Design of Location Service for a Hybrid Network of Mobile Actors and Static Sensors*

Zhigang Chen, Min-gyu Cho and Kang G. Shin
EECS Department
University of Michigan
Ann Arbor, MI 48109-2122
Email: {zhigangc,mgcho,kgshin}@eecs.umich.edu

Abstract

Location services are essential to many applications running on a hybrid of wirelessly-networked mobile actors and static sensors, such as surveillance systems and the Pursuer and Evader Game (PEG). To our best knowledge, there has been no previous location service protocol for wireless sensor networks. A number of location service protocols have been proposed for mobile ad hoc networks, but they are not applicable to sensor networks due to the usually large per-hop latency between sensors.

This paper presents a distributed location service protocol (DLSP) for wireless sensor networks. Using a rigorous analysis of DLSP, we derive the condition for achieving a high packet-delivery ratio, and show how to configure the protocol parameters to ensure the scalability of DLSP. We find that DLSP is scalable if the mobile's speed is below a certain fraction of the packet-transmission speed, which depends on a movement threshold. For example, if the movement threshold for the location servers at the lowest level equals the radio range, the speed limit is one-tenth of the packet-transmission speed. The mobile's theoretical speed limit is one-fifth of the packet-transmission speed, beyond which DLSP cannot scale regardless of the movement threshold. Because of the high location-update overhead of DLSP, we propose an optimization, DLSP-SN, which can reduce the overhead by over 70%, while achieving high packet-delivery ratios. However, due to the griding effect, the packet's path length of DLSP-SN may be longer than that of DLSP, incurring higher data-delivery cost.

*The work reported in this paper was supported in part by the US Army Research Office under Grant W911NF-05-1-0421, and by the NSF under CNS-0435023.

1 Introduction

There are a growing number of sensor network applications that require communication between mobile actors and stationary sensors. For example, in the PEG (Pursuer and Evader Game) and surveillance systems, hundreds or thousands of sensors may be *statically* deployed to monitor certain areas or physical infrastructures, and a few dozens of actor nodes may move around and interrogate static sensors for information at multiple spots of interest.

Geographic routing (or location-based routing) [9, 12] has been widely used in mobile ad hoc networks (MANETs) as well as sensor networks, because it incurs low communication and memory overheads of maintaining routing information. A mobile periodically reports its (geographic) location to selected nodes, called *location servers*. Other nodes can acquire the mobile's location from one of its location servers and then deliver data to the mobile receiver using geographic routing. A number of location-service protocols have been proposed for MANETs, such as grid location service (GLS) [13], distributed location management (DLM) [21], geographic hashing location service (GHLS) [5], column row location service (XYLS) [19], DREAM [3], Twins [20], hierarchical location service (HLS) [10], and home-zone-based location service [18].

These location service protocols, however, may not be applicable to sensor networks due to the usually high per-hop latency in a sensor network which ranges from a few hundred milliseconds to a few seconds [14, 23], while that of a MANET is an order-of-magnitude lower (tens of *ms*) [7, 11]. The high per-hop latency can be attributed to the two factors – scheduling delay and transmission time. First, wireless communication consumes much more energy than other operations for (severely energy-constrained) sensor nodes. Hence, energy-efficient MAC protocols avoid idle listening and overhearing by scheduling transmission and listening periods (e.g., S-MAC [24] and T-MAC [4]),

Proceedings of the 27th IEEE International Real-Time Systems Symposium (RTSS'06)
0-7695-2761-2/06 $20.00 © 2006 IEEE

or low-power channel polling (e.g., WiseMAC [6] and BMAC [15]), or both (e.g., SCP [23]). As a result, the radio's duty cycle can be limited to a few percentages. Thus, a packet has to be held for some time before its transmission to the next hop. Second, a sensor node's radio usually has a lower bandwidth, incurring a longer transmission time. For example, Mica2 (MicaZ) has a bandwidth of 19.5 kbps (250 kbps), while MANETs typically use wireless LAN cards of 11 Mbps or 54 Mbps.

This high per-hop latency makes packet transmission in a sensor network much slower than in MANET. Moreover, a sensor network is usually of much larger scale than a MANET. Therefore, the location-service protocols are unlikely to perform well in sensor networks, because, during the nontrivial duration of delivering a message from a source node to a location server, then to the mobile receiver's location obtained from the location server, the mobile could have moved too far away to receive the message directly as in GHLS or even by using forward pointers as in GLS.

In this paper we present a distributed location service protocol (DLSP) for a hybrid wireless network of stationary sensor nodes and mobile actors. Like GLS, DLSP is built on a hierarchical grid structure. A mobile selects multiple location servers at each level, and sends location updates more frequently to the location servers at lower levels than to those at higher levels. A location query, i.e. a data packet may go through multiple rounds of "lookup-and-chase" to reach the mobile receiver. Through a rigorous analysis, we derive the condition to achieve a high query-delivery ratio (which equals data success rate in DLSP), and show how to configure the protocol parameters to ensure the scalability of the location service. In this paper, scalability means that, as the network size increases, the location service protocol retains high query-delivery ratio and the protocol overhead is proportional to $O(log(N))$, where N denotes the network size. We find that, in order to retain high query-delivery ratio, the mobile's speed should be below a certain fraction of the packet-transmission speed, which depends on the underlying movement threshold. For example, if the movement threshold for the lowest-level location servers is the same as the node's radio range, the mobile's speed limit is one-tenth of the packet-transmission speed. The theoretical speed limit is one-fifth of the packet-transmission speed beyond which DLSP cannot scale regardless of the movement threshold.

Like GLS, DLSP incurs a high location-update overhead because a mobile needs to update multiple location servers at each level with its location information. Therefore, we propose an optimization, called *DLSP with a selected neighbor* (DLSP-SN), in which the mobile updates the location server in at most one neighbor square at each level. The selection of a neighbor square is determined by the mobile's trajectory. DLSP-SN achieves a significant reduction of update overhead. However, due to the griding effect[1], DLSP-SN may incur more rounds of lookup-and-chase than DLSP, thus making the average path length of location queries greater than that of DLSP and increasing data-delivery cost.

The rest of this paper is organized as follows. Section 2 describes DLSP. Section 3 derives the condition for achieving a high packet-delivery ratio under DLSP, and Section 4 analyzes the overhead of DLSP, and presents an optimization, DLSP-SN. To validate our analysis results, we simulate the performance of location services in Section 5. We summarize the related work in Section 6, and conclude the paper and discuss future directions in Section 7.

