GLANCE: A Lightweight Querying Service for Wireless Sensor Networks

Murat Demirbas
Computer Science & Engineering
University at Buffalo, SUNY
Buffalo, NY

Anish Arora Vinod Kulathumani
Computer Science & Engineering
The Ohio State University
Columbus, OH

Abstract

By exploiting basic geometry concepts, we present a lightweight, distance-sensitive, and tunable querying service, *Glance*, for dense wireless sensor networks. Glance ensures that a "query" operation invoked within d hops of an event intercepts the event's "publish" operation within $d * s$ hops, where s is a "stretch-factor" tunable by the user. A significant feature of our service is that it can be implemented easily without any localization information.

1 Introduction

A major application area for wireless sensor networks (WSNs) is environmental monitoring [1, 2, 13, 17, 18]. The vision grandeur for these applications is to scatter thousands of wireless sensor nodes across an area of interest upon which the nodes self-organize into a network and enable monitoring and querying of events in the area. An example application is a disaster evacuation scenario where the rescue workers query the network to learn about fire or chemical threats in the area.

There are two main modes of operation in most WSN monitoring applications. The first mode is "centralized monitoring and logging". For monitoring and logging purposes it is important to gather information about events in the network. This can be easily satisfied by enforcing events to advertise metadata to a basestation that could forward the data to a monitoring and control center. In our disaster evacuation scenario, the control and command center needs to get metadata about events for logistical purposes, such as deciding how many rescue workers to send to each region and coordinating the rescue efforts. These data are also valuable for keeping logs and statistics of events.

The second mode of operation is "in-network querying" or "location-dependent querying". In the context of the evacuation scenario, the rescue workers in each region would need to query the network for nearby events, such as fire/chemical threats, and vital statistics from victims. It is inefficient and unscalable, for most cases, to force the queriers to learn about events only from the basestation, since it would compel a querier that is very close to an event to communicate all the way back to a basestation to learn about that event. The inefficiency of the scenario is amplified if the querier needs to get a stream of data from the event. Using long routes for forwarding data not only increases the latency but also depletes the battery power of the relaying nodes in the network quickly. Using the basestation as a broker also leads to a communication bottleneck for the network. For these reasons it is important to be able to discover short (local) paths from queriers to nearby events.

In this paper we describe a querying service that addresses these two modes of operation in WSN monitoring applications. Such a querying service involves two operations. *Publish operation* is invoked by the nodes that detect an event, and it aims to inform any potential nodes (including the basestation) interested in the event. *Query operation* can be invoked by any node in the network, and it aims to inform the querying node about a matching event and construct a path from the querying node to the event.

Contributions of the paper. We show that, for geometric dense networks, it is possible to devise a simple, lightweight, and efficient querying solution by exploiting basic geometry concepts. Our querying service *Glance* is distance-sensitive and tunable: It ensures that a query operation invoked within d hops of an event intercepts the event's publish information within $d * s$ hops, where s is a "stretch-factor" tunable by the user. Another significant feature of our service is that it can be implemented without localization information or any hierarchical partitioning of the network. Based on our simulation results, we also suggest a simple way to implement Glance in WSNs without a significant degradation in distance-

sensitivity guarantees.

Overview of Glance. Let C be a distinguished basestation node in the center of the network. Let d_q be the distance (in terms of number of hops) between a querying node q and C, d_e the distance between an event e and C, and finally d the distance between q and e. For the cost of query operations there are two possible cases with respect to the angle z formed by locations of q, C, and e. Figure 1 illustrates these two cases.

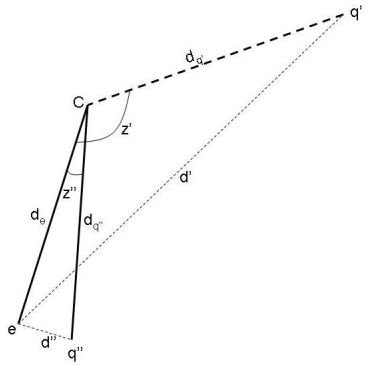

Figure 1: Two cases with respect to z

1. z is larger than a threshold: A large z implies that d is large relative to d_q and d_e. Thus, it is acceptable for the query to go to C to learn about the event, since the stretch-factor s can still be satisfied this way. For example, in Figure 1 $d_{q'} \leq d' * s$.

2. z is smaller than the threshold: A small z implies that d is small relative to d_q and d_e. Thus, it is unacceptable for the query to go to C, since this violates the stretch-factor property. For example, $d_{q''} > d'' * s$.

Our publish operation in Glance seeks to optimize for the above two cases (see Figure 2):

- The publish operation advertises the event on a cone boundary for some distance. The exploration angle z is calculated based on the the stretch-factor s as $arcsin(1/s)$ (as described in Section 3.2). This cone-advertisement accounts for potential queriers q with a small angle z'', whose $d_q > d'' * s$.

- After the cone-advertisement, it becomes cost-effective to go directly to C since it still satisfies the stretch-factor for potential queriers with a large angle z'.

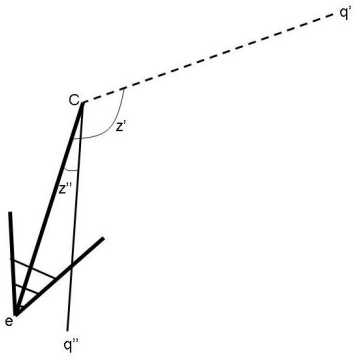

Figure 2: Publish operation for s=2

The query operation is simply a *glance* to the center; it creates a straight path from the querying node to C. Once a publish path for a matching event is found, the query operation stops forwarding the query any further and informs the querying node by reversing the path it traversed. The querying node can then use this path to learn more about the event.

