, which receive the data messages, keep them in queue, and rebroadcast them. Intuitively, this approach achieves a lower data delivery delay at the cost of more traffic overhead and energy consumption.

Similar queuing models as discussed in Sec. III-A can be employed for analyzing this flooding approach. Compared with Basic Approach I where message arrival depends on message generation only, a sensor in the flooding approach not only generates its own data messages but also receives messages from other sensors, resulting in a higher λ. On the other hand, since a sensor may transmit to other sensors in addition to the sinks, the service rate is also higher. The queue length and queuing delay can be derived accordingly.

In the Basic Approach I, the queuing delay is the same as the data message delivery delay because a sensor transmits to the sink nodes only. In the flooding approach however, they are different, due to the duplicate messages at multiple sensor nodes. To analyze the message delivery delay, we consider a data message generated by a sensor with infinite buffer space. For simplicity, we assume the sensor's activation period to be a constant T, within which the sensor can transmit its messages to its neighbors that are activated at the same time. We assume the bandwidth is high enough such that the sensor can always transmit its data messages when it meets other active sensors or the sink nodes. We also assume the mobility is high enough and the network is large enough such that the sensor always meets different neighbors when it wakes up. We study a sequence of activation periods after the message is generated. p is the probability that a sensor can communicate with at least one sink node when it is activated. As we have discussed in Sec. III-A, $p = A = 1-(1-a)^n \approx na$. Denote p_j to be the probability that the message is not delivered to the sink nodes in the first $j-1$ periods and at least one copy of the message is delivered to the sink in the j^{th} period. Let N_j



(a) System size.

(b) Dropping rate.

Fig. 3. Performance of direct transmission with *finite* buffer space under $N = 100$, $n = 10$, $K = 20$, T=20, $w = 20$, $a = 0.0314$.

denote the number of sensors that have a copy of the message in the j^{th} period if the message has not been delivered to the sink. N_j is calculated as follows:

$$N_j = \begin{cases} (N-1)A+1, & j=1 \\ (N-N_{j-1})(1-(1-A)^{N_{j-1}})+N_{j-1}, & j>1. \end{cases} \quad (14)$$

Consequently, p_j is derived below,

$$p_j = \begin{cases} p, & j=1 \\ (1-(1-p)^{N_{j-1}})(1-\sum_{i=1}^{j-1} p_i), & j>1. \end{cases} \quad (15)$$

Thereupon, the average delay of delivering the data message is expressed by

$$\omega = T\sum_{j=1}^{\infty} j \times p_j. \quad (16)$$

Note that, when $N_1 = N_2 = ... = 1$, the above analysis turns into an alternative model for the Basic Approach I, where the sensor transmits its data messages to the sink directly, and thus there is only a single copy of a message in the network.

Since many copies of a given message exist in the network and a sensor is not aware whether the sink has received it or not, the message is eventually received and transmitted once by every sensor node, resulting in totally N copies. Accordingly, the average power consumption per message is proportional to the network size, namely,

$$E = O(J \times N). \quad (17)$$

2) Optimized Flooding: In the simple flooding scheme, each sensor aggressively propagates its data messages to any neighboring nodes, resulting in the lowest delivery delay. At the same time, however, it also incurs very high overhead (i.e., the number of message copies) and energy consumption. Here we introduce an optimized flooding scheme that may significantly reduce flooding overhead and energy consumption.