2 Distributed Location Service Protocol

We now present the details of DLSP. We assume that a large number of stationary sensor nodes have been randomly and uniformly deployed in a field of interest and a relatively smaller number of mobile actors move around within this field. Geographic routing (e.g., GPSR [9]) is used for multi-hop routing. Each sensor node can determine its location by using a GPS receiver, or by invoking a localization service [8, 17]. Likewise, each mobile either is equipped with a GPS receiver or can estimate its location using the neighbor sensors' location information.

Table 1 summarizes the notations used in this paper.

2.1 Selection and Update of Location Servers

A sensor network is assumed to have been deployed in an $L \times L$ square field with the lower-left corner at (X_0, Y_0), as was assumed in GHT [16]. Similar to GLS [13], the entire square field is partitioned into a grid as shown in Figure 1. Four level-0 squares make up one level-1 square, four level-1 squares make up one level-2 square, and so on. To avoid overlap between two squares of the same size, a particular level-k square is part of one and only one level-$(k+1)$ square. For simplicity, we assume that the field is perfectly gridded, i.e., the field is a square of edge length $L = 2^h \ell$, where h is an integer. We will discuss how this restriction can be relaxed in Section 5. Each node is preloaded with L, ℓ, and (X_0, Y_0) upon which it can calculate the entire grid structure.

At time T, a mobile R uses a well-known hash function, $H(P, i, j, ID_R)$ [2] to compute a location, $L_{0,j}(P(R,T), ID_R)$

[1] 'Griding effect' means that the source and destination nodes across but close to the boundary of a high-level square may require the query to travel many hops up to the common square containing both nodes. Both GLS and DLSP-SN suffer from the griding effect, but DLSP does not.

[2] The hash function can be defined in many forms. For example, $H =$

Proceedings of the 27th IEEE International
Real-Time Systems Symposium (RTSS'06)
0-7695-2761-2/06 $20.00 © 2006 IEEE

$P(S)$	Location of a stationary sensor node S
ID_R	ID of a mobile node R
$P(R,T)$	Location of R at time T
$S_{k,0}(P)$	the level-k square in which P falls
$S_{k,j}(P)$	8 level-k neighbor squares adjacent to to $S_{k,0}(P)$ where $j = 1,\ldots,8$
$L_{k,j}(P(R,T),ID_R)$	The location that R picks in the square $S_{k,j}(P(R,T))$ at time T
$LS_{k,j}(P(R,T),ID_R)$	The level-k location server of mobile R in $S_{k,j}(P(R,T))$, i.e., the sensor node closest to $L_{k,j}(P(R,T),ID_R)$
L	Edge length of the square field of interest
(X_0,Y_0)	The location of the lower-left corner of the field
ℓ	Edge length of level-k square is $2^k\ell$
m	The movement threshold of location update at level-k location servers is $2^{k-m}\ell$
τ	The time threshold of location update at level-k location servers is $2^k\tau$
t_h	Per-hop latency, including transmission/retransmission time, and scheduling delay
p	Per-hop progress; the average decrease of Euclidean distance to the destination per hop
r	Radio range
$\bar{v}$	Mobiles' average speed
$dist(P_1,P_2)$	Distance between two locations, P_1 and P_2

Table 1. Summary of notations

in each level-0 square $S_{0,j}(P(R,T))$ $(j = 0,\ldots,8)$. Using the hash function, different mobile nodes are like to select different sensor nodes as their location servers, so the workload as well as energy consumption is distributed. The sensor node closest to this location is chosen as the mobile's level-0 location server, denoted as $LS_{0,j}(P(R,T),ID_R)$. A neighbor square is omitted if it is out of the field boundary. At level-1, R picks a location server from each of its neighbor squares, $S_{1,j}(P(R,T))$. There is no location server in $S_{1,0}(P(R,T))$, as it has been fully covered by the level-0 location servers, and so on. The location servers at *different* levels are updated at *different* rates. Suppose R has sent a location update to level-k location servers at time T_1. It will then send the next update to the level-k servers at $T + \Delta T$ if and only if $dist(P(R,T),P(R,T+\Delta T)) \geq 2^{k-m}\ell$ (i.e. the movement threshold) or $\Delta T \geq 2^k\tau$ (i.e. the timeout). R sets the *lifetime* of its location servers to be slightly greater than $\Delta T = min(2^k\tau, \frac{2^{k-m}\ell}{\bar{v}})$. If a location server does not receive a new update from the mobile R before this lifetime expires, it is no longer a location server for R.

2.2 Processing of Location Queries

When a sensor node S sends a data message to R, it only knows its own location and R's ID, and encapsulates the data into a location query. The query is first sent to a location server, and then to R's location found from the server. This *lookup-and-chase* process is illustrated by an example in Figures 2 and 3.

In Figure 2, S first assumes that R resides in $S_{0,j}(P(S))$ at some time, T_0, i.e. R and S are in the same level-0 square or two adjacent level-0 squares at T_0. Then $S_{0,0}(P(S),ID_R) = S_{0,j}(P(R,T_0),ID_R)$. Hence, S sends the query to the sensor node (N_1) closest to $L_{0,0}(P(S),ID_R)$. N_1 is not an R's location server, because it has not received any location update from R or the R's location information has expired. To explore the larger square $S_{1,0}(P(S))$, N_1 sends the query to N_2, i.e. the node closest to $L_{1,0}(P(S),ID_R)$, and so on. Suppose the query eventually reaches a location server, denoted as $LS_{2,0}(P(S),ID_R)$, which has the R's location at time T_1, denoted as $P(R,T_1)$. $LS_{2,0}(P(S),ID_R)$ (i.e., $LS_{2,4}(P(R,T_1),ID_R)$) then sends the query to $P(R,T_1)$. This process of looking up the location of, and chasing, a mobile is called a *round*.

If R moves fast and if S and R are far apart, by the time the location query reaches this location, R could have moved too far away from $P(R,T_1)$ to receive the location query. Then the query will be received by the node A closest to $P(R,T_1)$. Unlike GLS, A does not maintain any forward pointer under DLSP. Instead, it starts a new *round*. As shown in Figure 3, the query first goes to the node N_3 closest to $L_{0,0}(P(R,T_1),ID_R)$, then to $LS_{1,0}(P(R,T_1),ID_R)$ (i.e., $LS_{1,6}(P(R,T_2),ID_R)$), which has more recent R's location, $P(R,T_2)$. Finally, the query catches up with R near $P(R,T_2)$.