Outline of the paper. Next, we discuss related work. In Section 3 we present Glance and we discuss how Glance can be implemented without the need for localization in Section 4. In Section 5, we present simulation results. We conclude the paper with a discussion of future work in Section 6.

2 Related Work

There has been several work in querying (also known as information brokerage or data-centric routing) in WSNs. One of the earlier works, Geographic Hash Tables (GHT) [15], stores and retrieves information by using a geographic hash function on the type of the information. GHT is not distance-sensitive: it can hash the event information far away from the nearby event-query pair and thus violates the stretch-factor. GLS [11] uses a hierarchical clustering of the network by using a quadtree structure. GLS is more local than GHT, however it still cannot achieve distance-sensitivity due to the multi-level partitioning problem: In a hierarchical partitioning it is possible that a query-event pair nearby in the network might be arbitrarily far away in the hierarchy due to multi-level partitioning between them. Hence, GLS cannot guarantee distance-sensitivity for all pairs. Stalk [4] uses a hierarchical partitioning also, but to account for the multi-level partitioning effects a querying node performs lateral searches to neighboring clusterheads

at every level to reach the event information in a distance-sensitive manner. Recently, Funke et. al. [7] also considered distance-sensitive information brokerage problem and achieve distance-sensitivity by using a lateral exploration technique similar to that used in Stalk. They adapt the discrete centric hierarchy method [9] to construct the hierarchical partitioning without the need for localization information.

One of the drawbacks of the hierarchical querying services is the resultant backward links that lead to undesirable long paths: Even though an event information is eventually forwarded towards the top level "center node" in the hierarchy, the information may travel in the opposite direction to the center while being forwarded to the next level clusterheads in the hierarchy. Another drawback of hierarchical querying services is the cost of contacting neighboring clusterheads at increasingly higher levels of the hierarchy. These lateral explorations are done over increasingly longer links for higher levels, and it is cumbersome to maintain these extra communication infrastructure among clusterheads. Glance overcomes these drawbacks by fully exploiting the geometry of the network.

Rumor routing [3] provides a tunable data-centric querying mechanism for networks without localization information. In this approach, an event employs a set of agents to do a random walk of the network creating paths that lead to the event. Querying node also sends out query agents which randomly traverse the network. Whenever a query agent encounters a path set up by an event agent with a matching interest a route is established between the query and the event. Gossiping among event agents is employed for improving the odds of the meeting of query and event agents. The scheme is tunable in that for guaranteeing higher reliability it is possible to increase the number of agents sent from each event and query.

Glance improves over rumor routing by providing a more structured approach to publishing and querying. Since both the publish and query operations now target a common center the possibility of their meeting increases greatly compared to a random walk strategy. In addition, using the stretch-factor idea and the cone-advertisement the meeting distance of the publish and query are optimized. Glance also avoids wasting energy by not advertising the event in the regions where meeting at C is already an acceptable solution for the query and event.

Combs and Needles paper [12] also investigates efficient strategies for supporting on-demand information dissemination and querying in WSNs. The paper shows that the optimal routing structure depends on the rate of queries and events in the network and investigates combining the advantages of both pull and push strategies in order to adapt to the relative rates of query and events. In our paper, we assume that number of queries is more than number of events, and thus, the cone-advertisement is provided by the publish operation and not by the query. Had the rate of events been more than rate of queries, we would have assigned a cone-search to the query operation instead of burdening the publish operation with a cone-advertisement. Note that Glance also combines the advantages of both push and pull strategies in that both publish and query do some work to optimize the overall outcome. In contrast to Combs&Needles, Glance exploits a center node to focus the dissemination of information and forwarding of the queries, and achieve efficient, distance-sensitive, and tunable (with respect to the stretch factor) querying. Also in contrast to Combs&Needles Glance does not require an underlying localization service and is applicable to a wider-range of WSNs.

3 Glance Algorithm

3.1 Model

We consider a multihop wireless network where communication is bidirectional over a singlehop. We use $dist(j,k)$ to denote the hop distance between two nodes j and k. We restrict ourselves to a geometric graph model, where the triangle inequality holds: for any node j,k,l, $dist(j,k) + dist(k,l) \geq dist(j,l)$. In this paper, for simplicity, the sensor nodes lie in a 2-D coordinate space, however, our results also apply to higher dimensions. We assume a dense, connected network. We define the cost of a communication over d hops as $O(d)$.

We assume a distinguished basestation node C at the center of the network. We denote the distance between a querying node q and C as d_q, and event e and C as d_e. $z_{q,e}$ denotes the angle formed by location of q, center C, and the location of e.

We use a calculational proof notation [5] where a proof of $K \equiv M$ can be expressed as:

$$\begin{array}{ll} & K \\ \equiv & \{ \text{ reason why } K \equiv L \} \\ & L \\ \equiv & \{ \text{ reason why } L \equiv M \} \\ & M \end{array}$$

3.2 Details of the Publish Operation

Here we explain the cone-advertisement needed for the publish operation in detail, and discuss how the publish operation ensures distance-sensitivity for a given stretch-factor, s. For the discussion in this section we pretend that localization is available (every node knows its coordinates in the plane), later in Section 4 we explain how the need for localization information is avoided.