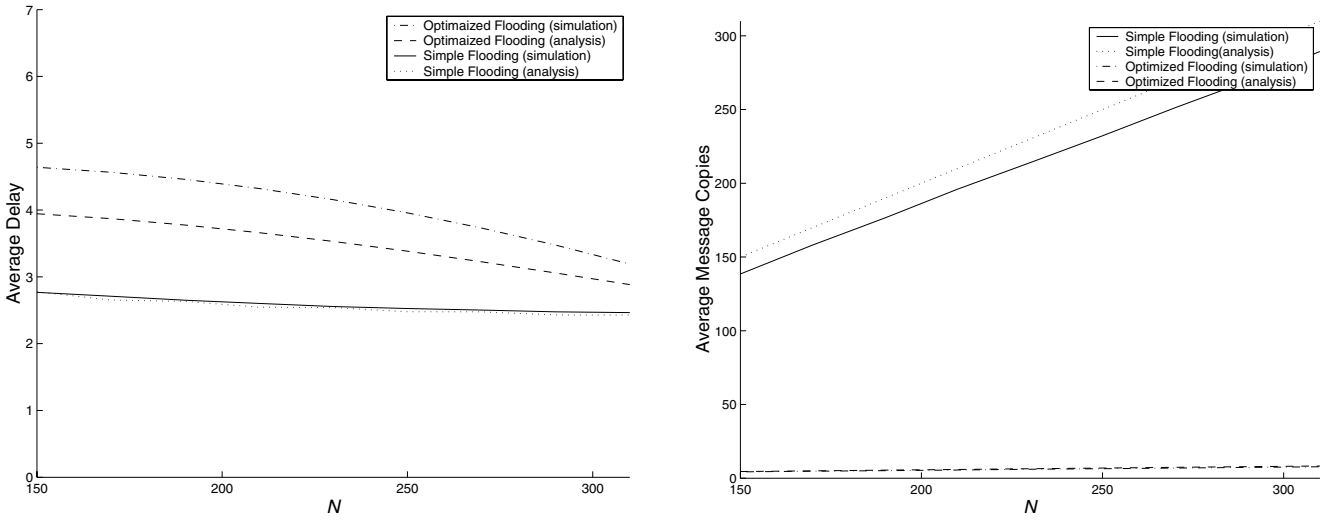
The basic idea of the optimized flooding scheme is to estimate the message delivery probability and stop further propagation of a message if its delivery probability is already high enough in order to reduce transmission overhead. Similar to our discussion on simple flooding, we consider a sequence of activation periods. Assume the message's propagation is terminated after period d (i.e., the sensor that has a copy of the message does not transmit it to any other nodes except the sinks after the d^{th} period). Our objective is to minimize d such that the message delivery probability in total D ($D \geq d$) periods is higher than a given threshold, i.e., $p_D \geq \gamma$.

Since the sensors stop broadcasting the message after d periods, N_j is given by,

$$N_j = \begin{cases} (N-1)A+1, & j=1 \\ (N-N_{j-1})(1-(1-A)^{N_{j-1}})+N_{j-1}, & d \geq j > 1 \\ N_d, & j > d. \end{cases} \quad (18)$$

Similar to the analysis for simple flooding, $p_j = [1-(1-p)^{N_{j-1}}](1-\sum_{i=1}^{j-1} p_i)$ with $p_1 = p$. For a given threshold γ, one can derive the minimum d such that $p_D \geq \gamma$. Accordingly, the average delay is $\omega = T\sum_{j=1}^{\infty} j \times p_j$.

After determining the optimal value of d, we can estimate the average number of message copies made during the d periods, M_d. Note that N_j is the number of copies in the j^{th} period, given that the message has not been delivered to the sink in the first $j-1$ periods. Thus, N_d is not equivalent to M_d. Since the message is not propagated any more after the d^{th} period, the number of copies reaches its maximum at the d^{th} period. Let U_j denote the number of nodes which have a copy of the message but have not transmitted to the sink nodes yet at the j^{th} period, and V_j denote the number of copies that



(a) Delivery delay.

(b) Duplicate copies.

Fig. 4. Performance of flooding schemes.

have been sent to the sinks. We have

$$U_j = \begin{cases} (N-1)A^{1-p} + 1 - p, & j = 1 \\ (1-p)U_{j-1} + (N - U_{j-1} - V_{j-1}) & \\ \times (1-(1-A)^{(1-p)U_{j-1}}), & d \geq j > 1 \end{cases} \tag{19}$$

and

$$V_j = \begin{cases} p, & j = 1 \\ V_{j-1} + p \times U_{j-1}, & d \geq j > 1. \end{cases} \tag{20}$$

Therefore, the average number of message copies made during the d periods is

$$M_d = U_d + V_d. \tag{21}$$

3) Numeric Results: We have simulated and compared the two flooding approaches discussed above. The network is deployed in an area of 100×100 m^2 with 3 sink nodes, and the transmission range of each node is 9 m. For simplicity, the sensor nodes and the sink nodes are all randomly moving, and the message buffer of each sensor is large enough so that no message is dropped. $\gamma = 0.7$ and $D = 5$.

