

Combs, Needles, Haystacks: Balancing Push and Pull for Discovery in Large-Scale Sensor Networks

Xin Liu
Department of Computer Science
University of California
Davis, CA 95616, USA
Emails: liu@cs.ucdavis.edu

Qingfeng Huang and Ying Zhang
Palo Alto Research Center (PARC) Inc.
3333 Coyote Hill Road
Palo Alto, CA 94304, USA
Emails: {qhuang, yzhang}@parc.com

ABSTRACT

In this paper we investigate efficient strategies for supporting on-demand information dissemination and gathering in large-scale wireless sensor networks. In particular, we propose a “comb-needle” discovery support model resembling an ancient method: use a comb to help find a needle in sands or a haystack. The model combines push and pull for information dissemination and gathering. The push component features data duplication in a linear neighborhood of each node. The pull component features a dynamic formation of an on-demand routing structure resembling a comb. The comb-needle model enables us to investigate the cost of a spectrum of push and pull combinations for supporting discovery and query in large scale sensor networks. Our result shows that the optimal routing structure depends on the frequency of query occurrence and the spatial-temporal frequency of related events in the network. The benefit of balancing push and pull for discovery in large scale geometric networks are demonstrated. We also raise the issue of query coverage in unreliable networks and investigate how redundancy can improve the coverage via both theoretical analysis and simulation. Last, we study adaptive strategies for the case where the frequencies of query and events are unknown *a priori* and time-varying.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Protocols, Wireless Communications.

General Terms

Algorithms, Design

Keywords

Query, routing, data dissemination, sensor network, pull, push

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SenSys'04, November 3–5, 2004, Baltimore, Maryland, USA.
Copyright 2004 ACM 1-58113-879-2/04/0011 ...\$5.00.

1. INTRODUCTION

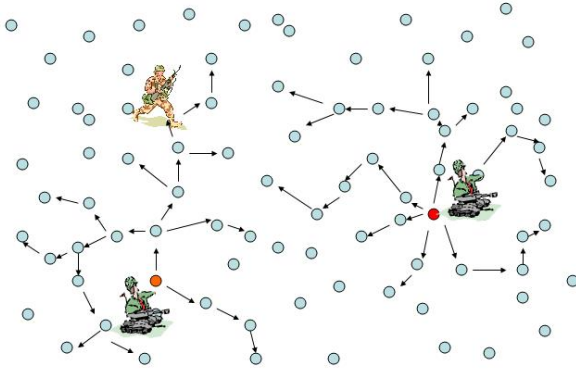
The goal of this work is to investigate efficient and reliable routing mechanisms for supporting queries in large scale ad hoc wireless sensor networks. Recently, wireless sensor network has attracted a lot of attention from the research community, e.g., [1, 2, 3, 4, 5, 6, 7, 8]. Many emerging sensor network applications involve dissemination of observed information to interested clients and thus demand efficient dissemination mechanisms due to resource limitations. For instance, a sensor network might be deployed for enhancing soldiers' battle field situation awareness when visibility is low (either at night or due to smoke). A soldier might be interested in where the tanks are in the battlefield. The nodes detecting the tanks can periodically push (broadcast) the information throughout the network in anticipation of the soldier's needs, as shown in Figure 1(a). This push-based information dissemination strategy is efficient when there are many soldiers in the network constantly in need of the information. However, a lot of broadcast bandwidth is wasted when the demand for the information is low. Alternatively, the system may choose a pull-based information dissemination strategy, as shown in Figure 1(b). The soldier broadcasts a query for the information when it is needed. When the nodes that have the information receive query, they send the information to the soldier. This pull-based strategy is relatively more efficient than the push-based strategy when the frequency of query is relatively low compared to the frequency of the interested events, because communication takes place only when it is needed. A natural question is: can we combine the advantages of both push and pull strategies and build an efficient query-support mechanism that adapts to the frequencies of query and events?

In this paper, we propose a novel “comb-needle” query support model that integrates both push and pull data dissemination, and analyze its performance in large scale ad hoc wireless networks. Using this model, we are able to explore a whole spectrum of push and pull strategies shown in Fig.2. In the comb-needle model, each sensor node pushes its data to a certain neighborhood and the query is disseminated only to a subset of the network. More specifically, the query process builds a routing structure dynamically that resembles a comb and the sensor nodes push the data duplication structure like a needle, as illustrated in Figure 3. One can view this query process as combing for needles in the sands or haystack. The desirable properties of this mechanism include:

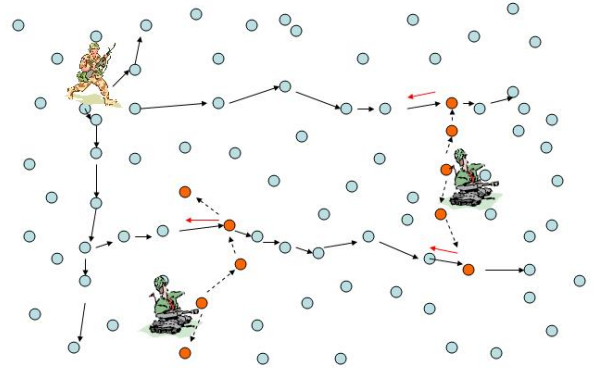
- In most cases, the comb-needle strategy is more effi-



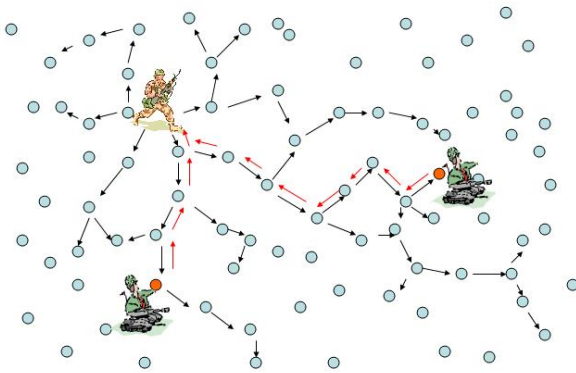
Figure 2: A Spectrum of Push-Pull Strategies



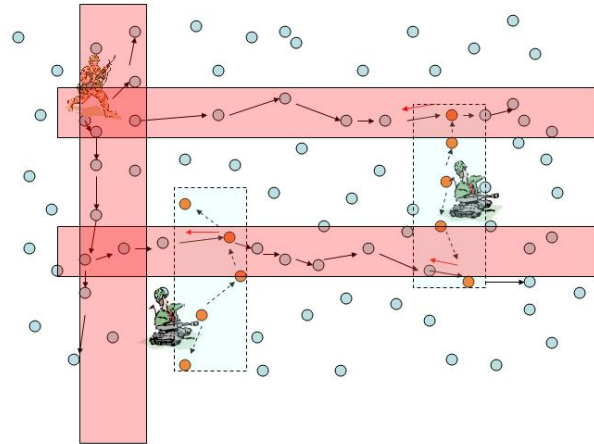
(a)



(a)



(b)



(b)

Figure 1: (a) Pure push-based strategy (b) Pure pull-based strategy

cient than both pure push and pure pull strategies. Pure push and pull strategies can be consider as the end points of the combing strategy.

