

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.

In this report, we present 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 prove 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 lowest-level location servers is the same as the radio range, the mobile's 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 DLSP suffers from a high location-update overhead, we propose an optimization, called DLSP with the Selected Neighbor (DLSP-SN), which can reduce the update overhead by more than 70%, while achieving a high packet-delivery ratio. Due to the gridding effect, the average packet's path length of DLSP-SN is longer than that of DLSP. This increases data-delivery cost for continuous data streams. In order to make a tradeoff between update and data-delivery costs, we present a greedy adaptation mechanism, called DLSP-

ASN, which can make a significant improvement of overall energy-efficiency.

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, 11] 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) [12], distributed location management (DLM) [20], geographic hashing location service (GHLS) [5], column row location service (XYLS) [18], DREAM [3], Twins [19], and home-zone-based location service [17]. Das *et al.* [5] presented a quantitative model and compared the performance of these protocols.

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 [13, 22], while that of a MANET is an order-of-magnitude lower (tens of *ms*) [7, 10]. This long communication latency of sensor networks can be attributed to the following two facts. First, wireless

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

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 [23] and T-MAC [4]), or low-power channel polling (e.g., WiseMAC [6] and BMAC [14]), or both (e.g., SCP [22]). 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 node. 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 a forward pointer as in GLS.

In this report 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 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 packet-delivery ratio, and show how to configure the protocol parameters to ensure the scalability of DLSP, i.e., successfully delivering messages to the mobile receiver which moves around in a very large network. DLSP is proven to scale well if the mobile’s speed is 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. Moreover, in DLSP-SN, the delivery of a data packet may need to take multiple rounds of lookup-and-chase, thus making the average path length of location queries greater than that of DLSP. This increase of data-delivery cost may become significant, especially for continuous data streams. In order to make a tradeoff between location-update and data-delivery costs, we present a greedy adaptation mechanism, called DLSP-ASN, to improve the overall energy-efficiency.

The rest of this report is organized as follows. Section 2 describes DLSP, while Section 3 derives the condition for achieving a high packet-delivery ratio under DLSP and proves DLSP’s scalability when the mobile’s speed is below a certain fraction of the packet-transmission speed. Section 4 analyzes the overhead of DLSP, and presents an optimization, DLSP-SN. Section 5 proposes a greedy adaptation mechanism, DLSP-ASN. To validate our analysis results, we simulate the performance of location service in Section 6. We summarize the related work in Section 7, and conclude the report and discuss future directions in Section 8.

2 DLSP: Distributed Location Service Protocol

We now present the details of DLSP for a hybrid network of mobile actors and stationary sensors. 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. To deliver a message to a node that is multiple hops away from the sender node, geographic routing (e.g., GPSR [9]) is used. Each sensor node can determine its location by using a GPS receiver if it has, or by invoking a localization service [8, 16] if it doesn’t. 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 report.

2.1 Selection and Update of Location Servers

A sensor network is assumed to have been deployed in an $L \times L$ square field as was assumed in GHT [15]. Similar to GLS [12], the entire square field is partitioned into a grid as shown in Fig. 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

$P(S)$	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 $j = 1, \dots, 8$
$L_{k,j}(P(R, T), ID_R)$	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 $L_{k,j}(P(R, T), ID_R)$
$ALS_k(P(S), ID_R)$	The adaptive location server for a data source S in the square $S_{k,0}(P(S))$
L	Edge length of the square field of interest
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
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

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 6.

At time T , a mobile R uses a common hash function to compute a location, $L_{0,j}(P(R, T), ID_R)$ in each level-0 square $S_{0,j}(P(R, T))$ ($j = 0, \dots, 8$). 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 $dist(P(R, T), P(R, T + \Delta T)) \geq 2^{k-m} \ell$ or $\Delta T \geq 2^k \tau$. 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. This location query is first sent

to a location server, and then to R 's location found from the location server. This *lookup-and-chase* process is illustrated by an example in Figs. 2 and 3.

In Fig. 2, S first assumes that R resides in $S_{0,0}(P(S))$ at some time, and hence, S sends the query to the sensor node (N_1) closest to $L_{0,0}(P(S), ID_R)$. Note that $L_{k,0}(P_1, ID_R) = L_{k,0}(P_2, ID_R)$ if P_1 and P_2 are both located within the same level- k square. 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. In order to explore a larger square $S_{1,0}(P(S))$, N_1 sends the query to N_2 , which is 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 and chasing, a mobile is called a *round*.

If R moves fast and if S and R are far apart, then by the time the location query reaches this location, R could have moved too far away from $P(R, T_1)$ for R 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 of dropping the query, it starts a new *round*. As shown in Fig. 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 . For overall energy-efficiency, such a decision should depend on the sender's transmission rate, as discussed in Section 5.

3 Conditions for Achieving 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 prove 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. We find that the theoretic speed limit is one-fifth of the packet-transmission speed. Last, we present the condition for achieving a high packet-delivery ratio in GHLS, and show that GHLS is not scalable regardless of the update frequency.

