Geocast for Wireless Sensor Networks

***Abstract*—An important but relatively less studied class of network layer protocol for wireless sensor networks is geocast. It allows a sensor node to send messages to all nodes in a given geographical area without the sender node having any knowledge about which nodes are present in that area. Developing a robust geocast protocol for practical sensor networks poses several challenges. Geocast messages should be reliably delivered to the destination area in the presence of unreliable wireless links, a typical characteristic of practical sensor network deployments. The protocol should minimize the number of radio transmissions and avoid control traffic to save energy, which is a scarce resource in sensor networks. The protocol should be robust against a wide range of network densities. This paper presents the design, implementation, and evaluation of *SGcast* — a reliable, robust, and energy-efficient geocast protocol that achieves these goals. For a wide range of experiments conducted using networks of real sensor nodes and simulations, we show that compared to a recent geocast protocol, SGcast achieves up to 11.08x reduction in energy consumption and up to 2.17x improvement in successful delivery of geocast messages to the destination area, while being extremely robust against a wide variability in network densities.**

***Keywords*-Geocast, Sensor networks.**

I. Introduction

In Wireless Sensor Network (WSN) community, there has been extensive research interests in two categories of protocols at the network layer — *Collection* and *Dissemination*. Collection is a many-to-one routing protocol which allows sensor nodes to send their data (e.g. sensor readings) to one or a small number of collection points (say *base stations*) in the network. Dissemination is a one-to-many protocol that allows a base station to disseminate information (e.g. code updates, network commands, queries, etc.) to all or a subset of sensor nodes in the network. A different category of network layer protocol that has received relatively little attention is *geocast*. Geocast allows a node to send messages to all nodes within a given geographic area without the sender node having any knowledge about which nodes are present in that area. The destination nodes may be located outside the radio transmission range of the source node. Thus the geocast message may need to be forwarded by intermediate nodes to the destination area.

In many practical WSNs, the ability to assign location-specific tasks to nodes is important. For example, using geocast, a user can instruct all nodes in an area where fire is spreading to report temperature readings at a faster rate. In an earthquake situation, geocast can be used to communicate with sensors in a heavily damaged area of a building. In many practical cases, it may be difficult or even impossible to know which nodes are present in a given area at a given time because of ad-hoc nature of deployment (e.g. sensors dropped from an airplane), node mobility, incremental node deployment (addition of new sensors to the network to adapt to changing user and network requirements), large number of nodes, etc. As a result, unicast or multicast communications are not possible. Even if a user knows which nodes are present in a given area at a given time, sending separate unicast messages to each node in the area is very inefficient because of high bandwidth usage and energy consumption. Multicasting is very difficult because sensor nodes do not know beforehand, which multicast groups it should be a member of. Thus geocast is a useful communication primitive in many practical scenarios.

In this paper, we present the design, implementation, and evaluation of a geocast protocol for WSNs called *SGcast* that has four important goals: 1) *Reliability*: The protocol should achieve a high *hit%* — percentage of sensor nodes in the destination area that successfully receive the geocast packets. 2) *Robustness*: The protocol must be robust against a wide range of network conditions, mainly link disparities and node densities, which are typical characteristics of practical WSNs. 3) *Energy efficient*: The protocol should use a small number of radio transmissions to save energy — a scarce resource in WSNs. 4) *No control traffic*: The protocol should not use any control traffic (to learn about the current network conditions and adapt the protocol accordingly) since control packets incur additional energy and bandwidth overheads. Although several geocast protocols have been proposed for general purpose wireless mobile ad-hoc networks (MANETs) [1], [2], [3], [4], [5], [6], very little work has been done for WSNs. Existing works in WSNs are entirely simulation-based [6], do not achieve high hit% and energy-efficiency simultaneously, or use control traffic [7], [8]. To the best of our knowledge, SGcast is the first geocast protocol for WSNs that simultaneously achieves all four goals mentioned above and is extensively evaluated not only using simulations but also on testbeds of real sensor nodes.

Developing a robust and efficient geocast protocol for WSNs poses several challenges. First, many prior research works as well as experimental and practical WSN deployments have shown that wireless links in sensor networks are very complex, non-ideal, and unreliable [9], [10], [11], [12], [13], [14]. Temporal and spatial link dynamics, harsh and unpredictable nature of the environment in which the network may be deployed, low power radios, link asymmetry, external interference (e.g. IEEE 802.15.4 based radio transceivers used by many WSNs share the unlicensed 2.4 GHz ISM band with many wireless systems like WiFi, bluetooth, and even home appliances like microwaves), etc. are common in WSNs deployed in the *wild*. Thus reliably delivering geocast packets to a destination area through a series of intermediate nodes in the presence of unreliable radio connectivity is challenging.

The most important, and often the only, performance metric of a geocast protocol in general purpose MANETs is hit%. In WSNs, the geocast protocol should not only have a high hit% but also be energy efficient. Sensor nodes generally have limited energy supply (e.g. they are battery-powered) and may be deployed in places which are difficult to access physically. Thus, recharging sensors after deployment may not be feasible. Since radio transmissions are generally the most energy-expensive operations in WSNs [15], [16], number of radio transmissions should be reduced to increase network lifetime. Thus a geocast protocol for WSNs needs to fulfill two competing goals — allow sufficient nodes to forward the geocast packet in order to overcome the inherent unreliability of the wireless medium, and yet limit the number of redundant radio transmissions to save energy. Many existing geocast protocols exhibit a tradeoff between hit% and energy efficiency. They achieve good hit% at the cost of large number of (redundant) radio transmissions (or vice-versa). Sacrificing energy in favor of hit% may be acceptable in general wireless networks. For example, mobile phones and laptops can be recharged regularly, while it may not be possible in WSNs.

A geocast protocol for WSNs should be robust against different node densities. Depending on the nature of the application, WSNs can operate in a wide range of node densities. Even within a single network, node density can vary, both temporally and spatially. Random ad-hoc nature of node deployment, transient node and link failures, node mobility, incremental node deployment, etc. all contribute to variability in node density. In sparse networks, sufficient number of nodes should forward the geocast packets to increase the probability of the packet delivery to nodes in the destination area. On the other hand, in dense networks, the protocol should limit the number of nodes that forward the geocast packets, thereby decreasing network congestion and increasing delivery probability.

Another constraint in WSNs is that these challenging goals need to be achieved using limited resources. Compared to general wireless devices, sensor nodes are extremely resource constrained. For example, TMote, a popular sensor node, has 10KB RAM, 48KB program memory, 8MHz processor, and 250 Kbps radio. Thus the geocast protocol should be simple enough so that it can fit within memory, processing, and bandwidth constraints of practical WSNs.

