Mobile Sampling of Sensor Field Data Using Controlled Broadcast

Juzheng Li, Sol M. Shatz, *Senior Member*, *IEEE*, and Ajay D. Kshemkalyani, *Senior Member*, *IEEE*

Abstract—Mobile objects can be used to gather samples from a sensor field. Civilian vehicles or even human beings equipped with proper wireless communication devices can be used as mobile sinks that retrieve sensor-data from sampling points within a large sensor field. A key challenge is how to gather the sensor data in a manner that is energy efficient with respect to the sensor nodes that serve as sources of the sensor data. In this paper, an algorithmic technique called Band-based Directional Broadcast is introduced to control the direction of broadcasts that originate from sensor nodes. The goal is to direct each broadcast of sensor data toward the mobile sink, thus reducing costly forwarding of sensor data packets. The technique is studied by simulations that consider energy consumption and data deliverability.

Index Terms—Sensor data sampling, mobile object, directional broadcast, sensor networks.

1 Introduction

Recently, the concept of employing mobile objects (sometimes referred to as mobile sinks) to query a sensor network has been proposed [6], [17], [18], [21], [25], [32]. Applications can exploit this mobility to dynamically sample a sensor field. One high-level application scenario is illustrated in Fig. 1. A mobile object (car) is traveling along a path, and at some time and location (for example, T_0) it decides to take a sample of the sensor field, i.e., collect sensor data from "near-by" sensor nodes. The larger circle denotes the sampling region. Each sensor in that region will consequently be activated and reply with its locally sensed data. As the mobile object continues its travel, it reaches another location at time T_1 from which it initiates another sampling task.

There are three interesting features associated with the task of sensor field data sampling. First, due to the mobility of the sampling object, there are many options for selecting a sampling region, as opposed to the static sampling region associated with a static sink. Second, it is possible to employ commonly existing mobile objects, for example, taxis or buses, to help increase the coverage of the sensor field. So, it is possible to deliberately choose a mobile object and finely tailor its sampling regions to optimize a sampling task. Finally, in comparison to sensor nodes, mobile objects have relatively large (and adjustable) transmission ranges. Thus, they can trigger sampling-region sensors by the single-hop transmission of a sampling signal. This feature is elaborated in Section 3.2. The sampling distance is only constrained by the mobile object's transmission range, which should be more than sufficient for "local sampling" applications. For more "remote querying" of sensor data, an alternative scheme based on mobile-to-mobile node cooperation can be used, as in [25].

However, there are also challenges that arise from using these mobile sinks to gather sampled sensor data. One challenge is in controlling the process that sensors use to respond to a request for sensor data from a mobile sink. Because sensor nodes have a significantly smaller transmission range than mobile objects, sensor nodes must rely on multihop transmission of sensor data when they respond to the single-hop reception of the mobile object's sampling signal. This asymmetric communication property prevents sensor nodes from using a straightforward routing technique like reverse path forwarding. The fact that the mobile object continues to move after injecting its sampling signal further complicates the use of any explicit destination-oriented routing. Furthermore, general routing-tree based protocols [26], [31], are not well suited for this situation because route discovery implies high energy cost, and a discovered route might not be easily reused when faced with a series of highly dynamic sampling tasks. Also, because sensor networks are typically very large in scale, they do not naturally allow for a global IP address for each sensor node. This impedes use of traditional IP-based routing methods used in classical communication and wireless ad hoc networks. In addition, power and cost constraints make it impractical to assume GPS capability for very low-cost sensors needed for large-scale sensor network applications, and efficient accurate localization techniques are still in the research stage. Thus, for sampling large-scale sensor networks, it is not desirable to depend on routing protocols that require sensors to be location-aware, such as location-based GAF (Geographic Adaptive Fidelity) [30], and cluster-based LEACH (Low-Energy Adaptive Clustering Hierarchy) [11].

Finally, an implied requirement for sensor field sampling is that there is a time constraint imposed by the mobility of the sink object. To facilitate the collection of sensor data from the sampling region, it is helpful if all sensor data can

- *The authors are with the Department of Computer Science (MC 152), University of Illinois at Chicago, 1120 Science and Engineering Offices, 851 S. Morgan Street, Chicago, IL 60607-7053. E-mail: alex3278@gmail.com, {shatz, ajay}@uic.edu.*

Manuscript received 6 May 2009; revised 13 Aug. 2010; accepted 1 Oct. 2010; published online 2 Dec. 2010.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2009-05-0161.
Digital Object Identifier no. 10.1109/TMC.2010.233.

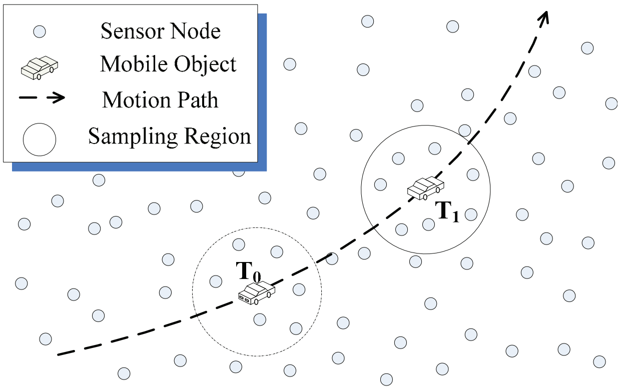


Fig. 1. Sensor field sampling.

be routed to the mobile object before the object has deviated significantly from the location at which it initiated the sampling task. This suggests that sensors should respond quickly upon receiving a sampling request, and the sensor-data propagation method should be highly efficient. In this paper, we make no assumptions about the nature of the sensor data, allowing for the possibility that sensors are heterogeneous with regard to data type (e.g., each sensor measures a different environmental property). Thus, deliverability of sensor data takes priority over performance of data aggregation operations.

The approach used in this paper is based on traditional software-based broadcast. Although it is understood that broadcast is to be generally avoided in sensor networks due to the problems associated with message flooding, there are significant advantages to using this basic mechanism, especially for the application at hand, sensor field sampling: broadcast is simple and does not require that sensor nodes be configured with special dedicated hardware; broadcast can be initiated immediately after receiving the sampling task since it requires no routing table or tree setup; and broadcast can naturally handle the mobile sink scenario since a sensor-data packet can reach the mobile object as long as the object is within transmission range of some broadcast, or rebroadcast, of that packet. The primary problem with using broadcast for gathering sensor data is that broadcast does not consider direction, and left unchecked would flood an excessively large geographic region. Note that this flooding could even extend beyond the intended sampling region, which means the omni-directional broadcast suffers from very low energy efficiency.

In this paper, we discuss a new broadcast-based sensor-data gathering mechanism, as introduced in [17]. The mechanism is optimized for the purpose of sensor-field data sampling by a mobile object. It is called *Band-based Directional Broadcast* since it uses the concept of bands created by partitioning the sampling region using multiple concentric circles (see Fig. 2 for a quick look). These bands are used to help control the direction of data flow of sensor data packets, without the need for sensor nodes having any sophisticated directional antenna [5], [10], [12]. The key idea is that our approach will reduce the propagation of packets that flow away from the sink mobile object—thus reducing broadcast events and sensor node energy consumption. This is accomplished by preventing packets that originate

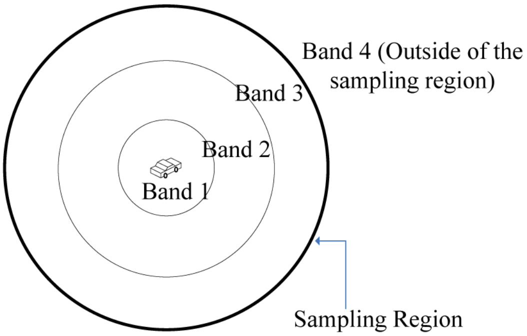


Fig. 2. A 4-band configuration.

from a sensor in any band from being propagated (rebroadcast) by sensors in a higher numbered band.

We know that media access control plays an important role in sensor node routing protocols. This is in part due to the fact that sensor nodes are low-power and have only a single signal-reception channel. One type of problem occurs when more than one sender simultaneously sends a packet to a common receiver. When no MAC scheduling protocols are used, packet collisions at the receiver side can occur, causing the receiver to obtain no useful signal. It has been shown that packet collisions significantly impair the performance of wireless sensor networks. There are two important conditions that increase the chances for packet collisions at a receiver node. The first condition is a large volume of broadcast activity within the vicinity of the receiving node; and the second condition is that these broadcast events occur within a short time-frame. Since our band-based scheme prunes many of the rebroadcast packets, it is expected to also reduce opportunities for packet collisions. Intuitively, less broadcasts/rebroadcasts will lead to fewer collisions.

Our band-based broadcast scheme can handle packet collisions by scheduling sensors in different bands to begin their initial broadcasts at different times. Using such a band scheduling technique introduces an explicit time drift between packets sent by nodes in two different bands, which consequently weakens the impact of the second condition for forming packet collisions.

The rest of this paper is organized as follows: Section 2 discusses related work. In Section 3, the design philosophy for Band-based Directional Broadcast is presented, and a practical method for implementing the approach is introduced. Section 4 explores and optimizes our scheme's capability for handling packet collisions via the concept of band scheduling. Section 5 provides a simulation study of the proposed approach, comparing it with existing protocols. Section 6 summarizes the results and suggests some future research opportunities.

2 Related Work

2.1 Broadcast Mechanisms and Scheduling

Since our sensor-data routing protocol is based primarily on broadcast, we now summarize several popular broadcast-based mechanisms and examine their applicability to the problem of sensor field data sampling by a mobile object.

Other nonbroadcast schemes that support transmitting sensor data to a sink node do not consider the case of a mobile sink and/or require the transmitting nodes to have some location awareness.

Simple flooding serves as the baseline of all broadcast mechanisms. In this protocol, a node rebroadcasts exactly once each message it receives. The rebroadcast (relaying) terminates when there are no more messages to broadcast. Generally, simple flooding has the best reliability and deliverability but the worst efficiency in terms of energy consumption [27]. A generalization of simple flooding is Probability-based Broadcast [27]. Upon receiving a packet that it has not previously received, a node rebroadcasts the packet with a probability of p, but discards it with probability $(1 - p)$. Simple flooding sets $p = 1$.

It is believed that there is an inverse relationship between the number of times a packet is received at a node and the probability of the node being able to reach additional areas on a rebroadcast [19]. So, in Counter-based Broadcast, a node maintains a counter and a timer for each unique packet it receives. The timer is used to control how long the node holds a packet before considering rebroadcasting of the packet. When the timer expires, the node checks how many duplicate copies of this specific packet have been received. If this number exceeds a previously assigned threshold, the packet is dropped; otherwise, a rebroadcast is initiated. In general, for a dense network, nodes will be less likely to rebroadcast packets, in comparison to sparse networks [19]. However, Counter-based Broadcast is inherently slow in terms of reaction time due to the need to wait for timer expiration before any rebroadcasts.