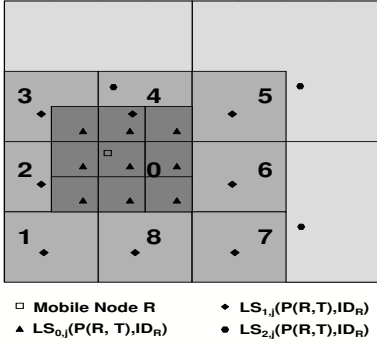


Figure 1. The location servers selected at three levels of the grid

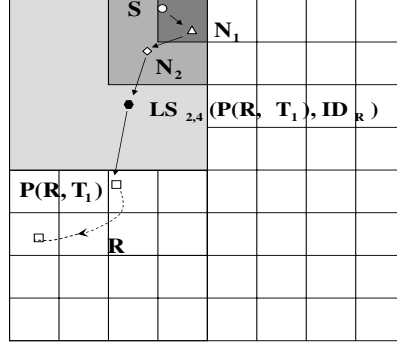


Figure 2. Round 1 of location query processing

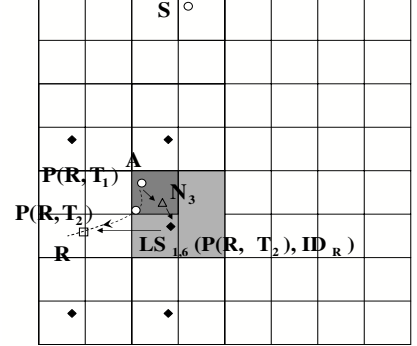


Figure 3. Round 2 of location query processing; only the location server in the shaded level-1 neighbor square is visited.

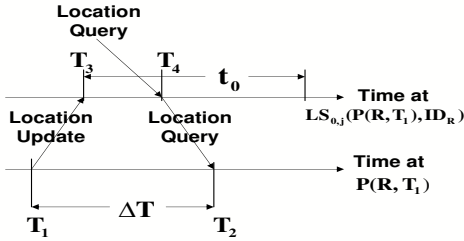


Figure 4. The timeline of events for location query processing at level-0.

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 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, $LS_{0,j}(P(R, T_1), ID_R)$, $j \in \{0, 1, \dots, 8\}$. 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 Fig. 4.

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

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

Suppose $\Delta T = T_2 - T_1$, then $\text{dist}(P(R, T_1), P(R, T_2))$ 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 transmission time 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 obsolescence of the location information on at location server; $T_2 - T_4$ denotes the transmission time 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., $\text{dist}(P(R, T_1), L_{0,j}(P(R, T_1), ID_R))$. Since the R and the level-0 location server can be located in anywhere of the level-0 square, d_0 is the average of the distance between two random points in the level-0 square. $d_0 \approx 1.27\ell$ 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,

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

$$= \begin{cases} \frac{1}{2}\tau\bar{v} + 2\frac{d_0}{p}t_h\bar{v} & \text{if } \bar{v} < \frac{2^{-m}\ell}{\tau} \\ 2^{-m-1}\ell + 2\frac{d_0}{p}t_h\bar{v} & \text{if } \bar{v} \geq \frac{2^{-m}\ell}{\tau}. \end{cases} \quad (5)$$

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,

$$\begin{cases} \tau\bar{v} + \frac{5\ell}{p}t_h\bar{v} \leq 2r & \text{if } \bar{v} < \frac{2^{-m}\ell}{\tau} \\ 2^{-m}\ell + \frac{5\ell}{p}t_h\bar{v} \leq 2r & \text{if } \bar{v} \geq \frac{2^{-m}\ell}{\tau}. \end{cases} \quad (6)$$

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

$$k_{i+1} \leq k_i - 1. \quad (8)$$

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:

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

Similar to Eq. (2), we get

$$\begin{aligned} \Delta T' &= T_2' - T_1' \\ &= \frac{1}{2}2^{k_i}t_0 + 2\frac{2^{k_i}d_0}{p}t_h. \end{aligned} \quad (10)$$

So, we have

$$\begin{aligned} \text{dist}(P(R, T_1'), P(R, T_2')) &\leq \frac{1}{2}2^{k_i}t_0\bar{v} + 2\frac{2^{k_i}d_0}{p}t_h\bar{v} \\ &= \begin{cases} 2^{k_i-1}\tau\bar{v} + 2.5\frac{2^{k_i}\ell}{p}t_h\bar{v} & \text{if } \bar{v} < \frac{2^{-m}\ell}{\tau} \\ 2^{k_i-m-1}\ell + 2.5\frac{2^{k_i}\ell}{p}t_h\bar{v} & \text{if } \bar{v} \geq \frac{2^{-m}\ell}{\tau} \end{cases} \end{aligned} \quad (11)$$

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,

$$\begin{cases} \tau\bar{v} + \frac{5\ell}{p}t_h\bar{v} \leq \ell & \text{if } \bar{v} < \frac{2^{-m}\ell}{\tau} \\ 2^{-m}\ell + \frac{5\ell}{p}t_h\bar{v} \leq \ell & \text{if } \bar{v} \geq \frac{2^{-m}\ell}{\tau}. \end{cases} \quad (12)$$

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

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

3.2 Configuration of Protocol Parameters for DLSP

Using 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.