In order to improve the protocol performance, some geocast protocols [2], [3], [4], [7] use control traffic to learn about current network conditions and adjust the protocol accordingly. In such schemes, nodes need to periodically broadcast *advertisement* messages. Upon receiving such messages, each node infers important information about its neighbors, e.g. neighbor locations, local node density, qualities of its links to each neighbor (by using, say, RSSI values). When an intermediate node receives a geocast packet, it uses the neighborhood information and some heuristics to forward the packet to the *best* neighbor — a neighbor that it finds most likely to forward the packet to the destination area with the least cost (different schemes can define costs differently). The problem with such schemes is that the frequency at which the nodes need to exchange the advertisement messages is proportional to the rate at which network conditions change. It is generally not possible for a node to learn about the qualities of its links to the neighbors on demand — when it has a geocast packet to forward. Such information needs to be learned over time [17], [18]. In practical WSNs, network conditions can be very dynamic because of a variety of reasons mentioned above. Geocast protocols may utilize schemes like Trickle [19] which exchange control messages less frequently during steady-state (when network conditions are stable) and more frequently otherwise. However, since the control messages need to be exchanged *periodically* (even when no geocast communication is taking place), the monotonically increasing energy cost of broadcasting them can be very significant over time [20]. Furthermore, the control traffic also grows proportionally with node density. In contrast, SGcast is a very light-weight protocol that does not maintain neighbor table or broadcast any control packets. Sensor nodes use a set of local rules to make packet forwarding decisions, without requiring any local or global knowledge about the network.

SGcast uses various novel techniques to efficiently deliver geocast packet to nodes in the destination area. It uses a novel concept of *negative hops* (a measure of divergence of geocast propagation from the destination area) to choose potential forwarding nodes such that redundant transmissions are avoided. It also uses a new idea that different nodes should use different rules to make packet forwarding decisions based on the progress the packet has made towards the destination area. It applies an intelligent backoff scheme so that proper nodes are chosen to forward the geocast packet, and transmissions from undesired nodes are suppressed.

Our contributions in this paper are as follows:

1) Using novel ideas like negative hops, different forwarding rules based on propagation characteristic of the geocast packets, intelligent backoff schemes, etc., we achieve four goals simultaneously — reliability, low energy consumption, robustness to network conditions, and no use of control traffic.

2) Most existing geocast protocols are developed for general purpose MANETs. Few existing protocols for WSNs are evaluated using mathematical analysis or simulation only. We present a complete design and implementation of the protocol in real sensor nodes.

3) We perform extensive evaluation of SGcast using networks of real sensor nodes as well as simulations. Unlike existing protocols that show a tradeoff between energy consumption and hit%, we show that SGcast acheives a significant reduction in energy consumption (upto 11.08 times less than a recent protocol [1]), while achieving very high hit% (upto 2.17 times more than [1]).

4) SGcast is a completely distributed protocol requiring no global knowledge of current network conditions.

II. Related Work

Geocast protocols for MANETs can be broadly classified in two categories: those which require nodes to broadcast control packets periodically [2], [3], [4], [7] and those which do not [1], [21] . The focus of this paper is the latter category of protocols since they do not incur extra bandwidth and energy overheads of control traffic. A simple approach in this category is to use flooding, where each node broadcasts the geocast packet that it receives until the packet reaches the destination area. Flooding, however, is not scalable [22]. Practical WSNs can have hundreds of sensor nodes and flooding can cause the network to be congested, leading to a poor hit% and high energy consumption. Ko and Vaidya [21] proposed various controlled flooding schemes by introducing the concept of forwarding zones. Only nodes in the forwarding zones broadcast the packet. Forwarding zones can be defined in different ways, e.g. smallest rectangle enclosing the originator and the destination area, smallest rectangle enclosing the forwarding node and the destination area, etc [21].

In this paper, we compare SGcast with a recent work [1] (referred to as *HallGeocast* hereafter) that improves Ko and Vaidya's scheme by defining a set of local heuristics that must be satisfied by a node to forward the packet. HallGeocast suppresses redundant transmissions in the forwarding zone, and yet allows sufficient number of nodes to forward the geocast packet so that hit% is not negatively affected. [1] provides an excellent survey of existing geocast protocols for MANETs and a qualitative discussion of how HallGeocast outperforms many of the existing geocast protocols. In the next section, we present a brief description of HallGeocast. Since HallGeocast was developed for MANETs, it is not designed to tackle many of the problems (mentioned in the previous section) that are specific to WSNs. We implement both HallGeocast and SGcast for WSN and show that SGcast significantly outperforms HallGeocast under a wide range of network conditions.

Note that the evaluation in [1] was done for a class of scenarios representative of general MANET usage, and SGcast has been evaluated for a class of scenarios more typical of WSN usage. It is an open question, and subject for future work, how well SGcast performs in general MANET usage scenarios.

III. Brief overview of HallGeocast

The closest existing geocast protocol to SGcast is HallGeocast, which was developed for general purpose MANETs. For completeness of the discussion, we provide a brief description of HallGeocast in this section. In HallGeocast, when a node generates a gecoast packet, it broadcasts the packet, with the packet header including originator location, a globally unique identifier[1], and a definition of the destination area (e.g. center and radius for a circular destination area). When a node n_1 hears the broadcast geocast packet with globally unique identifier *gid*, it enqueues the packet in a queue for possible transmission. When the packet reaches the head of the queue, n_1 broadcasts the packet if the following heuristic is satisfied:

$$Heuristic \equiv FZ \wedge (M \vee T \vee CD) \tag{1}$$

The heuristic consists of four sub-heuristics. *FZ* heuristic is true if n_1 lies in the forwarding zone. Different schemes can define forwarding zones differently. A basic requirement for defining forwarding zones is that the nodes present in such zones should provide network connectivity between originator of the geocast packet and nodes in the destination area. *M* heuristic is true if the node has heard less than m number of geocast packets with the same *gid*. The rationale behind *M* heuristic is to increase redundancy to at least a specified minimum level for reliability. *T* heuristic is true if all geocast packets previously heard by n_1 for this *gid* are from nodes which are at least t distance away from n_1. The main idea behind *T* heuristic is to spread the geocast to a wide area so that the probability of the geocast packet reaching the destination area is high, even if some packet broadcasts fail to reach their neighbors. *CD* heuristic is true if n_1 is closer to the destination area than all other nodes from which n_1 has heard geocast packets for this *gid*. *CD* heuristic enhances the progress of the geocast packet to the destination area by making sure that the node closest to the destination area always forwards the packet.