- The combing granularity is dynamically adjustable based on demand. No infrastructure is built *a priori*.

We analyze the communication cost of the comb-needle strat-

Figure 3: A Comb-Needle Example

egy based on query frequency and event frequency, explore the issue of query coverage and reliability in the case of faulty networks, and study adaptive strategies for the case where the frequencies of query and events are unknown *a priori* and time-varying.

This study adopts a few assumptions about the environment, the network, and the application scenario:

- the event of interest can occur randomly at anywhere

at any time.

- the query entry point can be anywhere in the network and occur at any time.
- the network is ad hoc and uniform in the sense that all nodes are equivalent and the network does not have build-in hierarchy.
- all nodes in the network have information of their locations in the 2D space.

The remainder of the paper is organized as follows: in Section 2, we describe and analyze the comb-needle mechanism. In Section 3, we present simulation results for supporting queries on random grid networks. In Section 4, we explore the coverage issues in an unreliable network and present more robust strategies, which is followed by an adaptive comb-needle strategy in Section 5. Section 6 includes related work, which is followed by a discussion section and a conclusion section.

2. BALANCING PUSH AND PULL

2.1 Motivation

We consider the global discovery type query such as “where are all the tanks?”, and “what locations have a temperature exceeding 90 degrees?” etc. For such queries, the whole network need to be traversed by the underlying query resolution mechanism in order to get a complete response. Note that not all queries require a global discovery. Queries like “are there 5 tanks in the region?” could only traverse part of the network (up to the point where the answer is positive.) Further, we assume that query entry can be any nodes in the network. Such query generation mode can support mobile information-gathering agencies (e.g., soldiers in battlefield) or hierarchical networks where higher hierarchies are more intelligent and may demand information.

For resolving a global query, a frequently used query dissemination mechanism is the flooding-based querying (FBQ)[9]. In FBQ, the underlying query support mechanism floods the request to the whole network. All nodes with relevant information respond. The flooding often represent the most significant message overhead for such queries. Note that we are only interested in one-shot query, rather than continuous query. In a continuous query, directed diffusion type of schemes work sufficiently well since the initial flooding cost can be amortized over the duration of the continuous information flow from the sensor nodes to the querier.

FBQ represents a pull-based query support mechanism. As noted in the introduction, there is a potential advantage in combining the push based strategy with the pull based strategy to increase query efficiency. For instance, let’s consider a simple grid network like the one in Figure 4. Nodes (4, 2) and (7, 6) detected some abnormal event E , push the information vertically in the networks, and have it temporarily stored on the nodes traversed. Node (0, 8) is an entry point of a discovery query interested in such events. Note that the query only needs to be disseminated horizontally to be resolved, i.e., the information pull only need to occur horizontally rather than throughout the network. Assume that the events are only detected once at the two nodes, and the query only occurred once as well. The total message cost for supporting the query is about 30 by this push-pull

model, comparing the approximately 100 messages needed for the FBQ method. In the present hardware state, communication cost often weights much more than the storage cost, so this push-pull strategy seems desirable. One also can observe that the relative benefit of this push-pull strategy depends on the relative rates of occurrence of the events and query in the network. When there are more queries, there are more savings in the message overhead for supporting the queries. In this section we introduce a more general pull-push model and analyze its performance.

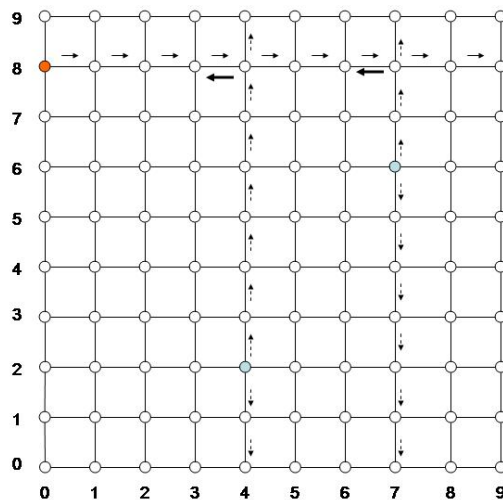


Figure 4: A Special Push-Pull Strategy

2.2 The Combing Strategy

The push-pull query support model we propose resembles the action of combing for needles in a haystack or in a pool of sand, and is thus dubbed as the “comb-needle” model. The comb-needle query support model combines both push and pull in the following way: each sensor node pushes its data to a certain neighborhood and the query node disseminates its request to a subset of the network. More specifically, the query process builds a routing structure dynamically that resembles a comb and the sensor nodes push the data duplication structure like a needle, as illustrated in Figure 3.

To analyze mathematically the cost and benefit of the query support mechanism, we need to have a model of the world that the sensor network is monitoring. In this paper, we assume the following random event and query models. Events occur uniformly in space and time across the sensor network. The discovery query entry point can be any node in the network with the same probability. Accordingly, we define two parameters:

f_q : the arrival frequency of discovery queries.

f_e : the arrival frequency of relevant events.

A life-time parameter could also be included in the query message to indicate whether it is interested in the events happened in the past half-an-hour or in the next five minutes. In this paper, we consider homogeneous query messages that are interested in events in one time unit.

2.3 Analysis of the Combing Strategy

For simplicity, like in the earlier example, first consider a grid network with n^2 nodes located at (i,j) where $0 \leq i, j < n$. Each node sends its state update to $2l$ of its vertical neighbors. For instance, node (i, j) will send its state update to nodes $(i, j+1), (i, j+2), \dots, (i, j+l)$ and $(i, j-1), (i, j-2), \dots, (i, j-l)$. An example is shown in Fig. 5 where the node in the middle duplicates its data to 4 neighbors (i.e., $l = 2$). Each data push incurs cost

$$C_l \sim 2l \quad (1)$$

in terms of the number of hops. We assume per-hop communication costs for the query propagation and data propagation are the same.