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, Eq. (7) is independent of the field edge length, L . This indicates that, 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. That is, if $\ell \geq 2r$, DLSP is guaranteed to be scalable. 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), $\frac{5\ell}{p}t_h\bar{v}$ 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 $m = 1$, $\bar{v} < \frac{r}{5\ell} \frac{p}{t_h} = \frac{p}{10r t_h}$, which is one-tenth of the packet transmission speed. If the movement threshold for location updates is reduced, the speed limit can be lifted. 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 Scalability of DLSP

Theorem 1 *Let r be the radio range, and d the movement threshold. If the edge length of the smallest square is $2r$, and $\bar{v} \leq \frac{2r-d}{10r} \frac{p}{t_h}$ holds, DLSP is scalable.*

Similar to the analysis, this proof consists of the base case and the inductive step. Due to space limitation, we omit the full proof.

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

GHLS can be considered as a trivial case of DLSP, in which $\ell = L$. The analysis of GHLS is the same as that of

the base case in DLSP, except that $d_0 \approx 0.5L$ because there is only one location server in the $L \times L$ square.

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

$$d + \frac{2L}{p}t_h\bar{v} \leq 2r. \quad (14)$$

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 the 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 $\frac{1}{t_k} = \frac{2^{-k}}{t_0}$. 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

$$\begin{aligned} U &\geq \sum_{k=0}^{h-1} 3u_k \frac{1}{t_k} + u_0 \frac{1}{t_0} \\ &= \sum_{k=0}^{h-1} 3.8 \frac{\ell}{pt_0} + 0.5 \frac{\ell}{pt_0} \\ &= (3.8(h-1) + 0.5) \frac{1}{pt_0} \\ &\geq (3.8(h-1) + 0.5) \frac{2^{m_v}}{p} \end{aligned} \quad (15)$$

Hence, like GLS, 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

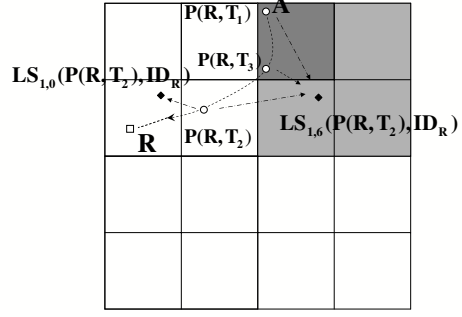


Figure 5. R sends updates to two level-1 location servers at $P(R, T_2)$, because $P(R, T_3)$ is in the selected neighbor square of $P(R, T_2)$.

a location query ‘‘chases’’ the mobile receiver, the mobile’s trajectory determines which location servers to visit.

This observation is illustrated in Fig. 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 the Selected Neighbor* (DLSP-SN).

To illustrate how DLSP-SN works, let us zoom in the lower-left level-2 square of Fig. 3 in Fig. 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.

- Suppose the highest level is h . DLSP updates $LS_{0,j_1}(P(R,T),ID_R)$ ($j_1 = 0, 1, \dots, 8$), and $LS_{k,j_2}(P(R,T),ID_R)$ ($k = 1, 2, \dots, h-1$ and $j_2 = 0, 1, \dots, 8$). DLSP-SN updates $LS_{k,0}(P(R,T),ID_R)$ ($k = 0, 2, \dots, h$), as well as the location server in the selected neighbor square.
- 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 Adaptation of Location Service

DLSP-SN reduces its update overhead, but may extend the query path, increasing data-delivery cost. This increase of data-delivery cost may become significant for delivering continuous data streams. To achieve overall energy-efficiency with DLSP-SN, we propose an adaptive location-update scheme in which a mobile adaptively sends its location updates based on the varying distribution and rate of the data sources. We then analyze the parameter configuration for the adaptation to ensure a high query-delivery ratio and present a greedy algorithm to improve overall energy-efficiency. Finally, we summarize the comparison among the hierarchical location service protocols, DLSP, DLSP-SN, DLSP-ASN, and GLS.

5.1 Adaptive Location Update

In MANETs, most data communication is one-to-one. After a mobile node receives a location query, it can periodically send location updates directly to the source node. The source can cache the location information and send data directly until the location information expires. In a hybrid wireless networks of static sensors and mobile actors, however, a mobile node may receive data from multiple data sources located in the areas of interest. If the mobile receiver sends location updates to each of these data sources, the location-update overhead can be prohibitive. Fortunately, the data sources in an area of interest may be spatially close. Therefore, the mobile node can send updates to only a few location servers shared by the data sources.