Areas where stretch-factor is readily satisfied. We mentioned in the Introduction that there are two possible cases for the cost of a query operation invoked at a node q, for an event e, with respect to the angle $z_{q,e}$. To account for the case where $z_{q,e}$ is less than the threshold angle x, the publish operation needs to advertise on a cone boundary. For $z_{q,e}$ greater than x no action is required as the stretch-factor is readily satisfied even when q contacts C directly, incurring a d_q cost. In order to be able to determine the boundaries of the advertisement cone, we first need to calculate the threshold angle x for a given stretch-factor s. Here we show how we calculate x by determining the areas for which stretch-factor is readily satisfied.

As a simple example, let's take $s = 1$. We calculate the region where the stretch-factor is readily satisfied by taking successively larger circles centered at e and C and intersecting them. Figure 3 illustrates this method. There $A2$ is the region for which the stretch-factor is readily satisfied by contacting C directly (since a querying node in $A2$ is already closer to C than it is to e), and $A1$ is where the stretch-factor may be violated.

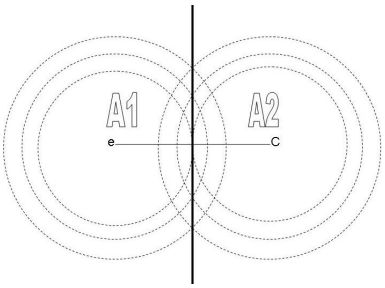

Figure 3: Area where s=1 is readily satisfied

For $s > 1$, the same method is used for calculating the areas where stretch-factor is readily satisfied: we let a circle with radius r centered at e intersect with a circle with radius $s * r$ centered at C. Figure 4 shows an example for $s = 2$. In the figure the stretch-factor is readily satisfied for areas $A2, A3$, and $A4$. For area $A1$ stretch-factor may be violated, and cone-advertisement should account for the querying nodes in this region. Note that $A1$ is bounded not only in the direction of C but also in the opposite direction. This is because for $r \geq d_e$, all the circles centered at e are subsumed by circles with radius $2 * r$ centered at C.

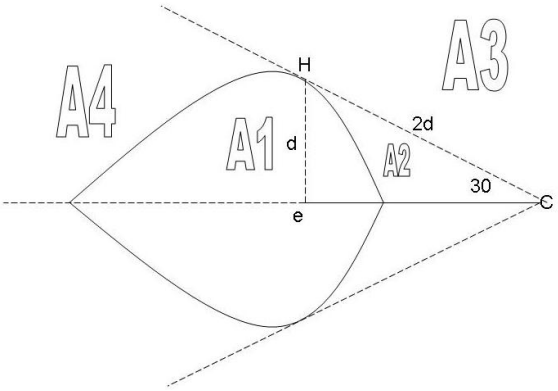

Figure 4: Areas where s=2 is readily satisfied

Also note that for intersection point H, where $\widehat{HeC}$ forms a right angle, the ratio between the radius of the circle centered at e and that centered at C is $s = 2$ as expected. This point determines the maximum angle between any intersection point and e with respect to C. Therefore, the threshold angle for $s = 2$ is calculated as 30° from the HCe triangle. In general x is calculated as $arcsin(1/s)$, since $x = arcsin(|eH|/|CH|)$. For $s = 1$ there is no feasible solution since $x = 90°$. For $s = 2$, $x = 30°$, and $s = 4$, $x = 14.5°$.

The algorithm for query. The query operation is simply a *glance* to the center; it creates a straight path from the querying node to C. Once a publish path for a matching event is found, the query operation stops forwarding the query any further and informs the querying node by reversing the path it traversed. The querying node can then use this established path to learn more about the event.

The algorithm for publish. Publish operation has two phases, advertisement and informing the center, as depicted in Figure 8. We assume that $|EC|$ is known at E the location of e.

Advertisement. In the first phase, advertisement is performed on a cone boundary to intercept the query operations of nearby queriers for which going to the center strategy is unacceptable. The angle

for the advertisement cone is selected to be equal to x ($\arcsin(1/s)$). Figure 5 shows the cone-advertisement for the publish operation for $s = 2$. The lateral advertisements inside the cone are to account for the querying nodes in area $A1$ that also fall within the cone boundaries. Consider a query within unit distance of e, by drawing the first lateral link at distance s from e, the stretch-factor is satisfied for the query, and also other queries in Case 3 within distance s of e. The second lateral link, drawn at distance s^2, handles queries within distance s–s^2, and so on. The final lateral link is at the boundary of A1, which is $d_e/s + 1$ distance away from e.

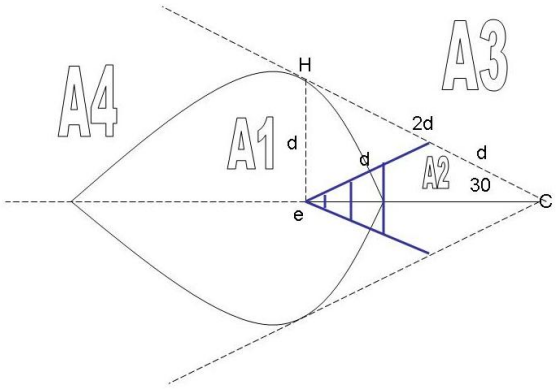

Figure 5: Local advertisement for $s = 2$

Lemma 1. A query operation invoked in A1 within d hops of an event intercepts the event's publish information within $d * s$ hops.

Proof: There are three cases.

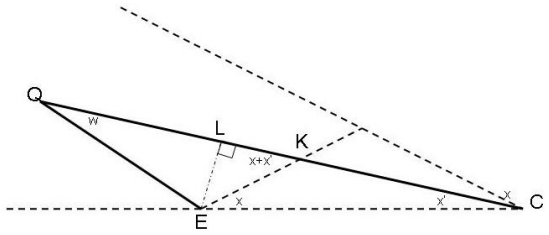

Figure 6: Advertisement, case 1

Case 1 (see Figure 6): We prove that in Figure 6, $|QK| < s|QE|$.