Fig. 4 compares analytic and simulation results of both approaches. As can be seen, the simulation results and the analytic results match well. As shown in Fig. 4(a), the message delivery delay of both approaches decreases slightly with increase in network density. This is somewhat expected because under higher network density, the message is broadcasted to more neighbors and thus is propagated faster. We notice that the message delay of the optimized flooding is slightly higher than that of the simple flooding approach because the sensors stop forwarding the message after d periods. At the same time, the optimized flooding scheme introduces much fewer duplicate messages compared with its simple flooding counterpart (see Fig. 4(b)), and thus significantly reducing energy consumption. The increase of network density leads to a linear increase in the number of duplicate messages when simple flooding is employed. In contrast, the number of duplicate messages of the optimized flooding approach increases only marginally, because d is optimized to lower flooding overhead.

IV. Proposed DFT-MSN Data Delivery Scheme

We have studied two basic approaches so far. The direct transmission approach minimizes transmission overhead (i.e., the number of message copies) and energy consumption, at the expense of a long message delivery delay (with large buffer space) or a low message delivery ratio due to a high message dropping rate (with small buffer space). In contrast, the flooding approach minimizes the message delivery delay. At the same time however, it results in very high transmission overhead and energy consumption. Note that, although the optimized flooding scheme may significantly reduce the number of message copies, it is based on the assumptions of unlimited buffer space and globally synchronized activation periods. Those assumptions usually don't hold in practical DFT-MSNs.

An efficient DFT-MSN data delivery scheme will take into consideration the tradeoff between delivery delay/ratio and transmission overhead/energy. In particular, the following three key issues need to be addressed.

- *When to transmit data messages?* When a sensor moves into the communication range of another sensor, it needs to decide whether to transmit its data messages or not, in order to achieve a high message delivery ratio, and at the same time, minimize transmission overhead.
- *Which messages to transmit?* The data messages generated by the sensor itself or received from other sensors are put into the sensor's data queue. After deciding to initiate data transmission, the sensor needs to determine which messages to transmit if there are multiple messages with different degrees of importance in its queue.

- *Which messages to drop?* A data queue has a limited size. When it becomes full (or due to other reasons as to be discussed later), some messages have to be dropped. The sensor needs to decide which messages to drop according to their importance in order to minimize data transmission failure.

The proposed 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 employed to address the issues discussed above. Then, we introduce the queue management and data transmission schemes.

A. 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,

$$\xi_i = \begin{cases} (1-\alpha)[\xi_i] + \alpha\xi_k, & \textit{Transmission} \\ (1-\alpha)[\xi_i], & \textit{Timeout}, \end{cases} \tag{22}$$

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. However, unlike other typical store-and-forward networks where the packets are deleted from the buffer after they are transmitted to the next hop successfully, 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, we discuss two approaches to define the fault tolerance of a message.

Delivery Probability-Based Approach. We may define the fault tolerance of a message 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 $\mathcal{F}_{\psi_z}^j$,

$$\mathcal{F}_{\psi_z}^j = 1 - (1 - [\mathcal{F}_i^j])(1-\xi_i) \prod_{m=1,\ m\neq z}^{Z} (1-\xi_{\psi_m}), \tag{23}$$

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

$$\mathcal{F}_i^j = 1 - (1 - [\mathcal{F}_i^j]) \prod_{m=1}^{Z} (1-\xi_{\psi_m}), \tag{24}$$

where $[\mathcal{F}_i^j]$ 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 larger fault tolerance.