IV. SGcast design

Like previous works, we assume that all nodes know their physical locations. Although GPS can be very energy-expensive for sensor nodes, there are many localization schemes proposed in the literature that do not require GPS [23], [24], [25]. In this section, we first present the key insights used by SGcast to achieve four goals mentioned above, followed by the detailed description of the protocol.

A. Key insights

Avoid explicit forwarding zones: Using small number of radio transmissions and achieving a high hit% are often conflicting goals because of the *forwarding node selection* (FNS) problem. A geocast protocol should increase the number of forwarding nodes to combat inherent unreliability of wireless links in sensor networks to achieve a high hit%. However, this also increases the total number of radio transmissions, and thus the energy consumption, in the network. On the other hand, decreasing the number of forwarding nodes reduces energy consumption, but also degrades hit%. Existing protocols attempt to tackle this tradeoff by defining forwarding zones. In [21], only nodes in forwarding zone broadcast the geocast packets. In HallGeocast [1], in addition to being in the forwarding zone, a node has to satisfy some heuristics in order to forward the packet.

Although the use of forwarding zone can limit the number of redundant broadcasts, it just transforms the FNS problem to a problem of defining the forwarding zone — the size of the zone should be large enough to allow sufficient nodes to forward the packet to achieve a high hit%, and small enough to reduce the number of radio transmissions. Furthermore, the

[1] A tuple consisting of originator node id and a monotonically increasing sequence number can serve as a globally unique identifier.

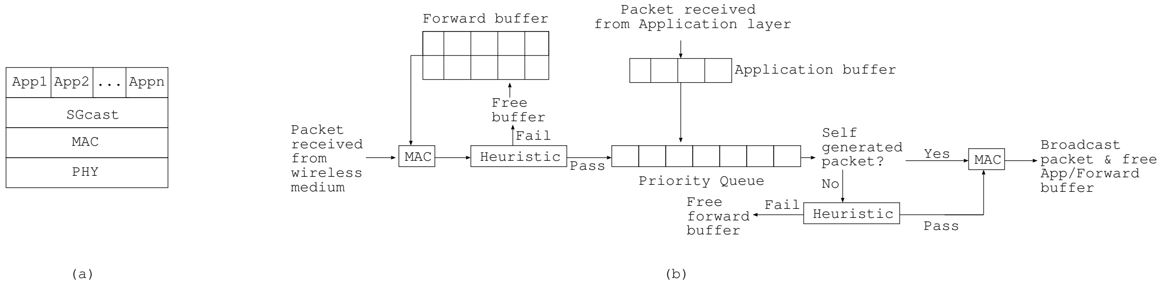


Fig. 1. (a) SGcast sits between application and MAC layers, (b) SGcast packet flow framework.

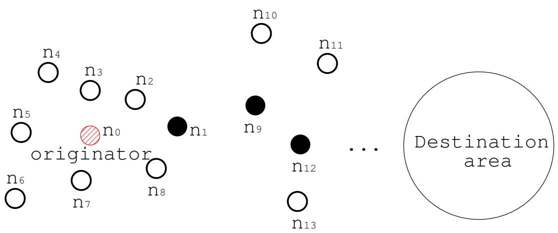


Fig. 2. Node n_0 is the originator of the geocast packet and the circular area on the right is the destination area. In SGcast, dark nodes forward the packet, while in HallGcast, all nodes shown in the figure can forward the packet in the worst case.

definition of forwarding zone should be a function of current network conditions, mainly node density and link qualities. In networks which are sparse or have bad links, forwarding zone should be large, while in dense networks or networks with good links, it should be small. Since node density and link qualities can vary significantly in WSNs (both temporally and spatially), forwarding zones should be defined dynamically. Existing schemes define forwarding zones statically[2] and thus are not scalable with respect to node density, link qualities, etc. Defining forwarding zones dynamically is difficult because sensor nodes need to learn about the current network conditions which requires periodic control traffic. Moreover, the knowledge about single-hop neighborhood is not sufficient. Current network conditions in an entire area between the originator node and the destination area should be known to define a good size and shape of the forwarding zone. It is difficult for a node to obtain such semi-global information in WSNs with frequently changing network conditions.

Instead of defining forwarding zone explicitly, SGcast introduces a novel concept of *negative hop* — a communication hop that causes the geocast packet to diverge away from the destination area. More precisely, when a node n_2 receives a geocast packet from a node n_1, the $n_1 - n_2$ hop is called a negative hop if n_1 is closer to the destination area than n_2. In SGcast, the geocast packet header includes a *NumNegHops* field, which represents the number of negative hops the packet has traversed. The originator of the geocast packet initializes this field to 0 and each intermediate node increments this field if the neighbor from which it receives the packet is closer to the destination area than itself (location of the transmitter is also included in the geocast packet header). SGcast limits the propagation of geocast beyond a threshold value n of *NumNegHops*. Thus by using *NumNegHops*, SGcast implicitly learns about the current network conditions by using the propagation path already traversed by the geocast packet. The *implicit* forwarding zone defined by using *NumNegHops* is dynamic because the value of *NumNegHops* is a function of current network density and link qualities.

[2]Note that adaptive forwarding zone defined in [21] is not *dynamic* because the number of nodes that lie in the forwarding zone can vary significantly based on link qualities and node density.

Dynamic heuristics: In existing protocols, a node uses static heuristics to decide whether it should forward the geocast packet. In [21], a node forwards the packet if it is in the forwarding zone. In [1], the heuristic mentioned in equation(1) is used. A key observation that we use in SGcast is that different nodes should use different heuristics depending on the *amount* of progress the packet has made towards the destination area. *NumNegHops* is a very good and natural metric that quantifies this. In SGcast, the *severity* of heuristics used by a node to make a packet forwarding decision is proportional to *NumNegHops*. For example, if a node receives a geocast packet with *NumNegHops*=0, it knows that the packet is progressing well towards the destination and thus uses a mild heuristic that can be easily satisfied. On the other hand, if *NumNegHops*=5, the node knows that 5 hops followed by the packet have deviated from the destination area, so it applies a strict rule to inhibit making a forwarding decision.

Distance based backoff: As shown in figure 1-a, SGcast is a network layer protocol that sits between application and MAC layers. Unlike many existing geocast protocols which rely on the backoff mechanism of MAC layer [1], [21], SGcast introduces its own backoff scheme. Existing MAC layers for sensor networks (e.g. widely used IEEE 802.15.4) are not designed with geocast in mind. They are meant to provide fair medium access to all nodes in a local neighborhood. In SGcast, nodes which are closer to the destination area backoff for a shorter period than those which are further away.