Fig. 6 provides an illustrative example. Suppose sensor nodes, S_1 and S_2 , reside in the same level-0 square $S_{0,0}(P(S_1))$, and continuously report data to a mobile node R . Instead of sending location updates to S_1 and S_2 individually, R picks an adaptive location server, $ALS_0(P(S_1), ID_R)$, in $S_{k,0}(P(S_1))$, and periodically updates it as well as the other location servers. When S_1 or S_2 send data to R , a location query is processed exactly the same as in Section. 2 except for the 1st round. R 's location can be obtained at $ALS_0(P(S_1), ID_R)$ instead of the level-2 location server, $LS_{2,j}(P(R, T_1), ID_R)$, as shown in Fig. 2. Thus, the communication cost of data delivery is reduced at the expense

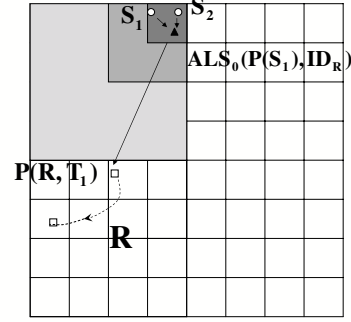


Figure 6. Location queries (or data packets) from S_1 and S_2 travel less hops during round 1 with adaptive location updates.

of extra adaptive location updates. To justify the need for adaptive location updates, we want to derive the condition for achieving a high query-delivery ratio, and analyze the overall energy-efficiency.

5.1.1 Condition for High Query-Delivery Ratio

Suppose R updates $ALS_0(P(S_1), ID_R)$ at time T_1' , and the location query reaches $P(R, T_1')$ at time T_2' . Let $T_a = 2^{k_a} t_0$ be the period of the location updates to $ALS_0(P(S_1), ID_R)$, and D be the distance between $L(P(S_1), ID_R)$ and $P(R, T_1')$.

Similar to the analysis of Eq. (10), we get

$$\begin{aligned} \Delta T' &= T_2' - T_1' \\ &= \frac{1}{2} 2^{k_a} t_0 + 2 \frac{D}{p} t_h. \end{aligned} \quad (16)$$

We simply let $\Delta T' \bar{v} \leq \frac{D}{2}$. That is,

$$2^{k_a-1} t_0 \bar{v} + 2 \frac{D}{p} t_h \bar{v} \leq \frac{D}{2}. \quad (17)$$

In Section 3, we have derived Eq. (4) and the speed limit, $v < \frac{p}{10t_h}$ with the movement threshold r . In order to satisfy Eq. (17), we have need:

$$2^{k_a-2} \ell + \frac{D}{5} \leq \frac{D}{2}. \quad (18)$$

So, approximately,

$$k_a \leq \log_2 \left(\frac{D}{\ell} \right). \quad (19)$$

¹In GLS, one square considers the three adjacent squares belonging to the same parent square as its neighbors; in DLSP protocols, all the eight adjacent squares are considered neighbors.

²'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.

Protocols	DLSP	DLSP-SN	DLSP-ASN	GLS
ID-to-LS Mapping	Hash the mobile node ID into a location within a square, and the node closest to the location becomes a location server	Same as DLSP	Same as DLSP	Select the node whose ID is closest to the mobile node ID within a square
LS in Neighbor Squares	One LS in each of the eight neighbor squares	At most one location server from a neighbor square	Same as DLSP-SN	One LS from each of the three neighbor squares
Cross-Boundary Updates	Location updates can only be triggered by timeout or exceeding the movement threshold	Same as DLSP, but the selected neighbor depends on whether or not a boundary of certain level is crossed	Same as DLSP-SN	Need to update the three neighbor location servers ¹ from level-0 to level- k if the boundary of a level- k square is crossed
Data Source Adaptive-ness	No	No	Yes. A mobile node may send additional location updates to the location servers near the data sources according to a greedy algorithm	Yes. Two communicating mobile nodes piggybacks their location information on data packets to each other. Also, a mobile node directly sends its location to the data source in one-way communication. However, there is no adaptive mechanism of updating location servers according to the data traffic.
Griding Effect ²	No	Yes	Not after the additional location updates are sent out	Yes
Multi-Round	Yes, with restrictions (1) the query needs to obtain a newer location of the mobile node at each round; (2) do not proceed if the location is obtained at a level-0 location server in the previous round; (3) the location server in the previous round should be at a higher level than that at this round.	Yes, with restrictions (1) and (2)	Same as DLSP-SN	No

Table 2. The comparison of hierarchical location services

When k_a is small, there is more update overhead but shorter less data-delivery cost; when k_a is large, there is less update overhead but higher data transmission cost. So, k_a needs to be configured on-the-fly to achieve overall energy-efficiency.