It is generally expected that with an increase in situational awareness, there is a benefit in performance. This is true for broadcast mechanisms [27]. When sensor nodes are granted more power—for example, the ability to acquire precise location information or 1-hop (even 2-hop) neighbor information—the broadcast methods become increasingly efficient [4], [19], [27]. However, sophisticated broadcast mechanisms (including area based, neighbor based, and distance based) do not fit well with the sensor sampling problem due to the requirements they would impose on the individual resource-constrained sensor nodes.

Another type of optimized broadcast can be collectively referred to as directional broadcast [5], [10], [12]. These methods generally require "enhanced" sensor nodes—nodes equipped with dedicated directional antenna, GPS, or other localization devices. For many applications, such a requirement may not be feasible due to cost issues or deployment methods. In this paper, we propose a directional broadcast scheme that does not rely on sensor nodes having location information via any special hardware or complex localization algorithms. Our method is a lightweight software-based scheme that can also work collectively with other hardware-based approaches under various sensor node setups.

Recent research [13] also exploits the potential use of scheduling methods to reduce transmission collisions in wireless environments. However, the basic network considered in [13] is quite different from that needed in this paper. In that work, the broadcasting involves a set of homogeneous nodes, or a single-layer network; and the goal is for a root node to broadcast and reach all other nodes in that layer. In contrast, the network environment in this paper is a heterogeneous architecture, with mobile objects on an "upper" layer injecting sampling signals into the underlying sensor-node layer to request retrieval of sensor data. Thus, our proposed scheduling approach exploits this unique characteristic—by using the sampling signal as a basis for sensor nodes to self-determine their broadcast schedules. Furthermore, the approach of [13] strictly requires that every node know its precise location, while we do not require such an assumption on any of the sensor nodes. Finally, [13] requires a virtual backbone construction procedure, which is not practical for highly dynamic, mobile sampling.

2.2 Use of Bands in Sensor Networks

There are some previous works that use similar notions of bands, but in different contexts. In [1], bands are introduced to help measure and compare the energy consumption of sensors at different distances from a sink. An algorithm is then proposed to avoid the sink-hole problem. Sensors are statically deployed into specific bands with adjusted transition ranges to achieve uniform energy depletion. In contrast to that work, our research focuses on dynamic band-computation and on using band knowledge to reduce rebroadcast of sensor data.

In [26], an idea for using bands to help conduct routing is introduced. The sensor field is divided into many slices (formed by coronas, which are like our bands, and wedges, which cut across bands). Routing trees are then constructed with the help of these slices. However, as pointed out in Section 1, although tree-based routing can achieve good performance, the building of a routing tree requires high energy cost and additional setup time. Furthermore, a discovered routing tree for one specific sampling task cannot be reused by other sampling tasks. The work of [26] mainly focuses on a static sink, fixed query region, and continuous monitoring. In that context, such overhead might be reasonable, but for a sequence of "one-shot," highly dynamic sampling tasks, such overhead cannot be justified. The mobility of sink nodes further demands a rapid response by sensor nodes.

2.3 Collision Handling

The problems associated with packet collisions in wireless systems have been broadly studied and motivated different collision handling methods, applicable to various situations [27]. From the perspective of collision handling, what is unique about our work is that it addresses collision handling within the specific context of our band-based broadcast mechanism. In particular, we demonstrate that our approach reduces the probability of packet collisions by explicitly reducing packet broadcast/rebroadcast events and by providing a natural mechanism for scheduling the transmission of packets based on band identification. Within each collision domain (each band), we employ a fairly conventional means for packet collision reduction—using random delays before broadcast [27]. It is useful to note that a range of other, more sophisticated collision avoidance protocols, such as the techniques used in [3], [8],

TABLE 1
Key Notations

Symbol	Definition
N	number of bands
R_{MOBILE}	sampling range
W	width of a band
R_{SENSOR}	sensor communication range
$T_{1\text{-}HOP}$	one-hop transmission time for sensor packets
θ	compensation factor for non-linear propagation of packets during multi-hop routing
T_{SIS}	time required for SIS injection by a mobile object
Ω_D	maximum random-delay value for band scheduling
$t(i)$	time for a SIS to reach all sensors in Band *i*, and for flooding of sensor data-packets in lower numbered bands
V	upper bound on mobile object's speed without band scheduling
$SV[1..N]$	band scheduling vector
S_i	Band *i*'s stage in a band schedule
$\|S_i\|$	duration for Stage *i*
ORT	overall reaction time to complete all stages of a band schedule
$ET(i)$	elapsed time until completion of Stage *i* in a band schedule
V_i	speed limitation imposed by Band *i* due to its stage in a schedule
V_{max}	maximum speed for a mobile object, imposed by a schedule
deliverability	fraction of sampling-region sensor readings received by mobile object

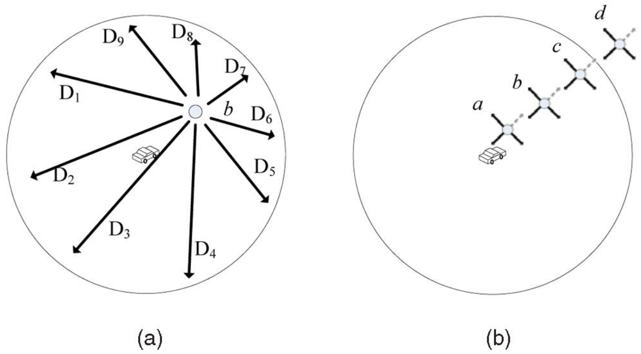
[14], [23], [24], could also be adopted by our approach to handle intraband collisions. However, in general, these more advanced approaches will impose additional requirements on the sensor nodes, such as carrier sensing capability, multichannels [3], [8], [14] or directional antenna [23], [24].

3 Band-Based Sensor Data Sampling

We first list in Table 1 the key notations used in this paper.

3.1 Overview of Band-Based Directional Broadcast

As discussed previously, upon receiving a request for sensor data, sensors in the sampling region will immediately react by broadcasting their sensed data. However, a fundamental problem here is that broadcast does not consider direction, and left unchecked would flood an excessively large geographic region. Considering Fig. 3a as an example, sensor b will flood its reply in all directions, illustrated by the nine different arrows shown in the figure. Note that although it is not explicitly shown, this flooding could even extend beyond the intended sampling region. Intuitively, it makes sense to try and control this flooding so that it is directed toward the mobile object, to minimize energy consumption associated with transmitting and receiving messages. For example, ideally we would like to constrain the flooding to the directions of D_2 and D_3.



(a) (b)

Fig. 3. Broadcasting sensor data.

A closer look at the flooding situation is provided in Fig. 3b. Note that only some of the sensor nodes and their broadcast/rebroadcast are depicted. To simplify the presentation of the general idea, we initially assume that the mobile object is static. (The impact of the object's mobility will be discussed in Section 3.3.) As desired, sensor b's response will be rebroadcast by sensor a and received by the mobile object; but b's packet will also propagate to other sensor nodes, for example, c, or even node d, which is outside of the sampling region. The rebroadcasts of b's data by nodes other than node a are not of direct benefit in terms of delivering the sensor data to the mobile object. Ideally, it would be desirable if each broadcast could avoid sending packets in a direction that is "away from" the location of the mobile object (those directions depicted by the dotted line segments). However, without the support of a directional antenna on individual sensor nodes, a packet broadcast propagates in all directions.

Despite this, we can seek to control the flooding at the receiver side. For example, upon receiving a packet from b, node c can choose to discard the packet, rather than initiating a rebroadcast. The challenge is for nodes to distinguish the arrival of packets from nodes that are located closer to the mobile object, without the assumption that nodes are location aware. In our solution, we only rely on nodes knowing their own bands. Thus, any node can identify received packets that originated from a different band, i.e., those packets that moved between bands.

Given a specific sampling task, we partition the entire sampling region into multiple concentric circles as shown in

Fig. 2, with the space between circles defining bands. The center of the circles is the mobile object's location at the time it initiates a sampling task. Each band has two radii associated with it—an inner radius and an outer radius that define the width of the band. We denote the innermost band as Band 1 and the outermost band as Band N ($N = 4$ in Fig. 2). Note that there are two special cases: 1) Band 1's inner radius is 0; and 2) Band N's outer radius is ∞, and its inner radius defines the boundary for the sampling region. Each sensor node has an associated band number corresponding to the band that contains the location of that sensor node. All bands within the sampling region, i.e., Band i, for $i < N$, have the same width.

Now, when a node broadcasts a packet, it should also attach its band number. Upon receiving a packet, a node will make the rebroadcast decision based on the band number attached to the packet. If the node's band number is less than or equal to the packet's band number, the node will rebroadcast that packet; otherwise the node discards the packet. Note that in achieving this directional broadcast property, there is a chance for packet-loss due to routing paths that are not allowed by the method. This issue will be explored in Section 5.3.3 of the simulation study.

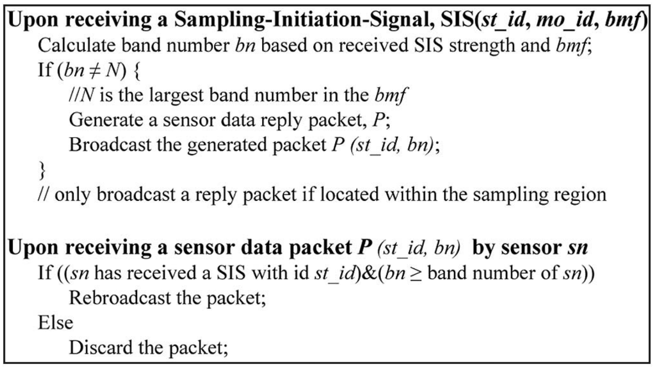
3.2 Band Identification and Sensor Protocol

While various methods can be used to associate sensor nodes with bands, including the techniques used in [1] and [26], we suggest an alternative method that is highly efficient and natural for the sensor-sampling problem. Each time a mobile object decides to sample a region of the sensor field, it issues a Sampling-Initiation Signal (SIS), which is broadcast with an intended sampling range, R_{MOBILE}. Using this sampling signal, sensors obtain partial and relative knowledge of their locations, and thus determine a band number. It is well-known that a radio signal attenuates as the distance between the transmitter and receiver increases [7]. Thus, when a mobile object issues a SIS, it can attach a function that maps signal strengths to band numbers. We assume that 1) the mobile object has knowledge of its own signal's attenuation pattern in its environment, and 2) the object also defines the number of bands to be used.