Message Hop Count-Based Approach. The fault tolerance can also be defined according to the hop count of the message. Denote h_j to be the number of times that the message j has been forwarded. A message with larger h_j usually has more copies in the network. More specifically, the number of copies of the message j is proportional to h_j^2. Thus, we may let $\mathcal{F}_i^j = h_j^2/H^2$, where H is the maximum hop count. For a new message, $\mathcal{F}_i^j = 0$ since $h_j = 0$. If a message has just been sent to the sink, $\mathcal{F}_i^j = 1$. The approach based on the message hop count is simpler but, at the same time, less accurate when compared with the delivery probability-based approach. Their performance outcomes will be compared in Sec. IV-C.

B. 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. (1) After the sensor acquires data from its sensing unit, it creates a data message, which is inserted into its data queue; (2) When the sensor receives a data message from other sensors, it inserts the message into its data queue; (3) 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_i^m 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 $B_i(x) = K - \sum_{m=0}^{x} k_i^m$. 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 $\Xi' = \{\psi_z \mid 1 \le z \le Z'\}$ denote 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 is determined by the following algorithm, where γ is a threshold, $\mathcal{F}_i^j$ is the fault tolerance of the message j at Sensor i, and $B_{\psi_z}(\mathcal{F}_i^j)$ is the number of available buffer slots at Node ψ_z for messages with fault tolerance $\mathcal{F}_i^j$.

By following Algorithm 1, Sensor i sends Message j to a set of neighbors with higher delivery probabilities (i.e., $\xi_i < \xi_{\psi_z}$),

Algorithm 1 Identification of receiving sensors.

```
Φ = ∅.
for z = 1 : Z' do
  if ξ_i < ξ_{ψ_z} AND B_{ψ_z}(F_i^j) > 0 then
    Φ = Φ ∪ ψ_z.
  end if
  if 1 − (1 − F_i^j) ∏_{m∈Φ}(1 − ξ_m) > γ then
    Break.
  end if
end for
```

and at the same time, controls the total delivery probability of Message j (i.e., $1-(1-\mathcal{F}_i^j)\prod_{m\in\Phi}(1-\xi_m)$) just enough to reach γ in order to reduce unnecessary transmission overhead. 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., $B_{\psi_z}(\mathcal{F}_i^j)$) before data transmission.

Clearly, this message transmission scheme is equivalent to direct transmission when the network is just deployed, because the delivery probability is initialized with zero and thus the sensors transmit to the sink nodes only. As the delivery probability is gradually updated with none zero values, multihop relaying will take place.

C. Simulation Results

Extensive simulation has been carried out to evaluate the performance of the proposed DFT-MSN data delivery schemes. In our simulation, 3 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 zones, each with an area of 40×40 m^2. A sensor node is initially resided in its home zone. It moves with a speed randomly chosen between 0 and 5 m/s. Whenever a node reaches the boundary of its zone, it moves out with a probability of 20%, and bounces back with a probability of 80%. After entering a new zone, the sensor repeats the above process. However, if it reaches the boundary to its home zone, it returns to its home zone 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 50 bits. The channel bandwidth is 2500 bps. The fault tolerance threshold is set to be $\gamma = 0.8$. The above default simulation parameters are summarized in Table I.

The sensor node transmits its data messages according to our proposed DFT-MSN data delivery schemes. We first study the effectiveness of delivery probability updating. For clarity, we set the delivery probability of the sensor into 5 discrete levels. Level i ($1 \le i \le 5$) represents the successful delivery probability between $(i-1) \times 0.2$ and $i \times 0.2$. Fig. 5(a) shows DFT-MSN at the initial stage, where each sensor node has a delivery probability of 0. With the proposed protocol running, each node updates its delivery probability. The results after 1000 seconds are illustrated in Fig. 5(b), where the nodes