5.1.2 Analysis of Overall Energy-Efficiency

Let Eff denote the energy-efficiency without adaptive location updates, and Eff_a denote the efficiency with adaptive location updates. The mobile maintains a moving window to compute the average data rate, R_{data} , the average hop count, C_{hops} , and the average distance, D_{src} , from the two sources.

$$\begin{aligned} \Delta Eff &= Eff - Eff_a \\ &= R_{data} \cdot \left(C_{hops} - \frac{D_{src}}{p} \right) - \frac{D_{src}}{p} \frac{1}{2^{k_a} t_0} \end{aligned} \quad (20)$$

R periodically searches for k_a in $[0, \log_2(\frac{D}{l})]$ such that ΔEff is maximized.

In general, the rate and distribution of data sources can be complex. It is very difficult to compute the optimal solution that combines data sources and sets k_a for each combination. The mobile can use some simple heuristics to find good solutions. For example, only the data sources in the same level-0 are combined at their level-0 location server. We can then use the above analysis to set k_a for each level-0 location server.

5.2 Comparison of Hierarchical Location Services

The comparison of hierarchical location services is summarized in Table 2.

6 Evaluation

Using extensive simulation, we comparatively evaluate the performance of location-service protocols. We have implemented the DLSP protocols (DLSP, DLSP-SN, DLSP-

ASN) 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; (4) Overall Energy Cost—the total number of messages transmitted for both queries and location updates.

6.1 The Simulation Scenario

The transmission range for the radio of all the nodes is set to 100m, which is adopted from the characteristics of MicaZ [1] wireless sensor nodes. Also, we set the per-hop latency as 0.5s. 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 $100 \times 100 \text{ m}^2$. 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 node density, the average per-hop progress is approximately 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 we do not consider interactions among mobiles, only one mobile is simulated in our evaluation, and its movement follows the modified random way-point mobility model [24]. The mobile’s speed is set to 4, 6, 8, 10, 15, 20, 30, and 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, DLSP-ASN, GHLS, and GLS is set to 100m. The timeout for triggering location updates for the location service protocols except GLS 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 $\bar{v} = \frac{\text{dist}(P(R,T),P(R,T'))}{T'-T}$, 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 is $\ell = 200\text{m}$. 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 outside of 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 $(x = 900\text{m}, y = 900\text{m})$. Then, the level-2 square $\{(0,0), (800,800)\}$ becomes its replacement.

Ten sensor nodes are deployed for each network size. With each deployment, we generate a movement scenario for 4, 6, 8, 10, 15, 20, 30 and 40m/s, respectively. 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. Although this is reasonable for a MANET, for which GLS is designed, this may cause unnecessary communications to stationary sensor nodes. Thus, 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.

6.2 The Simulation Results

6.2.1 Query-Delivery Ratio

Since the per-hop latency is 0.5s, and the per-hop progress is about 70m, the packet transmission speed is 140m/s. For DLSP, the speed limit with the movement threshold of 100m is 14m/s. Fig. 7 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, 14m/s. We have also run tests with per-hop latency of 0.25s, and with the movement threshold of 50m. The results are consistent with our analysis, and thus omitted.

The query-delivery ratio of DLSP-SN, as shown in Fig. 8, 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

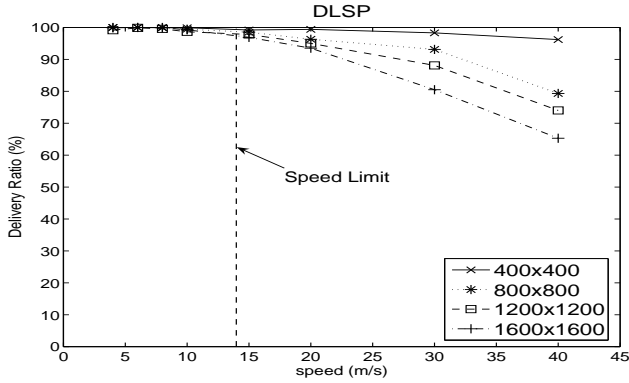


Figure 7. The query delivery ratio of DLSP is above 96% for all network sizes if the mobile’s speed $\leq 15m/s$. The speed limit from our analysis is 14m/s.

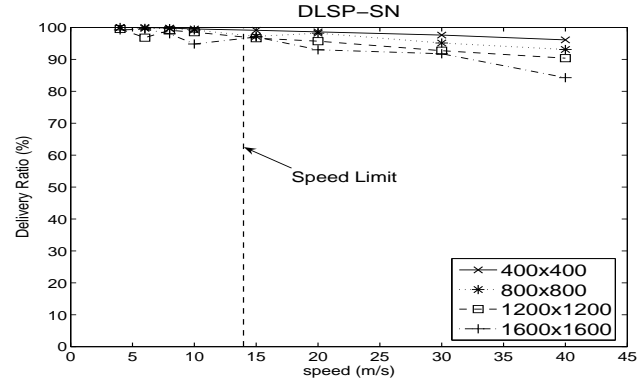


Figure 8. The query delivery ratio of DLSP-SN is close to that of DLSP below the speed limit, and noticeably better in case of high speeds.