$|QK| < s|QE|$

$\equiv$ { $|QK| = |QL| + |LK|$ and $|LK| = |LE|\cot(x + x')$}
$|QL| + |LE|\cot(x + x') < s|QE|$

$\equiv$ { $|QL| = |QE|\cos(w)$ and $|LE| = |QE|\sin(w)$}
$|QE|\cos(w) + |QE|\sin(w)\cot(x + x') < s|QE|$

$\equiv$ { $\sin(x) = 1/s$ (multiply both sides with $\sin(x)$) also eliminate $|QE|$ }
$\sin(x)\cos(w) + \sin(x)\sin(w)\cot(x + x') < 1$

$\equiv$ { $\cot(\alpha) = \cos(\alpha)/\sin(\alpha)$ }
$\sin(x)\cos(w) + \sin(x)\sin(w)\cos(x + x')/\sin(x + x')$
< 1

$\equiv$ { Definition of $\cos(\alpha + \beta)$ and $\sin(\alpha + \beta)$}
$\sin(x)\cos(w) + \sin(x)\sin(w)(\cos(x)\cos(x')$
$-\sin(x)\sin(x'))/(\sin(x)\cos(x') + \cos(x)\sin(x'))$
< 1

$\equiv$ { Arithmetic }
$\sin^2(x)\cos(w)\cos(x') + \sin(x)\cos(x)\cos(w)\sin(x')$
$+\sin(x)\cos(x)\sin(w)\cos(x') - \sin^2(x)\sin(w)\sin(x')$
$< \sin(x + x')$

$\equiv$ { Arithmetic }
$\sin^2(x)(\cos(w)\cos(x') - \sin(w)\sin(x'))$
$+\sin(x)\cos(x)(\sin(w)\cos(x') + \cos(w)\sin(x'))$
$< \sin(x + x')$

$\equiv$ { Definition of $\cos(\alpha + \beta)$ and $\sin(\alpha + \beta)$}
$\sin^2(x).\cos(w + x') + \sin(x)\cos(x)\cos(w + x')$
$< \sin(x + x')$

$\equiv$ { Arithmetic, definition of $\sin(\alpha + \beta)$}
$\sin(x)\sin(x + w + x') < \sin(x + x')$

$\equiv$ { Arithmetic }
$\sin(x + w + x') < \sin(x + x')/\sin(x)$

Note that $0 \le x' \le x \le 90°$, also $x + x' + w < 180°$ as they are in a triangle. There are two cases.

Case A ($x+x' \le 90°$): Then, $\sin(x+x')/\sin(x) > 1$ is satisfied due to property of *sine* for angles between $0° - 90°$. Since $\sin(\alpha) \le 1$, for any α, we have $\sin(x + w + x') < \sin(x + x')/\sin(x)$.

Case B ($90° \le x + x' < 180°$): Note that, $\sin(x + x')/\sin(x) > \sin(x + x')$, since $\sin(x) \le 1$, for any x. Also, $\sin(x + x') > \sin(x + w + x')$, since $90° < x+w+x' < 180°$ and as angle increases sine decreases in that interval.

Case 2 (see Figure 7): We prove that in Figure 7, $|QK| < s|QE|$.

$|QK| = |LK| - |QL|$. Note that $|QL| = |QE|\cos(180 - w) = -\cos(w)|QE|$.

$|QK| < s|QE|$

$\equiv$ { $|QK| = |LK| - |QL|$}
$|LK| - |QL| < s|QE|$

$\equiv$ { $|QL| = |QE|\cos(180 - w) = -\cos(w)|QE|$}
$|LK| + |QE|\cos(w) < s|QE|$

$\equiv$ { $|LK| = |LE|\cot(x + x')$ and

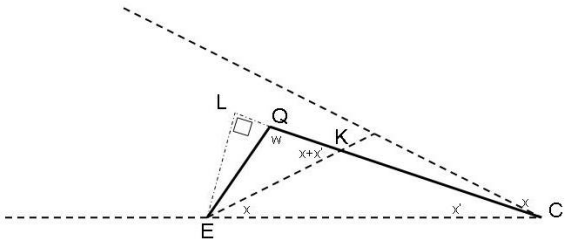

Figure 7: Advertisement, case 2

$$|LE| = |QE|\sin(180 - w) = |QE|\sin(w)\}$$
$$|QE|\sin(w)\cot(x + x') + |QE|\cos(w) < s|QE|$$
$$\equiv \quad \{ \text{ Same inequality as in Case 1}\}$$
$$\sin(x + w + x') < \sin(x + x')/\sin(x)$$

Since $0 \leq x' \leq x \leq 90$ both subcases in Case 1 apply without modification.

Case 3: In this case the querying node is in area A1 in Figure 4, and falls within the cone. The lateral links in the cone satisfy the stretch-factor for these queries as discussed above. $\diamond$

Informing the center. The second phase starts after the first phase completes cone-advertisement. In this phase the publish operation goes straight to C. Figure 8 summarizes both phases.

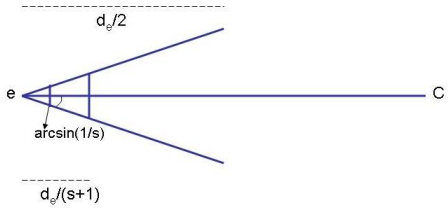

Figure 8: Publish operation

Theorem 1. A query operation invoked within d hops of an event intercepts the event's publish information within $d * s$ hops.

