Asymptotics of Query Strategies over a Sensor Network

Sanjay Shakkottai

Abstract— We consider the problem of a user querying for information over a sensor network, where the user does not have prior knowledge of the location of the information. We consider three information query strategies: (i) a Source-only search, where the source (user) tries to locate the destination by initiating query which propagates as a continuous time random walk (Brownian motion); (ii) a Source and Receiver Driven "Sticky" Search, where both the source and the destination send a query or an advertisement (both propagating as random walks), and these leave a "sticky" trail to aid in locating the destination; and (iii) where the destination information is spatially cached (i.e., repeated over space), and the source tries to locate any one of the caches. After a random interval of time with average t, if the information is not located, the query times-out, and the search is unsuccessful.

For a source-only search, we show that the probability that a query is unsuccessful decays as $(\log(t))^{-1}$. When both the source and the destination send queries or advertisements, we show that the probability that a query is unsuccessful decays as $t^{-5/8}$. Further, faster polynomial decay rates can be achieved by using a finite number of queries or advertisements. Finally, when a spatially periodic cache is employed, we show that the probability that a query is unsuccessful decays no faster than t^{-1}. Thus, we can match the decay rates of the source and the destination driven search with that of a spatial caching strategy by using an appropriate number of queries.

On the other hand, we argue that the memory requirement for spatial caching is larger (in an order sense) than that for "sticky" searches. This indicates that the appropriate strategy for querying over large sensor networks would be to use multiple queries and advertisements using the "sticky" search strategy.

keywords: Stochastic processes/Queueing theory

I. INTRODUCTION

With the availability of cheap wireless technology and the emergence of micro-sensors based on MEMS technology [10], [21], sensor networks are anticipated to be widely deployed in the near future. Such networks have many potential applications, both in the military domain (eg. robust communication infrastructure or sensing and physical intrusion detection), as well as commercial applications such as air or water quality sensing and control. Such networks are usually characterized by the absence of any large-scale established infrastructure, and nodes cooperate by relaying packets to ensure that the packets reach their respective destinations.

An important problem in sensor networks is that of querying for information. The query type could range from trying to determine the location of a particular node to querying for particular information. This problem has recently received a lot

of attention [8], [3], [2], [20], and is also related to routing over such networks [6], [19], [9], [7].

We consider a problem where a querying node (the source/user) transmits a query for some information (the destination), which is located at a (normalized) distance of '1' from the source. However, we assume that the source has no knowledge of the location of the information (the destination), nor that they are separated by a distance of '1' (i.e., *no destination location information is available*). Further, we assume that the nodes *do not have any "direction" information.* In other words, nodes only know who their local neighbors are, but do not have their geographical position or direction information.

We consider three search/query strategies: **(i)** A Source-only search, where the source (i.e., the user) tries to locate the destination by initiating a query which propagates as a "continuous time" random walk (Brownian motion); **(ii)** a Source and Receiver Driven "Sticky" Search, where both the source and the destination send a query or a advertisement (both propagating as random walks), and these leave "sticky" trails to aid in locating the destination; and **(iii)** where the destination information is spatially cached (i.e., repeated periodically over space), and the source tries to locate any one of the caches.

As an aside, we note that if partial destination location information was available, one could design strategies which explicitly use this knowledge. For instance, if it was known a-priori that the destination was at a distance '1' from the source, constrained flooding would be a candidate strategy for locating the position of the destination. Further, the availability of direction information (for instance, with GPS equipped nodes) would enable strategies similar to the "sticky" search, but route along "straight-lines" in the network [2] instead of random walks.

In this paper, we study query strategies in the absence of location/direction information. We consider the search strategies based on random walks, and associate an (exponentially distributed) random time-out interval, after which a query ceases to propagate. Thus, if the destination is not found before the time-out, the query is unsuccessful. We derive the asymptotic behavior of the querying strategies discussed above, and discuss their relative performance.

A. Related Work

As discussed earlier, there has been much work on querying and routing with information constraints in sensor networks. There have been various protocols and algorithms that have been recently developed for these networks [8], [3], [2], [20], [6], [19], [9], [18], [12], [13]. These protocols span various types – from using geographical information [9], [12], [8], [6] to (limited) flooding [18], [20] to gossip [13], [18] – and use
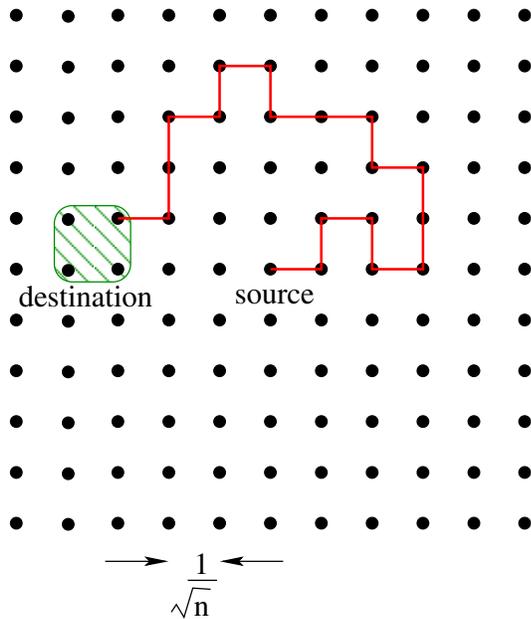
Fig. 1. A random walk based search on a sensor grid