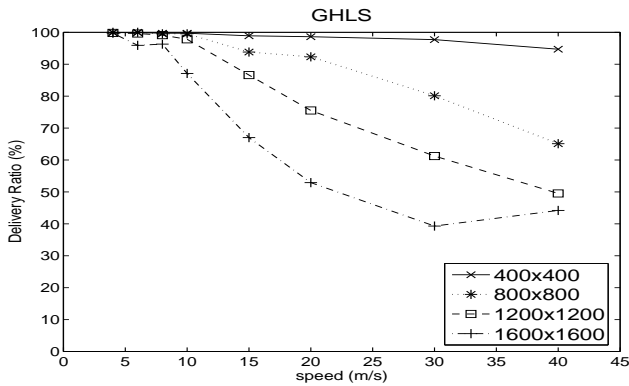


Figure 9. There is no single speed limit for different network sizes in GHLS because it does not scale.

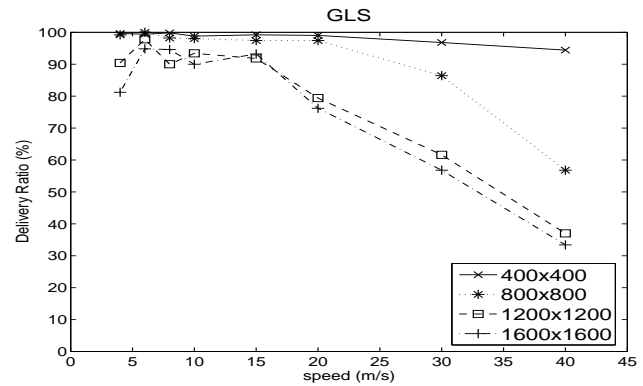


Figure 10. The delivery ratio of GLS degrades because many forward pointer messages are lost.

restriction and can take more rounds of lookup-and-chase. Fig. 9 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). That is, when the query reached the location it obtained from the location server, the mobile has larely moved too far away from that location to receive it. Hence, the message must be dropped.

Fig. 10 shows the delivery ratio of GLS, which degrades significantly as the network size and the mobile’s speed increase. This is 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 forwarding pointers 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.

This is 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 will be 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 location 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.

6.2.2 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 Figs. 11 – 13. 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 Figs. 11 and 12. More importantly, the normalized overhead of DLSP-SN decreases as network size increases. This can be explained as follows. Suppose the network size increases from $2^h \ell$ to $2^{h+1} \ell$. Then, the average number of packets transmitted for a location update should double in GHLS. But for DLSP-SN, the increase of update overhead is only $\frac{1}{h}$. In a network of 1200 nodes, the overhead of DLSP-SN is approximately equal to that of GHLS. This ensures DLSP-SN not to have serious collision problem even if there are a number of mobile nodes.

In Fig. 13, 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. Again, for the same reason described above, the normalized overhead of GLS decreases as the network size increases.

Compare Figs. 13 and 12. GLS has about 5 times as much overhead as DLSP-SN, because it updates 3 location servers at each level and incurs additional updates when the mobile crosses a square boundary.

Let’s compare the tests of 400×400 networks in Figs. 13 and 11. GLS is shown to have a much higher overhead than DLSP, because DLSP only has 4 level-0 squares in a 400×400 square, thus allowing each square to have only 3 neighbors (the same as GLS). GLS incurs an additional overhead in boundary-crossing.

6.2.3 Query Path Length

The results plotted in Fig. 14 are also from the same tests for the query-delivery ratio. Due to the gridding effect, the query path length of DLSP-SN is 40 – 45% longer than that of DLSP in large networks.

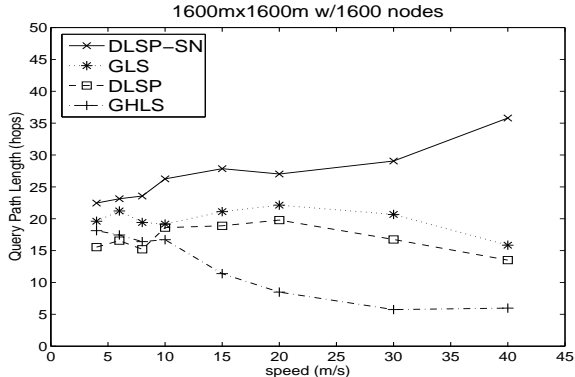


Figure 14. DLSP-SN has longer query path length due to gridding effect.

In Fig. 14, 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 Figs. 7, 9 and 10.

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 gridding effect.

The results of smaller networks show the same trends with smaller gaps.

6.2.4 Adaptation

To evaluate how DLSP-ASN improves overall energy-efficiency over DLSP-SN, we use the same deployments and movements as in previous tests, but change the workload such that only one sensor node sends data to the mobile at a constant rate. Figs. 15 and 16 show the results from a network of 1600 nodes and the data rate of 1 message per second.