When a sensor node receives the SIS, it calculates its own band number based on the signal strength of the received signal and the mapping function attached to that signal. For now, we simply assume an ideal open-air environment, resulting in perfect circular bands as shown in Fig. 2. In Section 5.3.4, we will relax this assumption and study the impact of band assignment errors caused by imprecision in location estimation.

Definition 1. *A* Sampling-Initiation Signal *is a message broadcast by a mobile object in order to initiate the gathering of locally accessible sensor data within a given sampling region. The signal is represented as a 3-tuple, SIS = (ST_ID, MO_ID, BMF), where ST_ID is a unique identifier for the sampling task, MO_ID is the identifier of the mobile object, and BMF is a Band Mapping Function that maps signal strength to band number (i.e., BMF(SIS_Strength) → Band_Number).*

By piggy-backing the BMF on the SIS, each sensor node that receives the signal can use the mapping function to determine its own band. The BMF function is precalculated before the sampling signal is issued by the mobile object, and it is based on the size of the target sampling region, the desired total number of bands, and the characteristics of the mobile object's transmitter. A generic implementation of this function is shown below:



```
Upon receiving a Sampling-Initiation-Signal, SIS(st_id, mo_id, bmf)
    Calculate band number bn based on received SIS strength and bmf;
    If ($bn \neq N$) {
        //N is the largest band number in the bmf
        Generate a sensor data reply packet, P;
        Broadcast the generated packet P (st_id, bn);
    }
    // only broadcast a reply packet if located within the sampling region

Upon receiving a sensor data packet P (st_id, bn) by sensor sn
    If ((sn has received a SIS with id st_id)&($bn \geq$ band number of sn))
        Rebroadcast the packet;
    Else
        Discard the packet;
```

Fig. 4. Sensor-node broadcast protocol.

$$Band\_Number = \begin{cases} 1 & when\ \lambda_1 \leq SIS\_Strength < \lambda_0 \\ i & when\ \lambda_i \leq SIS\_Strength < \lambda_{i-1} \\ N & when\ \lambda_N = 0 \leq SIS\_Strength \\ & \quad < \lambda_{N-1}. \end{cases}$$

Note that in the above formula, λ_{N-1} establishes a threshold. Any sensor that receives the SIS with a signal-strength less than λ_{N-1} views itself as being outside of the sampling region. Consequently, this sensor would not reply to this SIS and not rebroadcast packets intended for this SIS, as will be presented later.

Definition 2. *Band* i $(1 \leq i \leq N) = \{x, y$ *coordinates* $|$ *a sensor node located at position (x,y) will receive the sampling signal with a signal-strength greater than or equal to* λ_i *but less than* $\lambda_{i-1}\}$.

A key idea of this paper is the way that bands are used to control flooding. We now describe the core behavior of sensor nodes by giving the sensor node broadcast protocol in Fig. 4. To simplify the presentation, we only consider one sampling task; handling multiple simultaneous tasks is straightforward due to the unique task ID in each SIS.

Sensor nodes react to two events: reception of a SIS sent by a sampling mobile object, and reception of sensor-data packets sent by other sensor nodes. In response to a SIS, a sensor node computes its own band number, which is then used to determine if the sensor is located within the sampling region. If so, the sensor node broadcasts a "response" packet containing its own sensor data, and also including its band number, which serves as a type of "location-stamp." When other sensor nodes receive this sensor-data broadcast message, they can determine from what band the message originated. This leads to the second event-driven action of a sensor node—responding to sensor-data packets. When a sensor node that is located in Band b_r receives a sensor-data packet, if the received packet originated in Band b_s ($b_s < b_r$), the packet should not be broadcast further; it is discarded. Otherwise the packet is rebroadcast, to continue its route toward the sampling mobile object. Our scheme aims to minimize energy

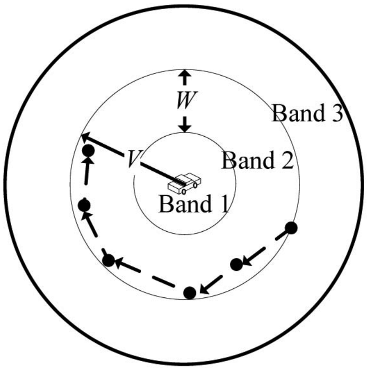


Fig. 5. A packet propagation path.

consumption by reducing the number of message broadcasts and message receptions. Like other broadcast algorithms, our protocol in Fig. 4 can be easily refined to use packet ids or hop counts to prevent packets from circulating within a band.

It is interesting to note that sometimes there is no neighbor node with band number same as or less than the current node, yet there may be a route to the sampling mobile object via a neighbor node in a higher band. To use such a route, the node may have to send the packet to neighbor nodes in higher bands to see if they can find routes toward the mobile object. But in our scheme, such detour is eliminated. We will study its impact in Section 5.

3.3 Impact of Band-Based Broadcast on Sink-Object Mobility

Our Band-based Directional Broadcast scheme can handle a mobile object as the sink. This capability comes from the nature of broadcast. As per the broadcast protocol, a sensor *sn* in Band i will flood its sensor data among the sensors in Band j, $1 \leq j \leq i$. So, as long as the mobile object moves within a "reasonable" speed range, it will have the opportunity to receive *sn*'s reply. This means that sensor nodes need not provide for cache management of packets that they forward, for the purpose of possible later delivery of those packets to the mobile object. This is significant since it further supports our goal of providing a low-complexity sensor node protocol. How fast is this "reasonable" speed? To avoid loss of any data packet (due to the packet not being able to reach the mobile object), a packet from Band i must be able to flood the entire set of bands $\{\text{Band } i, \text{Band } i-1, \ldots, \text{Band } 1\}$ before the mobile object moves out of the region associated with those bands, once the mobile object has injected a SIS. Assume that each band, excluding the outermost Band N, has the same width, W, and the mobile object moves at a speed V.

As an example, consider Fig. 5, where the mobile object is moving away from some sensor node located at the outer edge of Band 2. For the mobile object to receive a data packet originating from this sensor node, the packet must propagate to some sensor node that is located in a position where it is able to directly communicate with the mobile object. Note that in using the band-based broadcast protocol, this packet will not be able to propagate into Band 3. Thus, in terms of a worst-case situation, the packet will have to follow a maximum length propagation path, as illustrated in Fig. 5; and the packet must arrive at the shown destination node before the mobile object has moved out of transmission range of that destination node.

To guarantee the opportunity to receive a packet from a sensor located in Band i, the mobile object must not move any further than the outer edge of Band i. Thus, if $t(i)$ denotes the time required for a SIS to reach all sensors in Band i, plus the time for the sensor data-packets to flood the set of bands $\{\text{Band } i, \text{Band } i-1, \ldots, \text{Band } 1\}$, we can obtain the following relationship:

$$W \times i \geq V \times t(i). \tag{1}$$

Relationship (1) expresses that the distance traversed by the mobile object—after it has injected a SIS and during the time required for sensor data-packet propagation—must not exceed the distance defined by Band i. The time function in (1) can be characterized by the following (2):

$$t(i) = \frac{\pi \times W \times i}{R_{SENSOR}} \times T_{1-HOP} \times \theta + T_{SIS}. \tag{2}$$

The term $\pi \times W \times i$ represents the worst-case distance, which is half of the circumference of the ith band, as we noted previously in the example of Fig. 5, where $i = 2$. R_{SENSOR} is the communication range of a sensor node, T_{1-HOP} is the time required for a sensor node to broadcast a packet for one hop, θ is an error-margin parameter used to compensate for nonlinear propagation of packets during multihop routing [15], [25], [29], and T_{SIS} is the time required for SIS injection by the mobile object. Since individual sensor nodes only report a reading of their own sensor data, we assume each sensor-data response is small enough to fit into one data packet.

The first term in (2) is the flooding time for sensor data using multihop propagation by sensor nodes, while the second term is the time for only a single transmission by the mobile object. Since the first term dominates, we can ignore the second term, and simplify (2) as follows:

$$t(i) = \frac{\pi \times W \times i}{R_{SENSOR}} \times T_{1-HOP} \times \theta. \tag{2'}$$

By substituting (2)′ into (1), we obtain the following upper bound on the mobile object's speed:

$$V \leq \frac{R_{SENSOR}}{\pi^* \theta^* T_{1-HOP}}. \tag{3}$$

Surprisingly, this speed is not related to the band number i from where the broadcast of the sensor-data packet originated. Using the properties of contemporary sensor nodes [20], and a θ value of two,[1] (3) approximates the maximum speed as follows, which suggests that the approach is appropriate for environments using conventional mobile objects:

$$V \leq \frac{50 \text{ meters}}{\pi^* 2^* 50 \text{ ms}} \approx 160 \text{ meters/second}.$$

[1] This allows for a potential doubling of the multihop path length, in comparison to the optimal shortest path that exists when there is sufficient sensor node density to realize a linear path. This value is consistent with the results reported in [15].

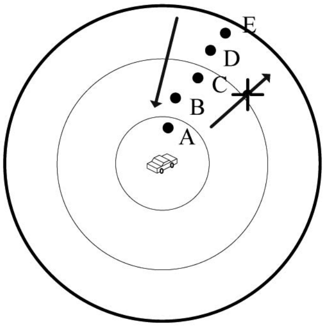


Fig. 6. Inbound versus outbound packets.

4 Packet Collision Handling

4.1 Band Scheduling

Media access control plays an important role in sensor-node routing protocols. High-end nodes, which are equipped with advanced hardware, can handle media access control easily by employing multichannels for sending and receiving. In this case, the requirements for media access control can be significantly relaxed. But, low-end nodes, which are designed to be low power and low cost, only have a single channel for receiving signals. For large-scale sensor networks that deploy low-end nodes, the problem of media contention must be addressed.

A serious type of contention happens when more than one sender simultaneously sends a packet to a common receiver. The result can be packet collision at the receiver side, and the receiver obtains no useful signal then. It has been shown that packet collision significantly impairs the performance of the wireless sensor networks—by expending more energy if retransmissions are used; or by reducing the number of successfully delivered packets if no retransmissions are allowed. As our approach does not use retransmissions, we are concerned with the latter effect. We capture the impact of the latter effect by defining a metric called *deliverability*.

Definition 3.

$$deliverability = \frac{\#\ of\ sensor\ replies\ from\ sensors\ in\ the\ sampling\ region}{\#\ of\ sensors\ in\ the\ sampling\ region}.$$

Notice that deliverability is affected by the number of collisions as well as the number of rebroadcasts (due to the effect mentioned at the end of Section 3.2).