Proof: There are two cases. If querying node is in $A1$, due to Lemma 1, the stretch-factor is satisfied. If querying node is in $A2$, $A3$, or $A4$ then due to geometry, $|QC| < |QE| * s$, and stretch-factor is readily satisfied. $\diamond$

3.3 Cost of publish and query

From Figure 8, we calculate the cost of publish operation as follows. The publish operation goes all the way to C as a straight line, which induces d_e cost. In addition, the two cone boundaries induce $2 * (d_e/2) * \cos(x)$ cost, where $x = \arcsin(1/s)$. The cost for the lateral explorations inside the cone for the $d_e/(s+1)$ distance is calculated as $\sum_{i=0}^{\log_s(d_e/(s+1))} s^i * \tan(x)$. Thus the overall cost comes up to $d_e + d_e * \cos(x) + \sum_{i=0}^{\log_s(d_e/(s+1))} s^i * \tan(x)$. The cost for publish depends on the distance d_e of the event from a basestation C. Note that, $d_e * \cos(x)$ is always less than d_e, but for x very close to 90° (i.e., for s very close to 1) the last term can get high as $\tan(x)$ gets large. For $s = 1.44$, the cost of publish is less than $3.5 * d_e$, and for $s = 2$ the cost is less than $2.5 * d_e$.

As proved in Theorem 1, the worst case cost for query is $d * s$.

4 Glance Without Localization

Since Glance requires only an approximation for the direction to the center, it is possible to avoid localization information as follows. After the deployment of the sensor network, the basestation node C starts a one-time flood that annotates each node in the network with its hopcount from C and creates a spanning tree rooted at C. To send the query or publish as a straight line, it is enough to route the message to the parent node along a branch in this tree.

The problems with a spanning tree construction in WSN are discussed and analyzed extensively in [8]. Experiments with flooding protocols display a large number of anomalous situations. *Stragglers*, nodes that miss transmissions even though they would be expected to receive a packet with high probability, leads to *backward links* in the spanning tree, where recipient of the flood is closer to the base-station than the transmitter. Also the opportunistic, earliest-first parent selection when constructing the tree results in highly-clustered trees, where most nodes in the tree end up having few descendants while a significant few have many children. These anomalies are due to collisions when multiple nodes transmit simultaneously and due to non-deterministic and non-isotropic nature of radio transmission. Our first experiments with constructing the tree were also susceptible to these stragglers and backward links even though we used a best-effort minimum spanning tree (MST) protocol and asserted that a node should adapt a neighboring node with smallest hopcount as its parent. Figure 9 shows one such example.

To design a lightweight MST protocol while avoiding the problems due to stragglers, backward links,

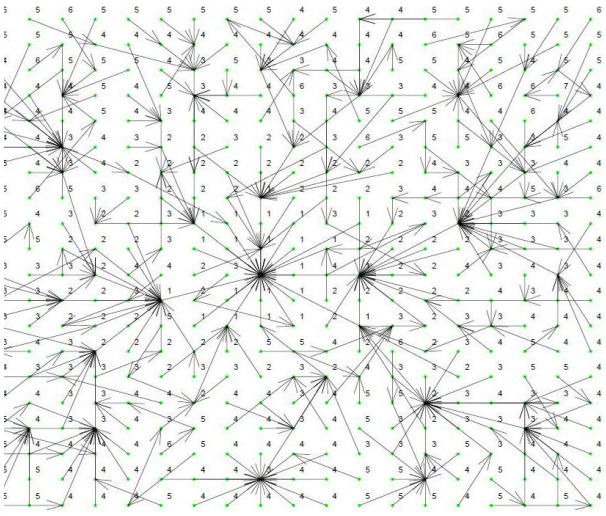

Figure 9: Spanning tree rooted at C

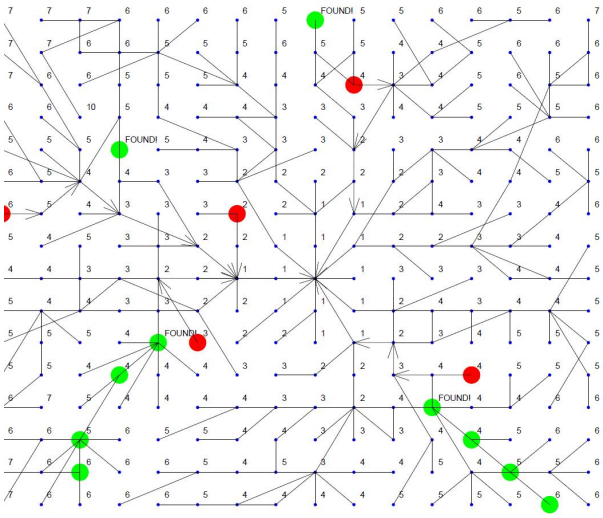

Figure 10: Optimized MST rooted at C

and highly clustered nodes, we exploited snooping and randomization. Due to stragglers and backward links, it is possible to have a node with a hopcount of 5 just next to a bunch of nodes with hopcount 1. Our repairing strategy is a reactive one. When a node j with hopcount x hears a transmission with hopcount greater than x+2 this is an indicator of a straggler in the vicinity. j may then choose to retransmit a message to correct the straggler (and hence the backward link) based on a randomized outcome (to avoid multiple nodes to respond at once). In order to avoid highly clustered trees, a node j with hopcount x that receives a broadcast from k with hopcount x-1, may randomly choose to select k as its parent to achieve a more balanced tree. Note that had k's broadcast be with hopcount less than $x-1$, j had to adapt k as its parent since we are aiming to construct an MST.