Fig. 15 shows that, with adaptive location updates, DLSP-ASN has slightly higher delivery ratio than DLSP-SN. This is because, at the first round, each query travels less hops in DLSP-ASN than in DLSP-SN. This indicates that the query reaches the mobile’s known location earlier, and the mobile has moved less away from the location. Starting at the second round, the query has a higher chance to be delivered successfully. Fig. 16 shows the total energy cost normalized by the total cost of GHLS. The adaptive mechanism can improve the overall energy-efficiency by as much as 40%. For the range of 4 to 15m/s, the total energy cost of DLSP-ASN is comparable to, or even lower than, that of GHLS. This is because most queries are delivered in one round. During the first round, a query obtains the mobile’s location information within a small square in DLSP-ASN, but in GHLS, it has to travel more hops to a

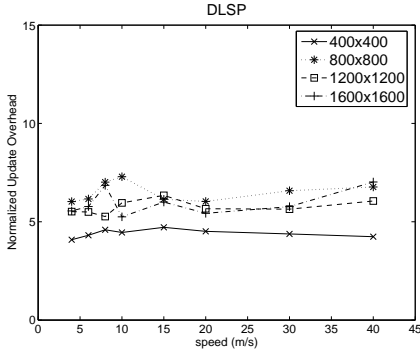


Figure 11. DLSP has a very high update overhead because there may be as many as 8 location servers at each level.

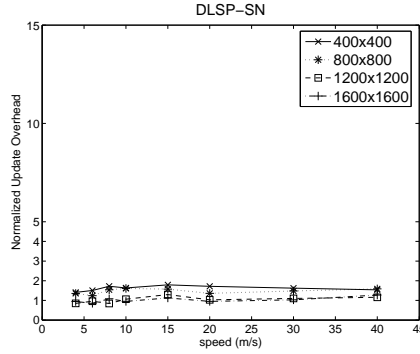


Figure 12. DLSP-SN reduces the update overhead by 70% or more. Its overhead is comparable to that of GHLS in a network of 900 nodes or more.

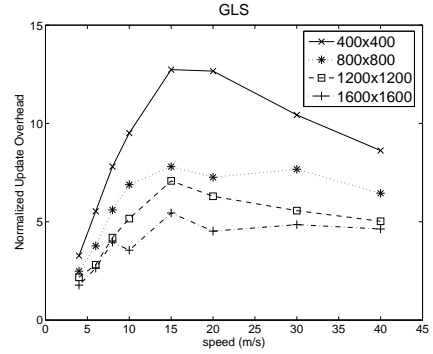


Figure 13. GLS has a very high update overhead because each level has 3 location servers, and because there is an additional overhead incurred by boundary-crossing.

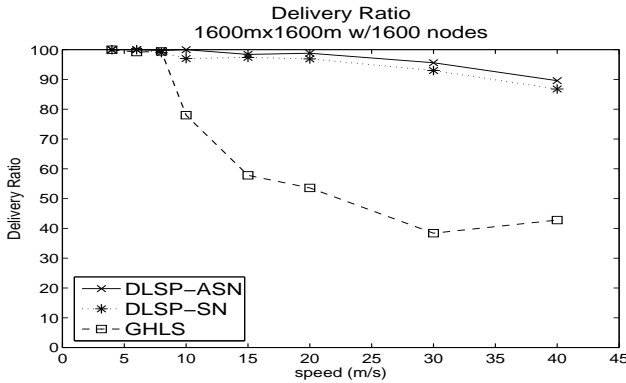


Figure 15. The delivery ratios of DLSP and GHLS match the results in previous figures.

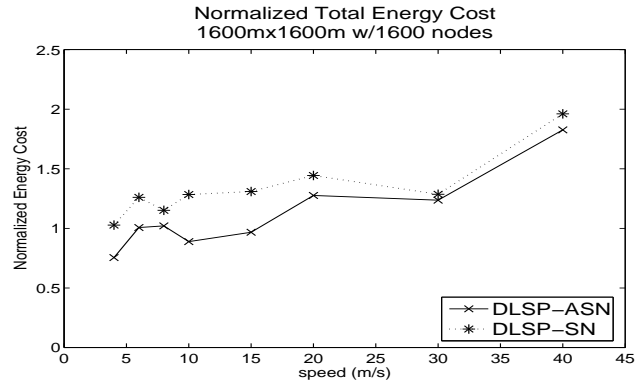


Figure 16. The energy cost of DLSP-ASN is even less than GHLS when the speed is below 15m/s, while it provides much higher packet delivery ratio.

randomly picked location server within the largest square. As the speed increases, the improvement decreases because DLSP-ASN also needs to take more rounds to deliver the queries. In the test of smaller networks and lower data rates, as we expected, DLSP-ASN shows less improvement on overall efficiency.