TABLE I

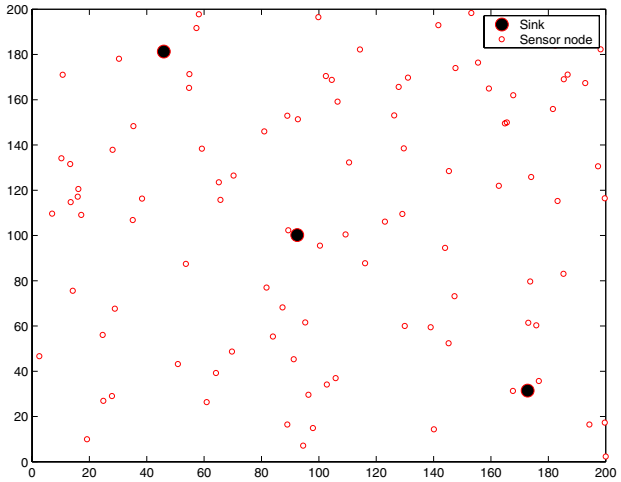
DEFAULT SIMULATION PARAMETERS

Maximum sensor transmission range	10 m
Number of sensor nodes	100
Number of sink nodes	3
Size of network area	200×200 m^2
Size of a zone	40×40 m^2
Probability to move out of a zone	20%
Probability to move back to home zone	100%
Maximum queue length	200
Message generation rate	$0.01/s$
Message length	50 $bits$
Bandwidth	2500 bps
Nodal moving speed	$0-5m/s$
γ	0.8

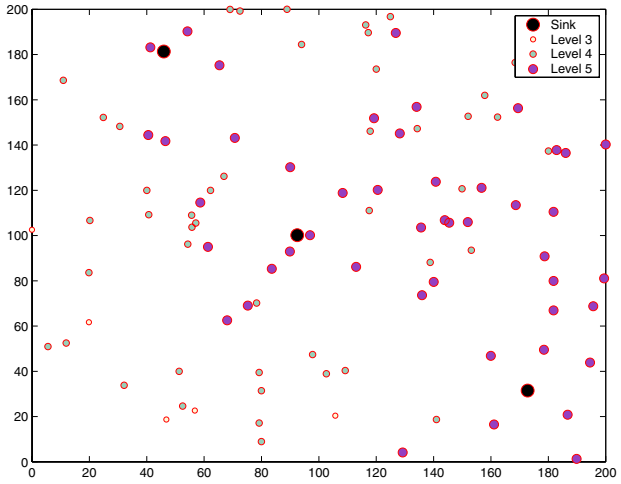
closer to the sinks usually have higher delivery probabilities as expected.

We vary several parameters to observe their impacts on the performance. Fig. 6 compares the performance of the simple flooding approach, the direct transmission approach, the hop count-based approach, and the delivery probability-based approach, by varying the number of sink nodes in DFT-MSN. As shown in Fig. 6(a), the proposed delivery probability-based approach always has a higher delivery ratio than other approaches, especially when a small number of sinks are deployed. With a large number of sinks, the delivery probability-based approach, the hop count-based approach, and the direct transmission approach yield similar 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%. As expected, the flooding approach has a much lower delivery ratio than other approaches because it generates too many message copies, which leads to excessive buffer overflow and message dropping. Fig. 6(b) demonstrates that the average delay of every approach decreases quickly 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, because the hop count-based approach generates more copies for each message, it has a slightly lower delay than the delivery probability-based approach. Clearly, the direct transmission approach suffers the longest delay since messages can then 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. 6(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 is not shown here, because it generates excessive copies which are several orders of magnitude higher than those of other approaches. The delivery probability-based approach makes around 10 copies for each message, and the number of copies decreases slightly with an increase in the number of sink nodes. The results of the hop count-based approach first increases and then decreases. This is explained below. At first, with more sink nodes deployed, the sensor nodes' delivery probabilities increase.



(a) Initial deployment.