After receiving the query, R may decide whether or not to send its location information to S, which caches the location for later packets. In this paper, we intend to examine the performance of location services in essence, and leave it as future work how caching affects the performance of location services.

3 Conditions for High Packet-Delivery Ratio

In this section, we first derive the condition for achieving a high packet-delivery ratio under DLSP. Then, we discuss how to configure the parameters of DLSP to make it scalable. We find that DLSP is scalable if the mobile's speed is lower than a certain fraction of the packet-transmission speed, which depends on the movement threshold used. Last, we present the condition for achieving a high packet-delivery ratio in GHLS, and show that GHLS is not scalable.

3.1 Analysis of Conditions for High Packet-Delivery Ratio under DLSP

Our analysis of DLSP consists of the *base* case and the *inductive* step. The base case analyzes how a location query

(H_x,H_y). $H_x(P,i,j,ID_R) = C_x(P,i,j) + f_h(ID_R)\cdot 2^i\ell$. $C_x(P,i,j)$, the X axis of the lower-left corner of $S_{i,j}(P)$, can be calculated by P_x, X_0, ℓ, i, j, and ℓ. $f_h(ID_R)$ is a uniform function ranging between $(0,1)$. Similar is H_y.

Proceedings of the 27th IEEE International
Real-Time Systems Symposium (RTSS'06)
0-7695-2761-2/06 $20.00 © 2006 IEEE

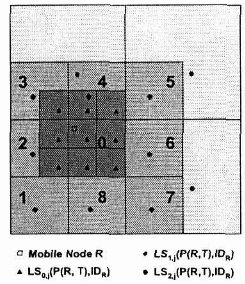

Figure 1. The location servers selected at three levels of the grid

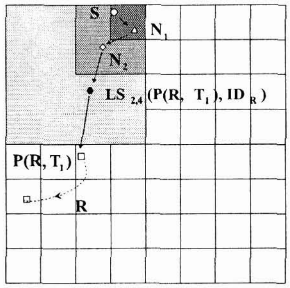

Figure 2. Round 1 of location query processing

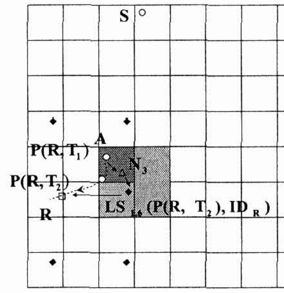

Figure 3. In round 2, only the location server in the shaded level-1 neighbor square is visited

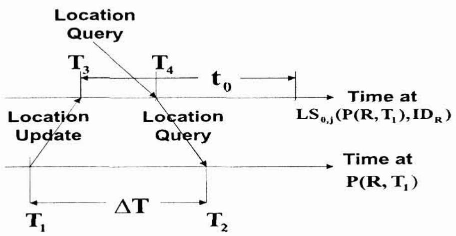

Figure 4. The timeline of events for location query processing at level-0.

can catch up with the mobile receiver after obtaining its location information from a level-0 location server. The inductive step analyzes how the location query can get closer to the mobile node after completing each round.

3.1.1 The Base Case

Suppose, at Time T_1, R sends its location, $P(R,T_1)$, to a level-0 location server, $LS_{0,j}(P(R,T_1),ID_R)$, $j \in \{0,1,\ldots,8\}$. The location server receives the location update at time T_3. At time T_4, it receives a location query and forwards the query to $P(R,T_1)$. The location query reaches $P(R,T_1)$ at time T_2. The timeline of these events are shown in Figure 4.

In order to have R receive the query at T_2, the following condition must be satisfied:

$$dist(P(R,T_1),P(R,T_2)) \leq r. \quad (1)$$

Suppose $\Delta T = T_2 - T_1$, then $dist(P(R,T_1),P(R,T_2))$ is bounded by $\Delta T\bar{v}$, because the distance is maximized when R moves on a straight line between T_1 and T_2. The average speed is computed as the length of the trajectory curve between T_1 and T_2 over ΔT. ΔT can be broken into three items, $T_3 - T_1$, $T_4 - T_3$, and $T_2 - T_4$. $T_3 - T_1$ denotes the average latency of the location update from $P(R,T_1)$ to $LS_{0,j}(P(R,T_1),ID_R)$; $T_4 - T_3$ represents the average obsoleteness of the location information on at location server; $T_2 - T_4$ denotes the average latency of the location query from $LS_{0,j}(P(R,T_1)$ to $P(R,T_1)$.

Let d_0 be the average distance between R and a level-0 location server, i.e., $dist(P(R,T_1),L_{0,j}(P(R,T_1),ID_R))$. Consider R as a random point in an $\ell \times \ell$ square, and the location server a random point in the same square or one of the eight adjacent squares, $d_0 \approx 1.27\ell$ according to the numerical analysis. Also, we let t_0 be the update interval for level-0 location servers. We have $T_3 - T_1 = T_2 - T_4 = \frac{d_0}{p}t_h$, and $T_4 - T_3 = \frac{1}{2}t_0$ because T_4 ranges from T_3 to $T_3 + t_0$. So,

$$\Delta T = \frac{1}{2}t_0 + 2\frac{d_0}{p}t_h. \quad (2)$$

Also, from Section 2, we have

$$t_0 = \begin{cases} \tau & \text{if } \bar{v} < \frac{2^{-m}\ell}{\tau} \\ \frac{2^{-m}\ell}{\bar{v}} & \text{if } \bar{v} \geq \frac{2^{-m}\ell}{\tau}. \end{cases} \quad (3)$$

From Eq. (3), we have

$$\bar{v}t_0 \leq 2^{-m}\ell. \quad (4)$$

Therefore,

$$dist(P(R,T_1),P(R,T_2)) \leq \frac{1}{2}t_0\bar{v} + 2\frac{d_0}{p}t_h\bar{v} \quad (5)$$