7 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*, and the update frequency is adapted to its frequency. When a node needs to send packets to a mobile destination, 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 [18], 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 [19], Home-Zone-Based Location Service [17], and GHLS [5]

all use a centralized location server for handling location updates and queries. In Twins and Home-Zone-Based Location Service, a mobile hashes its ID to determine its home region or cluster, which acts as the location server. GHLS uses the geographic hashing of GHT [15] to pick only one node as the location server. Both GLS [12] and DLM [20] are hierarchical location service protocols based on a grid structure. They both determine the frequency of updating the location servers by the movement of a mobile, and needs to update location servers of multiple levels when the boundary of a high-level square is crossed. Both protocols suffer from the “gridding” effect. The differences between the two protocols are as follows. GLS selects 3 location at each level of grids, which results in a non-uniform distribution of location servers; DLM selects a location server in each of level- m squares. In GLS, 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 query is guided by the hierarchical address of the destination.

Das *et al.* [5] proposed a quantitative model and compared the performance of XYLS, GLS, and GHLS. Their analysis and simulation results show that GHLS beats XYLS and GLS with regards to both update overhead and packet-delivery ratio. Their 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 [21] takes a different approach to data delivery to mobile sinks in sensor networks. 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. The request eventually stops at the source or a dissemination node which has the requested data (upon request from other sinks). Such a forwarding process also relays information of the path to the sink, so the source can send data directly to the mobile in the future.

8 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 report, 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 prove 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 30 – 45% longer than that of DLSP, indicating a significant increase of data-delivery cost when sensor nodes send continuous data stream to a mobile. In order to make a tradeoff between update and data-delivery costs, we present a greedy adaptation mechanism, DLSP-ASN, which can significantly improve overall energy-efficiency.

Our future work will address how data buffering and location caching can affect the performance of location services. By “data buffering” we mean that location servers can buffer data packets and send them to the mobile receivers as soon as their location updates are received. This allows a mobile to determine its location-update timing and frequency based on the requirement of data-packet latency and the constraint of memory space at the location servers.

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 *ACM MOBICOM*, 1998.
- [4] T. Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of ACM SenSys*, 2003.
- [5] S. M. Das, H. Pucha, and Y. C. Hu. Performance comparison of scalable location services for geographic ad hoc routing. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 217–228, 2005.
- [6] A. El-Hoiydi and J. Decotignie. WiseMAC, an ultra low power mac protocol for the WiseNET wireless sensor network. In *IEEE International Symposium on Computers and Communications*, 2004.
- [7] T. Goff, N. B. Abu-Ghazaleh, D. S. Phatak, and R. Kahvecioglu. Preemptive routing in ad hoc networks. In *Mobile Computing and Networking*, pages 43–52, 2001.
- [8] L. Hu and D. Evans. Localization for mobile sensor networks. In *Proceedings of 10th Annual International Confer-*

- ence on Mobile Computing and Networking (MobiCom'04), pages 45–57, New York, NY, USA, 2004. ACM Press.
- [9] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of 6th Annual International Conference on Mobile Computing and Networking (MobiCom'00)*, pages 243–254. ACM, 2000.
- [10] H. Kim and J. Hou. Improving protocol capacity with model-based frame scheduling in IEEE 802.11operated WLANs. In *ACM MobiCom*, 2003.
- [11] S. Lee, B. Bhattacharjee, and S. Banerjee. Efficient geographic routing in multihop wireless networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 230–241, New York, NY, USA, 2005. ACM Press.
- [12] 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: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 120–130, New York, NY, USA, 2000. ACM Press.
- [13] Y. Li, W. Ye, and J. Heidemann. Energy and latency control in low duty cycle MAC protocols. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, New Orleans, LA, USA, March 2005.
- [14] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of ACM SenSys*, 2004.
- [15] 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*, 2002.
- [16] A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of 7th Annual International Conference on Mobile Computing and Networking (MobiCom'01)*, pages 166–179, New York, NY, USA, 2001. ACM.
- [17] S. Sivavakeesar and G. Pavlou. Scalable location services for hierarchically organized mobile ad hoc networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 217–228, New York, NY, USA, 2005. ACM Press.
- [18] I. Stojmenovic. A routing strategy and quorum based location update scheme for ad hoc wireless networks. Technical report, University of Ottawa, 1999.
- [19] 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: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 193–204, New York, NY, USA, 2005. ACM Press.
- [20] Y. Xue, B. Li, and K. Nahrstedt. A scalable location management scheme in mobile ad-hoc networks. In *Proceedings of IEEE Conference on Local Computer Networks*, pages 102–111, Nov 2001.
- [21] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 148–159. ACM Press, 2002.
- [22] W. Ye and J. Heidemann. Ultra-low duty cycle MAC with scheduled channel polling. Technical Report ISI-TR-2005-604b, USC/Information Sciences Institute, July 2005.
- [23] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of INFOCOM*, 2002.
- [24] J. Yoon, M. Liu, and B. Noble. Sound mobility models. In *Proceedings of ACM/IEEE MobiCom 2003*, Sept. 2003.