The query process first sends the query vertically then fans out horizontally. For instance, assume that node $(0,0)$ in Fig. 5 is the query source. A query is sent vertically from

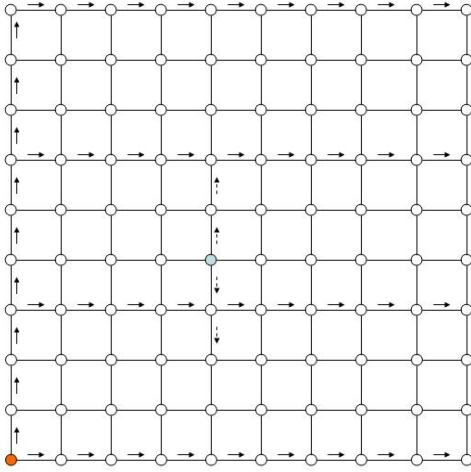


Figure 5: Combing

$(0,0)$ to $(n-1,0)$, and then fanned out horizontally from nodes $(0,0), (s,0), (2s,0), \dots, (\lfloor \frac{n-1}{s} \rfloor s, 0)$, and $(n-1,0)$. The resulting routing structure looks like a comb. The query dissemination cost C_{qd} is

$$C_{qd} = n - 1 + (n - 1)(\lfloor \frac{n-1}{s} \rfloor + 1) \quad (2)$$

Note that if $s = 2l + 1$, the query dissemination reaches its finest level, in the sense that it can collect the sensing data from every node.

The cost of query response depends on the reply scheme including data aggregation strategies used, if any. For simplicity, we first consider the reply scheme without data aggregation. Each node on the comb replies immediately if it has the matching data, including those data pushed from the nodes not on the comb. The reply paths are the reverses of the comb growing paths for the simplicity of analysis. In reality, the reply path could be a shorter one found by any ad hoc geometric routing algorithm, since the location of the destination is known.

Assume there is a match in node (i, j) , where $i = 0, s, 2s, \dots$. The reply cost is

$$c_{qr} = j + i \quad (3)$$

In the case when there is more than one matching data in a time window of τ , and they are uniformly distributed in the network with density $f_e \tau / n^2$, the total reply cost is about:

$$C_{qr} \sim (n - 1) * f_e \tau \quad (4)$$

Assume that the query frequency is f_q , the data event frequency is f_e , and $\tau = 1$ (in a unit time window). The total cost per query:

$$C = C_{qd} + C_{qr} + f_e * C_l / f_q \quad (5)$$

$$\simeq C_{qr} + 2(n - 1) + \frac{(n - 1)^2}{s} + s * \frac{f_e}{f_q} \quad (6)$$

$$\geq C_{qr} + 2(n - 1) + 2(n - 1) \sqrt{\frac{f_e}{f_q}} \quad (7)$$

This minimum of the cost is around

$$s_{optimal} \sim (n - 1) \sqrt{\frac{f_q}{f_e}} \sim \sqrt{N} \sqrt{\frac{f_q}{f_e}} \quad (8)$$

where N is the total number of nodes in the network. One can see that the result indicates that if the data frequency is relatively higher compared to the query frequency, the comb should be finer, i.e., with smaller inter-spike spacing. In other words, when we have more queries and less related events/data, the ‘‘combing degree’’ s should be larger and the ‘‘push degree’’ l should be larger, i.e., more push and less pull is better in such scenarios. On the other hand, when there are more events and less queries, l and s should be smaller.

When $f_q < f_e$, the global-pull-local-push model as shown in Figure 5 is an optimal one. If $f_q = f_e$, then the optimal strategy is the one shown in Figure 4. If $f_q > f_e$, then a reversed combing strategy is desired as shown in Figure 6, which is local-pull-global-push. An extreme of the reverse combing is the pure pull-based strategy where each data node broadcasts its event to all other nodes. This highlights the principle of the comb-needle structure: adjust the communication strategy based on the frequency of the query and events. In particular, the lower the (relative) frequency of the query/event, the larger the number of nodes it propagates to. Combined with the reverse comb, the comb strategy covers the whole spectrum of the push and pull strategies as shown in Figure 2. In this paper, we focus on the $f_q \leq f_e$ scenario.

Note that the baseline cost (using FBQ) per query is $O(N)$. So, comparing to the naive flooding based approach, this comb-needle scheme dramatically reduces query cost to $O(\sqrt{N})$. Note that the possibility of achieving this cost reduction partly relies on knowledge about the expected data and query frequency. Note also that the duplication of data in neighboring nodes incurs some storage cost as well. In this study, we focus on communication cost by assuming communication (energy) is a more restricted resource than storage in sensor networks. Last, throughout the paper, we count the number of hops as an indication of the power consumption, and assume that the per-hop communication costs for query propagation, data duplication, and data report are the same. In practice, the power consumption is more complicated: it consumes energy for a node to listen to a transmission or to be idle, the broadcast nature of wireless communications can be reduced the number of transmissions, and sleep-awake patterns are often implemented to reduce the


```

received query  $q$  at  $w$  do
  if  $new(q)$  and  $atComb(q.x, q.y, w.x, w.y, S, W)$  then
    broadcast  $q$ ;
     $events = getEvents(q)$ ;
    if  $events \neq \emptyset$  then
      broadcast  $r(events, q.x, q.y, w.x, w.y)$ ;
    end
  end
end

received event  $e$  at  $w$  do
  if  $new(e)$  and  $atNeedle(e.x, e.y, w.x, w.y, L, W)$  then
    copy  $e$  to memory
    broadcast  $e$ ;
  end
end

received report  $r(events, q_x, q_y, s_x, s_y)$  at  $w$  do
  if  $new(r)$  and  $q_x = w.x$  and  $q_y = w.y$  do
    return events;
  end
  if  $new(r)$  and  $atPath(q_x, q_y, s_x, s_y, w.x, w.y, W)$  then
    broadcast  $r$ ;
  end
end

function  $out = atComb(q_x, q_y, w_x, w_y, S, W)$ 
 $d = \text{mod}(|q_y - w_y|, S)$ ;
 $out = |q_x - w_x| < W$  or  $\min(d, S - d) < W$ ;

function  $out = atNeedle(e_x, e_y, w_x, w_y, L, W)$ 
 $out = |e_x - w_x| < W$  and  $|e_y - w_y| < L + W$ ;

function  $out = atPath(q_x, q_y, s_x, s_y, w_x, w_y, W)$ 
 $out = |s_y - w_y| < W$  and  $(w_x - s_x)(q_x - w_x) > 0$  or
 $|q_x - w_x| < W$  and  $(w_y - s_y)(q_y - w_y) > 0$ 

```

Figure 7: Constrained geographical flooding for the comb-needle query model

nodes in the grid. Figure 8 shows an instance of the network connectivity of this radio model for a grid network with small random offset. Two separate tests are done in this environment. The first test is to discover what is the best spacing for the comb, given query rate and event rate, in order to verify the theoretical result in the paper. The second test is to show the robustness of the protocol with a varying query width for a grid network with large random offset (Figure 9).