In order to satisfy Eq. (1), we simply let $\frac{1}{2}t_0\bar{v} + 2\frac{d_0}{p}t_h\bar{v} \leq r$. That is,

$$\begin{cases} \tau\bar{v} + \frac{5.08\ell}{p}t_h\bar{v} \leq 2r & \text{if } \bar{v} < \frac{2^{-m}\ell}{\tau} \\ 2^{-m}\ell + \frac{5.08\ell}{p}t_h\bar{v} \leq 2r & \text{if } \bar{v} \geq \frac{2^{-m}\ell}{\tau}. \end{cases} \quad (6)$$

Approximately, Eq. (6) can be satisfied if

$$2^{-m}\ell + \frac{5\ell}{p}t_h\bar{v} \leq 2r. \quad (7)$$

3.1.2 Analysis of the Inductive Step

Consider the case of requiring multiple rounds of lookup-and-chase. Suppose the query looks up R's location from a level-k_i location server in round i, and from a level-k_{i+1}

server in round $i+1$. To ensure the query makes progress towards R, we need to satisfy

$$k_{i+1} \leq k_i - 1. \tag{8}$$

Suppose the query gets R's location, $P(R,T_1')$ in round i and reaches $P(R,T_1')$ at time T_2'. $k_{i+1} \leq k_i - 1$ holds if the following inequality holds:

$$dist(P(R,T_1'),P(R,T_2')) \leq 2^{k_i-1}\ell. \tag{9}$$

Similar to Eq. (2), we get

$$\Delta T' = T_2' - T_1' = \frac{1}{2}2^{k_i}t_0 + 2\frac{2^{k_i}d_0}{p}t_h. \tag{10}$$

So, we have

$$dist(P(R,T_1'),P(R,T_1')) \leq \frac{1}{2}2^{k_i}t_0\bar{v} + 2\frac{2^{k_i}d_0}{p}t_h\bar{v} \tag{11}$$

In order to satisfy Eq. (8), we simply let $\frac{1}{2}2^{k_i}t_0\bar{v} + 2\frac{2^{k_i}d_0}{p}t_h\bar{v} \leq 2^{k_i-1}\ell$. That is,

$$\begin{cases} \tau\bar{v} + \frac{5.08\ell}{p}t_h\bar{v} \leq \ell & \text{if } \bar{v} < \frac{2^{-m}\ell}{\tau} \\ 2^{-m}\ell + \frac{5.08\ell}{p}t_h\bar{v} \leq \ell & \text{if } \bar{v} \geq \frac{2^{-m}\ell}{\tau}. \end{cases} \tag{12}$$

Again, because of Eq. (4), Eq. (12) can be satisfied if

$$2^{-m}\ell + \frac{5\ell}{p}t_h\bar{v} \leq \ell. \tag{13}$$

3.2 Configuration of Protocol Parameters for DLSP

The above analysis provides some insights into what parameters affect the packet-delivery ratio and how they can be configured to achieve the scalability of DLSP w.r.t. query delivery.

3.2.1 Configuration of ℓ

Consider the condition of the base case, Eq. (7), and that of the inductive step, Eq. (13). The condition of the base case is stronger than that of the inductive step if $\ell \geq 2r$. Moreover, both Eq. (7) and (13) are independent of the field edge length, L. Therefore, as long as data can be delivered within a small region (level-0 squares) of edge length $\ell \geq 2r$, it can be delivered from an arbitrarily far away node. In fact, ℓ should be set to $2r$, because the overhead of location updates increases as ℓ increases (in Section 4).

3.2.2 Configuration of m

In Eq. (7), $\frac{5\ell}{p}t_h v$ is always a positive term since t_h is not negligible. So, m must be a positive integer. Again, the overhead of location updates is proportional to 2^m when the mobile's speed is above the threshold. Therefore, m should be set to 1, and the movement threshold is r.

3.2.3 Limit of the Mobile's Speed

From Eq. (7), if $m = 1$, $\bar{v} < \frac{r}{5\ell}\frac{p}{t_h} = \frac{p}{10t_h}$, which is one-tenth of the packet transmission speed. If the movement threshold for location updates is smaller, the location updates are more frequent, and a mobile node is allowed to move faster. However, $\bar{v} < \frac{2r}{5\ell}\frac{p}{t_h}$ must always hold, and the speed can never be greater than $\frac{p}{5t_h}$. So, the theoretic speed limit of the mobile is one-fifth of the packet transmission speed, no matter how frequently the location servers are updated.

3.3 Analysis of Conditions for High Packet-Delivery Ratio in GHLS

GHLS can be considered as a trivial case of DLSP, in which $\ell = \mathrm{L}$. The analysis of GHLS is the same as that of the base case in DLSP, except that $d_0 \approx 0.5\mathrm{L}$ because the mobile node and its location server are considered two random points in the $\mathrm{L} \times \mathrm{L}$ square.

Suppose the movement threshold for updating the location server is $\bar{v}t_0 \leq d$. We need to satisfy

$$d + \frac{2\mathrm{L}}{p}t_h\bar{v} \leq 2r. \tag{14}$$

Because t_h is nontrivial, Eq. (14) may not hold for large networks and fast moving nodes, no matter how small d might be. Therefore, GHLS is not scalable.

4 Analysis of Location-Service Overhead

In this section, we first analyze the overhead of location updates under DLSP and then propose a design optimization, called DLSP with a Selected Neighbor (DLSP-SN) which makes a significant reduction of location-update overhead.

4.1 Analysis of Location-Update Overhead

Let U denote the total overhead of location updates, and u_k the overhead of updating a level-k location server. The location-update frequency for the level-k location servers is $t_k = 2^k t_0$. The average distance between R and a level-k location server ($LS_{k,j}(P(R,T),ID_R)$ is $1.27 \cdot 2^k\ell$, and the average distance between R and the level-0 location server, $LS_{0,0}(P(R,T),ID_R)$ is 0.5ℓ. Since there are at least 3 neighbor squares at each level except the highest, we have

$$\begin{aligned} U &\geq \sum_{k=0}^{h-1} 3 \cdot 1.27 \cdot 2^k \ell \frac{1}{t_k} + 0.5\ell\frac{1}{t_0} \\ &\geq (3.8h + 0.5)\frac{2^m v}{p} \end{aligned} \tag{15}$$

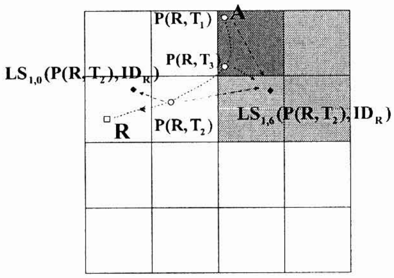

Figure 5. R sends updates to two level-1 location servers at $P(R,T_2)$, because $P(R,T_3)$ is in the selected neighbor square of $P(R,T_2)$.

