Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN): A New Paradigm for Pervasive Information Gathering

Yu Wang, *Member*, *IEEE*, and Hongyi Wu, *Member*, *IEEE*

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 trade-off between data delivery delay/ratio and transmission overhead, we introduce an optimized flooding scheme that minimizes transmission overhead in flooding. Then, we propose two simple and effective DFT-MSN data delivery schemes, namely, the *Replication-Based Efficient Data Delivery Scheme (RED)* and the *Message Fault Tolerance-Based Adaptive Data Delivery Scheme (FAD)*. The RED scheme utilizes the erasure coding technology in order to achieve the desired data delivery ratio with minimum overhead. It consists of two key components for data transmission and message management. The former makes the decision on when and where to transmit data messages according to the *delivery probability*, which is the likelihood that a sensor can deliver data messages to the sink. The latter decides the optimal erasure coding parameters (including the number of data blocks and the needed redundancy) based on its current delivery probability. The FAD scheme employs the message *fault tolerance*, which indicates the importance of the messages. The decisions on message transmission and dropping are made based on fault tolerance for minimizing transmission overhead. 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 both schemes achieve a high message delivery ratio with acceptable delay. The RED scheme results in lower complexity in message and queue management, while the FAD scheme has a lower message transmission overhead.

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

1 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 regarding the flu virus (or any epidemic disease in general) from an area with high human activity in order to monitor and prevent the explosion of devastating flu. Another example is air quality monitoring, where the goal is to track the average toxic gas taken in by people every day. 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 from 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 statistical 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 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 propose 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

• *The authors are with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA 70504-4330. E-mail: {yxw1516, wu}@cacs.louisiana.edu.*

Manuscript received 19 Jan. 2006; revised 3 Nov. 2006; accepted 13 Nov. 2006; published online 7 Feb. 2007.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0020-0106. Digital Object Identifier no. 10.1109/TMC.2007.1006.

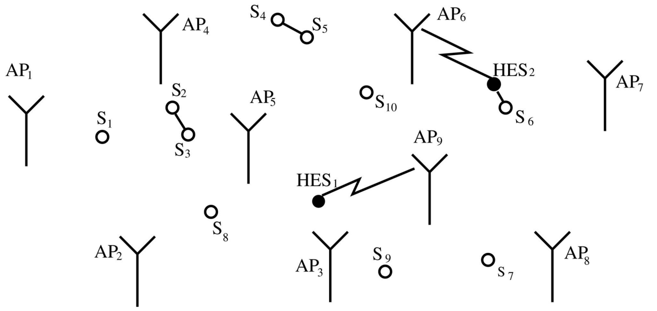


Fig. 1. An overview of the integrated self-configurable wireless mesh network and delay/fault-tolerant mobile sensor system. S_1-S_{10}: sensors; HES_1-HES_2: high end sensors (sinks); AP_1-AP_9: access points of backbone network.

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 preconfiguration.

Although it has 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).
- **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. A similar idea has also been found in [2] to deal with the trade-off between transport reliability and energy consumption in sensor networks.
- **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 a short radio transmission range, low computing capability, and limited battery power.

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 a certain success ratio [3]. 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) [4] and pertinent work to be discussed in Section 2, 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 trade-off between data delivery delay/ratio and transmission overhead, we introduce an optimized flooding scheme that minimizes transmission overhead in flooding. Then, we propose two simple and effective DFT-MSN data delivery schemes, namely, *Replication-Based Efficient Data Delivery Scheme (RED)* and *Message Fault Tolerance-Based Adaptive Data Delivery Scheme (FAD)*. The RED scheme utilizes the erasure coding technology tailored for DFT-MSN in order to achieve the desired data delivery ratio with minimum overhead. It consists of two key components for data transmission and message management, respectively. The former makes a decision on when and where to transmit data messages according to the *delivery probability*, which is the likelihood that a sensor can deliver data messages to the sink. The latter decides the optimal erasure coding parameters (including the number of data blocks and the needed redundancy) based on its current delivery probability. The FAD scheme employs the message *fault tolerance*, which indicates the importance of the messages. The decisions on message transmission and dropping are made based on fault tolerance for minimizing transmission overhead. The system parameters are carefully tuned according to thorough analyses to optimize network performance. Extensive simulations have been carried out for performance evaluation. Our results show that the proposed DFT-MSN data delivery schemes achieve a high message delivery ratio with acceptable delay and transmission overhead.

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 presents our studies on two basic approaches. Section 4 and Section 5 introduce the proposed data delivery schemes. Section 6 presents the simulation results and discussion. Finally, Section 7 concludes the paper.

2 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 interoperability 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. That work also extends the DTN architecture with the concept of transaction abort.

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 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 interoperable with the Internet protocols [11]. Ho and Fall [12] propose employing 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 stores 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 [14] employs the mobile sensors to support wildlife tracking for biology research. The ZebraNet project targets 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. The pioneering work of ZebraNet has motivated our research on mobile sensor networks. 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, does not 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. In many practical applications, however, different nodes may have different probabilities to reach the sink and, thus, SWIM may not work efficiently. Worse 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 the direct contact probability between sensor and sink and are, thus, inefficient. In addition, an erasure coding-based data forwarding algorithm is proposed for opportunistic networks in [17]. The simulation results show that this algorithm provides the best worst-case delay performance with a fixed amount of overhead. However, it neither explains how to determine the optimal value of replication overhead nor discusses the distribution scheme for the coded messages.