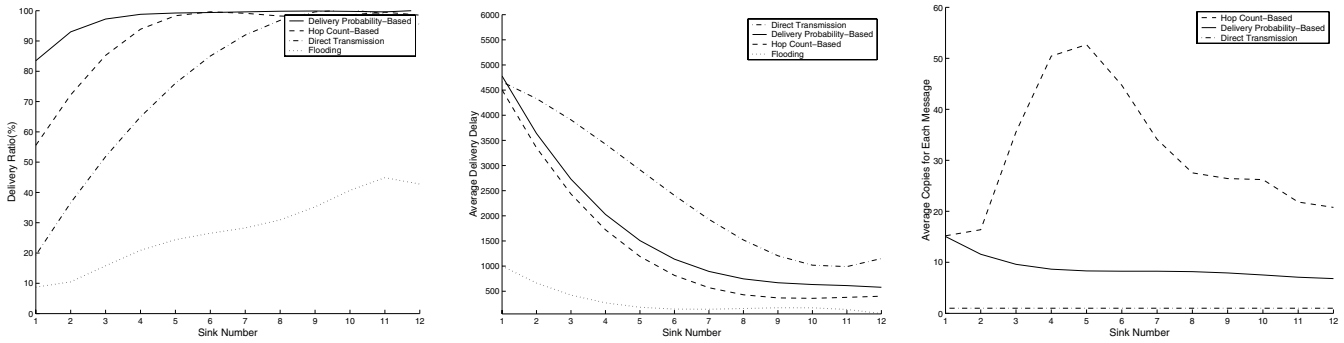


(b) 1000 seconds later.

Fig. 5. Update of delivery probability.

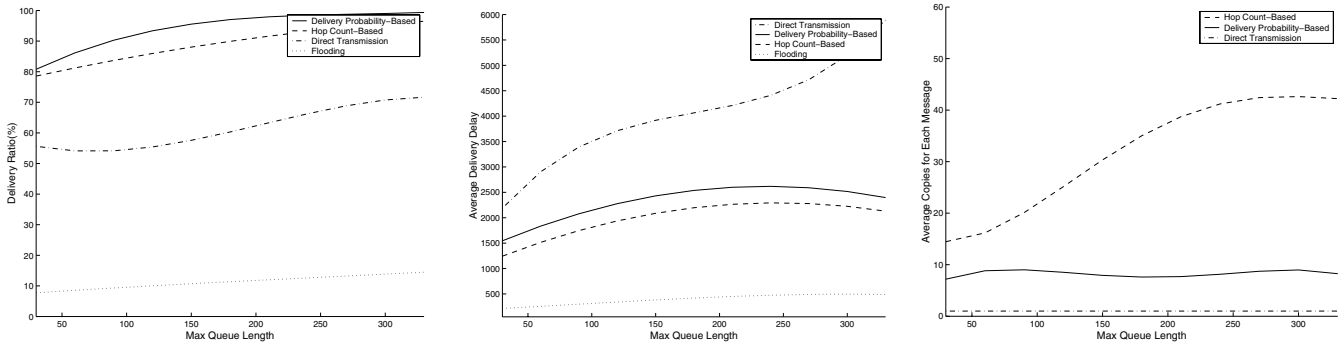
As a result, the nodes farther away from the sinks may have more neighbors with larger delivery probabilities. More copies are thus generated for each message. Then, with a continuous increase in the number of sink nodes, more and more sensors can reach the sinks within small numbers of hops. Therefore, the number of message copies eventually decreases.

We also vary the maximum queue length of each sensor in our simulations, with results presented in Fig. 7. With an increase in maximum queue length, the delivery ratio increases slowly for all approaches, as expected (see Fig. 7(a)). As shown in Fig. 7(b), the queue length doesn't have a significant impact on the delay of the simple flooding approach, the hop count-based approach, and the delivery probability-based approach. The delay of the direct transmission approach, however, increases sharply with the longer queue length, because more data messages will then reside in the queue for a longer time before being delivered. It is also noticed that the delivery probability-based approach can well control its transmission overhead (i.e., the number of copies generated)



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

Fig. 6. Impact of the number of sink nodes.



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

Fig. 7. Impact of maximum queue length.