There are two important conditions that increase the chances for packet collisions at a receiver node. 1) The first condition is a large volume of broadcast activity within the vicinity of the receiving node; 2) the second condition is that these broadcast events occur within a short time-frame. Since the algorithm in Section 3 prunes naturally the broadcast/rebroadcast of "outbound" packets that cross band boundaries (see Fig. 6), it is expected to also reduce opportunities for packet collisions. Intuitively, less broadcast/rebroadcast will lead to fewer collisions since it weakens the impact of the first condition.

Note that since the goal of the sampling task is to route sensor data to the mobile object, messages from higher bands must still propagate through lower bands, where collisions can still occur. By scheduling the sensor nodes to begin their broadcasts at different time slots, we can extend our band scheme to also reduce the negative impact of such packet collisions by reducing the "time-correlation" of the broadcast packets. To achieve this objective, we introduce the concepts of *stage* and *band-scheduling*. By using band scheduling, we introduce an explicit time lag between the broadcasting of packets sent by nodes in two different bands, which consequently weakens the impact of the second condition for causing packet collisions.

Definition 4. *For sensor nodes in Band i, there exists a time window, called the band's* stage *and denoted S_i, during which these nodes can broadcast their own sensor readings. Outside of this time window, these sensor nodes can only forward packets that originated in other (higher) bands.*

Employing a conventional means for packet collision reduction (during some specific stage), we assume that each sensor node delays for some random time before broadcasting its own sensor data [27]. This will reduce collisions within a band. The maximum random-delay value is denoted as Ω_D and is used later in formulating the duration of a stage.

Definition 5. *For an N-band configuration, there are $(N-1)!$ unique orderings of the $N-1$ stages for the bands that lie within the sampling region. Any one of these stage orderings is called a* band schedule *and the process of enforcing a particular band schedule is called* band scheduling.

Sensor node synchronization is explicitly required for band scheduling. Periodic sensor synchronization is also required because local sensor clocks can drift over time. Various techniques have been proposed to synchronize distributed sensor nodes [22]. One way to achieve synchronization is by using some explicit synchronization signal. Two drawbacks of this approach are that it can be difficult to predict when to issue such a synchronization signal and there is a relatively high cost associated with the need to issue such a special signal. However, in our scheme, the SIS itself can naturally serve as a synchronization signal, and this provides a number of advantages. First, the synchronization signal is automatically issued at the start of each round of sensor data gathering, which is exactly when the sensors need to synchronize. Furthermore, the SIS is issued by the sampling mobile object, which is much less energy constrained than individual sensor nodes. Finally, since the SIS is confined to the sampling region, for any sensor field data sampling task, only those sensors that are required to be synchronized do engage in synchronization.

There are $(N-1)!$ band schedules for an N-band configuration. For notational purpose, we introduce an array of stages, called a *Scheduling Vector* $SV[1..N-1]$. $SV[j]=k$ defines that Stage S_k is scheduled jth in the band-schedule sequence. A central issue regarding the concept of stages is how to decide the duration of a stage. Intuitively, the duration of a stage S_i (denoted $|S_i|$) should be large enough to allow all packets generated in Band i to complete their propagation, i.e., the packets complete the flooding of Bands i through 1. If this is the case, then when the next stage begins, nodes in Band i can be used exclusively for the purpose of forwarding packets that originate in outer bands, but not for originating any sensor-data packets. A similar calculation to the one used in Section 3.3 is now used to determine the duration of a stage. As before, we consider the worst-case propagation distance for sensor data packets, plus the worst-case delay value prior to the initial broadcast along that worst-case propagation path.

Specifically, the duration for Stage i is characterized as follows:

$$|S_i| = t(i) + \Omega_D = \frac{\pi^* W^* i}{R_{SENSOR}} * T_{1-HOP} * \theta + \Omega_D. \tag{4}$$

For an N-band configuration, the response to a mobile sink's SIS is complete when all $N-1$ bands have completed their stages, in a ordering defined by a given band schedule. The sum of the $N-1$ stage-durations is a constant, independent of the band schedule that is used. We call this sum the *Overall Reaction Time*, denoted as *ORT*.

Definition 6. *For an N-band configuration using band scheduling, the* ORT *is the elapsed time required to complete the sampling task.*

$$\begin{aligned} ORT &= \sum_{i=1}^{N-1} (t(i)) + (N-1)^* \Omega_D \\ &= \sum_{i=1}^{N-1} \left(\frac{\pi^* W^* i}{R_{SENSOR}} * T_{1-HOP} * \theta \right) + (N-1)^* \Omega_D. \end{aligned} \tag{5}$$

4.2 Tradeoffs

From the user's perspective, three parameters are of interest:

1. minimize *ORT*,
2. maximize deliverability, and
3. maximize speed.

From the system designer's perspective, minimizing energy consumption (measured in terms of number of message broadcasts/rebroadcasts and receives) is of interest. Of the parameters under control, assuming a fixed R_{MOBILE} and R_{SENSOR}, N is the most important. (Recall that the SIS allows a mobile object to define the number of bands used to cover a sampling region.) We now identify some tradeoffs in optimizing our objective metrics.

A higher N, which gives a lower W (assuming fixed R_{MOBILE}), implies the following:

1. A higher *ORT*, because $ORT \propto N^2 W \propto N$ from (5). As was done in Section 3.3, we can use properties of contemporary sensor nodes to compute *ORT* for different numbers of bands. Assuming a sampling region with radius 250 meters, which is the configuration for the simulation study in Section 5, Fig. 7 plots the resulting relationship between *ORT* and N.
2. Fewer collisions, which results in higher deliverability.
3. Less rebroadcasts, which results in a) lower energy consumption and b) some reduction in deliverability, if some packets cannot reach the mobile object using only paths allowed by the band-based scheme.
4. A decrease in the maximum speed allowed for a mobile object (See Section 4.3).
5. Less tolerance to errors in estimating their band numbers, leading to a decrease in deliverability.

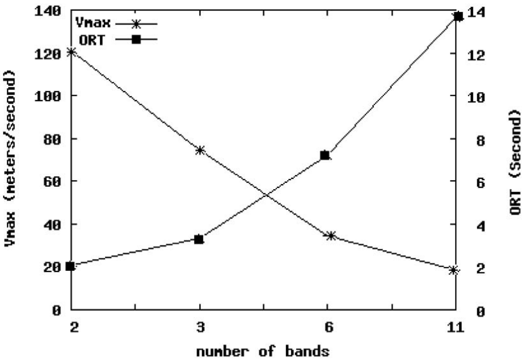


Fig. 7. Impact of bands on speed and reaction time.

Section 5 studies some of these trade-offs using simulation analysis.

4.3 Impact of Band Scheduling on Sink-Object Mobility

In (3), we established a theoretical upper limit on the speed of a mobile sink, with the requirement that the mobile sink is able to receive all sensor replies to its injected SIS from some *one* band, and this speed is independent of i. To ensure that the mobile object has the opportunity to receive all packets generated in Band i, packets from Band i should flood all of Band i through Band 1 before the mobile object (which is initially located at the center of Band 1) moves to a "higher" band than Band i. Otherwise those sensor-data packets will not be able to reach the mobile object since the band-based technique will prevent those packets from being transmitted to a higher band.

Intuitively, using band scheduling will extend the time required to collect sensor data packets from *all* bands, thus reducing the tolerable speed of a mobile sink. Also note from (4) that with band scheduling, the stage delay $|S_i|$ depended on i. We now study how the maximum speed of a mobile object is affected by band scheduling.

For any SV, if $SV[j] = k$, define $Index(k, SV) = j$, i.e., the *Index()* function maps a stage number to its position in the SV array. For any band schedule SV, the critical V for any Stage i, denoted as V_i, is as follows:

$$V_i = \frac{W^* i}{ET(i)} = \frac{W^* i}{\sum_{j=1}^{index(i,SV)} |S_{SV[j]}|}. \tag{6}$$

Here, E*T(i)* denotes the elapsed time required to complete Stage i, i.e., starting from the time of initial injection of the SIS. Clearly, the value of ET(i) depends on Stage i's position within the schedule. For example, if the stage is scheduled early, its ET-value will be relatively small.

The maximum allowable speed for the mobile object in that schedule is determined as the minimal value of V_i ($i \in [1, N-1]$) because exceeding any V_i value would result in the loss of packets belonging to Band i. Thus,

$$V_{max} = \min(V_1, V_2, \ldots V_{N-1}). \tag{7}$$

It is easy to understand that for the same N-band configuration, each of the $(N-1)!$ band schedules can result in different V_{max} values. We can identify the upper-bound and lower-bound of the allowed V_{max} for all possible schedules to obtain a deeper understanding of band scheduling. For example, a higher V_{max} value associated with some band schedule implies that that schedule has less impact (limitation) on the sampling mobile object's speed. This is an important advantage.

- Lower bound on V_{max}: To obtain the lower bound of V_{max}, we need to consider what band schedule will produce the minimum possible value for some V_i ($i \in [1, N]$). For (6), the numerator is minimized when $i = 1$. The denominator is maximized when the overall elapsed time to complete Stage i is maximized, i.e., when the denominator equals ORT. So, the lower bound occurs with any band schedule for which Stage 1 is scheduled last, i.e., $SV[N-1] = 1$. In this case, the lower bound on $V_{max} = V_1 = W/ORT$, independent of how other stages are scheduled. Intuitively this makes perfect sense—if we schedule Band 1 last, then the mobile object must move quite slowly if it is still going to be located within Band 1 at the time that sensors in Band 1 perform their data broadcasts. Note, $V_{max} = V_1 = W/ORT \propto W/WN^2 \propto 1/N^2 \propto R_{MOBILE}/WN^3$.
- Upper bound on V_{max}: We characterize this by the following theorem.

Theorem 1. *For an N-band configuration,*

1. *V_{max} is bounded by $W^*(N-1)/ORT$.*
2. *This bound is tight and is achieved when "inside-out" band scheduling is used, i.e., $SV[i] = i$.*

Proof. For an N-band configuration, the outermost band within the sampling region is Band $N-1$.