The first test was run on the network shown in Figure 8. We first let event rate be 1 packet per second (p/s), and query rate be 0.1 p/s ($f_q/f_e = 0.1$). Let the push length l be 0, 1, 2, 3 and comb spacing s be $1 + 2l$, i.e., 1, 3, 5, 7, respectively. The simulation runs 100 seconds. Figure 10 shows the energy consumption in the simulation. In this case, $\{l = 1, s = 3\}$ results in the minimum energy consumption. This result matches well with the predication in Eq. (8). We then reduce the event rate to 0.1 ($f_q/f_e = 1$). Figure 11 shows the energy consumption. In this case, $s = 5$ and $s = 7$ are close (with $s = 5$ slightly better), which consume the minimum amount of energy. It is partly due to the boundary effect, since both $s = 5$ and $s = 7$ have

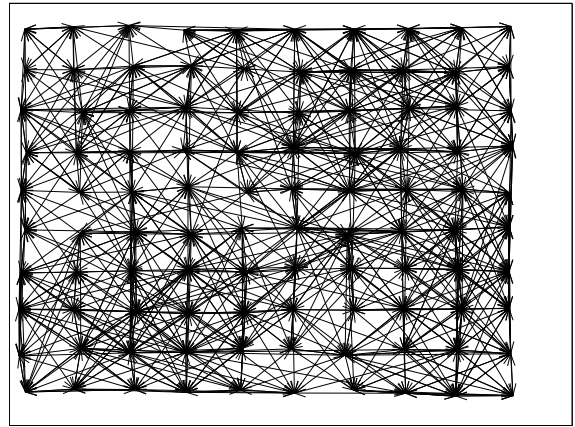


Figure 8: An instance of network connectivity

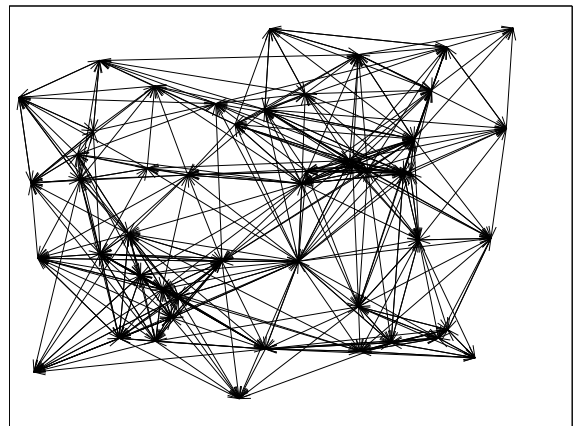


Figure 9: A random network (grid with large offset). Each rectangular box highlights the location of a node.

the same number of nodes in the comb ($\lfloor \frac{9}{5} \rfloor = \lfloor \frac{9}{7} \rfloor$), while $s = 5, l = 2$ has shorter needles.

The second test was run on the network shown in Figure 9. In this case, we let $f_q = 0.1$, $f_e = 1$, $l = 1$, $s = 3$, and the query width w be 0.5, 0.6, 0.7, 0.8, and 0.9. Figure 12 shows the success rate with respect to different CGF width. We can see that the wider the w is, the higher the success rate. Figure 13 shows corresponding the energy consumption; the wider the cost is higher. Clearly there is a trade-off between the success rate and the communication cost. Note that when w goes to 1.5, the combing is similar to flooding.

This simulation results also raise the issue of reliability in the query support routing mechanism. Due to the nature of wireless communication in distributed networks, the link between the nodes are not totally reliable. For instance, there is certain probability that collisions may occur and data transmission is not successful. This led us to investigate the impact of link failures on the routing structure and the issue of query support reliability.

4. RELIABILITY

As discussed in the last section, wireless links are unreli-

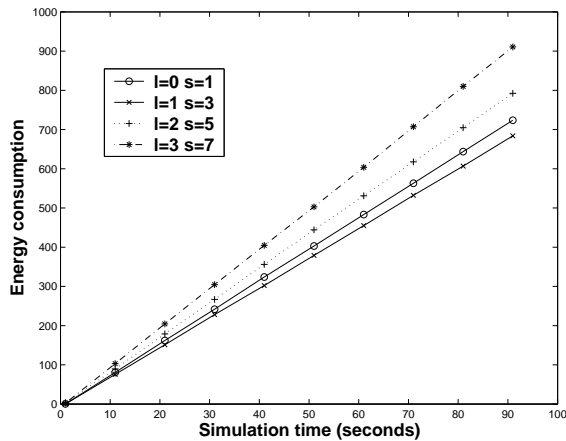


Figure 10: Energy consumption: $f_q = 0.1$, $f_e = 1$

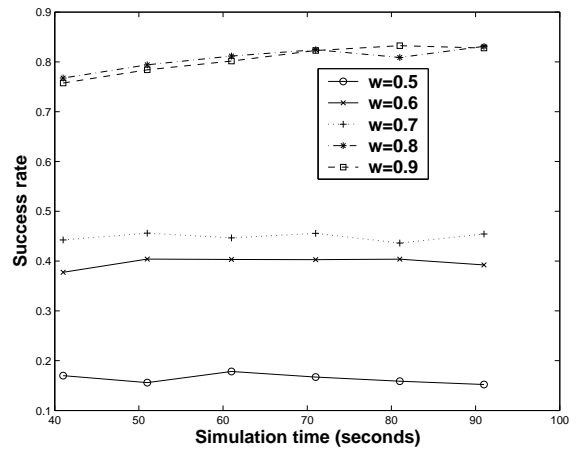


Figure 12: Success rate for different query widths

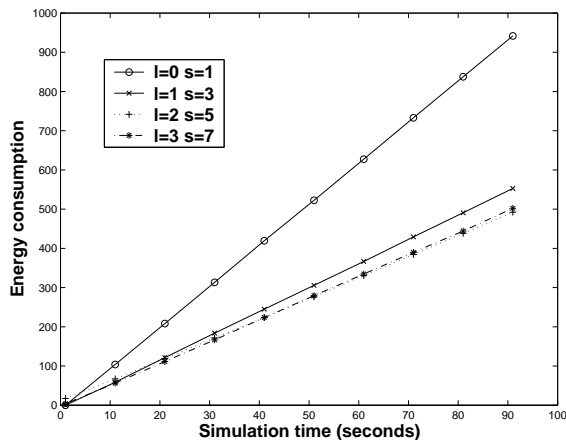


Figure 11: Energy consumption: $f_q = 0.1$, $f_e = 0.1$

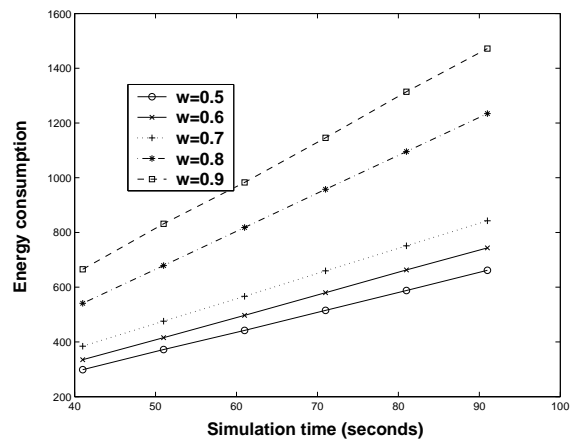


Figure 13: Energy consumption for different query widths

able due to the nature of wireless communications. In addition, sensor nodes are deployed in harsh environment unattended, and thus prone to failures. In this section, we first analyze the effect of node failures on the query and report, and then study reliable query-and-report strategies, namely, local re-enhancement and spacial redundancy schemes.