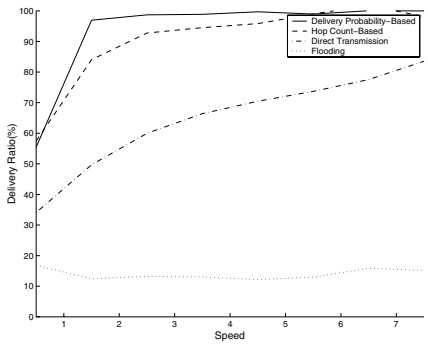
even when the available queue size is large. On the other hand, far more duplicated copies are generated under the hop count-based approach (see Fig. 7(c)) as a result of of increasing the maximum queue size .

Fig. 8 depicts the impact of nodal moving speed. As the speed increases, the delivery ratios of all approaches but the simple flooding rise, while the delivery delays of all approaches decrease. This is because the node with a higher speed has a better opportunity to meet other nodes and also with higher probabilities to reach the sink nodes. Thus, the messages have a better chance to be delivered before they are dropped. It is also noticed that the transmission overhead of the proposed delivery probability-based approach is almost constant with the increase of nodal speed (as shown in Fig. 8(c)), making it most suitable for the network with varying nodal speeds.

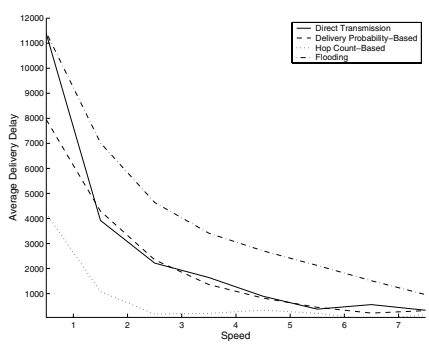
γ is crucial to the performance of our proposed delivery probability-based approach. With a larger γ, the message propagation becomes more aggressive. In other words, a sensor has higher probabilities to transmit data messages to other nodes. As a result, both the delivery ratio and the transmission overhead become higher (see Fig. 9(a) and Fig. 9(c)). Besides, we notice that the delay of the successfully delivered message also increases with a larger γ. This is explained as follows. When γ is low, most successfully delivered messages are those generated by nodes close to the sinks, causing the delay to be low. When γ rises, the messages produced by nodes farther away from the sink nodes can also be successfully delivered, but with a longer delay.

V. Conclusion

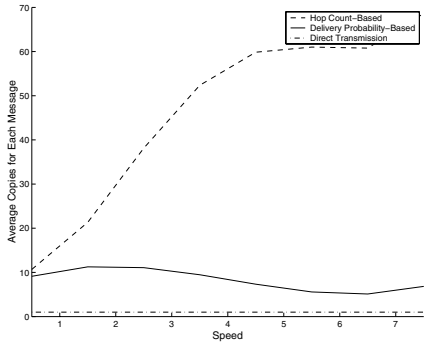
This paper deals with the *Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN)* for pervasive information gathering. DFT-MSN has several unique characteristics such as sensor mobility, loose connectivity, fault tolerability, delay tolerability, and buffer limit. We have first studied two basic approaches, namely, direct transmission and flooding, using queuing theory and statistics. Based on the analytic results that show the tradeoff between data delivery delay/ratio and transmission overhead, we have introduced an optimized flooding scheme that minimizes the transmission overhead of flooding. Then, we have proposed a simple and effective DFT-MSN data delivery scheme, which consists of two key components respectively for data transmission and queue management. The former makes decision on when and where to transmit data messages based on the nodal *delivery probability*, while the latter decides which messages to transmit or drop based on the *fault tolerance*. Extensive simulations have been carried out for performance evaluation. Our results show that the proposed DFT-MSN data delivery scheme achieves the highest message delivery ratio with acceptable delay and transmission overhead.



(a) Average delivery ratio.