Consider an example in figure 2. Here node n_0 wants to send a geocast packet to all nodes (not shown in the figure) in the destination area represented by a large circle in the figure. Among nodes $n_1, n_2, ..., n_8$, that hear the geocast packet broadcast by node n_0, the backoff period generated by SGcast layer at node n_1 is smallest because it is closer to the destination area than $n_2, n_3, ..., n_8$. As a result, node n_1 broadcasts the packet first. Various factors make it very difficult for nodes $n_2, n_3, ..., n_8$, to make a forwarding decision. First, *NumNegHops* is now 1, so stricter heuristic is applied. Second, *M* heuristic (described in the next section) will have less chance of being true. (*M* heuristic is true if a node has heard less than m number of geocast packets for a given *gid*.) Third, when node n_9 hears node n_1's transmission, it will transmit early since it is closer to the destination area, thereby making it even more difficult for nodes to the left of it to make a forwarding decision. In contrast, in protocols that rely on MAC layer's backoff mechanism, nodes $n_1, n_2, ..., n_8$ all have equal chance of transmitting the geocast packet first. Thus "non optimal" nodes (like $n_1, n_2, ..., n_8$ which do not enhance the progress of the packet towards the destination area) may transmit the packet before more optimal nodes like n_1, n_2, and n_3, thereby increasing the number of radio transmissions and possibly causing congestion if the network is dense. The dark nodes in figure 2 forward the packet in SGcast, while in schemes like HallGeocast that rely on MAC layer's backoff mechanism, depending on the order in which nodes transmit the packet, all nodes shown in figure 2 may forward the packet in the worst case.

One potential issue with protocols that greedily choose nodes closer to the destination area to forward the geoacast packet is the *local maximum* problem — geocast packet reaches a node that does not have any neighbor closer to the destination area than itself. This can prevent the packet from reaching the destination area in schemes that selects *only one* neighbor to forward the packet (e.g. protocols that use unicast communication to deliver the packet from the originator node to the destination area). However, in SGcast the severity of this problem is significantly reduced because of various inherent redundancies present in the protocol. First, generally there are multiple paths with *NumNegHops=0*. So, a given geocast can reach the destination area via multiple paths having *NumNegHops = 0*. Second, even if *NumNegHops* is greater than 0 (but less than the threshold value n), it does not necessarily mean that the packet will be dropped. With higher *NumNegHops*, progressively stricter heuristics are applied. Those heuristics can still be satisfied, especially for small values of *NumNegHops*.

B. Formal protocol description

When an application generates a geocast packet, it includes the destination area in the geocast packet header. In our current implementation, the destination area is a circle and is specified by center and radius of the area. Destination area can be specified differently and SGcast is not tied to a particular way of defining the destination area. Then the application layer gives the geocast packet to SGcast at the network layer. SGcast adds following fields to the geocast packet header — globally unique geocast id (*gid*), *NumNegHops*, and its own location. In our current implementation, *gid* is a tuple consisting of originator node id and a monotonically increasing sequence number. *NumNegHops* is the number of negative hops the packet has traversed. It is initialized to 0 by the originator. Then the originator broadcasts the geocast packet.

When a node receives the broadcast geocast packet, it uses the locations of the transmitter, destination area (both are present in the geocast packet header) and its own location to compare its distance to the destination area with that of the transmitter. In our current implementation, distances to the center of the destination area are calculated. If the node is further away from the destination area than the transmitter, it increments *NumNegHops*. If *NumNegHops* is greater than n, it drops the packet. The rationale is that the packet has traversed n hops which have deviated from the destination area and has incurred many (possibly) redundant transmissions. Then it calculates the backoff period proportional to its distance to the destination area. When the backoff period expires, the node checks a set of heuristics depending on the value of *NumNegHops*. The idea is to increase the severity of the heuristic as *NumNegHops* increases. In our current implementation, we use the following two levels of heuristics:

$$\text{If NumNegHops is 0}, Heuristic \equiv M \vee CD \quad (2)$$

$$\text{If NumNegHops is 1}, Heuristic \equiv M \wedge CD \quad (3)$$

Here each heuristic has 2 sub-heuristics. *M* heuristic is true if the node has heard less than m number of geocast packets with the given *gid*. *CD* heuristic is true if the node is closer to the destination area than any other nodes from which it has heard geocast packets for this *gid* [3]. Note that SGcast provides a flexible framework so that different levels of severity can be used for different values of *NumNegHops*. We leave exploring different combinations of heuristics for different values of *NumNegHops* as future work.

C. Geocast packet flow framework

Figure 1-b shows SGcast packet flow framework. SGcast maintains two packet buffers — forward buffer for geocast packets that are received from other nodes and application buffer for packets that are received from applications running on the same node. The use of separate buffers for these two types of geocast packets helps to reduce the memory usage. Generally, the number of packets that a node needs to forward is more than that it generates itself. Consider a single geocast. It is generated by a single node, but forwarded by many. By using smaller application buffer and larger forward buffer instead of a single buffer, the amount of unused buffer space can be minimized.

Forward packet flow: When SGcast receives a geocast packet to forward from the wireless medium via the underlying MAC/PHY layer, instead of copying the contents of the packet to the forward buffer, it returns a new buffer (pointer) from the forward buffer to the MAC/PHY layer and keeps the received buffer. Quickly returning the buffer (instead of performing the time-consuming memcpy( ) operation to copy the packet

[3]Note that when a node n_2 receives a packet from a node n_1 and if this is a negative hop, *CD* heuristic is always false. So, if the current hop is the negative hop and *NumNegHops* $< n$, only *M* heuristic is checked. For further hops, *M* $\wedge$ *CD* is used.

contents to the forward buffer) to the MAC/PHY layer prevents the content of the received packet from being corrupted due to the node receiving multiple packets in a quick succession. Note that SGcast is only a small component of the entire software stack running on the sensor node. Many other applications and protocols use the same MAC/PHY layer. If the buffer is not quickly returned to the MAC/PHY layer, a subsequently received packet (geocast packet or a packet from other any other protocol) may corrupt the earlier packet. If the forward buffer is full and SGcast receives a request to forward a packet from the MAC/PHY layer, it drops the packet.

After the geocast packet is buffered in the forward buffer, a heuristic check is performed based on the value of *NumNegHops* as described in the previous section. If the heuristic check fails, the packet is dropped. Otherwise, a backoff interval (proportional to the node's distance to the destination area) is computed and the packet is enqueued in a priority queue. The priority queue uses the backoff interval as the priority. A packet with smallest backoff interval is placed at the head of the queue, the next spot is occupied by the packet with second smallest backoff period, and so on. The size of the priority queue is the sum of the sizes of forward and application buffers.