1. We partition all $(N-1)!$ schedules into two classes (a) and (b) as follows: Let the last scheduled band be k. Then either $k \neq N-1$ or $k = N-1$, which determines the two classes.

   a. $k \neq N-1$ (i.e., the outermost band, Band $N-1$, is not scheduled last ($k < N-1$)). Based on (6), for any schedule in this class:

   $$V_{N-1} = \frac{W^*k}{|S_1| + |S_2| + \cdots + |S_{N-1}|} = W^*k/ORT.$$

   From (7), $V_{max} \leq V_{N-1} = W^*k/ORT \leq W^*(N-2)/ORT$. So, for all schedules in this class, $V_{max} \leq W^*(N-2)/ORT$.

   b. $k = N-1$ (i.e., the outermost band, Band $N-1$, is scheduled last). For any schedule in this class,

   $$V_{N-1} = \frac{W^*(N-1)}{|S_1| + |S_2| + \cdots + |S_{N-1}|} = W^*(N-1)/ORT.$$

   So from (7), for all schedules in this class, $V_{max} \leq W^*(N-1)/ORT$.

   Therefore, the maximum attainable value for V_{max} across all $(N-1)!$ schedules is bounded by $W^*(N-1)/ORT$. This value may be attainable only when the outermost band is scheduled last.

2. We prove that this bound is tight by showing that the "inside-out" band scheduling method achieves the $W^*(N-1)/ORT$ value. Based on (4) and (6), we can obtain the following relationships for the "inside-out" band schedule:

$$V_1 = \frac{W^*1}{|S_1|} > V_2 = \frac{W^*2}{|S_1| + |S_2|} \cdots > V_{N-1} = \frac{W^*(N-1)}{|S_1| + |S_2| + \cdots + |S_{N-1}|} = W^*(N-1)/ORT.$$

- Therefore, when "inside-out" band scheduling is used, $V_{max} = V_{N-1} = W^*(N-1)/ORT$ from (7). Note, $V_{max} = V_{N-1} = W(N-1)/ORT \propto 1/N \propto R_{MOBILE}/WN^2$.

□

Because the "inside-out" schedule imposes the minimum constraint on the speed of a mobile object that is serving as a collector of sensor data, this band schedule is considered the *optimal schedule* for sensor field sampling. Compared with the maximal speed allowed for a mobile object in the absence of band scheduling (see (3)), the current V_{max} is smaller. Fig. 7 plots the relationship between V_{max} and N, the number of bands, assuming a sampling region with radius 250 meters. For a sampling task using a 3-band configuration, V_{max} is approximately 70 meters/second, less than half of the previously calculated value in Section 3.3 (other parameters have the same value). Once the number of bands reaches 11, V_{max} is reduced to 20 meters/second, which is approximately the lower bound for highway speed. Thus, from a practical concern this sets a reasonable limitation on the number of bands to be used. We use an 11-band configuration as the upper bound for N in our simulation study.

It is interesting to note that for conventional sensor network routing with a stationary base, the opposite band schedule—"outside-in"—might be more appropriate since it has the natural capability to facilitate data aggregation for a sampling task. Future research can seek to further exploit this potential for data aggregation in combination with our band scheduling.

5 Simulation Study

5.1 Simulation Setup

To evaluate the effectiveness of our protocol, simulations were conducted using a custom simulator. All simulation results were averaged over 100 simulation runs and include 95 percent confidence interval data. Since the simulation experiments only considered a single sampling task at a time, the monitored environment (i.e., sensor field) was set as a 1,000 by 1,000 meter square, with the sampling signal injected at the center (i.e., sampling region was centered at

(500,500)). The sampling region radius, λ, was set to be 250 meters. Although the mobile object's movement was based on a Random Way Point mobility pattern [2], [9], this does not impact the results of our particular study (in particular, the results in Sections 5.4.3 and 5.4.4). This is because we were concerned only with events that took place during each sampling task interval, and these intervals are sufficiently short in duration so that the probability of a mobile object changing direction or speed is very low. Specifically, the time intervals of concern are from when the mobile object injects an SIS packet to when the forwarding of sensor-data packets is complete, as measured by the *ORT* metric (Definition 6). Thus, for each simulation run, the collected data approximates a mobile object traveling with a fixed speed and direction.

Unless explicitly specified, the speed range was set to be 0 to 30 meters/second,[2] and direction was randomly chosen from one of four directions on a grid: east, west, south, and north. In our simulation, one time unit represented 50 milliseconds, the approximate time to send a packet on the 10 Kbit radio of Motes [16]. Also, as noted in Section 3.3, we assumed that sensor data generated by any sensor node is small enough to fit in one packet. We assumed that sensors had a 100 percent success rate for communicating with a mobile object located within 50 meters, while the sensor-to-sensor communication success rate depended on a selected communication model. We used two communication models which are described below. We varied the number of sensor nodes within the whole sensor field from 300 to 1,100 in increment steps of 200. It is easy to deduce that this configuration corresponds to a network degree (average number of neighbors in a node's communication range) ranging from 2.355 to 8.635.

- The first communication model assumed a perfect and simplified MAC layer with no collisions and a "binary" sensor-to-sensor communication model. We assumed that there was a near-perfect (95 percent) success rate for communication between nodes located within 50 meters of each other, but a 0 percent success rate otherwise.
- The second communication model assumed a more realistic situation. First, packet collisions were detected at the receiving side, and when there was a packet collision, those packets were discarded (i.e., we did not employ any retransmission). Second, a "decay" sensor-to-sensor communication model was used, meaning that the success rate for communication was a function of the distance between the sending and receiving sensors. We used the link quality model [28], but with the "effective" transmission range scaled to 50 meters, which is appropriate for state-of-art hardware platforms such as the TelosB node [20]. Finally, the range of the random delay was set to [0, 10] time units ($\Omega_D = 10$).

5.2 Studied Algorithms

We studied four different configurations of our band-based broadcast approach, using 2 bands (2B), 3 bands (3B), 6 bands (6B), and 11 bands (11B). The simulation results compare the Band-based Directional Broadcast with the two existing broadcast methods discussed in Section 2, Simple Flooding (SF) and Counter-based Broadcast (CB). The timer and counter were set to 3 and 10, respectively, for the latter algorithm. Simple Flooding and Counter-based Broadcast were selected for the comparison study because those techniques, like our approach, satisfy some base conditions in terms of being suitable for sensor sampling applications—they do not require sensors to be location-aware; they do not rely on the construction of routing tables; and they do not depend on any specialized hardware.

To prevent Simple Flooding and Counter-based Broadcast from flooding the entire sensor field (the whole 1,000 by 1,000 environment), the concept of a Return Hop Counter (RHC) [25] was used. Whenever a sensor node generates a reply in response to receiving a sampling signal, it will set an initial RHC value for the generated sensor-data packet and then broadcast the packet with that RHC value appended. Each time that packet is rebroadcast, the RHC value is decreased by one. When the RHC reaches zero, the packet is discarded. The calculation of the initial RHC value is based on the following formula, intended to capture the number of hops needed to route the packet from a source sensor node to the mobile object:

Definition 7. $\mathrm{R_{HC}} = R_{MOBILE}/R_{SENSOR}*\theta.$

Recall that we set $\theta = 2$ (See Section 3.3). For $\theta > 2$, the number of packets sent and received would increase for both Simple Flooding and Counter-based Broadcast.

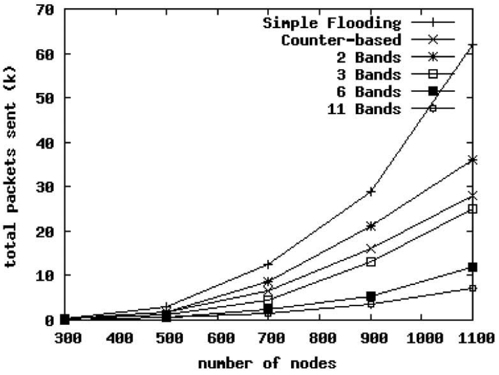
5.3 Band-Based Broadcast without Collisions

To understand the baseline behavior of our band-based broadcast method, we first studied the first communication model that did not consider packet collisions or use the band scheduling feature. Note that a key motivation behind the band-based approach is to reduce energy consumption due to rebroadcast of sensor data packets. To measure this property, we considered the following two criteria: total packets sent and total packets received, for servicing each sampling task. But, as we will see, inhibiting the rebroadcast of some packets might also negatively impact the successful delivery of some data packets to the mobile object. Thus, we also study deliverability for the different scenarios.

5.3.1 Total Packets Sent

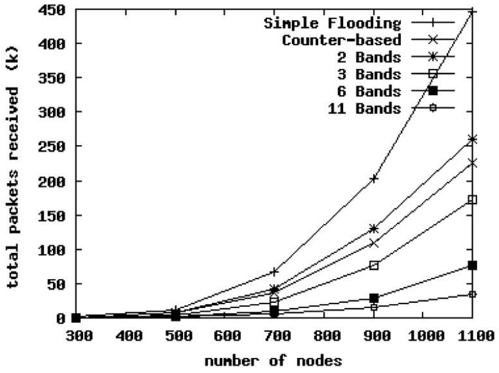
As an important factor of sensor energy consumption, packet sending plays a vital role. The simulation results for this feature are plotted in Fig. 8. As can be seen, our band-based broadcast approach performed well in this study, and the savings were more pronounced when the network density was higher. This is due to the fact that with Simple Flooding and Counter-based Broadcast, every generated packet will flood the entire region without any restriction on direction. Further, by increasing the number of bands, the total number of packets sent decreased. This is because

[2] We consider mobile objects across the spectrum of pedestrians walking (at a rate of approximately 1.25 meters/second) to high-speed vehicles (up to 30 meters/second). Since the lower bound is not significantly different from a stationary object, when considering the distance that can be traveled during any sampling task's overall reaction time, we set the speed range from 0 to 30 meter/second.



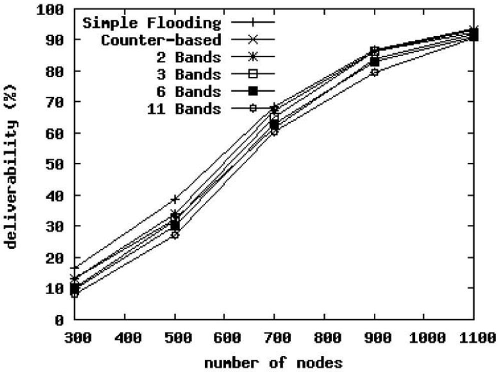
	N=300	N=500	N=700	N=900	N=1100
SF	357±41	2835±207	12736±554	29970±1135	62334±1413
CB	257±25	1693±129	6830±327	16023±535	27600±647
2 B	238±23	1800±220	8528±774	20527±908	35428±1033
3 B	207±21	1238±120	4606±418	13186±907	25029±1179
6 B	135±12	637±53	2428±179	5525±331	12540±651
11 B	117±11	490±36	1530±101	3726±216	7439±348

Fig. 8. Total packets sent: graphs and confidence interval data.