4.1 Query Coverage

There are usually two types of failures: transient failure and permanent failure. Because wireless channels are unreliable and time-varying [11, 12], one possible cause of a transient failure is the fading environment, which is modelled by Eq. (10) in our simulation in Section 3. When the channel between the transmitter and the receiver is in deep fading or collisions occur during the transmission, link-layer failures occur. Examples of permanent failures include the situations where nodes run out of energy or are physically damaged. We assume that each node has a failure probability p , which is independent of other nodes. (The independent assumption is discussed later.) Note that p can be a function of time to model the time-evolving reliability of the network: as time elapses, more nodes run out of battery and become unreliable. We look at a snapshot of the network,

and classify nodes as failure nodes and active nodes.

We consider the general case where any node can generate a query request. The question is how large an area can receive the query given node failures. The route of a query propagation is built based on the comb structure discussed earlier. We assume that if a node fails, then the query propagation stops. In Figure 14, we show the average probability of a node being covered by the query generated at node (i, j) where $0 \leq i, j \leq n - 1$. In the figure, we set $p = 0.005$, $n = 60$, and $s = 3$. When there exists a single route between the query node and the data node, any failure in the route will result the failure of the query. As the number of hops increases, the failure probability increases dramatically.

4.2 Local Enhancement

There are usually two types of schemes to enhance the reliable transmission of a message. Many research efforts have been devoted to reliable data delivery through vulnerable sensor networks. For instance, the authors of [14] propose the usage of interleaving paths to increase robustness. CFG used in the simulation in Section 3 exploits this type of strategy in a random network. In particular, all

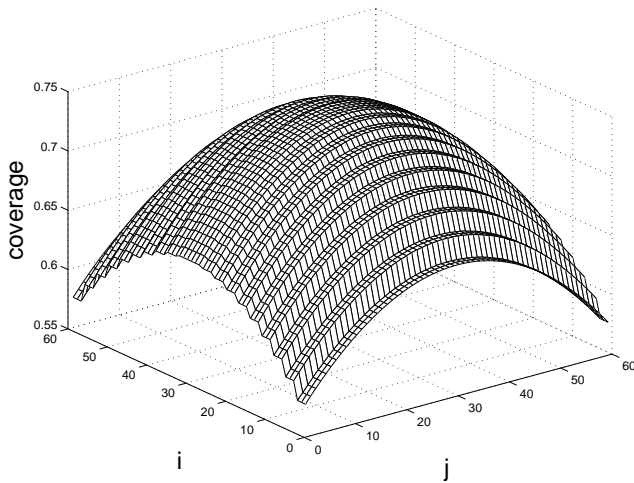


Figure 14: Query Reachability in the Network

nodes send data to all nodes at the next hop in an interleaving mesh. The performance of such an interleaving mesh is shown to be very robust against both independent link-layer packet loss and node failures[14]. Another approach to provide more reliable query and report is to develop alternative routes. Algorithms studied in the literature for developing alternative/backup routes can be applied here to enhance the transmission reliability.

Both link enhancement and route re-construction have their advantages and disadvantages. Link enhancement is simple and has low overhead. There is a minimum requirement on the MAC/routing protocols. For example, no re-transmission or acknowledgment is required. The functionality of each node can simply be listening-and-forwarding. On the other hand, route re-construction requires more routing functionality at each node and possibly a routing table to be attached to the message. To develop alternative routes incurs additional overhead and delay in exchanging information. As discussed earlier, there are transient failures and permanent failures. For transient failures, especially in small time-scale, simple redundancy scheme may perform better while reroute is preferred for permanent failures. Another important factor is the query source. If the queries are generated by a small number of fixed nodes, it may be energy-efficient to develop alternative routes, especially in the existence of permanent node failures. If the queries are generated by a large number of nodes randomly, it may be more cost-effective to use the link enhancement schemes.

4.3 Spacial Diversity

We note that failures may not be independent and random in a network as usually modelled. Node failure can be highly correlated. For example, sensor nodes in one area may have a higher probability of failure than other areas due to some unusual incidents, such as flooding and overheating. In addition, node failure can propagate. When a node fails, its neighboring nodes may have to share the sensing and communication burden of the failed node. They consume more energy and die quickly. Thus, node failures propagate in the network. The previous schemes, such as interleaving mesh, are not designed specifically to address such correlated node

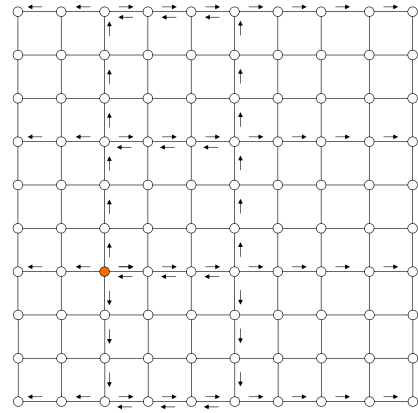


Figure 15: level-2 redundancy

failures in the network. Thus, we propose the use of spacial diversity to improve the reliability and coverage of the query process.

In Figure 15, we show a construction of query paths with spatial diversities. Basically, an additional vertical query propagation path is constructed. In each horizontal line, between the two vertical lines, the query message can be passed in either direction. If a node receives the same query message from both directions, it stops the message propagation. The basic idea is to provide spatial diversity and redundancy while local enhancement can be applied in each path. If one part of the network is seriously damaged beyond local enhancement, say one of the vertical lines is gone, nodes in other part of the network can still receive query messages propagated from the other vertical line.