$h = O(log(\mathrm{L} \times \mathrm{L}))$, and the total number of nodes, N is proportional to $\mathrm{L} \times \mathrm{L}$ for a given node density. So $U = O(log(N))$. That's DLSP is asymptotically scalable w.r.t. the protocol overhead. However, like GLS in small and median networks, DLSP suffers from a high update overhead because there are multiple location servers at each level of the hierarchy.

4.2 Optimization of DLSP

Our optimization goal is to reduce the location-update overhead while preserving the high packet-delivery ratio. The key observation is that it is unnecessary to update the location servers in all neighbor squares. This is because, as a location query "chases" the mobile receiver, the mobile's trajectory determines which location servers to visit.

This observation is illustrated in Figure 3. At time T_2, R updates the 5 location servers in the neighbor squares. Therefore, at round 2, the query can obtain a more recent location, $P(R,T_2)$, and catch up with R. Since A is in $S_{1,6}(P(R,T_2))$, the query relayed by A can only go through $LS_{1,6}(P(R,T_2),ID_R)$, not the other four level-1 location servers. That is, only the update to the location server in the neighbor square, $S_{1,6}(P(R,T_2))$ is useful for delivering this query. So, the design optimization is called *Distributed Location Service Protocol with a Selected Neighbor* (DLSP-SN).

To illustrate how DLSP-SN works, let us zoom in the lower-left level-2 square of Figure 3 in Figure 5. Suppose R needs to send location updates to level-1 location servers at $P(R,T_1)$, $P(R,T_3)$, and $P(R,T_2)$ consecutively. At $P(R,T_3)$, it checks if its previous location $P(R,T_1)$ was in the level-1 square, $S_{1,0}(P(R,T_1))$. If so, it only updates $LS_{1,0}(P(R,T_1),ID_R)$ (i.e., $LS_{1,6}(P(R,T_2),ID_R)$. At $P(R,T_2)$, R finds that its previous location $P(R,T_3)$ is in the neighbor square, $S_{1,6}(P(R,T_2))$, so it sends updates to both $LS_{1,0}(P(R,T_2),ID_R)$ and $LS_{1,6}(P(R,T_2),ID_R)$. Note that the locations of two consecutive level-k updates must be in the same level-k square or two neighbor level-k squares, because the movement threshold for level-k updates, $2^{k-m}\ell$, is strictly less than the edge length of level-k square, $2^k\ell$.

The difference between DLSP and DLSP-SN is summarized as follows. (1) Suppose the highest level is h. DLSP updates $LS_{0,j_1}(P(R,T),ID_R)$ ($j_1 = 0,1,\ldots,8$), and $LS_{k,j_2}(P(R,T),ID_R)$ ($k = 1,2,\ldots,h-1$ and $j_2 = 0,1,\ldots,8$). DLSP-SN updates $LS_{k,0}(P(R,T),ID_R)$ ($k = 0,2,\ldots,h$), as well as the location server in the selected neighbor square. (2) Suppose k_i and k_{i+1} are the levels of location servers DLSP and DLSP-SN obtains location information at round i and $i+1$. DLSP requires $k_i > k_{i+1}$, but DLSP-SN does not have this restriction. To avoid endless chasing, DLSP-SN requires that, at each round, the query gets more recent location information than the previous round.

DLSP-SN is less restrictive in the sense of obtaining location information, because it selects much fewer location servers than DLSP. As a result, DLSP-SN incurs more rounds and longer query path.

5 Evaluation

Using extensive simulation, we comparatively evaluate the performance of location-service protocols. We have implemented the DLSP protocols (DLSP, DLSP-SN) and GHLS in *ns-2* [2], and also ported GLS to the same version of *ns-2* we use for other protocols.

The following metrics are evaluated for the location service protocols: (1) Query Delivery Ratio—the percentage of location queries successfully delivered to the mobile receiver; (2) Update Overhead—the number of update packets transmitted with each hop counted as one packet transmission; (3) Query Path Length—the number of hops each successfully-delivered query takes.

5.1 The Simulation Scenario

The transmission range for radio communication is 100m, which is adopted from the characteristics of MicaZ [1] devices. Using the 802.11 MAC in *ns-2*, the transmission time plus the backoff delays ranges from 0.001 to 0.02s. Without a low-power MAC at hand, we add a fixed link-layer delay as 0.5s (or 0.25s). Thus, the actual per-hop latency ranges from 0.5 to 0.52s (or 0.25 to 0.27s), which resembles the per-hop latency in low-power MACs [14,23]. We assume the radio link is symmetric, and only collision may cause message loss. Typically, the raw radio of sensor nodes (e.g., Mica2, MicaZ) is lossy and asymmetric, but we rely on the underlying MAC or routing protocols to provide reliable transmission through scheduling and retransmission.

Sensors are uniformly distributed over a square area, with a density of 6.25 nodes per $100 \times 100\mathrm{m}^2$. Such a high node density is chosen because in low node-density networks, geographic routing (e.g., GPSR) suffers from relatively high packet losses, which may distract the readers from our main focus on the performance of location services. Given this high node density, the average per-hop progress is approximately $0.7r$, i.e. 70m. Our tests are run on networks of 400×400, 800×800, 1200×1200, and $1600 \times 1600\text{m}^2$, which include 100, 400, 900, and 1600 sensor nodes, respectively. Since interactions among mobiles is not considered, only one mobile is simulated in our evaluation, and its movement follows the modified random way-point mobility model [25]. The mobile's speed ranges from 4 to 40m/s, and the mobile's pause time is 0.

The beacon period for stationary sensor nodes is 10s, and 1s for the mobile. When a sensor node receives a beacon from the mobile, it replies with a beacon by a random delay ranging from 0 to 1s. The movement threshold for triggering location updates in DLSP, DLSP-SN, GHLS, and GLS is set to 100m (i.e. $m = 1$). The timeout for triggering location updates for the location service protocols except GLS (i.e. τ) is 8s. GLS does not have any timeout. Instead of using the instantaneous speed, the mobile node uses its average speed over a moving window. Suppose R sends two consecutive updates to its level-k location servers at time T and T'. The average speed $\bar{v} = \frac{dist(P(R,T),P(R,T'))}{T'-T}$, although R's trajectory can follow an arbitrary curve. To determine the timeout for the location information sent to a level-k location server, the mobile uses the average speed to predict the update interval $t_k = 2^k t_0$ by Eq. (3).