DTN technology has also been employed in mobile ad hoc networks. A Context-Aware Routing (CAR) algorithm is proposed in [18] to provide asynchronous communication in partially-connected mobile ad hoc networks. In [19], LeBrun et al. 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. Zhao et al. [20] propose 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 nonrandomness to help data delivery. In PROPHET [21], 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 the delivery predictability vector and the excessive message copies generated during forwarding. Hui et al. [22] study the human mobility patterns. They reveal that some nodes are more likely to meet with each other so that the network may be better described by a community model. Ghosh et al. [23] study the sociological movement pattern of mobile users and propose a series of sociological orbit-based routing protocols.

3 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, $A = 1 - (1-a)^n \approx na$.

3.1 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 served (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 is equal to 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 variations 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 (per time slot), 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

$$F_{\mathbf{X}}(x) = \sum_{i=0}^{x-s} \binom{s+i-1}{s-1} p^s (1-p)^i. \tag{1}$$

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

3.1.1 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 modeled 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 a 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 joules to transmit a message and ignore the data processing power. The average power consumption to deliver a message to the sink is

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

3.1.2 Finite Buffer Space

With finite buffer space (e.g., by assuming each sensor able to keep maximum K messages in its queue), the data generation and transmission can be modeled 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 Section 3.1.1. 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 (6) into (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,

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

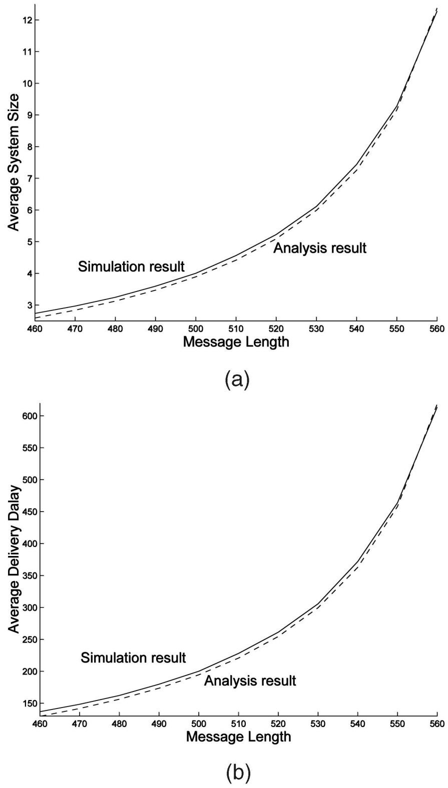


Fig. 2. Performance of direct transmission with *infinite* buffer space under $N = 100$, $n = 10$, $T = 50$, $w = 150$, and $a = 0.0314$. (a) System size. (b) Delivery delay.

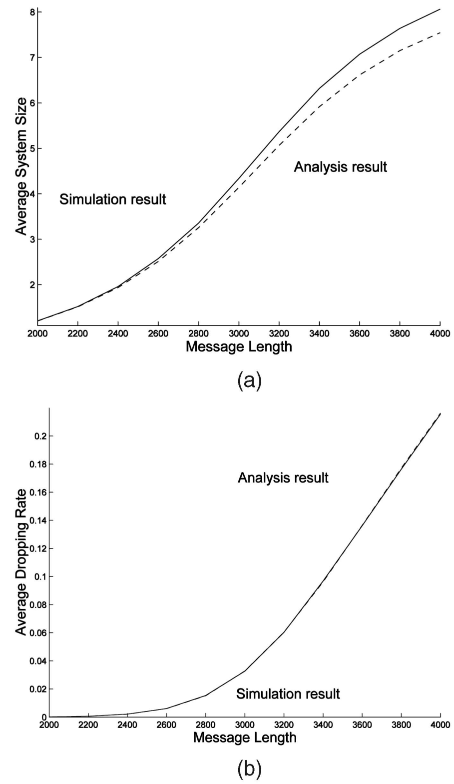


Fig. 3. Performance of direct transmission with *finite* buffer space under $N = 100$, $n = 10$, $K = 20$, $T = 20$, $w = 20$, and $a = 0.0314$. (a) System size. (b) Dropping rate.

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

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

3.1.3 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 Sections 3.1.1 and 3.1.2 to obtain approximate results.

Assume that the 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 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 Section 3.1.4.

3.1.4 Numeric Results

We have carried out simulations to validate our analytic models. The network is deployed in an area of $100 \times 100\ \mathrm{m}^2$, and the transmission range of each node is $10\ \mathrm{m}$. 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 Fig. 2 and Fig. 3.

Fig. 2 depicts the results for the network 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, thus resulting in a longer average system size (i.e., the total number of messages that are currently being served or waiting in the queue) and a longer average message delivery delay.

For the network with finite buffer space, we also observe a good match between simulation and analytic results, with a maximum difference of around 6 percent on system size and 1 percent on dropping rate (see Fig. 3). Since the buffer size is limited, a fraction of arrival 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.