The query-report success probability can be further improved by extending the needle length. For instance, we can extend the vertical data duplication in both directions (approximately) equally and let the needle length be $2s$. Thus, each report can reach two nodes that are on the query propagation path. If one of them receives the query successful, the query-report will be successful. We call this scheme a 2-level redundancy scheme. In the basic comb scheme, we have one vertical query propagation path, one horizontal query propagation path, and one vertical data duplication path from the query node to a data note. To improve the reliability, redundancy is introduced to all three of them, i.e., we allow one of the vertical lines, one of the horizontal lines, and one of the data duplication lines to fail. The failure probability can be approximated by

$$\begin{aligned}
 P_f(i, j|(x, y), k) &\approx a_1 + a_2 + a_3 - a_1 a_2 - a_2 a_3 - a_3 a_1 \\
 &\quad + a_1 a_2 a_3 \\
 &\approx a_1 + a_2 + a_3,
 \end{aligned}$$

where a_1 , a_2 , and a_3 approximate the error probabilities of failures of both vertical lines, both horizontal lines, and both data duplication lines, respectively. We have

$$\begin{aligned}
 a_1 &= (1 - (1 - p)^{|y-j|})(1 - (1 - p)^{|y-j|+k}) \\
 a_2 &= (1 - (1 - p)^{|x-i|})^2 \\
 a_3 &= (1 - (1 - p)^q)^2.
 \end{aligned}$$

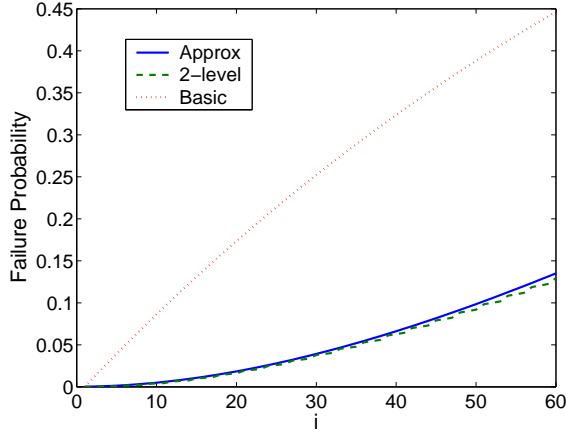


Figure 16: Compare the difference between the calculation and approximation.

Furthermore, we can provide high-level redundancy protection by providing more protections for vertical query, horizontal query, and vertical data duplications.

In Figure 16, we show the error probabilities for query generated by node $(0, 0)$ with $k = 1$ for data nodes (i, i) , $1 \leq i \leq n - 1$. We set $p = 0.005$, $n = 60$, and $s = 3$. Node $(0, 0)$ is chosen as the query node because it has the highest failure probability on average. It is not surprising that nodes closer to the query node has lower failure probability and nodes far away has high failure probability. It shows that the above approximation works well. In the figure, we also compare the redundancy scheme with the basic scheme. The spacial redundancy scheme significantly decreases the error probability especially for smaller values of p . Local enhancement (such as interleaving mesh) can be used to decrease the value of p .

In summary, in this section, we study the effect of node failures on the query-report process. Reliable transmissions studied in the literature can be used to enhance reliability locally and spacial redundancy can be implemented to further improve the query coverage of wireless sensor networks for correlated network failures.

5. ADAPTIVE COMB-NEEDLE STRATEGY

The basic comb-needle strategy assumes the knowledge of the query and event frequencies and analyzes the performance in a snap shot. In practice, the query and event frequencies may be time-varying, and thus a good query strategy should adapt to such changes. However, the values of the query and event frequencies may not be given. More importantly, other sensor nodes may not know the value of the inter-spike interval (s) used by a query node. This motivates the development of the adaptive comb scheme, as discussed in this section. Simulation results show that our adaptive scheme can track the changes well and keep a high success rate while maintaining a low communication cost.

We consider a time-slotted system where a query node is only interested in the event happened in the most recent time slot, i.e., $\tau = 1$. Let f_q be the probability that a query

is generated in a time slot. Let

$$f_d = \frac{f_e}{n^2}$$

be the probability that a sensor node detects an event in a time slot. In this case, we can rewrite Eq. (8) by

$$s_{optimal} \sim \sqrt{\frac{f_q}{f_d}}.$$

Thus, to calculate inter-spike interval and the needle length, a node only needs to estimate f_d and f_q .

We assume that the distribution of f_d and f_q are known to both query nodes and data nodes. If such information is not available, a default distribution can be assumed. For the query node, it has an estimate of f_d and f_q . (Initially, an unbiased estimate is the mean values of f_d and f_q .) The larger the number of queries the node performs, the better the estimate of f_d and f_q are. Based on the estimate, the query node calculates s .

Data nodes take a different approach. Let L be the total needle length at a data node. In particular, if the data node does not duplicate its data to any other nodes, then $L = 1$. In addition, we allow asymmetric data duplications to fine tune the needle length. For example, if the data is duplicated to one node south and to two nodes north, then $L = 4$. Analysis in Section 2 can be extended to such a case. Initially, each data node estimates its own data frequency f_d . Based on the estimation of f_d and the knowledge of the distribution of f_q , each data node calculates L such that

$$P(L > s) \geq P_m,$$

where P_m is a predefined threshold indicating the importance of the data.

Even if the needle length of a node is smaller than the inter-spike interval, its data duplication may still reach one query line if its distance to the line is smaller than the needle length in that direction. To elaborate, the success probability is L/s given $L < s$, and the over query success probability is

$$\begin{aligned} P_s &= P(L \geq s) + E\left(\frac{L}{s} | L < s\right) P(L < s) \\ &= E\left(\min\left(\frac{L}{s}, 1\right)\right). \end{aligned}$$

In general, the estimates of f_d and f_q at a data node may be poor and so does the value of L calculated based on the estimates, especially when f_d is small and the node misses some queries. Thus, the values of s obtained from previous queries are important and should be used to estimate L . A node learns about the value of s in two cases. First, a node may obtain information when its data is successfully queried. Second, we enhance the possibility that a node learns about the query by implementing a rotation of the query horizontal duplications. Suppose that the current horizontal queries occur on lines $0, s, 2s, \dots, [(n-1)/s] * s$, then the next query occur on lines $1, 1 + s, 1 + 2s, \dots, 1 + [(n-1)/s] * s$, and so on. A node may not generate an event that is successfully queried. However, when it is on a spike in the comb, it obtains the query information s . A node uses its most recent information on s as an important factor to calculate its needle length.

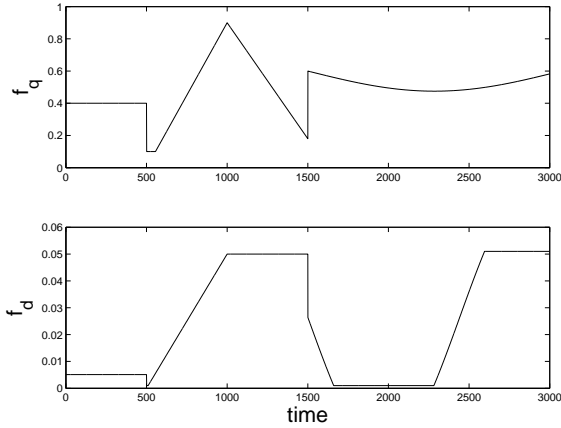


Figure 17: The query frequency and the event frequency.