The edge length of the smallest square in the DLSP protocols (i.e. ℓ) is 200m. In GLS, the smallest square size is set to 100m, because all nodes in the same smallest square should be within two hops. The network size in our tests, $1200 \times 1200\text{m}^2$, does not result in a perfect grid structure. In such a case, if an intended level-k square is within the network boundary, it is substituted by a neighbor level-k square inside the boundary. For example, the level-2 square may be outside of the boundary when the mobile is located at (900m,900m). Then, the level-2 square $\{(0,0), (800\text{m},800\text{m})\}$ becomes its replacement.

Ten deployments of sensor nodes are generated for each network size. With each deployment, we generate a movement scenario for each speed. All test results are the averages of 10 runs on all the deployments. Since the mobile's ID is the same in all tests, a seed is randomly generated in each run so that a sensor node can hash the mobile's ID into a different value for DLSP protocols and GHLS. As for the workload, a sensor node is randomly chosen to send a location query to the mobile once every 2s for a period of 200s, i.e., 100 queries are sent. All tests for the same network size use the same workload. In GLS, every node should publish its location to its location servers for the correct functioning of GLS. For fair comparison, we modify GLS such that the sensor nodes publish their location only during the initial warm-up period of 120s. These location updates during the warm-up period are not counted in the update overhead.

For all protocols, the workload starts at 120s and the simulation ends at 400s. The surplus 80s allows the last few queries to be delivered.

5.2 The Simulation Results

5.2.1 Query-Delivery Ratio

Since the per-hop latency is about 0.5s, and the average per-hop progress of a message is about 70m, the average packet-transmission speed is calculated as 140m/s. For DLSP, the speed limit with the movement threshold of 100m is 14m/s. Figure 6 shows DLSP to scale very well if the mobile's speed is less than or equal to 15m/s. In the network of 1600 nodes, the delivery ratios of both DLSPs drop below 90% beyond the theoretic speed limit, 28m/s. We have also run tests with different per-hop latencies and different movement thresholds. The results are consistent with our analysis, and thus omitted.

The query-delivery ratio of DLSP-SN, as shown in Figure 7, is close to that of DLSP below 20m/s and even higher above that speed because DLSP requires the query to obtain location information from a lower-level location server than the previous round, but DLSP-SN does not have this restriction and can take more rounds of lookup-and-chase. Figure 8 shows that the delivery ratio of GHLS degrades significantly as the network size and the mobile's speed increase. This is because, as the per-hop latency is nontrivial, the term $\frac{2L}{p} t_h \bar{v}$ easily exceeds the bound, $2r$, in Eq. (14). When the query reached the location it obtained from the location server, the mobile has already moved too far away from that location to receive it. Hence, the message must be dropped.

The delivery ratio of GLS, shown in Figure 9 degrades significantly as the network size and the mobile's speed increase, also because the mobile has moved too far away to receive the query when it reaches the location. In GLS, the mobile attempts to leave a forwarding pointer in the grid of which it moves out, so that a query may follow the mobile using the forwarding pointers. But the messages containing the forwarding pointers are likely to get lost, particularly when the mobile moves at a high speed, because the destination of these messages (i.e., the grid it moves out of) is in the opposite direction of the node movement. By geographic forwarding, the mobile picks the neighbor that is closest to the destination. But such a neighbor is most likely to be out of the mobile's radio range. When a forwarding pointer is lost, the chain of forwarding pointers is broken, and the query has to be dropped.

GLS also shows some performance degradation at the low speed for the following reason. Unlike the other lo-

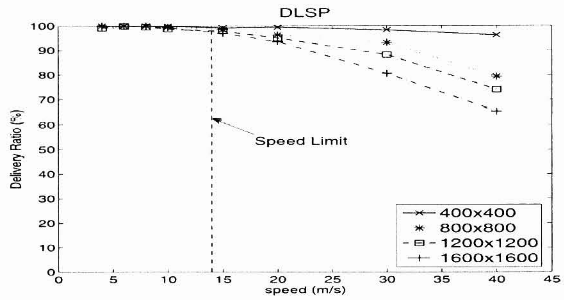

Figure 6. The query delivery ratio of DLSP is above 96% for all network sizes if the mobile's speed $\leq 15m/s$. The speed limit from our analysis is 14m/s.

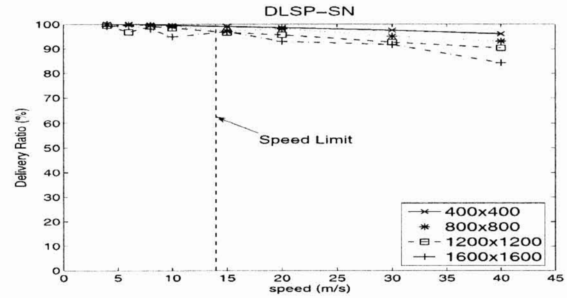

Figure 7. The query delivery ratio of DLSP-SN is close to that of DLSP below the speed limit, and noticeably better in case of high speeds.

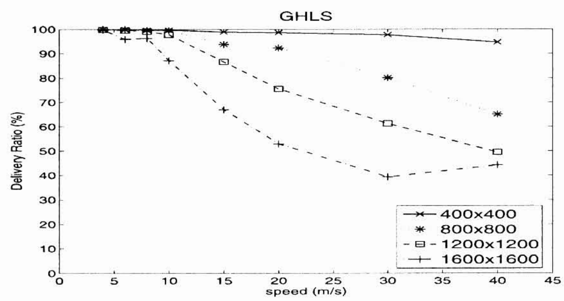

Figure 8. There is no single speed limit for different network sizes in GHLS because it does not scale.