3.2 Basic Approach II: Flooding

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

3.2.1 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 Section 3.1, 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 Section 3.1, $p = A = 1 - (1-a)^n \approx na$. 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 jth period. Let N_j denote the number of sensors that have a copy of the message in the jth 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 $N_1 = N_2 = \ldots = 1$, 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 a total of N copies. Accordingly, the average power consumption per message is proportional to the network size, i.e.,

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

3.2.2 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 dth 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 jth 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 dth period, the number of copies reaches its maximum at the dth 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 jth period, and V_j denote the number of copies that have been sent to the sinks. We have

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

and

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

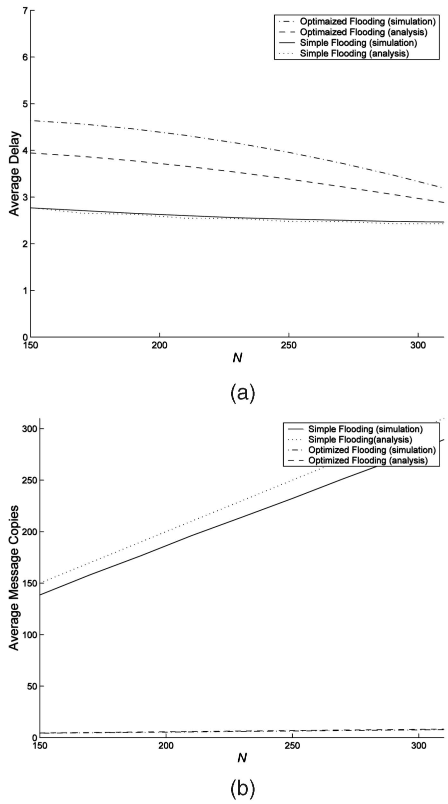


Fig. 4. Performance of flooding schemes. (a) Delivery delay. (b) Duplicate copies.

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

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

3.2.3 *Numeric Results*

We have simulated and compared the two flooding approaches discussed above. The network is deployed in an area of $100 \times 100\,\mathrm{m}^2$ with three sink nodes, and the transmission range of each node is $9\,\mathrm{m}$. 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 the analytic and simulation results of both approaches. As can be seen, the simulation results and the analytic results match well, with a maximum difference of around 17 percent on delay and 6 percent on the duplicate message number for the optimized flooding scheme. As shown in Fig. 4a, the message delivery delay of both approaches decreases slightly with an increase in network density. This is somewhat expected because, under higher network density, the message is broadcast 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 many fewer duplicate messages compared with its simple flooding counterpart (see Fig. 4b) and, thus, significantly reduces 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.

3.3 Observation from the Two Basic Approaches

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 do not hold in practical DFT-MSNs.

DFT-MSN is fundamentally an opportunistic network, where replication is necessary for data delivery, though at the expense of increased transmission overhead, in order to achieve a certain success ratio. An efficient DFT-MSN data delivery scheme will take into consideration the trade-off between delivery delay/ratio and transmission overhead/energy. In particular, the following three key issues need to be addressed:

- *When should data messages be transmitted*? 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 should be transmitted*? 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 should be dropped*? A data queue has a limited size. When it becomes full (due to other reasons, as will 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.

With the above issues taken into consideration, we will propose two data delivery schemes for DFT-MSN in the next two sections, namely, the *Replication-Based Efficient Data Delivery Scheme (RED)* and the *Message Fault Tolerance-Based Adaptive Data Delivery Scheme (FAD)*. Both schemes aim to minimize the overhead while achieving the required data delivery probability. In the former scheme, the replication is done by the source node via erasure coding. In the latter scheme, a message is replicated dynamically according to its fault tolerance by the source and the intermediate nodes.

4 Replication-Based Efficient Data Delivery Scheme (RED)

The proposed replication-based efficient data delivery scheme (RED) consists of two key components for data delivery and message management, elaborated below.

4.1 Data Delivery

4.1.1 Nodal Delivery Probability

The decision on data transmission is made based on the *delivery probability*, which indicates the likelihood that a sensor can deliver data messages to the sink. Note that the delivery probability is *not* simply the probability that a node meets the sinks.

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 could not 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 does not 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:

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

where $[\xi_i]$ is the delivery probability of sensor i before it is updated and $0 \le \alpha \le 1$ 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.

4.1.2 Data Transmission

The data messages are maintained in a first-in-first-out queue. The transmission is simple. Without loss of generality, we consider a sensor i, which has a message at the top of its data queue ready for transmission and is moving into the communication range of a set of sensors. Sensor i first learns their delivery probabilities and available buffer spaces via simple handshaking messages. Then, sensor i transmits its message to the neighbor, j, which has the highest delivery probability ($\xi_j > \xi_i$) and available buffers.

4.1.3 Further Discussion

During our experiments, we have found two potential inefficiencies in data delivery stemming from the approach for updating the nodal delivery probability. The first problem is *mutual reference*. Assume a node j with a slightly higher delivery probability than that of a node i. When nodes i and j are within transmission range, node i transmits a message to node j based on the data delivery scheme discussed above. Then, ξ_i increases due to a successful transmission. In the worst case, node j's delivery probability decreases because of timeout and, thus, ξ_j may become lower than ξ_i. Consequently, node j may transmit messages back to node i in the next transmission period, incurring fluctuation and unnecessary transmission overhead. In order to address this problem, we establish *Lemma 2* as follows:

Lemma 2. *For two nodes i and j with delivery probability ξ_i and ξ_j (assume $\xi_j \ge \xi_i$), $\xi_i < \frac{2-2\alpha}{2-\alpha}\xi_j$ is the necessary and sufficient condition to avoid the mutual reference problem.*