When the packet reaches the head of the queue and backoff period expires, the packet is dequeued and heuristic check is performed again. If the check is passed, the packet is given to the MAC/PHY layer for broadcasting. Otherwise, the packet is dropped. In either case, the forwarding buffer used by the packet is freed. Note that SGcast performs the heuristic check twice. The only purpose of the first check (performed immediately after receiving the packet from MAC/PHY and before enqueuing in the priority queue) is to save memory. A packet that fails the heuristic check before being enqueued in the priority queue fails the latter check anyway (performed after the packet is enqueued and reaches the head of the queue). By dropping the packet early, memory usage can be reduced. The second heuristic is performed *late* — just before transmitting the packet, thereby allowing sufficient time for the node to hear other packets transmitted by its neighbors that might cause the heuristic check to fail. The late heuristic check is very important to reduce the number of redundant radio transmissions [1].

Self-generated packet flow: When SGcast receives a request from the application layer to send a geocast packet and if the application buffer is full, it informs the application layer that the request is denied. Application layer can implement its own retransmission policy. If the application buffer is available, the packet is enqueued in the priority queue by selecting a backoff interval such that the packet is placed at the head of the queue[4]. Note that the goal of the backoff mechanism is to make sure that among the nodes in a local neighborhood that receive a geocast packet with a *given gid*, the node with the least distance to the destination area transmits the packet first. Since none of the nodes except the originating node has the geocast packet with this *gid*, the goal of the backoff mechanism is not compromised by giving a high priority to its own packet. Also note that the self generated geocast packet does not undergo any heuristic check because it is the first node that will transmit the geocast packet for this *gid* and thus the heuristic is always true.

[4]In fact in our current implementation, the packet is placed at the head of the queue if the queue is empty. Otherwise, it is placed just behind the head to let the packet already at the head to be transmitted first. This is to avoid packet corruption in the middle of a transmission.

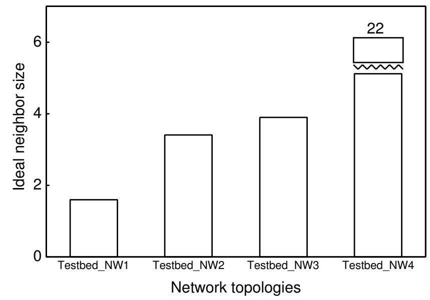


Fig. 3. Network density for testbed networks.

V. EVALUATIONS

We implement HallGeocast and SGcast in Tinyos [26] and compare them extensively using different networks of various node densities and link qualities. Testbed experiments are performed using TMote sensor nodes. For large scale evaluations, we use TOSSIM [27] simulations.

A. Testbed experiments

We use 4 networks of TMote sensor nodes of varying node densities and link qualities for testbed experiments.

1) *Testbed_NW1* is a small scale network of 10 sensor nodes deployed in two floors of a home building.
2) *Testbed_NW2* is a small scale network of 10 sensor nodes deployed in a single floor of a home building.
3) *Testbed_NW3* is a moderate scale network of 23 sensor nodes deployed in a single floor of a large office building.
4) *Testbed_NW4* is a moderate scale network of 23 sensor nodes deployed in a single room of an office building. In this network, all sensor nodes are within the transmission range of each other. We use this network to compare the robustness of SGcast and HallGeocast in a highly dense network.

In all experiments, the output transmission power of each node is set to the minimum possible value to increase the size of the network in terms of number of hops. In order to quantify densities of these networks, we performed the following experiment. In each network, each node broadcasts 100 packets. Each node counts the number of packets that it receives. Note that these are simple single hop broadcasts, not geocast communications. The time interval between any two successive packet broadcasts is sufficiently large so that no two broadcasts interfere with each other. The average number of nodes that receive a broadcast transmission (i.e. neighborhood size) is a measure of *ideal* node density (in the absence of interfering transmissions) of these networks. The average number of neighbors for each network is shown in figure 3.

TABLE I
PARAMETERS FOR THE EXPERIMENT.

Parameter	Value	Protocol
m	2	SGcast, HallGeocast
t	dynamically chosen	HallGeocast
R_1, R_2	5 * Transmission Range	HallGeocast
n	2	SGcast
Forward buffer	24	SGcast, HallGeocast
App buffer	5	SGcast, HallGeocast

Testbed_NW1 is the most sparse network while Testbed_NW4 is the most dense. Also, link qualities in home networks are different from those in office networks. Compared to the home environment, office environment has a large number of WiFi access points and terminals. Thus, Testbed_NW3 and Testbed_NW4 are subjected to more external interference than Testbed_NW1 and Testbed_NW2.

It is very difficult to accurately quantify network diameter in terms of number of hops because of variability of wireless link characteristics. In our experiments, we find that Testbed_NW1 is about 4 to 7 hops wide, Testbed_NW2 is about 2 to 3 hops wide, Testbed_NW3 is about 4 to 7 hops wide, and Testbed_NW4 is 1 hop wide.

Table I shows various parameter values used in the experiments. m is the parameter of *M* heuristic. n is the threshold value of *NumNegHops*. R_1 and R_2 are used to define forwarding zones in HallGeocast. All nodes within radius R_1 of the geocast originator node and within radius R_2 of the center of the destination area are in the forwarding zone. R_1 and R_2 are chosen as 5 times the transmission range of the nodes [1]. In WSNs, it is not possible to find an accurate value of the transmission range. Fixed disc model — where nodes within a certain distance of the transmitter receive the packet and those outside do not — does not hold true in practical WSNs. It has been shown [7] that in WSNs, packet reception rate (PRR) is very high within a certain distance, say d_1, from the transmitter. But in a *large* region between d_1 and d_2 ($d_2 > d_1$), PRR shows a wide variation. Beyond d_2, PRR is very low. In our experiments, we derive the approximate average transmission range by using the information about average number of neighbors (described above) and known distances between the nodes. Parameter t used in *T* heuristic by HallGeocast is also difficult to set. We find that the performance of HallGeocast, especially number of radio transmissions, depends strongly on the value of t. In each experiment, we choose t such that the number of transmissions is minimized. This also highlights the fact that it is very difficult to choose good parameter values in HallGeocast. Furthermore, they need to be adjusted when network conditions change, which can happen frequently in practical WSNs.

For each network, a node at one corner of the network generates geocast packets and transmits them to a circular destination area situated at another corner of the network. Each experiment is run for 15 minutes during which the originator transmits a geocast packet every 10 seconds. Each experiment is repeated 5 times. The results presented here are average values.