Note that the knowledge of the distribution of f_d and f_q is not critical. A more important issue is that the needle length of data nodes should synchronize with the inter-spike interval. Since the query node can learn the value of f_d and f_q faster and better, it is important for data nodes to obtain the value of s from the query node, which is enhanced by the query rotation strategy. Given the value of the most recently obtained s , the needle length at each data node is adjusted according to the importance of the data and the desired query success rate. The rotation scheme does not trigger additional communication cost, but does require memory spaces.

We simulate the adaptive scheme for 3000 time slots in a 20×20 regular grid. Figure 17 shows the query frequency and the data frequency. We assume node $(0, 0)$ is the query node, which generates query with probability f_q in each time slot. Each other node generates events independently with probability f_d in a time slot. In the adaptive scheme, both the query node and data nodes estimate the values of f_q and f_d using a moving average. The inter-spike interval and the needle length are calculated based on the estimations. In addition, the query rotation scheme is implemented in the simulation.

The adaptive scheme is compared with the ideal case, where the query node knows f_q and f_d precisely, and all data nodes know the value of current s . Such the ideal case consumes the least amount of energy to guarantee 100% query successes.

In this simulation, the query-report success ratio of the adaptive scheme is 99.33% and the total communication cost is 99.64% of the ideal case. The query success ratio is the ratio of the successful reports in the adaptive scheme over that of the ideal one. The communication cost is defined as the total cost to report events to the query node, which includes query cost, data duplication cost, and data report cost. We count one hop of query/event as one unit of communication cost. The estimated case consumes less energy at the cost of lower success ratio. In particular, there are cases where needles are not long enough to reach a query line. Such events are not reported back to the query node. There exists a trade-off between the success probability and the communication cost.

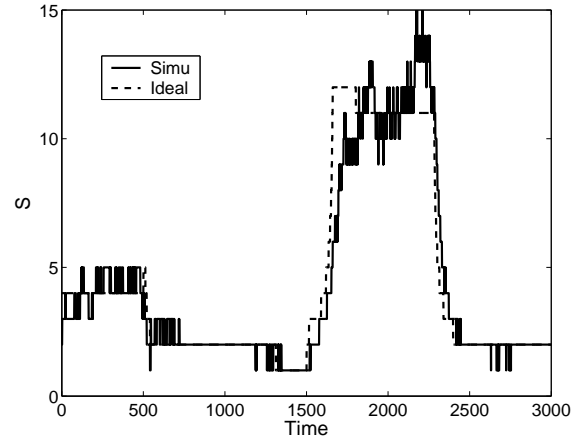


Figure 18: The ideal comb width and the estimated comb width.

Figure 18 compares the ideal comb size and the adaptive scheme where f_d and f_q are estimated. We can see that the adaptive scheme can track the changes in the ideal case. The fluctuation is due to the estimation errors on f_q and f_d , especially on f_q , at the query node.

Table 1 compares the comb scheme with that of push and pull in different time periods where the values of f_d and f_q differ. Each entry in the table represents a communication cost normalized over that of the ideal comb. We can observe that the saving of the comb strategy varies as f_d and f_q vary. The communication costs of pure-push and pure-pull strategies are 132% and 149% of that of the ideal case for the total 3000 time-slot period on average. In all time periods, the adaptive scheme achieves over 99% success rate with negligible additional communication cost compared to the ideal case.

In the simulation, we fix the query node to be $(0, 0)$. However, the exact strategy applies to the case where the query node is moving in the network, such as a soldier is moving within the deployed sensor network. The adaptive comb strategy also applies to the case where different nodes generate query requests. In such case, similar queries should be combined. In particular, a possible query node should estimate the value of f_q , which is the query frequency for all nodes, by observing the queries of its own and others. A query node can also use the current value of s in the network to estimate the inter-spike interval of its own query.

6. RELATED WORK

Sensor network applications often need support for periodic or sporadic monitoring of the environmental state. A simplest method for obtaining state from a large scale sensor network is to flood the query for the named data to the whole network and get responses from relevant sources. However, flooding in large-scale wireless networks is a very expensive operation and in general needs $O(N)$ radio transmissions, where N is the number of nodes in the network.

Many strategies have recently been proposed to reduce the cost of discovery and query in large-scale wireless networks. Most relevant to this work includes three different types of approaches.

Time	1-500	501-1000	1001-1500	1501-2000	2001-2500	2501-3000
Push	179%	131%	118%	169%	160%	114%
Pull	156%	138%	157%	131%	139%	153%

Table 1: Compare the comb strategy (ideal case) with pure-pull and pure-push strategies.

The first approach is to improve flooding efficiency by reducing the number of potentially redundant forwarding in the flooding process using neighborhood topological information [15, 16] or via probabilistic re-transmissions [17]. The goal of this approach can be viewed as to reduce the constant implicit in $O(N)$. In contrast, the comb-needle scheme achieves $O(\sqrt{N})$ in the best case by balancing push and pull. In another related work [18], Braginsky and Estrin proposed the interesting idea of rumor routing in which the query and the event information are propagated in the network via random walk with time-to-live parameters. A query “discovers” a relevant event when it meet the path traversed by the respective event agent. The comb-needle scheme could be regarded as a structured way to achieve the goal of rumor routing, in situations where node location information is available. Furthermore, very recently Shakkottai [19] showed that the probability that a query via rumor routing is unsuccessful decays as $t^{-5k/8}$, where the t is the average “propagation depth” of the query and events, and k is the number of rumor routing agents for each query and event.

The second approach in the literature is to reduce discovery/query cost by taking into account application semantics [20, 21]. For instance, discovery queries like “is there a tank in the field?” is potentially resolvable by only traversing part of the network in many cases. As a result, for this kind of queries, expanding-ring search or ACQUIRE [9] have been demonstrated to be more efficient than flooding. ACQUIRE considers the query as an active entity that is forwarded through the network either randomly or in some directed manner in search of a solution: nodes on the path that handle the active query use information from all nodes within d -hops in order to partially resolve the query. When the active query is fully resolved, the query forwarding process stops and a complete response is sent back to the querying node. This type of scheme limits the scope of necessary flooding by recognizing the specific nature of the query. Another outstanding example in this category is the Information-Driven Query Routing (IDSQ) scheme [22]. For instance, when tracking mobile entities, sensor nodes can keep a certain memory, so when a query like “where are the tanks?” comes in, the underlying system can quickly route the query to the nodes with most current information by following the traces of information exist in the network. This again eliminates the needs for flooding in many cases. In a recent work, [20] investigated the relative benefit of push and pull diffusion for queries of continuous type and arrived the conclusion of a need for a families of protocols catering to different application requirements. This paper investigates and demonstrates the benefit of taking into account both the application and the environmental characteristics in designing efficient data discovery and dissemination protocols.