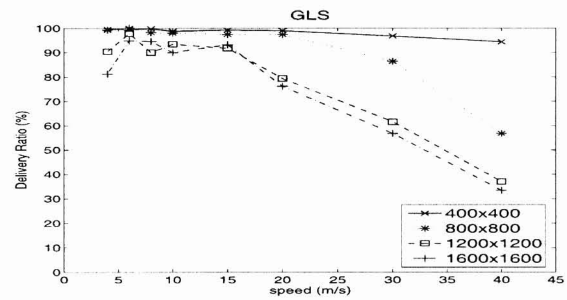

Figure 9. The delivery ratio of GLS degrades because many forward pointer messages are lost.

cation protocols we evaluate, the location updates are only triggered by the movement threshold in GLS. Therefore, when the mobile's speed is low, the update period is very long, especially for high-level location servers in large networks. Then, loss of a location update can disable these location servers for a very long time. Queries will be dropped if they reach these servers. At high speeds, the delivery ratio of small networks is noticeably better than that of large networks. This is because it is easier for a query to catch up with the mobile within smaller areas.

5.2.2 Location-Update Overhead

Because GHLS is shown to have the least update overhead in [5], we normalize the update overhead of DLSP, DLSP-SN, and GHLS by that of GHLS, illustrated by Figures 10 – 12. The results were obtained from the same tests for the query-delivery ratio. All the normalized overheads are relatively insensitive to the mobile's speed, because the tests of all protocols use the same movement threshold for triggering location updates. As the mobile's speed increases, the update overhead increases accordingly for all protocols.

Compared to DLSP, DLSP-SN reduces the location-update overhead by 70% or more, as shown in Figures 10 and 11. More importantly, the normalized overhead of DLSP-SN decreases as network size increases. This is because the normalized overhead of DLSP is $O(\frac{log(N)}{N})$, since the overhead of DLSP-SN is $O(log(N))$ and that of GHLS is $O(N)$. For this reason, GLS shows the similar trend in Figure 12. However, the trend is not clear for DLSP, which can be explained as follows. Because of network boundary, the number of location servers at any level increases as the network size grows. For example, the average number of level-0 (level-1) location servers increases from 4 to 6.25 (from 0 to 3) as N changes from 100 to 400. So the trend is offset by the increase of overhead due to additional location servers.

In Figure 12, the overhead of GLS increases almost linearly at low speeds for the following reason. GLS does not use any timeout for sending updates, so its update overhead always increases linearly with the mobile's speed. In GHLS, the timeout is 8s and the movement threshold is 100m, so, at low speeds, the mobile sends location updates every 8s, and the overhead of GHLS is constant even as the speed increases. Therefore, the normalized overhead of GLS increases linearly at low speeds. Compare GLS and DLSP-SN. The overhead of DLSP-SN is over %75 less than that of GLS, because it updates less location servers at each level and incurs less updates when the mobile crosses a square boundary. Compare GLS and DLSP. GLS is shown to have a much higher overhead than DLSP for $400 \times 400m^2$ networks, because GLS updates the same number of level-0 location servers (4) as DLSP does for this network size, and it incurs more overhead in boundary-crossing. As the network size grows, DLSP selects more location servers than GLS does, so its overhead catches up with or exceeds that

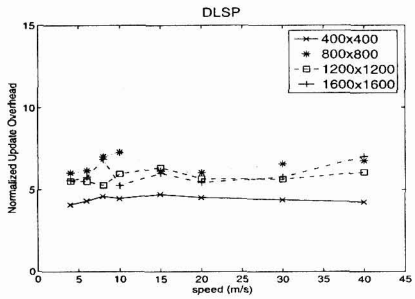

Figure 10. **DLSP has a very high update overhead because there may be as many as 8 location servers at each level.**

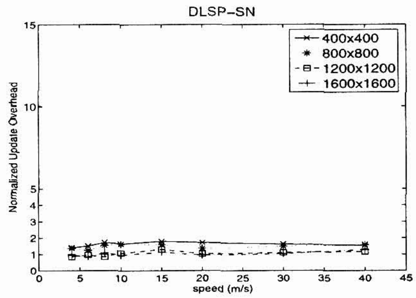

Figure 11. DLSP-SN reduces the update overhead by 70% or more. Its overhead is comparable to that of GHLS in a network of 900 nodes or more.

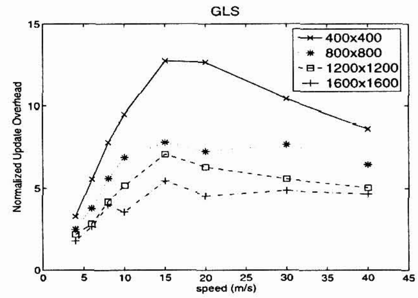

Figure 12. GLS has a very high update overhead because each level has 3 location servers, and because there is an additional overhead incurred by boundary-crossing.

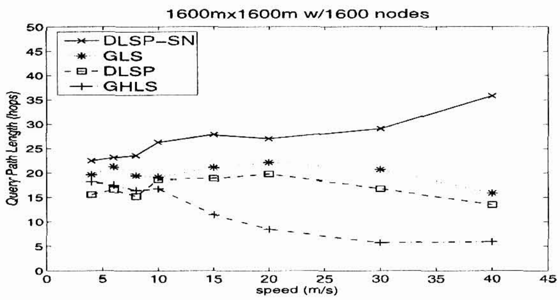

Figure 13. DLSP-SN has longer query path length due to griding effect.

of GLS.

5.2.3 Query Path Length

The results plotted in Figure 13 are also from the same tests for the query-delivery ratio. Due to the griding effect, the query path length of DLSP-SN is 40 – 45% longer than that of DLSP in large networks.

In Figure 13, the query-path length of GHLS decreases sharply beyond the mobile's speed of 10m/s, because more than 30% of the queries (most of them have a long path) are dropped and thus not counted. Similarly, the query-path lengths of DLSP and GLS decrease noticeably at 30 and 40m/s. These speed points are consistent with Figures 6, 8 and 9. Compared with GLS, DLSP-SN has a longer query path, because DLSP-SN uses less location servers than GLS. So, DLSP-SN suffers more from the griding effect. The results of smaller networks show the same trends with smaller gaps.

6 Related Work

To our best knowledge, there has been no previous work on location service in wireless sensor networks. A few location service protocols have been proposed in MANETs. Das *et al.* [5] categorizes these location services as flooding- or rendezvous-based.

In the flooding-based approach, such as DREAM [3], a mobile floods its location information to the nodes within a certain hop limit determined by *distance effect*. A location query is flooded towards the direction of the destination if the location is not available. This approach does not scale well due to the high overhead of flooding.