Proof. We prove the necessary condition first. Assume there exists a δ so that the mutual reference problem can be avoided if $\xi_j \ge \xi_i + \delta$. Thus, after node i transmits a message to node j, the following inequation must hold:

$$\xi_j' - \xi_i' > -\delta, \tag{23}$$

where ξ_j' and ξ_i' are the delivery probability of nodes j and i updated according to (22) after the transmission. In the worst case, $\xi_i' = (1-\alpha)\xi_i + \alpha\xi_j$ and $\xi_j' = (1-\alpha)\xi_j$. Plugging ξ_j' and ξ_i' into (23), we have

$$(1-\alpha)\xi_j - (1-\alpha)\xi_i - \alpha\xi_j > -\delta \tag{24}$$

and, thus, we arrive at

$$\delta > \frac{\alpha}{2-\alpha}\xi_j. \tag{25}$$

Therefore, we obtain the necessary condition: To avoid the mutual reference problem, node i should transmit to node j only if

$$\xi_i < \frac{2-2\alpha}{2-\alpha}\xi_j. \tag{26}$$

Similarly, we can show the sufficient condition: If $\xi_i < \frac{2-2\alpha}{2-\alpha}\xi_j$, $\xi_j' - \xi_i' > -\delta$ holds and, thus, the mutual reference problem is avoided. □

Another problem is *unnecessary propagation*. When a node meets a neighbor with higher delivery probability, it always sends messages to this neighbor according to the data transmission scheme discussed above, even when it already has a large enough probability to reach the sink node directly. This results in extra transmission and energy consumption. To avoid unnecessary propagation, each node maintains an additional parameter called the *direct delivery probability* denoted by ψ, which indicates how likely this node can transmit the messages directly to the sink node. If ψ is larger than a predefined threshold, the node only transmits messages to the sink node directly.

4.2 Message Management

As we have discussed in Section 3.3, replication is usually employed to improve the data delivery ratio and/or reduce the data delivery delay in opportunistic networks. In this research, we propose an erasure-coding approach tailored for DFT-MSN, which efficiently addresses the trade-off between delivery ratio/delay and overhead.

In the erasure coding approach [3], a message is first split into b blocks of equal size. Erasure coding is then applied to these b blocks, producing $S \times b$ small messages (which are referred to as *block messages*), where S is the replication overhead. The gain of erasure coding stems from its ability to recover the original message based on *any* b block messages.[1] Assuming that each block message has a constant delivery probability of p, then the delivery probability of the original message is

$$P = \sum_{j=b}^{Sb} \binom{Sb}{j} p^j (1-p)^{Sb-j}. \qquad (27)$$

Clearly, the erasure coding approach is reduced to simple whole message replication when $b = 1$.

Our objective is to determine the optimal erasure coding parameters (i.e., b and S) with given inputs (i.e., p) in order to meet the desired message delivery probability (denoted by H) while minimizing the transmission overhead.

According to (27), we can find the minimum S for given b and p so that P is no less than H, i.e.,

$$S(p,b) = min\left\{ S \;\middle|\; \sum_{j=b}^{Sb} \binom{Sb}{j} p^j (1-p)^{Sb-i} \geq H \right\}. \qquad (28)$$

In theory, b may vary from 1 to the length of the message. But, large b results in many small blocks, which also increases the processing overhead and decreases the bandwidth utilization. Thus, a minimum block size of m is adopted in our proposed approach. Let M denote the maximum message size. Thus, the maximum value of b is $\frac{b_{max}=M}{m}$. Fig. 5 shows $S(p,b)$ with p varying from 0 to 1 and b varying from 1 to $b_{max} = 50$. As we can see, the minimum S generally increases with the decrease of p. When p is very small, a large value of S is necessary for achieving the desired delivery ratio, i.e., H. This, however, may degrade the network performance due to the overwhelming overhead. Therefore, an upper bound of S (denoted by S_{max}) is enforced in our implementation. This approach is particularly useful in situations where nodes temperately have very low delivery ratio (for example, at the beginning of deployment).

The solid curve on the surface of Fig. 5 indicates the minimum S for each given p, while the dashed curve, which is the projection of the solid curve on the horizontal plane, indicates the optimal b for each p in order to minimize S. We notice that, when p is very low (less than a β_1, which is around 0.2) or very high (larger than a β_2, whose value depends on H), the optimal value of b is always 1, which means that the whole message replication is preferable. β_2 usually increases with H. For example, as can be observed by the comparison of Fig. 5a and Fig. 5b, β_2 increases from 0.7 to 0.9 when H increases from 0.9 to 0.99. When p is between β_1 and β_2, the optimal b varies widely and nonmonotonously between 1 and b_{max}, depending on the value of p. In our implementation, a list of p values and their corresponding optimal b values are kept in a table. Whenever a node i generates a data message, it checks its current delivery probability, ξ_i. Let $p = \xi_i$, node i looks up the table to determine the optimal b and minimum S, and accordingly encodes the data messages into $S \times b$ data blocks, which are then put to the queue independently for future transmission.