	N=300	N=500	N=700	N=900	N=1100
SF	1103±158	12961±1048	67024±3691	204595±9942	447457±14519
CB	740±99	7060±563	36078±1945	109804±4362	230137±6023
2 B	642±87	6944±987	43110±4209	126247±7703	264135±10338
3 B	540±71	4210±569	23921±2215	78984±6077	172553±998
6 B	281±39	1949±209	90576±885	27370±1870	75197±5143
11 B	187±25	1254±121	4847±429	14653±1124	35253±2327

Fig. 9. Total packets received: graphs and confidence interval data.



	N=300	N=500	N=700	N=900	N=1100
SF	16.56±2. 4	38. 89±3. 9	68. 35±4. 1	86.91±2. 19	93.54±0. 85
CB	13.29±2. 02	32.14±3. 1	61.68±4. 16	83.79±2. 2	91.82±1. 19
2 B	13.25±2. 3	33.89±4. 14	67.35±4. 39	86.50±3. 27	93.13±0. 48
3 B	10.29±2. 26	31.95±3. 49	65.04±4. 61	85.56±1. 76	92.14±1. 81
6 B	9.8±1. 8	30.2±3. 31	63.06±3. 84	82.78±2. 38	91.28±0. 91
11 B	8.22±2. 1	27.13±3. 4	60.39±3. 9	79.63±2. 3	90.45±1. 38

Fig. 10. Deliverability: graphs and confidence interval data.

with more bands, the broadcasts are further restricted within smaller regions.

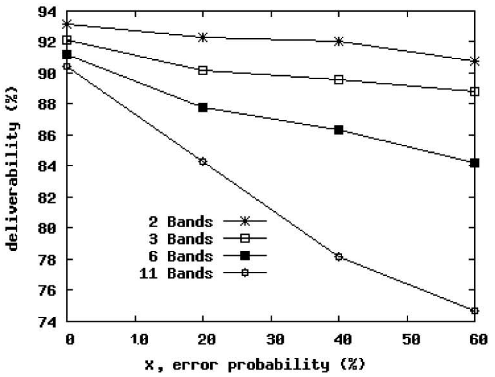
5.3.2 Total Packets Received

Another factor related to energy consumption of sensor networks is packet reception. As Fig. 9 shows, the trends for the studied protocols are similar to those for packet sending except that the y-scale now ranges up to 450 K total receiving packets. This is quite natural since one packet sent can cause multiple packet receptions. However, using our approach, the reception of a packet does not necessarily result in a packet being sent.

5.3.3 Deliverability

Since we have observed a significant reduction in the amount of packet sending and receiving with our band-based approach, it is important to also study how well our scheme performs when it comes to actual delivery of sensor-data to the sampling mobile object. A metric called *deliverability* is used (Definition 3).

Fig. 10 shows the simulation results for deliverability. One observation is that since our band-based scheme does eliminate some rebroadcasts, the delivery rate is slightly lower than with the other two approaches. This is because the band-based scheme might result in some lost connectivity (between a sensor node and the mobile object) due to the elimination of routing paths that rely on rebroadcasts that reach a higher numbered band. As an example, consider a node i that is located near the outer edge of Band 3 and whose only neighbor (within one hop) is some node j, located within Band 4. In this case, any multihop path from node i to the mobile object (located in Band 1) will not be



	x=0	x=20	x=40	x=60
2 B	93.13±0. 48	92.27±0. 44	92.01±0. 51	90.76±0. 38
3 B	92.14±1. 81	90.18±1. 28	89.57±0. 82	88.81±0. 92
6 B	91.28±0. 91	87.75±1. 12	86.34±0. 92	84.25±1. 37
11 B	90.45±1. 38	84.34±2. 18	78.19±2. 42	74.70±2. 65

Fig. 11. Deliverability with band assignment errors: graphs and confidence interval data.

realized using our band-based approach since node j will not rebroadcast any packet it receives from node i. A second observation is that this reduction in deliverability is more pronounced with lower sensor-node density and with more bands. This is expected, since both lower density and a larger number of bands increase the probability for situations like that in the above example.

Here again we can see the existence of a trade-off, this time between reducing packet rebroadcasts (and the associated consumption of energy) and increasing the chance of "lost" sensor-data packets. The loss-of-packets phenomenon appears to be relatively rare and is most likely to occur with low sensor node density and a high band count. A few points are important to note: First, in low density environments, any sensor-node routing technique will be vulnerable to packet loss due to "isolated" nodes, i.e., nodes with no single-hop neighbors. Second, since our focus is on *sampling* applications, we believe that the high-probability gain in energy savings is well worth the lower probability of not obtaining sensor-data results from every sensor node in the sampling region. But, if very high deliverability is critical, then the approach is most appropriate for high density situations. Finally, when we factored in the impact of packet collisions and band scheduling, the situation changes—our technique performed very well in terms of deliverability, especially with more bands. These results are presented in Section 5.4.2.

5.3.4 Impact of Band Assignment Errors

In the previously discussed simulations, we assumed that each sensor node correctly recognized its band, as if the sampling signal was transmitted in an ideal open-air environment. However, since signal strength can be affected by physical properties of the surrounding environment, there can be errors in band number assignments.

Since we use signal strength to determine bands, bands represent a type of coarse-location information. Thus, to account for band assignment errors that will occur in practice, we adopted a well-recognized error-precision probability model [7] for range-based localization: a node has $x\%$ probability to estimate its location with an error larger than y meter and $(100 - x\%)$ probability to estimate its location within an error of y meter. Based on [7], we set y to be 15 and varied $x\%$ among 0, 20, 40, and 60 percent. Thus, for each sensor in the environment, a probabilistically derived deviation was added to the sensor's actual location, and the sensor's band number was then determined. For this study, 1,100 nodes were deployed. The simulation results are plotted in Fig. 11. Note that a location estimation error does not imply a band assignment error. Consider a node locating at the center of a band with width W. As long as the location error is smaller than $W/2$, the node can still correctly recognize its band.

We found that as the error probability increased, the delivery rate decreased. However, the decrease was not highly significant, even with an error probability as high as 60 percent. This observation is consistent with a design feature of our band-based approach—it only needs coarse-grain, relative location information, rather than precise location information. As concluded in [7], a 20 percent error probability is a normal rate. Thus, we believe that our scheme performs well even if many nodes do not self-identify their accurate location.

Another observation is that when we employed more bands, the scheme became more sensitive to errors in band assignment. This is simply due to the fact that with more bands, the width W of each band gets smaller, and so the "error tolerance" capability $(W/2)$ also gets smaller. Therefore, using more bands increases the probability that a location estimation error will result in a band assignment error. Not unexpectedly, there is a trade-off to be considered when it comes to choosing the number of bands to use.

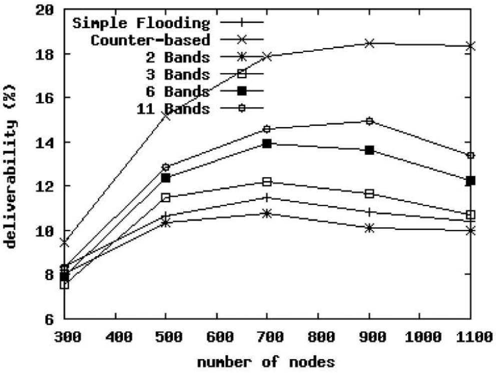
5.4 Band-Based Broadcast with Collisions

Our first set of simulations using the no-collisions and binary sensor-to-sensor communication model showed encouraging results. We then substituted the simplified communication model by the packet collision and decay communication model to study further the packet collision issues and our band scheduling feature.

5.4.1 Deliverability without Band Scheduling

We first investigated the deliverability issue under the collision-with-decay communication model. The results are plotted in Figs. 12 and 13. For these two figures, the band scheduling technique is not employed. Note that Fig. 12 is essentially a counterpart of Fig. 10 while Fig. 13 further explores the underlying properties of Fig. 12.

In Fig. 12, we noticed that in general there is a significant drop in deliverability in comparison to the simplified communication model (see Fig. 10). The deliverability was



	N=300	N=500	N=700	N=900	N=1100
SF	8.41±1. 08	10.64±0. 94	11.51±0. 71	10.81±0. 48	10.38±0. 49
CB	9.44±1. 16	15.16±1. 47	17.88±1. 02	18.48±0. 84	18.34±0. 59
2 B	8.03±1. 09	10.36±0. 96	10.77±0. 74	10.21±0. 53	9.98±0. 45
3 B	7.54±1. 1	11.46±1. 01	12.17±0. 69	11.64±0. 61	10.68±0. 43
6 B	7.91±1. 21	12.35±0. 90	13.93±0. 75	13.65±0. 59	12.27±0. 48
11 B	8.3±1. 15	12.86±0. 89	14.57±0. 68	14.92±0. 47	13.37±0. 43

Fig. 12. Deliverability without scheduling: graphs and confidence interval data.

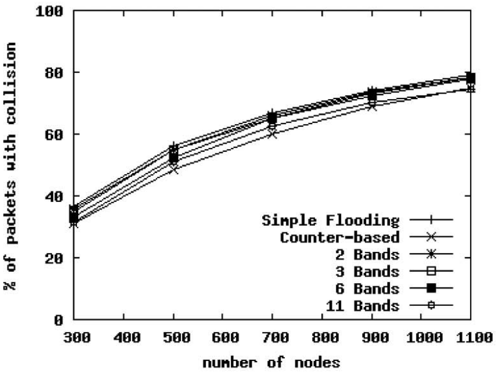
less than 20 percent, regardless of sensor density. We also found that the deliverability first slightly increases with the sensor density, but then drops. This is generally because at low network density, many sensors simply failed to find a neighbor to relay query results; as density increased, each node had more neighbors to help relay query results; however, there was then a higher probability of collisions from the routing of query results. Since we were using a MAC-layer protocol with receiver-side collision detection and no retransmission, collisions had a significant negative impact on the number of query results that were received at the mobile object, decreasing the total deliverability.

We also observed that for our band-based scheme, increasing the number of bands increases deliverability. Recall that this was not the case in Section 5.3, where packet collisions were ignored. We attribute this finding to the ability of our band-based scheme to reduce total packets sent and received. With more bands, fewer packets are broadcast, resulting in a lower probability for packet collisions.

Finally, we noticed that simple flooding has low deliverability while Counter-based Broadcast has the best deliverability. It is quite clear that the high volume of packets sent and received has a negative impact on performance for simple flooding. However, it is surprising to observe that Counter-based Broadcast behaves better even though it actually sends and receives more packets than our band-based scheme. Recall there are two conditions for forming a packet collision. In addition to reducing the number of broadcasts, the Counter-based Broadcast mechanism can also delay some packet rebroadcasts, thus negating the second condition for collisions. This is why the counter-based method outperforms our band-based scheme. Now it is interesting to see how the band scheduling feature, which can also weaken the second condition for collisions, can boost the performance of the band-based scheme. Fig. 13 shows the percentage of packets that encounter packet collisions. It basically shows a reversal of the pattern of deliverability in Fig. 12. The explanation follows from the above discussion, and reinforces our conclusion.