Figure 4-a compares hit% of HallGeocast and SGcast. We find that both protocols deliver geocast packets reliably to nearly 100% of nodes in the destination area for these small and moderate sized networks. However, as shown in Figure 4-b, the number of radio transmissions per geocast is significantly smaller in SGcast compared to HallGeocast. Note that the number of radio transmissions is a measure of energy consumption of the protocol. For these networks, we find that HallGeocast transmits 2.2 to 3.5 times more packets than SGcast. Figure 4-b also shows that the number of radio transmissions is strongly dependent on network conditions (density, size, link qualities, etc.) for HallGeocast, while SGcast is very robust to such conditions. Various design features of SGcast contribute in reducing the number of radio transmissions, in achieving high hit%, and in making the protocol robust to dynamic network conditions. Using *NumNegHops* to define implicit forwarding zones dynamically instead of static forwarding zones limits the spread of geocast propagation away from the destination area and makes the protocol robust against node densities. Using different heuristics for geocast paths with different values of *NumNegHops* and distance-based backoff mechanism ensure that those nodes are selected for forwarding which cause the geocast packet to progress towards the destination area. These techniques effectively suppress transmissions from undesired nodes, and yet provide sufficient redundancy so that even if some transmissions fail, the goecast packet still has a high probability of reaching the destination area.

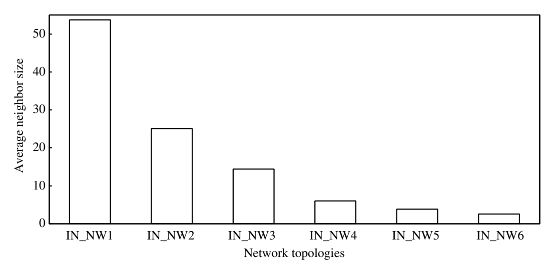


Fig. 5. Network density for indoor networks.

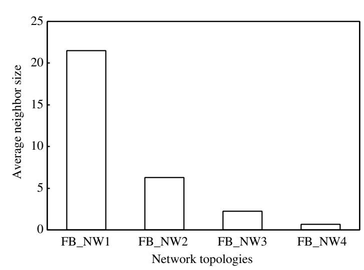


Fig. 6. Network density for outdoor networks.

Figure 4-c shows average number of hops required by HallGeocast and SGcast to deliver geocast packets to the destination area. In Testbed_NW2 and Testbed_NW4, the networks are spread in a small area and thus the number of hops in SGcast is only slightly smaller than that in HallGeo-

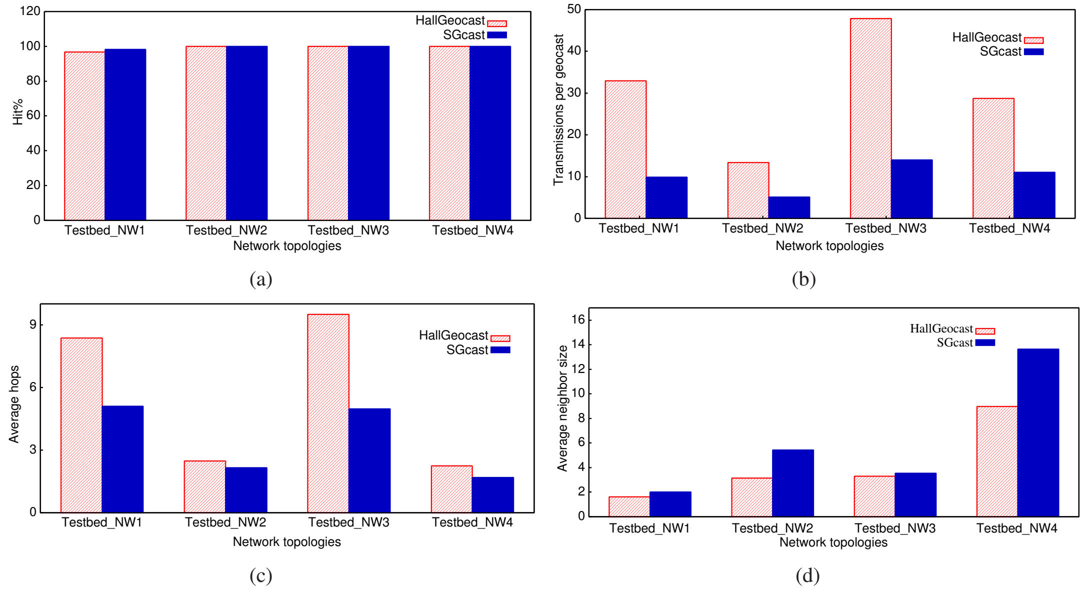


Fig. 4. Testbed experiments: Comparison between HallGeocast and SGcast in terms of (a) hit%, (b) number of transmissions per geocast, (c) average hops required to deliver the geocast packet to the nodes in the destination area, and (d) average neighborhood size.

cast. But in Testbed_NW1 and Testbed_NW3 where networks are spread across relatively large areas, SGcast significantly reduces the number of hops. In HallGeocast, due to static nature of the forwarding zone and reliance on MAC for the backoff mechanism, a large number of nodes participate in forwarding the geocast packets. Many of the hops are in negative directions — deviating away from the destination area. As a result, HallGeocast requires large number of hops and radio transmissions.

Figure 4-d also explains why SGcast performs better than HallGeocast. It compares these protocols in terms of average neighborhood sizes — the number of nodes that receive a geocast packet transmission. In SGcast, more nodes are able to successfully receive a geocast packet transmission than in HallGeocast. Since HallGeocast causes many nodes to forward the geocast packets, problems due to packet collisions, hidden terminal effects, etc. are more likely to occur in HallGeocast than in SGcast. As a result, only small number of nodes can receive a geocast packet successfully in HallGeocast compared to SGcast. We also find that average neighborhood size for SGcast is close to the ideal one (figure 3).

In summary, from our testbed experiments we find that SGcast significantly reduces energy consumption compared to HallGeocast, while delivering geocast packets to nearly 100% of nodes in the destination area.

B. Simulation experiments

It is essential to evaluate the performance of geocast protocols in large scale networks. In small networks, even simple flooding schemes may perform satisfactorily. We use TOSSIM simulations [27] to compare SGcast and HallGeocast in large networks. We consider two deployment scenarios — WSNs deployed in indoor office building and outdoor football field settings. Various parameters (like path loss exponent, white Gaussian noise parameters, shadowing components, etc.) that are used to characterize the radio environment are based on a theoretical model [28].