1. For simplicity, we assume the optimal erasure coding is used.

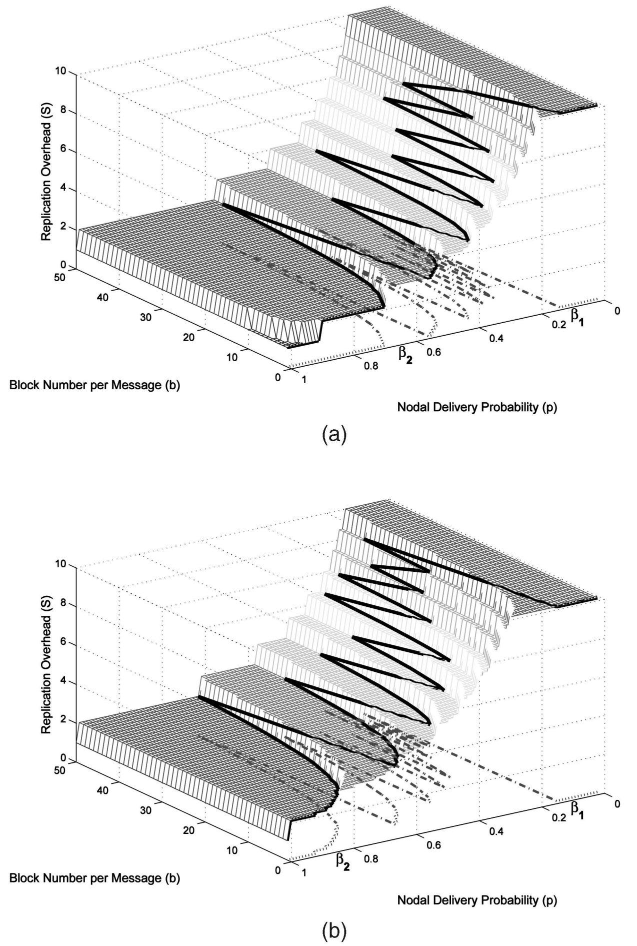


Fig. 5. Determining optimal b to minimize S in erasure coding. (a) $H = 0.90$. (b) $H = 0.99$.

5 Message Fault Tolerance-Based Adaptive Data Delivery Scheme (FAD)

The RED data delivery scheme proposed in Section 4 employs the erasure coding to improve delivery ratio. Its advantage is simplified message manipulation and queue management at intermediate nodes since all computation is done by the source node. At the same time, however, the optimization of erasure coding parameters is usually inaccurate because they are calculated according to the current data delivery probability of source node, especially when the source is very far away from the sinks. In addition, propagating many small messages in the network may incur further processing overhead and inefficiency of bandwidth utilization. In this section, we propose a Message Fault Tolerance-Based Adaptive Data Delivery Scheme (FAD) in order to avoid the above problems at the expense of increased complexity in message and queue management.

The proposed FAD data delivery scheme depends on two important parameters, namely, the nodal delivery probability and the message fault tolerance. The former has been discussed in Section 4.1.1 for the RED scheme. The latter indicates the amount of redundancy and the importance of a message. We first introduce the definition and updating algorithm of message fault tolerance. Then, the FAD data delivery scheme is elaborated.

5.1 FAD Parameter: Message Fault Tolerance

Unlike the RED data delivery scheme discussed in Section 4 (and most other typical data transmission schemes) where the packets are deleted from the buffer after they are transmitted to the next hop successfully, a sensor employing FAD 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. 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 us 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 a total of $Z+1$ 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 $\mathcal{F}_{\psi_z}^j$,

$$\mathcal{F}_{\psi_z}^j = 1 - (1 - [\mathcal{F}_i^j])(1 - \xi_i) \prod_{m=1,\, m \neq z}^{Z} (1 - \xi_{\psi_m}), \qquad (29)$$

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

$$\mathcal{F}_i^j = 1 - (1 - [\mathcal{F}_i^j]) \prod_{m=1}^{Z} (1 - \xi_{\psi_m}), \qquad (30)$$

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.

5.2 FAD Data Delivery Scheme

The proposed FAD data delivery scheme consists of two components for queue management and data transmission, discussed below.

5.2.1 Queue Management

Compared to the simple first-in-first-out queue in RED, the data queue management in FAD is more complicated. 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, and 3) after the sensor sends out a data message to a nonsink 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 send when the sensor meets another sensor and to determine which data message to drop 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. A message with the smallest fault tolerance is always at the top of the queue and transmitted first. A message is dropped on 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 (e.g., γ), 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 that 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 (where $0 \leq m \leq 1$). 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 decisions on data transmission, as discussed next.

5.2.2 Data Transmission

Data transmission decisions are 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 is determined by Algorithm 1, 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 $\mathcal{F}_i^j$.

Algorithm 1 Identification of receiving sensors.

```
Φ = ∅.
for z = 1 : Z' do
  if ξ_i < ξ_ψz AND B_ψz(F_i^j) > 0 then
    Φ = Φ ∪ ψ_z.
  end if
```

TABLE 1
Default Simulation Parameters

Maximum sensor transmission range	10 m
Number of sensor nodes	100
Number of sink nodes	3
Size of network area	$200 \times 200\ m^2$
Size of a zone	$40 \times 40\ m^2$
Probability to move out of a zone	20%
Probability to move back to home zone	100%
Maximum queue length	120 whole messages ($120 \times 200 bits$)
Message generation rate	$0.01/s$
Whole message length	200 $bits$
Bandwidth	10 $kbps$
Nodal moving speed	$0-5m/s$
γ	0.9
H	0.9
S_{max}	3
b_{max}	20

if $1-(1-\mathcal{F}_i^j)\prod_{m\in\Phi}(1-\xi_m) > \gamma$ **then**
  Break.
 end if
end for

By following Algorithm 1, sensor i sends message j to a set of neighbors with higher delivery probabilities (i.e., $\xi_i < \xi_{\psi_z}$) 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 no zero values, multihop relaying will take place.