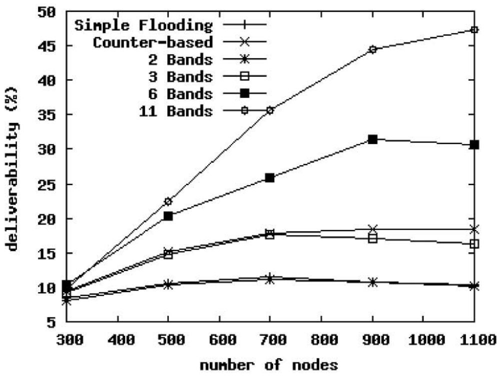
5.4.2 Deliverability with Band Scheduling

Our next study includes the band scheduling feature. The results are plotted in Figs. 14 and 15 using the optimal inside-out band scheduling method (see Section 4.2). We observed significant improvement in Fig. 14 over Fig. 12, especially when we deployed more bands. For example, the 6-band configuration shows about a 100 percent improvement over the result without band scheduling; for the 11-band configuration there is almost a 200 percent improvement. The curves for simple flooding and Counter-based Broadcast are reproduced again to show comparison. One interesting pattern in Fig. 14 is that as network density increases, deliverability decreases for 2-band and 3-band configurations; and it increases for 6-band and 11-band configuration. This is due to the fact that with more bands, more time (a longer *ORT*) is needed (see (5)). As a consequence, the probability for packet collisions is reduced. Fig. 15 shows the



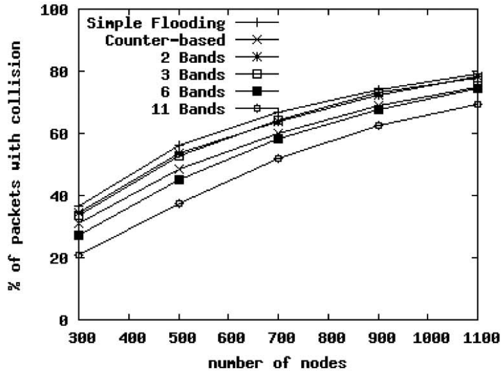
	N=300	N=500	N=700	N=900	N=1100
SF	36.76±1. 6	56.37±0. 69	66.85±0. 44	74.17±0. 26	79.13±0. 16
CB	31.14±1. 7	48.39±0. 85	59.95±0. 59	68.86±0. 33	74.86±0. 22
2 B	35.59±1. 99	54.95±1. 08	66.03±0. 72	73.76±0. 56	78.47±0. 37
3 B	34.69±2. 1	54.81±1. 19	65.31±0. 71	73.24±0. 44	78.48±0. 35
6 B	33.18±2. 18	52.48±1. 34	65.20±0. 62	72.39±0. 53	77.75±0. 35
11 B	31.62±2. 0	51.22±1. 29	62.59±0. 79	70.4±0. 48	74.38±0. 38

Fig. 13. Packet collisions without scheduling: graphs and confidence interval data.



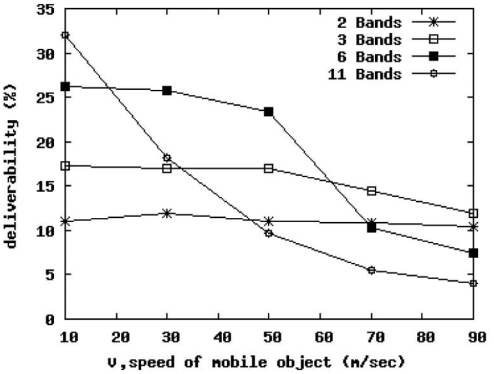
	N=300	N=500	N=700	N=900	N=1100
SF	8.41±1.08	10.64±0.94	11.51±0.71	10.81±0.48	10.38±0.49
CB	9.44±1.16	15.16±1.47	17.88±1.02	18.48±0.84	18.34±0.59
2 B	8.14±1.11	10.45±0.98	11.05±0.79	10.69±0.62	10.12±0.48
3 B	9.2±1.37	14.73±1.19	17.57±0.66	16.97±0.53	16.37±0.49
6 B	10.39±1.54	20.33±2.1	25.79±0.21	31.44±0.53	30.63±0.70
11 B	9.71±1.47	22.46±2.03	35.57±1.81	44.54±1.56	47.28±0.97

Fig. 14. Deliverability with scheduling: graphs and confidence interval data.



	N=300	N=500	N=700	N=900	N=1100
SF	36.76±1.6	56.37±0.69	66.85±0.44	74.17±0.26	79.13±0.16
CB	31.14±1.7	48.39±0.85	59.95±0.59	68.86±0.33	74.86±0.22
2 B	34.37±2.13	53.76±1.48	64.03±0.88	72.26±0.69	78.11±0.53
3 B	33.78±1.89	52.66±1.57	64.24±0.93	73.13±0.59	77.83±0.51
6 B	27.30±2.03	45.20±1.49	58.48±0.90	67.79±0.68	74.33±0.45
11 B	20.68±2.25	37.6±1.56	51.91±0.92	62.57±0.69	69.2±0.48

Fig. 15. Packet collisions with scheduling: graphs and confidence interval data.



	V=10	V=30	V=50	V=70	V=90
2 B	11.07±1.09	11.88±0.96	11.09±0.73	10.81±0.94	10.45±0.61
3 B	17.32±2.02	16.98±1.62	17.03±1.59	14.40±1.78	11.85±0.71
6 B	26.22±1.81	25.75±1.93	23.44±1.49	10.35±1.52	7.43±1.26
11 B	32.09±2.53	18.20±2.1	9.63±1.54	5.52±1.21	4.06±0.99

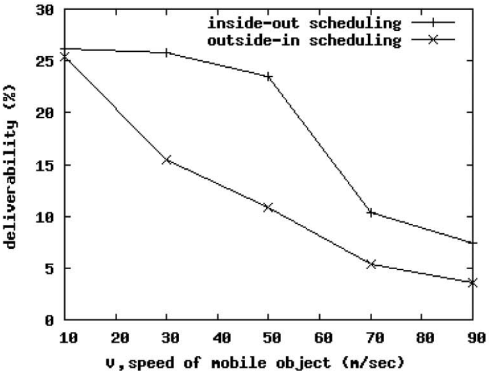
Fig. 16. Deliverability versus speed: graphs and confidence interval data.

percentage of packets that encounter packet collisions. It again basically showed a reversal of the pattern for deliverability in Fig. 14 and reinforced our conclusion.

5.4.3 Impact of Bands on Mobility

In Section 4.1, we analytically observed that as the number of bands increases, the sensor-data gathering delay increases, which can have a direct impact on sink object mobility. In particular, if the data gathering delay increases significantly, it may be necessary to constrain the speed of the mobile object in order to avoid a resulting reduction in deliverability of sensor-data packets. To evaluate this property, we performed a study with the results plotted in Fig. 16. We used 700 nodes (middle of our range) and the inside-out (optimal) scheduling method. As we noted previously, the duration of any sampling task is sufficiently short so that the results collected from each simulation run approximate a mobile object traveling with a fixed speed and direction. In both Figs. 16 and 17, the mobile object's speed is explicitly specified.

We found that for 2-band and 3-band configurations, the deliverability is not seriously affected by the mobile object's speed. However, for 6-band and 11-band configurations, as the mobile object's speed increased, the deliverability dropped significantly, especially for the 11-band scheme. Also, we found that when the mobile object traveled at higher speed ranges (50-90 meters/second), 6-band and 11-band configurations actually underperformed the 2-band and 3-band configurations. This is simply due to the growing *ORT* value. The mobile object can travel out of some band's region before sensor nodes in that region have flooded the entire region. Overall, the observed results confirmed our theoretical conclusions of Section 4.2.



	V=10	V=30	V=50	V=70	V=90
outside-in	25.41±1. 67	15.39±1. 64	10.90±1. 03	5.33±0. 74	3.61±0. 33
inside-out	26.22±1. 81	25.75±1. 93	23.44±1. 49	10.35±1. 52	7.43±1. 26

Fig. 17. Deliverability versus speed for IO and OI scheduling: graphs and confidence interval data.

5.4.4 *Band Scheduling Scheme*

One further interesting study focused on the optimal band scheduling method. Recall that the inside-out band scheduling is optimal in terms of imposing the least restriction on the mobile object's mobility. Fig. 17 shows the results of a study of two scheduling methods on a 6-band configuration with 700 nodes (the middle of our range). The two band schedules are the optimal (inside-out) schedule, IO, and the reverse (outside-in) schedule, OI.

We found that the optimal scheduling method consistently outperformed its peer, with a performance gain of approximately 100 percent when the mobile object travels at a relatively high speed (50 meters/second and over). Further, we noticed that the deliverability for the outside-in scheduling method drops sharply as the mobile object's speed increases. It is interesting to understand that this is due to the fact that since the outside-in scheduling method starts from the outermost band, as the mobile object moves it may move out of the range of some inner bands, even before those bands have a chance to initiate their stages. This can result in a complete loss of sensor readings from those bands. Although we only showed results for a 6-bands configuration; other band configurations have similar findings.

6 Conclusion

We proposed a simple, energy-efficient protocol to aid sensor-field sampling by a mobile object. The protocol exploits the concept of bands to limit the propagation of sensor data broadcasting, providing a form of directional broadcast based on software control. Methods for defining and using bands were presented. Extensive simulations under two communication models were conducted to evaluate the performance and trade-offs of our band-based scheme. The first communication model assumed no collisions and a binary sensor-to-sensor communication model. The second communication model assumed collisions, and a decay communication model. The simulations indicated that the band-based scheme is quite efficient in directional broadcast, and moreover, performs much better than default omnidirectional broadcast.

The scheme proposed in this paper is mostly a generalized model. It can be further optimized for different types of applications. For example, there are several directions for future work: 1) Handling data aggregation. Intuitively, some band schedules, for example, the outside-in schedule, have the natural capability to facilitate data aggregation for a sampling task. Their potential should be further exploited; 2) coordinating multiple sampling tasks. It is possible to have multiple sampling tasks, initiated by the same or different mobile objects, whose sampling regions overlap. It is desirable to have an efficient coordination mechanism such that overlapped regions need only reply once for the sampling tasks. The algorithm for multiroot multiquery optimization for a static network [33] is a potential candidate to be adapted to the mobile environment; 3) avoiding packet loss. With our band-based approach there may not always be a next-hop node located in the same band, or lower band, and this will stop the propagation of sensor-data packets. In this case, some sensor data packets may become lost in terms of reaching the mobile object. As we discussed, this is a trade-off of utilizing the band-based broadcast. Future research can consider some techniques to selectively allow rebroadcast of packets that would otherwise be disallowed by the currently proposed band-based mechanism. The purpose would be to avoid packet loss, while preserving a form of controlled broadcast.