In the rendezvous-based approach, one or multiple *location servers* are elected to store mobiles' location information. The mapping of the mobiles' IDs to location servers is pre-determined by the protocol. In XYLS [19], each location update is sent to a set of nodes in a *thick column*, and each location query is propagated along a row of nodes, which should intersect with the column. Then, the intersected nodes send back the location to the source. Twins [20], Home-Zone-Based Location Service [18], and GHLS [5] all use hash functions to select a centralized location server. In Twins (or Home-Zone-Based Location Service), a home region (or a cluster) acts as the location server, while GHLS picks only one node as the location server.

GLS [13], DLM [21], and HLS [10] are hierarchical location service protocols. The differences between these protocols are as follows. GLS selects three location servers at each level of grids, which results in a non-uniform distribution of location servers. Then, a location query travels up the hierarchy by going to the node whose ID is closest to the destination ID within each level of squares. In DLM, a location server is selected in each of level-*m* squares, and a query is guided by the hierarchical address of the destination. In HLS, a mobile selects a responsible cell (RC) at each level of square it resides, and sends updates to every RC. Then, a query is routed along the candidate tree, i.e. the set of RCs for the destination mobile.

Das *et al.* [5] proposed a quantitative model and compared the performance of XYLS, GLS, and GHLS. It is shown that GHLS beats XYLS and GHLS w.r.t. both update overhead and packet-delivery ratio. The most important conclusion is that GLS asymptotically scales better but suffers from very heavy location-update overhead, and GHLS is the best for networks of up to 25000 nodes.

Outside the domain of location services, TTDD [22] takes a different approach. The data sources (stationary sensors) proactively build a grid structure throughout the sensor field and set up dissemination nodes near the grid points. A mobile sink floods a request for specific data within its local grid square to reach a dissemination node, which then forwards the request to the its upstream dissemination node towards the source, and so on.

7 Conclusion

Wireless sensor networks may incur a nontrivial per-hop latency, which is much larger than that in mobile ad hoc networks (MANETs). Therefore, the location service protocols proposed for MANETs may not be applicable to sensor networks. In this paper, we present a distributed location service protocol (DLSP) for a hybrid wireless network of stationary sensor nodes and mobile actors. To our best knowledge, there has been no previous work on location service for sensor networks.

Through a rigorous analysis of DLSP, we derive the condition for achieving a high packet-delivery ratio, and show how to configure the protocol parameters to ensure the scalability of DLSP. We find that DLSP is scalable if the mobile's speed is below a certain fraction of the packet-transmission speed, which depends on a movement threshold. The theoretical mobile's speed limit is one-fifth of the packet-transmission speed.

The proposed optimization, DLSP-SN, can reduce the location-update overhead by 70% or more, while its query-delivery ratio is even better than DLSP in case of high speeds. Moreover, in large networks, the overhead of DLSP-SN is close to that of GHLS, but it can provide a much higher delivery ratio than GHLS when the mobile's speed is high. With DLSP-SN, however, the query-path length gets up to $30-45\%$ longer than that of DLSP in large networks, indicating a significant increase of data-delivery cost when sensor nodes send continuous data stream to a mobile.

In our future work, we will explore how to adaptively make a tradeoff between update and data-delivery costs, and to improve overall energy-efficiency. We will also investigate how caching location information affects the performance of location services.

Acknowledgements

We thank Daniel Kiskis and Jerome Kinlaw for giving us helpful comments and the simulation platform to run the extensive simulations.

References

[1] MicaZ. http://www.xbow.com/products/productsdetails.aspx?sid=101.

[2] The network simulator - ns-2. http://www.isi.edu/nsnam/ns/.

[3] S. Basagni, I. Chalamtac, and V. R. Syrotiuk. A distance routing effect algorithm for mobility (DREAM). In *MobiCom '98*.

[4] T. Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *SenSys '03*.

[5] S. M. Das, H. Pucha, and Y. C. Hu. Performance comparison of scalable location services for geographic ad hoc routing. In *INFOCOM '05*.

[6] A. El-Hoiydi and J. Decotignie. WiseMAC, an ultra low power mac protocol for the WiseNET wireless sensor network. In *ISCC '04*.

[7] T. Goff, N. B. Abu-Ghazaleh, D. S. Phatak, and R. Kahvecioglu. Preemptive routing in ad hoc networks. In *MobiCom '01*.

[8] L. Hu and D. Evans. Localization for mobile sensor networks. In *MobiCom '04*.

[9] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *MobiCom '00*.

[10] W. Kieß, H. Füßler, J. Widmer, and M. Mauve. Hierarchical location service for mobile ad-hoc networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(4), 2004.

[11] H. Kim and J. Hou. Improving protocol capacity with model-based frame scheduling in IEEE 802.11operated WLANs. In *MobiCom '03*.

[12] S. Lee, B. Bhattacharjee, and S. Banerjee. Efficient geographic routing in multihop wireless networks. In *MobiHoc '05*.

[13] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In *MobiCom '00*.

[14] Y. Li, W. Ye, and J. Heidemann. Energy and latency control in low duty cycle MAC protocols. In *WCNC '05*.

[15] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *SenSys '04*.

[16] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. GHT: A geographic hash table for data-centric storage. In *WSNA '02*.

[17] A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom '01*.

[18] S. Sivavakeesar and G. Pavlou. Scalable location services for hierarchically organized mobile ad hoc networks. In *MobiHoc '05*.

[19] I. Stojmenovic. A routing strategy and quorum based location update scheme for ad hoc wireless networks. Technical report, University of Ottawa.

[20] A. C. Viana, M. D. de Amorim, S. Fdida, Y. Viniotis, and J. F. de Rezende. Easily-managed and topology-independent location service for self-organizing networks. In *MobiHoc '05*.

[21] Y. Xue, B. Li, and K. Nahrstedt. A scalable location management scheme in mobile ad-hoc networks. In *LCN '01*.

[22] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *MobiCom '02*.

[23] W. Ye and J. Heidemann. Ultra-low duty cycle MAC with scheduled channel polling. Technical Report ISI-TR-2005-604b, USC/Information Sciences Institute, 2005.

[24] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *INFOCOM '02*.

[25] J. Yoon, M. Liu, and B. Noble. Sound mobility models. In *MobiCom '03*.