6 Simulation Results

Extensive simulation has been carried out to evaluate the performance of the proposed DFT-MSN data delivery schemes. In our simulation, three sink nodes and 100 sensor nodes are randomly deployed in an area of $200 \times 200\ \mathrm{m}^2$. The whole area is divided into 25 nonoverlapped zones, each with an area of $40 \times 40\ \mathrm{m}^2$. A sensor node initially resides in its home zone. It moves with a speed randomly chosen between 0 and $5\ \mathrm{m/s}$. Whenever a node reaches the boundary of its zone, it moves out with a probability of 20 percent and bounces back with a probability of 80 percent. 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 percent. The message size is 200 bits. Each sensor has a maximum transmission range of $10\ \mathrm{m}$ and a maximum queue size of 120 whole messages (or, totally, 120×200 bits). The data generation of each sensor

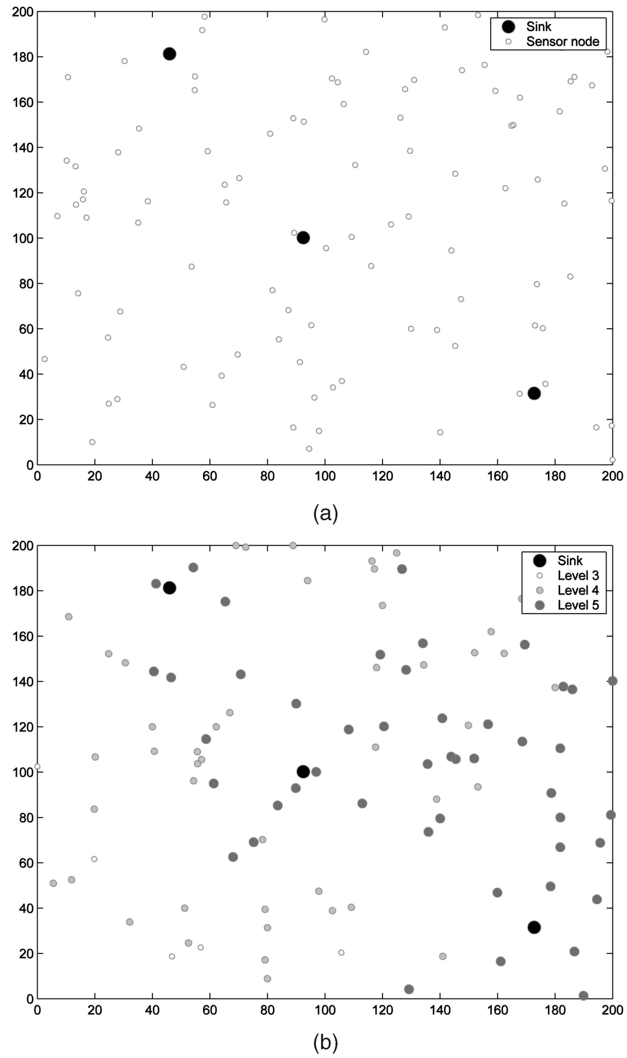
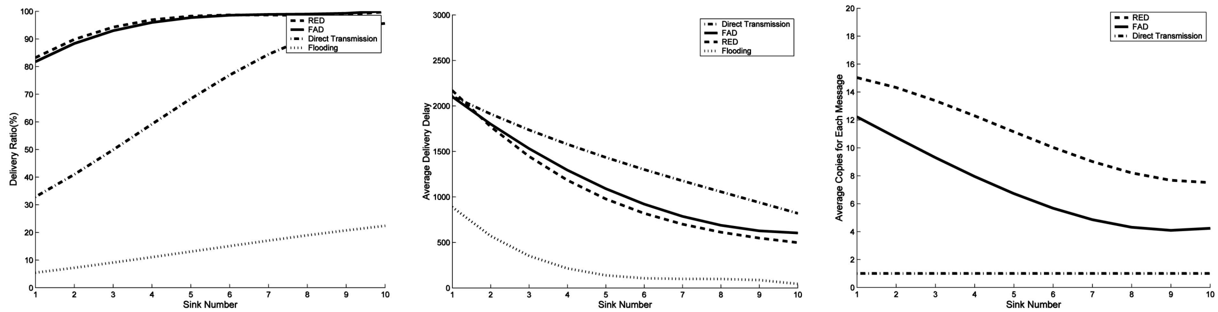


Fig. 6. Update of delivery probability. (a) Initial deployment. (b) 1,000 seconds later.

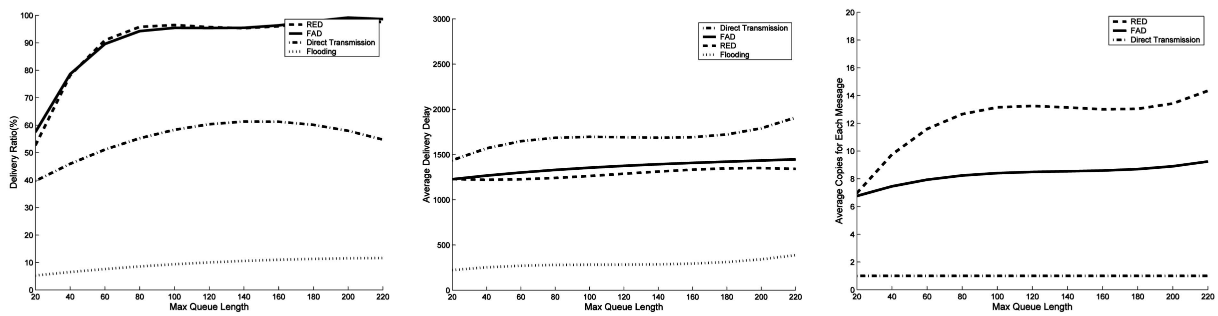
follows a Poisson process with an average arrival interval of $100\ \mathrm{s}$. The channel bandwidth is $10\ \mathrm{kbps}$. The fault tolerance threshold used in the FAD scheme is set to be $\gamma = 0.9$, while the delivery threshold used in RED scheme is set to be $H = 0.9$. In erasure coding, the maximum replication overhead (S_{max}) is set to 3 and the maximum number of blocks (b_{max}) is 20. The above default simulation parameters are summarized in Table 1.