Figure 10 shows an example of our optimized tree construction. In the resulting tree there is no backward links or highly-clustered nodes. The network is 16x16 and the diameter of the tree is 2*7. There are 5 events (represented as red [or dark gray] spots) and 4 queries (green [or light gray] spots) in the network. The events construct a trail of arrows to C, the query also follows parent pointers to C and if information about the event is intercepted, the query stops.

In the absence of localization it is infeasible to draw cone borders as in Figure 5. We replace that with occasional lateral exploration inside the cone. For this phase nodes with same hopcount are visited at predefined distances from the event. The resulting advertisement structure is in Figure 11. Of course this type of lateral exploration is only an approximation to drawing the cone borders in the absence of localization. In the next section, based on our simulation results, we suggest a simpler way to implement Glance in WSNs without a significant degradation in distance-sensitivity guarantees.

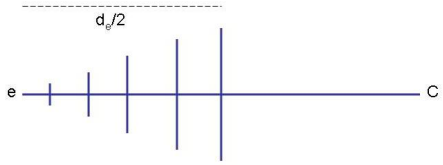

Figure 11: Publish operation without localization

5 Simulation Results

For our simulations, we use the Prowler wireless sensor network simulation tool [16]. Prowler realistically simulates the radio transmission, propagation, reception delays of Mica2 motes [14], including message collisions due to hidden node and interference problems.

Our experiments are performed on a grid topology where the distance between immediate neighbors in

the grid is taken as one unit. In our experiments we fixed the radio transmission radius as approximately two units. In our setup, the events arrive in the network first and form a trail of arrows to C using the MST rooted at C. Queries arrive later, and are forwarded to the parent node at each hop along a branch in the MST. If a node that is in an event trail receives a message regarding the forwarding of a query, it replies with the event information. When this information is received at the node where the query currently resides, the event is said to be "found", and the query operation is stopped. In our simulations we assume that all events are of interest to each query. Our experiments are performed with 3 events and 6 queries unless noted otherwise. Each data point in our graphs is an average over at least 10 runs of the experiment.

We provide two implementations for our querying service. The first implementation, Glance, avoids the cone-advertisement for a publish operation completely, and constructs the event trail by following parent pointers along a branch in the MST. The second implementation, GlanceP, approximates the cone-advertisement process by probing for lateral neighbors while constructing the event trail. The probing is done twice, at $d_e/4$ and $d_e/2$ hop-distance from e along the trail, by broadcasting a message and recruiting the singlehop neighbors that receive the message to store the event information. These recruited neighbors can respond with the event information later when they hear forwarding of a query. We do not provide a complete implementation for the cone-advertisement as described in Section 4 due to the extra complexity involved in programming. However, as our simulation results show below, we find that due to the inherent aggregation in an MST the cone-advertisement can be avoided in the implementation of publish operation without a significant degradation in the distance-sensitivity guarantees.

Comparison with Rumor routing. Figure 12 shows a sample run from the rumor routing algorithm with 3 events and 6 queries. Each event agent performs a random walk for 2*N hops in a network of NxN nodes. The query agents arrive later and perform random walk until they intercept an event information. As it is the case for Glance, a node in an event path replies with the event information if a query agent is overheard at the node.

Figure 13 shows comparisons of query hops for Rumor, Glance, and GlanceP. The average number of hops a query travels before finding any information

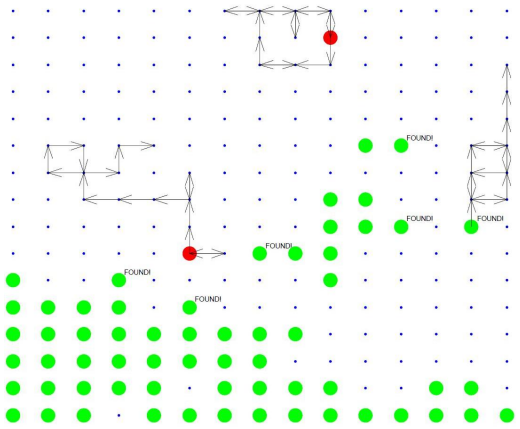

Figure 12: Rumor routing

about an event is high for Rumor routing and increases rapidly as the size of the network is increased. On the other hand, average query hops remain low for Glance (2.65 for 16x16, 5.78 for 30x30, and 10.0 for 60x60), and GlanceP (2.33 for 16x16, 4.53 for 30x30, and 9.86 for 60x60). The scalability of the average query hops for Glance and GlanceP are remarkable as the network size increases. Ideally the query hops would depend only on the distance between query and events. However, since we choose event locations from a uniform distribution over the network, the average distance between query-event locations tend to increase for larger networks, and as such query-hops is indirectly linked to the network size. Later in Figure 15 we show that the stretch-factor is independent of network size.

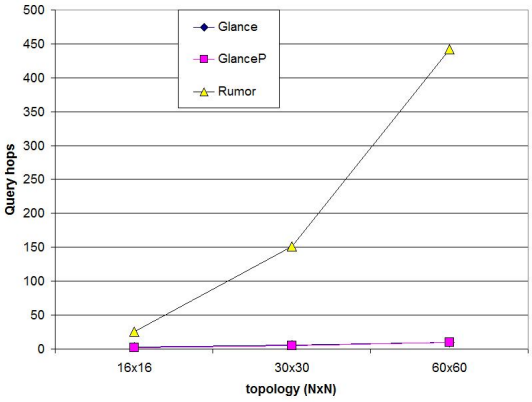

Figure 13: Query costs

Figure 14 shows comparisons of average publish hops for Rumor, Glance, and GlanceP. Since we had fixed the number of hops for event agents as 2N for a grid of NxN, the publish hops data for Rumor routing is uninteresting. The publish hops for Glance is equal to d_e, the cost of going to C, and is proportional to the depth of the MST constructed by C. As the graph shows, the depth of MST scales nicely with respect to the network size. The cost for publish for GlanceP includes two extra broadcast to recruit lateral neighbors and is only slightly more than that of Glance.