We consider 6 networks for indoor environment (*IN_NW1, ..., IN_NW6*) and 4 networks for outdoor football field environment (*FB_NW1, ..., FB_NW4*). Each network consists of 100 nodes arranged in a 10x10 grid fashion. The difference between different networks is the distance between two successive nodes in the grid. Like in testbed networks, we use the following experiment to quantify the ideal node densities of these networks. Each node in the network broadcasts 100 packets. Each node counts the number of packets that it receives. Figure 5 and figure 6 show the average number of neighbors in each of these networks. These figures show that the networks used for our experiments exhibit a wide variation in node densities.

For geocast experiments, each node selects a time instant randomly from a [5sec, 15sec] interval and sends a geocast packet to a randomly selected circular destination area in the network. This process is repeated for the experiment duration of one hour. Each experiment is repeated 3 times. The results presented here are average values. Simulation parameters are same as those mentioned in Table I.

Figure 7 compares HallGeocast and SGcast in terms of hit%, number of transmissions per geocast, average number of hops taken by geocast packets to reach the nodes in the destination area, and average neighborhood size. Like in testbed experiments, we find that SGcast is significantly better than HallGeocast. Some of the simulation results that are different from testbed results are as follows. First, although both HallGeocast and SGcast were able to achieve good hit% in small/moderate sized testbed networks, SGcast achieves significantly higher hit% for larger networks used in simulation

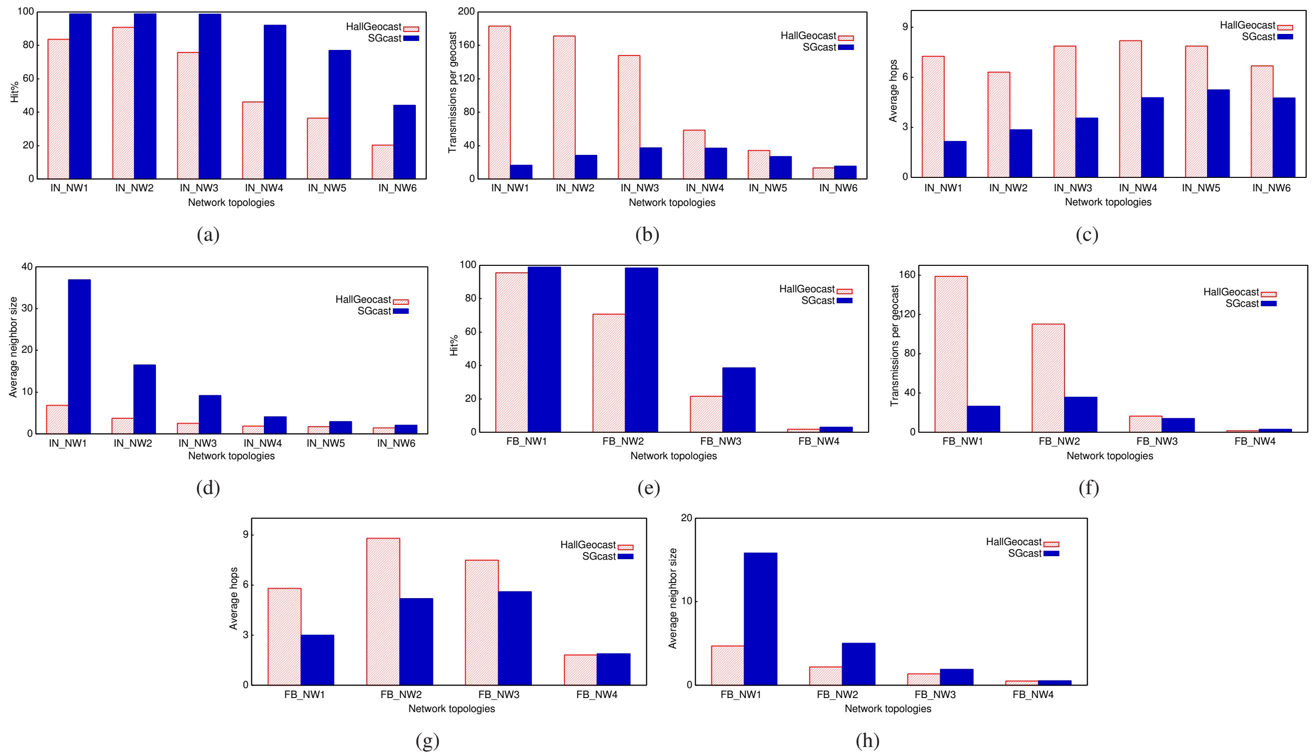


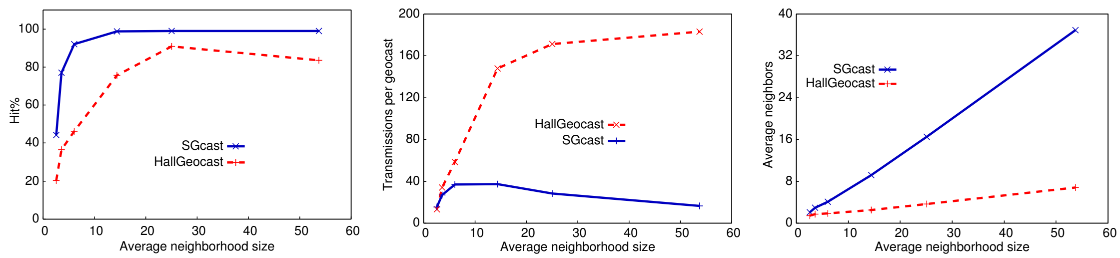
Fig. 7. Simulation experiments: Comparison between HallGeocast and SGcast in terms of (a) hit%, (b) number of transmissions per geocast, (c) average hops required to deliver the geocast packet to the nodes in the destination area, and (d) average neighborhood size for indoor networks. (e), (f), (g). and (h) are for outdoor networks.

than HallGeocast. In HallGeocast, many nodes are involved in forwarding the geocast packet. In large networks, this can lead to packet collisions, hidden terminal issues, and ultimately to network congestion. As a result, hit% degrades. Although SGcast's hit% is also decreased in simulation experiments compared to testbed experiments, they are still significantly higher than HallGeocast (by a factor of upto 2.17). In very sparse networks like IN_NW6, IN_FB3, IN_FB4, SGcast's hit% is quite low (still significantly better than HallGeocast), because, as shown in figures 5 and 6, the average node density is very low in these networks. We examined the neighborhood size distribution and found that the number of neighbors for many nodes at the edge of the network is significantly lower (close to 0) than the average value plotted in figures 5 and 6. As a result, many geocasts originated from edge nodes fail to proceed even a single hop beyond the originator node. Thus hit% is reduced.