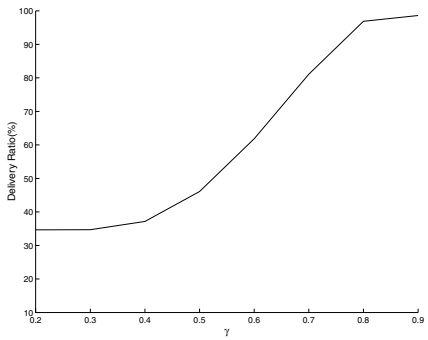


(b) Average delay.

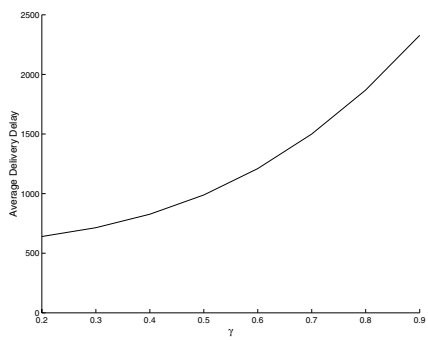


(c) Average overhead.

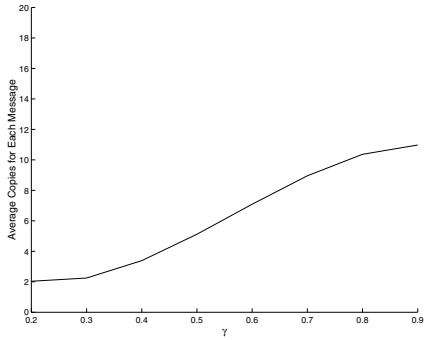
Fig. 8. Impact of nodal speed (m/s).



(a) Average delivery ratio.



(b) Average delay.



(c) Average overhead.

Fig. 9. Impact of γ.

REFERENCES

[1] I. Akyildiz, W. Su, and Y. Sankarasubramaniam, "A Survey on Sensor Networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] 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.

[3] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proc. of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, pp. 27–34, 2003.

[4] 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 Communications Magazine*, vol. 41, no. 6, pp. 128–136, 2003.

[5] 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.

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

[7] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *Proc. of ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pp. 88–97, 2002.

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

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

[10] M. Ho and K. Fall, "Poster: Delay Tolerant Networking for Sensor Networks," in *Proc. of IEEE Conference on Sensor and Ad Hoc Communications and Networks*, 2004.

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

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

[13] 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. of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC03)*, 2003.

[14] T. Small and Z. J. Haas, "Resource and Performance Tradeoffs in Delay-Tolerant Wireless Networks," in *Proc. of ACM SIGCOMM05 workshop on Delay Tolerant Networking and Related Topics (WDTN-05)*, 2005.

[15] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-Coding Based Routing for Opportunistic Networks," in *Proc. of ACM SIGCOMM Workshop on Delay Tolerant Networking*, 2005.

[16] Y. Wang and H. Wu, "Replication-Based Efficient Data Delivery Scheme (RED) for Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN)," in *Proc. of the First International Workshop on Ubiquitous & Pervasive Healthcare (UbiCare), in conjunction with PerCom'06*, 2006, in press.

[17] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks," in *Proc. of IEEE 6th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*, pp. 1–7, 2005.

[18] J. LeBrun, C.-N. Chuah, and D. Ghosal, "Knowledge Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks," in *Proc. of IEEE Vehicular Technology Conference (VTC) Spring*, pp. 1–5, 2005.

[19] W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," in *Proc. of the 5th ACM international symposium on Mobile ad hoc networking and computing (Mobihoc)*, pp. 187–198, 2004.

[20] A. Lindgren, A. Doria, and O. Scheln, "Probabilistic Routing in Intermittently Connected Networks," in *Proc. of the First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR 2004)*, 2004.

[21] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket Switched Networks and Human Mobility in Conference Environments," in *Proc. of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pp. 244–251, 2005.

[22] J. Ghosh, S. Yoon, H. Ngo, and C. Qiao, "Sociological Orbits for Efficient Routing in Intermittently Connected Mobile Ad Hoc Networks." Tech Report, Department of Computer Science and Engineering, University at Buffalo, The State University of New York, 2005.