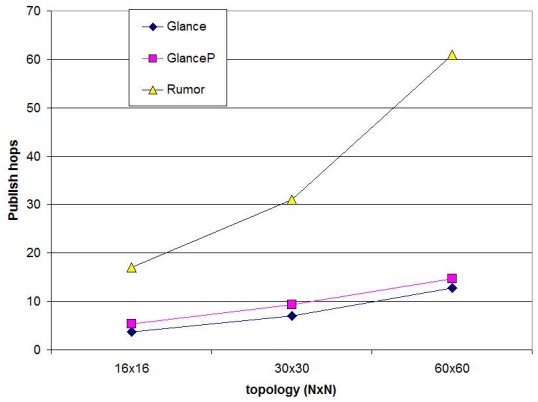

Figure 14: Publish costs

Comparison with Combs&Needles. In contrast to Rumor routing the performance of Combs&Needles is very predictable. For publish operation Combs&Needles builds a a network-wide routing structure that resembles a comb, and the query operation forms a line (i.e., needle structure) that is orthogonal to the publish structure. By arranging the distance between the teeth of the comb structure, Combs&Needles tunes the minimum length for the needle structure to guarantee that query operation intersects the publish operation. By using the average query hops for Glance from Figure 13, we calculate the minimum cost for constructing the comb structure to guarantee the same query hops in Combs&Needles as at least 14 for 16x16, 30 for 30x30, and 90 for 60x60. Thus, the event hops for Combs&Needles are significantly larger than those for Glance in Figure 14. Moreover, Combs&Needles requires an underlying localization service for constructing the comb structure and performing the query operation.

Experiments with Glance. Here we investigate the stretch-factor for Glance in more detail. Figure 15 shows the stretch-factors for Glance and GlanceP with respect to increasing network size. The experiments use one event and one query per run. As the graph demonstrates, the stretch-factors for both Glance and GlanceP are independent of the network size. Both Glance and GlanceP satisfy very low stretch-factors (less than 1.2) on average, and the numbers for Glance and GlanceP are close. The reason Glance performs well for distance-sensitivity is that in contrast to our abstract model where we calculated our theoretical results in Section 3, the simulation experiments with WSN employs an MST rooted at C that performs a significant amount of aggregation. Our abstract model was a purely geometric one: even though two points e and q are close to each other, the lines drawn from e to C and q to C did not intersect before C and resulted in large stretch-factors. On the other hand, in the presence of an MST, the information from two points e, q close to each other are bound to intermingle. The probability that ancestors of e and q are always more than 1-hop away from each other rapidly drops to zero due to the aggregation inherent in MST.

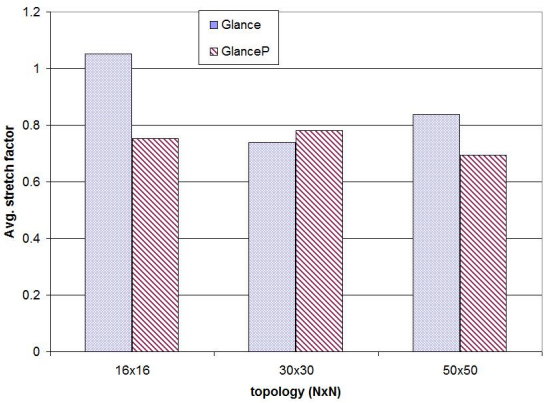

Figure 15: Stretch-factors with increasing network size

Figure 16 shows the stretch-factors for Glance with respect to increasing hop-distance between a query and an event in a 30x30 network. The graph demonstrates that as the distance d between the query and event increases the stretch-factor decreases. Recall that stretch-factor is defined as query-hops divided by d. As d increases the angle $\widehat{eCq}$ becomes large, and hence, the distance between q and C (d_q) becomes relatively small compared to d. For instance,

for "$300 > \widehat{eCq} > 60$" d_q is always less than d and stretch-factor is less than or equal to 1. In addition, for small distances between e and q, aggregation in MST ensures that query-hops remain low. In fact, our results show that the average stretch-factor for $d = 2$ is less than 1.8. The highest stretch-factors we encountered for $d = 2$ are an occasional 3 and 4. Note that for $d = 1$ the query finds the event immediately.

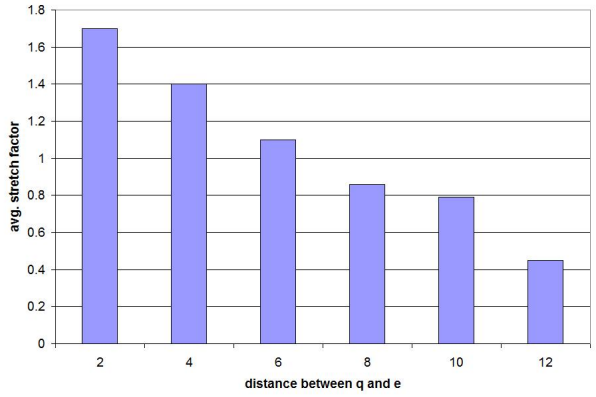

Figure 16: Stretch-factors with increasing distance

Finally, in Figure 17 we move the location of C from the center to one of the corners of the network, and perform experiments with increasing network size. The experiments are performed with 3 events and 6 queries in the network. The results show an almost two-fold increase in event-hops, as moving C to the corner effectively doubled the height of the MST. On the other hand, the query hops did not show a significant increase since the mean distance between query and events are unaffected by moving C to the corner and the query operation is distance-sensitive and practically independent of network size (as Figure 15 demonstrates).