In larger networks, the reduction in number of radio transmissions by SGcast (figure 7-b and 7-f) is even more (up to 11.08 times less than HallGeocast). Furthermore, these figures show that the number transmissions has a strong dependance on network density in HallGeocast, while SGcast shows less variation with respect to node density. This is an important property of SGcast since in practical deployments, WSNs can operate in a wide range of node densities.

Next we compare SGcast and HallGeocast with respect to node density (figure 8) for indoor networks. Note x-axes of these plots are *ideal* average node densities, i.e. average number of nodes that can hear a broadcast packet in the absence of interferring transmissions by nearby nodes (as shown in figure 5). (We omit the results for outdoor networks as they are similar to the indoor results). Fig 8-a shows that as node density increases, hit% increases in both SGcast and HallGeocast. But when the network is very dense, HallGeocast's hit% decreases slowly while that of SGcast remains steady at close to 100%. In very dense networks, large number of nodes lie in the statically defined forwarding zone, causing large number of nodes to forward the message. On the other hand, because of the use of *NumNegHops* to limit the divergence of geocast propagation, SGcast is very robust against node densities and hit% remains unaffected even in very dense networks. Furthermore, we notice that except for very sparse networks with neighborhood size less than 8, SGcast achieves hit% close to 100%. On the other hand, in HallGeocast, hit% is below 80% for most node densities.

Figure 8-b shows transmissions per geocast as a function of node density. With the increase in node density, many nodes are involved in forwarding geocast packet in HallGeocast. As a result, energy consumption increases. In contrast, number of radio transmissions is extremely robust to node density in SGcast. The reason for better performance of SGcast is also explained in figure 8-c. Here x-axis is the ideal node density while y-axis is the node density (average neighborhood size) observed by the two protocols. As node density increases, the number of nodes that can hear a geocast packet transmission increases for both SGcast and HallGeocast, but the increase



(a) (b) (c)

Fig. 8. Simulation results: Comparison of (a) hit%, (b) transmissions per geocast, and (c) average neighborhood size between HallGeocast and SGcast with respect to node density.

is more steep in SGcast. This shows that SGcast experiences very little network congestion as node density increases while the congestion is significant in HallGeocast.

VI. Conclusion

In this paper we presented the design, implementation, and an extensive evaluation of SGcast — a geocast protocol for WSNs. SGcast introduces many new concepts like negative hops, dynamic forwarding rules based on the path taken by geocast packets, intelligent backoff mechanism, etc. to achieve high energy efficiency without requiring local or global knowledge about the network conditions, and still maintains excellent hit%. SGcast is robust against a wide range of network conditions, mainly node densities and link disparities. As future work, we plan to integrate SGcast with a localization algorithm and study the effect of localization error on the performance of the geocast protocol. Also we intend to evaluate SGcast in general MANET usage scenarios.

References

[1] R. Hall, "An Improved Geocast for Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, pp. 254–266.

[2] B. Karp and H. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 243–254.

[3] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: Of theory and practice," *Proceedings of the twenty-second annual symposium on Principles of distributed computing*, pp. 63–72.

[4] W. Lou and J. Wu, "On reducing broadcast redundancy in ad hoc wireless networks," *IEEE Transactions on Mobile Computing*, pp. 111–123.

[5] M. Heissenbuttel, T. Braun, T. Bernoulli, and M. Walchli, "BLR: beacon-less routing algorithm for mobile ad hoc networks," *Computer communications*, vol. 27, no. 11, pp. 1076–1086.

[6] W. Liao, Y. Tseng, K. Lo, and J. Sheu, "Geogrid: A geocasting protocol for mobile ad hoc networks based on grid," *Journal of Internet Technology*, vol. 1, no. 2, pp. 23–32, 2000.

[7] M. Zamalloa, K. Seada, B. Krishnamachari, and A. Helmy, "Efficient geographic routing over lossy links in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 3, pp. 1–33, 2008.

[8] K. Seada and A. Helmy, "Efficient and robust geocasting protocols for sensor networks," *Computer communications*, vol. 29, no. 2, pp. 151–161, 2006.

[9] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex behavior at scale: An experimental study of low-power wireless sensor networks," *Tech report UCLA/CSD-TR,02-0013, Computer Science Department UCLA*.

[10] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 214–226.

[11] L. Sang, A. Arora, and H. Zhang, "On exploiting asymmetric wireless links via one-way estimation," *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pp. 11–21, 2007.

[12] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 1–13.

[13] A. Cerpa, J. Wong, M. Potkonjak, and D. Estrin, "Temporal properties of low power wireless links: modeling and implications on multi-hop routing," *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pp. 414–425.

[14] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and yield in a volcano monitoring sensor network," *Proceedings of the 7th symposium on Operating systems design and implementation*, pp. 381–396.

[15] P. Zhang, C. Sadler, S. Lyon, and M. Martonosi, "Hardware design experiences in ZebraNet," *Proceedings of the 2nd international conference on Embedded networked sensor systems (Sensys)*, pp. 227–238.

[16] C. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," *Proceedings of the 4th international conference on Embedded networked sensor systems*, pp. 265–278.

[17] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, pp. 1–14, 2009.

[18] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, "Four-bit wireless link estimation," *Proceedings of the Sixth Workshop on Hot Topics in Networks (HotNets VI)*, 2007.

[19] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle:A Self Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks," *NSDI*, pp. 15–28, 2004.

[20] R. Panta, M. Vintila, and S. Bagchi, "Fixed Cost Maintenance for Information Dissemination in Wireless Sensor Networks," *2010 29th IEEE International Symposium on Reliable Distributed Systems*, pp. 54–63.

[21] Y. Ko and N. Vaidya, "Flooding-based geocasting protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 7, no. 6, pp. 471–480, 2002.

[22] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Mobicom*, pp. 151–162, 1999.

[23] Z. Zhong and T. He, "Achieving range-free localization beyond connectivity," *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pp. 281–294, 2009.

[24] R. Stoleru, P. Vicaire, T. He, and J. Stankovic, "StarDust: a flexible architecture for passive localization in wireless sensor networks," *Proceedings of the 4th international conference on Embedded networked sensor systems*, pp. 57–70.

[25] R. Stoleru, T. He, J. Stankovic, and D. Luebke, "A high-accuracy, low-cost localization system for wireless sensor networks," *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pp. 13–26.

[26] http://www.tinyos.net.

[27] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: accurate and scalable simulation of entire tinyOS applications," *SenSys*, 2003.

[28] http://www.tinyos.net/tinyos-2.x/doc/html/tutorial/usc-topologies.html.