The sensor node transmits its data messages according to our proposed DFT-MSN data delivery schemes. We first study the effectiveness of the delivery probability updating scheme. For clarity, we set the delivery probability of the sensor into five 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. 6a 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 1,000 seconds are illustrated in Fig. 6b, where the nodes closer to the sinks usually have higher delivery probabilities as expected.



(a) (b) (c)

Fig. 7. Impact of the number of sink nodes. (a) Average delivery ratio. (b) Average delay. (c) Average overhead.



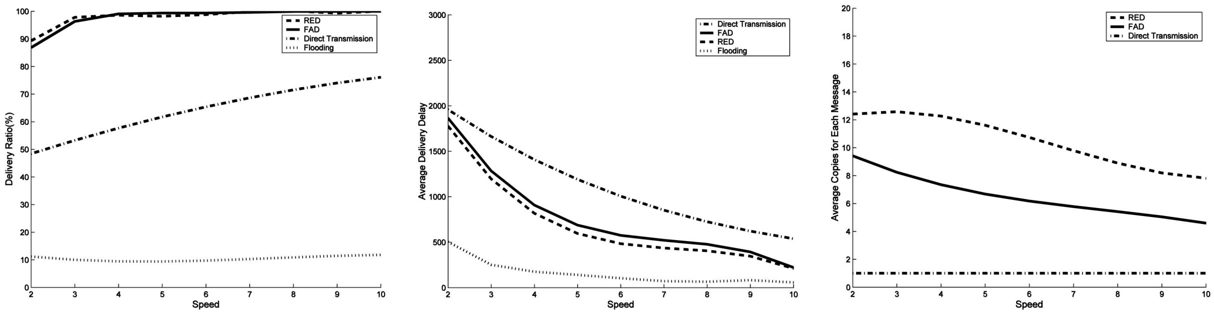
(a) (b) (c)

Fig. 8. Impact of maximum queue length. (a) Average delivery ratio. (b) Average delay. (c) Average overhead.

We vary several parameters to observe their impacts on the performance. Fig. 7 compares the performance of the proposed two data delivery schemes (i.e., RED and FAD) with the simple flooding approach and the direct transmission approach by varying the number of sink nodes in DFT-MSN. As shown in Fig. 7a, the delivery ratio increases with more sink nodes being deployed for all approaches. This is reasonable because the sensors then have high probabilities to reach the sink nodes, thus resulting in high delivery ratio. The proposed RED and FAD schemes always have higher delivery ratios than other approaches, especially when a small number of sinks are deployed. 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. 7b 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, since the RED scheme averagely transmits more copies for each message than the FAD scheme, it has a slightly lower delay than the FAD scheme. Clearly, the direct transmission approach suffers from the longest delay since messages can then be delivered only when the source node meets the sink.

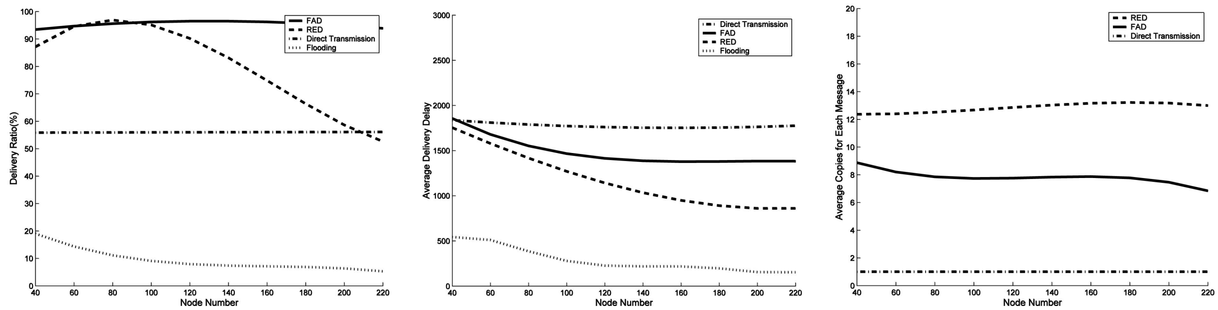
Energy consumption of the sensor is due mainly to data transmission. Thus, the more duplicated copies generated, the higher the energy consumption. As depicted in Fig. 7c, the number of message copies in direct transmission is always 1 since a sensor always transmits data messages to the sink directly. 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 both RED and FAD schemes with the increase of sink node density. This is because more sink nodes may shorten the message transmission path (in terms of number of hops) from the sensors to the sinks and accordingly reduce the transmission overhead. Additionally, the replication parameter S in RED becomes smaller when the nodal delivery probability (p) increases according to our discussion in Section 4.2. This may further compress the overhead of message transmission in the RED scheme. Meanwhile, we have also noticed that the FAD scheme always has smaller overhead than the RED scheme. This is reasonable because the message replication in FAD is done dynamically according to the delivery probabilities of the intermediate nodes and is thus more accurate compared with the RED scheme, where replication is performed by the source node only.

We also vary the maximum queue length of each sensor in our simulations, with results presented in Fig. 8. With an increase in maximum queue length, the delivery ratio increases for all approaches, as expected (see Fig. 8a). As shown in Fig. 8b, the queue length does not have a significant impact on the delay of the simple flooding approach, RED scheme, and FAD scheme. The delay of the direct transmission approach, however, increases 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 FAD approach can well control its transmission overhead (i.e., the number of copies generated) even when the available queue size is large. On the other hand, more duplicated copies are generated under the RED approach (see Fig. 8c) as a result of increasing the maximum queue size.