6 Concluding Remarks

In this paper we showed that, for dense geometric sensor networks, it is possible to devise simple and lightweight solutions for querying by exploiting basic geometry concepts. Our querying service, *Glance*, ensures that a subscribe operation invoked within d hops of an event intercepts the event's tracking information within $d*s$ hops, where s is a "stretch-factor" provided by the user. A significant feature of Glance is that it can be implemented without any localization information. We showed through our simulation results that by constructing an MST rooted at the basestation, Glance can circumvent the need for localization. Moreover, our simulation results also showed that due to the aggregation inherent in the MST, the need for cone-advertisement for events can also be avoided without a significant degradation in distance-sensitivity guarantees.

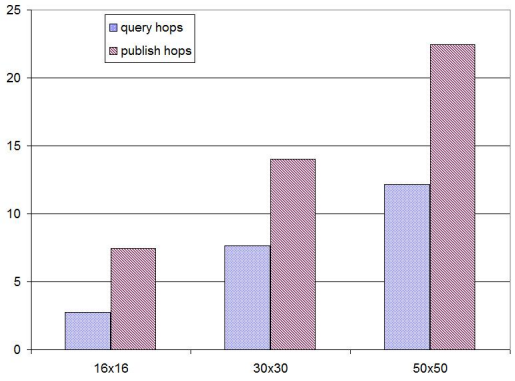

Figure 17: Changing the location of C

Some immediate extensions to our Glance protocol are the use of gossiping and multiple basestations for improving performance. Gossiping can improve the odds of discovery as investigated in the Rumor routing work [3]. For instance, it is probable that during local advertisement a publish operation would obtain information about other publish operations in the vicinity. We can employ gossiping so that later when the publish operation comes across a find interested in any of those earlier publish operations, the publish can inform the find about these. Secondly, by employing multiple basestations it is possible to scale Glance to very large areas easily. By partitioning a very large network into regions, and using a basestation for each region, Glance can be more energy-efficient and scalable to very large areas. This setup is of course more appropriate when queries are interested only at events within the current region.

In future work, we will try to extend the Glance idea for providing more efficient and lightweight solutions to the tracking problem [4, 6, 10, 11], where the event source is mobile. We believe that it is possible to improve Glance to address mobile events if a local solution can be devised for updating published information as an event relocates.

References

[1] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y-R. Choi, T. Herman, S. S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks (Elsevier)*, 46(5):605–634, 2004.

[2] A. Arora and et. al. Exscal: Elements of an extreme scale wireless sensor network. *11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, 2005.

[3] D. Braginsky and D. Estrin. Rumor routing algorthim for sensor networks. In *WSNA*, pages 22–31, 2002.

[4] M. Demirbas, A. Arora, T. Nolte, and N. Lynch. A hierarchy-based fault-local stabilizing algorithm for tracking in sensor networks. *8th International Conference on Principles of Distributed Systems (OPODIS)*, pages 299–315, 2004.

[5] E. W. Dijkstra. *A Discipline of Programming*. Prentice Hall, 1976.

[6] S. Dolev, L. Lahiani, N. Lynch, and T. Nolte. Self-stabilizing mobile node location management and message routing. *Self-Stabilizing Systems*, pages 96–112, 2005.

[7] S. Funke, L. J. Guibas, A. Nguyen, and Y. Wang. Distance-sensitive routing and information brokerage in sensor networks. In *DOCSS*, 2006.

[8] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks. Technical Report CSD-TR 02-0013, UCLA, 2002.

[9] J. Gao, L. J. Guibas, and A. Nguyen. Deformable spanners and applications. In *SCG '04: Proceedings of the twentieth annual symposium on Computational geometry*, pages 190–199, 2004.

[10] H. Lee, J. Welch, and N. Vaidya. Location tracking using quorums in mobile ad hoc networks. *Ad Hoc Networks*, 1(4):371–381, 2003.

[11] J. Li, J. Jannotti, D. S. J. De 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, 2000.

[12] X. Liu, Q. Huang, and Y. Zhang. Combs, needles, haystacks: balancing push and pull for discovery in large-scale sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 122–133, 2004.

[13] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. *ACM Int. Workshop on Wireless Sensor Networks and Applications*, September 2002.

[14] Crossbow technology, Mica2 platform. `www.xbow.com/Products/Wireless_Sensor_Networks.htm`.

[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 '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 78–87, 2002.

[16] G. Simon, P. Volgyesi, M. Maroti, and A. Ledeczi. Simulation-based optimization of communication protocols for large-scale wireless sensor networks. *IEEE Aerospace Conference*, pages 255–267, March 2003.

[17] R. Szewczyk, A. Mainwaring, J. Polastre, and D. Culler. An analysis of a large scale habitat monitoring application. *In Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2004.

[18] G. Tolle, J. Polastre, R. Szewczyk, N. Turner, K. Tu, P. Buonadonna, S. Burgess, D. Gay, W. Hong, T. Dawson, and D. Culler. A macroscope in the redwoods. *In Proceedings of the Third ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2005.