Acknowledgments

This material is based upon work supported by the US Army Research Office under grant number W911NF-05-1-0573.

References

[1] H.M. Ammari and S.K. Das, "Promoting Heterogeneity, Mobility, and Energy-Aware Voronoi Diagram in Wireless Sensor Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 19, no. 7, pp. 995-1008, July 2008.

[2] C. Bettstertter, G. Resta, and P. Santi, "The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks," *IEEE Trans. Mobile Computing,* vol. 2, no. 3, pp. 257-269, July-Sept. 2003.

[3] C. Chen, E. Seo, H. Kim, and H. Luo, "SELECT, Self-Learning Collision Avoidance for Wireless Networks," *IEEE Trans. Mobile Computing,* vol. 7, no. 3, pp. 305-321, Mar. 2008.

[4] A. Chiganmi, M. Baysan, K. Sarac, and R. Prakash, "Variable Power Broadcast Using Local Information in Ad Hoc Networks," *Ad Hoc Networks,* vol. 6, no. 5, pp. 675-695, July 2008.

[5] F. Dai and J. Wu, "Efficient Broadcasting in Ad Hoc Networks Using Directional Antennas," *IEEE Trans. Parallel and Distributed Systems,* vol. 17, no. 4, p. 335, Apr. 2006.

[6] E. Ekici, Y. Gu, and D. Bozdag, "Mobility-Based Communication in Wireless Sensor Networks," *IEEE Comm.,* vol. 44, no. 7, pp. 56-62, July 2006.

[7] E. Elnahrawy, X. Li, and R.P. Martin, "The Limits of Localization Using Signal Strength: A Comparative Study," *Proc. First Ann. IEEE Comm. Soc. Conf. Sensor and Ad Hoc Comm. and Networks,* Oct. 2004.

[8] J. Garcia-Luna-Aceves and A. Tzamaloukas, "Receiver-Initiated Collision Avoidance in Wireless Networks," *Wireless Networks,* vol. 8, nos. 2/3, pp. 249-263, Mar. 2002.

[9] M. Garetto and E. Leonardi, "Analysis of Random Mobility Models with Partial Differential Equations," *IEEE Trans. Mobile Computing,* vol. 6, no. 11, pp. 1204-1217, Nov. 2007.

[10] H. Gossain, C. Cordeiro, and D.P. Agrawal, "Minimizing the Effect of Deafness and Hidden Terminal Problem in Wireless Ad Hoc Networks Using Directional Antennas," *Wireless Comm. and Mobile Computing,* vol. 6, no. 7, pp. 917-931, Nov. 2006.

[11] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Sensor Networks," *Proc. Hawaii Int'l Conf. System Sciences,* Jan. 2000.

[12] C. Hu, Y. Hong, and J. Hou, "On Mitigating the Broadcast Storm Problem with Directional Antennas," *Proc. IEEE Int'l Conf. Comm.,* May 2003.

[13] C.H. Huang, P. Wan, J. Deng, and Y.S. Han, "Broadcast Scheduling in Interference Environment," *IEEE Trans. Mobile Computing,* vol. 7, no. 11, pp. 1338-1348, Nov. 2008.

[14] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-Hop Wireless Network Performance," *Wireless Networks,* Special Issue: Selected Papers from ACM MobiCom '03, vol. 11, no. 4, pp. 66-80, July 2003.

[15] J. Kuruvila, A. Nayak, and I. Stojmenovic, "Hop Count Optimal Position-Based Packet Routing Algorithms for Ad Hoc Wireless Networks with a Realistic Physical Layer," *IEEE J. Selected Areas in Comm.,* vol. 23, no. 6, pp. 1267-1275, June 2005.

[16] P. Levis and D. Culler, "Maté: A Tiny Virtual Machine for Sensor Networks," *Proc. 10th Int'l Conf. Architectural Support for Programming Languages and Operating Systems,* Oct. 2002.

[17] J. Li and S.M. Shatz, "Sampling Sensor Fields Using a Mobile Object: A Band-Based Approach for Directional Broadcast of Sensor Data," *Proc. IASTED Int'l Symp. Distributed Sensor Networks,* Nov. 2008.

[18] L. Lilien, "A Taxonomy of Specialized Ad Hoc Networks and Systems for Emergency Applications," *Proc. First Int'l Workshop Mobile and Ubiquitous Context Aware Systems and Applications,* Aug. 2007.

[19] W. Peng and X. Lu, "On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Netoworks," *Proc. ACM Int'l Symp. Mobile Ad Hoc Networking and Computing,* Aug. 2000.

[20] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research," *Proc. Int'l Symp. Information Processing in Sensor Networks,* Apr. 2005.

[21] N. Sadagopan, B. Krishnamachari, and A. Helmy, "Active Query Forwarding in Sensor Networks (ACQUIRE)," *Ad Hoc Networks,* vol. 3, no. 1, pp. 91-113, Jan. 2005.

[22] B. Sundararaman, U. Buy, and A.D. Kshemkalyani, "Clock Synchronization in Wireless Sensor Networks: A Survey," *Ad Hoc Networks,* vol 3, no 3, pp. 281-323, 2005.

[23] K. Sundaresan and R. Sivakumar, "A Unified MAC Layer Framework for Ad-Hoc Networks with Smart Antennas," *IEEE Trans. Networking,* vol. 15, no.3, pp. 546-559, June 2007.

[24] M. Takai, J. Martin, A. Ren, and R. Bagrodia, "Directional Virtual Carrier Sensing for Directional Antennas in Mobile Ad Hoc Networks," *Proc. ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc '02),* June 2002.

[25] S. Tian, S.M. Shatz, Y. Yu, and J. Li, "Querying Sensor Networks Using Ad-Hoc Mobile Devices: A Two Layer Networking Approach," *Ad Hoc Networks,* vol. 7, no. 5, pp. 1014-1034, July 2009.

[26] A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, and K. Jones, "Training a Wireless Sensor Network," *Mobile Networks and Applications,* vol. 10, nos. 1/2, pp. 151-168, Feb. 2005.

[27] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," *Proc. Third ACM Int'l Symp. Mobile Ad Hoc Networking and Computing,* June 2002.

[28] A. Woo, T. Tong, and D. Culler, "Taming the Underlying Challenges of Multihop Routing in Sensor Networks," *Proc. First ACM Conf. Embedded Networked Sensor Systems,* Nov. 2003.

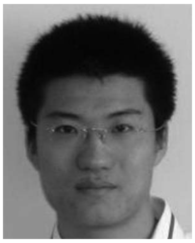
[29] G. Xing, C. Lu, R. Pless, and Q. Huang, "On Greedy Geographic Routing Algorithms in Sensing-Covered Networks," *Proc. Fifth ACM Symp. Mobile Ad Hoc Networking and Computing,* May 2004.

[30] Y. Xu, J. Heidemmann, and D. Estrin, "Geography-Informed Energy Conservation for Ad Hoc Routing," *Proc. Seventh Int'l Conf. Mobile Computing and Networking,* July 2001.

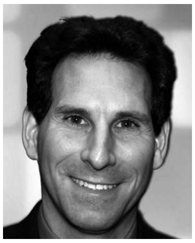
[31] Y. Zhang and Q. Huang, "A Learning-Based Adaptive Routing Tree for Wireless Sensor Networks," *J. Comm.,* vol. 1, no. 2, pp. 12-21, May 2006.

[32] Y. Zhang, J. Zhao, and G. Cao, "On Scheduling Vehicle-Roadside Data Access," *Proc. Fourth ACM Int'l Workshop Vehicular Ad Hoc Networks,* Sept. 2007.

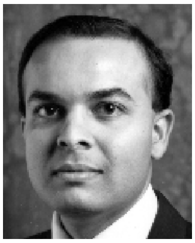
[33] Z. Zhang, A.D. Kshemkalyani, and S.M. Shatz, "Dynamic Multi-Root, Multi-Query Processing Based on Data Sharing in Sensor Networks," *ACM Trans. Sensor Networks,* vol. 6, no. 3, June 2010.



Juzheng Li received the bachelor's degree in software engineering from the Harbin Institute of Technology, China. He is currently working toward the PhD degree in the Department of Computer Science, College of Engineering, University of Illinois at Chicago (UIC). As of September 2010, he is a software development engineer at Yahoo! in Sunnyvale, California. His research interest is the design of ad hoc wireless networks, including sensor networks.



Sol M. Shatz received the BS degree in computer science from Washington University, St. Louis, Missouri, and the MS and PhD degrees in computer science from Northwestern University, Evanston, Illinois, in 1981 and 1983, respectively. He is currently a professor of computer science and an associate dean for Research and Graduate Studies in the College of Engineering at the University of Illinois at Chicago. He also serves as codirector of the Concurrent Software Systems Laboratory and pursues research in the areas of software engineering, sensor networks, and mobile computing. He has served on the program and organizing committees of many conferences, including program cochair and general chair for the International Conference on Distributed Computing Systems (ICDCS) in 2003 and 2007, respectively. He is a member of the editorial board for various technical journals, having served as a board member for the *IEEE Transactions on Software Engineering* from 2001 to 2005. He has received various teaching awards from the University of Illinois at Chicago, as well as being awarded a University Scholar Award from the University of Illinois in 2007. He is a distinguished scientist of the ACM and a senior member of the IEEE.



Ajay D. Kshemkalyani received the BTech degree in computer science and engineering from the Indian Institute of Technology, Bombay, in 1987, and the MS and PhD degrees in computer and information science from The Ohio State University in 1988 and 1991, respectively. He spent six years at IBM Research Triangle Park working on various aspects of computer networks, before joining academia. He is currently a professor in the Department of Computer Science at the University of Illinois at Chicago. His research interests are in distributed computing, distributed algorithms, computer networks, and concurrent systems. In 1999, he received the US National Science Foundation Career Award. He is currently on the editorial board of the Elsevier's *Computer Networks* and the *IEEE Transactions on Parallel and Distributed Systems*. He has coauthored a book entitled *Distributed Computing: Principles, Algorithms, and Systems* (Cambridge University Press, 2008). He is a distinguished scientist of the ACM and a senior member of the IEEE.