The third type of approach is to reduce the cost of search by using efficient distributed indexing schemes. For instance, [7] described a distributed indexing scheme called Semantic

Routing Tree (SRT). SRT only supports query from a fixed node and constructs a route tree based on historical data, so only applies well to slow changing sensor readings. In [23], the authors proposed a distributed index for multi-dimensional data (DIM) which allows query to be issued from any node. DIM achieves an average event insertion cost of $O(\sqrt{N})$ and average query cost of $O(\sqrt{N})$ in most cases. It is also mentioned in [23] that DIM out-performs flooding as long as the ratio of the number of insertions to the number of queries is less than \sqrt{N} . Earlier exploration in this direction includes GHT [24] and DIMENSIONS [25]. The push component in our comb-needle scheme is similar in spirit with DIM in the sense that the data generated at a node are hashed/pushed to different locations. However, our scheme is simpler and achieves similar cost savings. Furthermore, we investigate how the query support structure perform when links are unreliable, an important issue which is largely neglected in earlier works.

7. CONCLUSION

In this paper we proposed the comb-needle model, a simple yet efficient data discovery scheme for supporting queries in large-scale sensor networks. We also used the comb-needle model as a substrate for study the benefit of balancing push and pull in data gathering and dissemination in large-scale wireless networks. The comb-needle scheme (including the reverse one) covers a spectrum of the push and pull strategies, with the pure push-based and pure pull-based schemes in two extremes. We have demonstrated that in general, the comb-needle scheme performs better than both pure push-based and pure pull-based schemes. Furthermore, we raised and studied the issue of query reliability and query coverage when the underlying communication channel is not reliable. Finally, we proposed and studied an adaptive comb-needle strategy for cases where the utilization patterns and environmental activity frequency are time-varying. The contribution is highlighted not only by the theoretical analysis of the problems studied, but also by the confirmation of the theoretical observations via simulations. We note that the proposed scheme can be used in the hierarchical structure as well. Further, data aggregation and compression can be integrated into the comb-needle strategy to further reduce the communication cost. These are all interesting future work to explore. We also look forward to further validating this work in actual large-scale sensor network test-beds and explore new issues that might arise in the real world.

8. ACKNOWLEDGMENT

This work is funded in part by Defense Advanced Research Project Agency contract # F33615-01-C-1904. Xin Liu would like to thank Prof. R. Srikant for the initial discussion on the problem. The authors would like to thank the reviewers for their constructive comments.

9. REFERENCES

- [1] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1999, pp. 263–270.
- [2] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, 2000.
- [3] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications.*, Atlanta, GA, September 2002.
- [4] J. Liu, D. Petrovic, and F. Zhao, "Multi-step information-directed sensor querying in distributed sensor networks," in *Proceedings of the International Conference in Acoustics, Speech and Signal Processing (ICASSP)*, 2003.
- [5] D. Li, K. Wong, Y. Hu, and A. Sayeed, "Detection, classification and tracking of targets in distributed sensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, March 2002.
- [6] B. Blum, P. Nagaraddi, A. Wood, T. Abdelzaher, S. Son, and J. Stankovic, "An entity maintenance and connection service for sensor networks," The First International Conference on Mobile Systems, Applications, and Services (MobiSys), San Francisco, CA, May 2003.
- [7] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "The design of an acquisitional query processor for sensor networks," in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data.* ACM Press, 2003, pp. 491–502.
- [8] Q. Huang, C. Lu, and G.-C. Roman, "Spatiotemporal multicast in sensor networks," in *Proceedings of the First ACM Conference on Embedded Networks Sensor Systems (SenSys)*, Los Angeles, California, Nov. 2003.
- [9] N. Sadagopan, B. Krishnamachari, and A. Helmy, "Active query forwarding in sensor networks," *Elsevier Journal of Ad Hoc Networks*, 2003.
- [10] G. Simon, "Probabilistic wireless network simulator," 2003, <http://www.isis.vanderbilt.edu/projects/nest/prowler/>.
- [11] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proceedings of the First International Conference on Embedded Networked Sensor Systems*, Los Angeles, Nov 2003, pp. 1–13.
- [12] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *ACM Sensys 2003*, 2003.
- [13] S. M. Ross, *Introduction to probability models*, 7th ed. Harcourt Academic Press, 2000.
- [14] F. Ye, G. Zhong, S. Lu, and L. Zhang, "Gradient broadcast: A robust data delivery protocol for large scale sensor networks," *ACM Wireless Networks (WINET)*, vol. 11, no. 2, March 2005.
- [15] W. Peng and X. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," in *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2000.
- [16] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," INRIA, Tech. Rep. Research Report RR-3898, Feb. 2000. [Online]. Available: citeseer.nj.nec.com/qayyum00multipoint.html
- [17] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, August 1999, pp. 152–162.
- [18] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," in *Proceedings of the First Workshop on Sensor Networks and Applications (WSNA)*, Atlanta, GA, October 2002.
- [19] S. Shakkottai, "Asymptotics of query strategies over a sensor network," in *Proceedings of the IEEE INFOCOM*, Hongkong, China, March 2004.
- [20] J. Heidemann, F. Silva, and D. Estrin, "Matching data dissemination algorithms to application requirements," in *Proceedings of the first international conference on Embedded networked sensor systems.* ACM Press, 2003, pp. 218–229.
- [21] B. Krishnamachari and J. Heidemann, "Application-specific modelling of information routing in sensor networks," in *Proceedings of the IEEE International on Performance, Computing, and Communications Conference.* Phoenix, Arizona, USA: IEEE, April 2004, pp. 717–722.
- [22] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *International Journal of High Performance Computing Applications*, vol. 16, no. 3, 2002.
- [23] X. Li, Y. J. Kim, R. Govindan, and W. Hong, "Multi-dimensional range queries in sensor networks," in *Proceedings of the first international conference on Embedded networked sensor systems.* ACM Press, 2003, pp. 63–75.
- [24] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu, "Data-centric storage in sensornets with GHT, a geographic hash table," *Mob. Netw. Appl.*, vol. 8, no. 4, pp. 427–442, 2003.
- [25] D. Ganesan, D. Estrin, and J. Heidemann, "DIMENSIONS: Why do we need a new data handling architecture for sensor networks?" in *Proceedings of the First Workshop on Hot Topics In Networks (HotNets-I)*, Princeton, NJ, October 2002.