(a) (b) (c)

Fig. 9. Impact of nodal speed (m/s). (a) Average delivery ratio. (b) Average delay. (c) Average overhead.



(a) (b) (c)

Fig. 10. Impact of node density. (a) Average delivery ratio. (b) Average delay. (c) Average overhead.

Fig. 9 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 has a higher probability 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 FAD and RED schemes decreases slightly with the increase of nodal speed (as shown in Fig. 9c), making them most suitable for the network with varying nodal speeds.

Fig. 10 illustrates the impact of node density by varying the total number of sensor nodes in the network. As shown in Fig. 10a, the RED scheme is sensitive to the change of node density. Its delivery ratio first rises and then decreases sharply. This can be explained as follows: At first, the delivery ratio increases because more nodes participate in message relaying so that a message has a better chance to reach the sink node. When the node density becomes too high, however, excessive message propagation incurs a much higher buffer overflow due to limited queue size and limited communication bandwidth, especially for the nodes close to the sinks. As a result, the RED scheme is more suitable for a sparse network. In contrast, the FAD scheme has very steady performance with the increase of node density, exhibiting perfect scalability. The node density does not have a significant impact on the average overhead in both RED and FAD schemes, as shown in Fig. 10c. It is also noticed that data messages can be propagated faster in a network with higher nodal density, thus decreasing average delay in both RED and FAD schemes (if the message is delivered to the sink successfully), as shown in Fig. 10b.

7 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 first studied two basic approaches, namely, direct transmission and flooding. We analyzed 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 introduced an optimized flooding scheme that minimizes the transmission overhead of flooding. Then, we proposed two simple and effective DFT-MSN data delivery schemes, namely, the *Replication-Based Efficient Data Delivery Scheme (RED)* and the *Message Fault Tolerance-Based Adaptive Data Delivery Scheme (FAD)*. In order to achieve the desired data delivery ratio with minimum overhead, the former utilizes the erasure coding technology tailored for DFT-MSN, while the latter employs the idea of message *fault tolerance* in data transmission and queue management. Our results show that both schemes achieve a similar high message delivery ratio with acceptable delay. The RED scheme results in lower complexity in message and queue management, while the FAD scheme has a lower message transmission overhead.

REFERENCES

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

[2] O.B. Akan and I.F. Akyildiz, "Event-to-Sink Reliable Transport in Wireless Sensor Networks," *IEEE/ACM Trans. Networking,* vol. 13, no. 5, pp. 1003-1016, 2005.

[3] W.K. Lin, D.M. Chiu, and Y.B. Lee, "Erasure Code Replication Revisited," *Proc. Fourth Int'l Conf. Peer-to-Peer Computing,* pp. 90-97, 2004.

[4] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay Tolerant Network Architecture," IETF Internet draft, work in progress.

[5] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," *Proc. ACM 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 Comm. Magazine,* 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 IRB-TR-03-030, Intel Research, Berkeley, California, 2003.

[8] http://www.cens.ucla.edu/, 2006.

[9] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," *Proc. ACM Int'l Workshop Wireless Sensor Networks and Applications (WSNA '02),* pp. 88-97, 2002.

[10] http://down.dsg.cs.tcd.ie/sendt/, 2006.

[11] http://www.sics.se/cna/dtnsn/index.html, 2005.

[12] M. Ho and K. Fall, "Poster: Delay Tolerant Networking for Sensor Networks," *Proc. IEEE Conf. Sensor and Ad Hoc Comm. 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," *Proc. First Int'l Workshop Sensor Network Protocols and Applications,* pp. 30-41, 2003.

[14] http://www.princeton.edu/~mrm/zebranet.html, 2006.

[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)," *Proc. MobiHoc,* pp. 233-244, 2003.

[16] T. Small and Z.J. Haas, "Resource and Performance Tradeoffs in Delay-Tolerant Wireless Networks," *Proc. ACM SIGCOMM Workshop Delay Tolerant Networking and Related Topics,* 2005.

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

[18] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks," *Proc. IEEE Sixth Int'l Symp. World of Wireless, Mobile and Multimedia Networks (WOWMOM),* pp. 1-7, 2005.

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

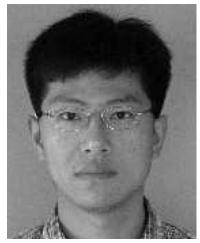
[20] W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," *Proc. MobiHoc,* pp. 187-198, 2004.

[21] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic Routing in Intermittently Connected Networks," *Proc. First Int'l Workshop Service Assurance with Partial and Intermittent Resources (SAPIR '04),* Aug. 2004.

[22] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket Switched Networks and Human Mobility in Conference Environments," *Proc. ACM SIGCOMM Workshop Delay-Tolerant Networking,* pp. 244-251, 2005.

[23] J. Ghosh, S. Yoon, H. Ngo, and C. Qiao, "Sociological Orbits for Efficient Routing in Intermittently Connected Mobile Ad Hoc Networks," technical report, Dept. of Computer Science and Eng., State Univ. of New York at Buffalo, 2005.

Yu Wang received the BS and MS degrees from the University of Electronic Science and Technology of China (UESTC) in 2000 and 2003, respectively. He is currently a PhD 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. He is a member of the IEEE.



Hongyi Wu received the PhD degree in computer science and the MS degree in electrical engineering from the State University of New York (SUNY) at Buffalo in 2002 and 2000, respectively. He received the BS 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 as the guest editor of two special issues of *ACM MONET*. He has published more than 50 technical papers in leading journals and conference proceedings. He received a US National Science Foundation CAREER Award in 2004. He is a member of the IEEE.