various combinations of the three schemes we have considered. In Section II, we shall discuss [2] in detail.

We use Brownian motion based models for analysis. Related work includes [7], where the author studies optimal placement of limited routing information over a sensor network. The author shows that uniform placement over space will not be optimal. Instead, the routing information needs to be "concentrated" according to a specific pattern (a bicycle wheel spokes like pattern). Other related work includes [14], where the authors consider an object at origin and an infinitely large number of mobile nodes (initially placed according to a spatial Poisson process) looking for the object. They show that the probability that the object is not located decays exponentially in time. For the schemes we have described in Section I, there does not seem to be a analytical comparison in literature. In this paper, we derive the asymptotic performance of these schemes, and discuss their trade-offs.

II. QUERYING MODELS FOR A SENSOR NETWORK

Let us consider a regular sensor grid network, with n sensor nodes per unit area as shown in Figure 1. For such a model, nodes are spaced regularly with the inter-node distance being $1/\sqrt{n}$. Let us assume that each node can communicate a distance $1/\sqrt{n}$. In other words, the nodes can communicate with each of it's adjacent neighbors.

We are interested in the situation where a node located at origin (also referred to as the *source node*) is interested in locating some particular information, which is located at some region or node on the grid, henceforth referred to as the *destination*. We assume that the destination is located at a (normalized) distance of '1' from the source node. However, we assume that the source node has *no knowledge* of the location of the destination.

We consider the following three query strategies for locating the destination.

(i) Source-only Search: In this strategy, the source transmits a query requesting the destination information. As the location of the destination is unknown, the query propagates through the network as a random walk. In other words, each intermediate node along the query path picks a neighbor at random (i.e., with equal probability of 0.25) and forwards the query to it (see Figure 1). The destination is located at a normalized distance of '1' from the source. We assume that all nodes that are a distance less than ϵ (for some fixed $0 < \epsilon < 1$) from the destination know the location of the destination, or equivalently, they form a spatial memory structure (for example, all the nodes in the shaded region in Figure 1 have the information, or possibly store the information in a distributed manner). Thus, if the query enters a region which is at a distance less than ϵ, the query is said to be successful. The destination can then pass the information back to the source node by various means depending on the infrastructure available (for example, using an addressing mechanism where the routing tables in intermediate nodes are updated by the source query).

Further, we assume that there is a time-out associated with the query. At each intermediate node along the path of the query, the node could do one of two things: it could decide to not forward the query with some positive probability, or it could decide to forward the query (to a random neighbor). Thus, for such a forwarding model, the query will terminate after a geometrically distributed number of hops (if the destination has not been found).

(ii) Source-Receiver "Sticky" Search: The second strategy we consider is a "sticky" search. Unlike the previous case where the source alone sends a query, we now consider the case where *both* the source and destination send "probes" into the network. As before, the source sends (one or more) queries. These queries propagate as a random walk, with a geometrically distributed time-out. However, in addition, the destination also sends probes into the network to advertise itself. These advertisements propagate over the network in a manner similar to that of a query, i.e., as a random walk with a geometrically distributed time-out. Henceforth, we will refer to queries and/or advertisements as *probes* that are transmitted by the source and/or destination.

In addition, the source and destination probes leave a "sticky" trail as they traverse the networks (see Figure 2). That is, each node in the network through which the source query passes through *remembers* that a query passed through it and is searching for some particular information. Similarly, each node in the network through which the destination advertisement passes through *remembers* that a a probe passed through it advertising the presence of some information. Thus, if the source (destination) probe passes through a node through which the destination (source) previously passed through, it can simply trace backward along the destination (source) probe's path to reach the destination (source), see Figure 2. This scheme has been proposed by Braginsky and Estrin [2], and the authors develop the rumor routing protocol for sensor networks based on this idea.

(iii) Spatially-periodic Caching: Finally, the third strategy we consider is spatial caching. As in the previous cases, we assume source initiates a query which propagates as a random
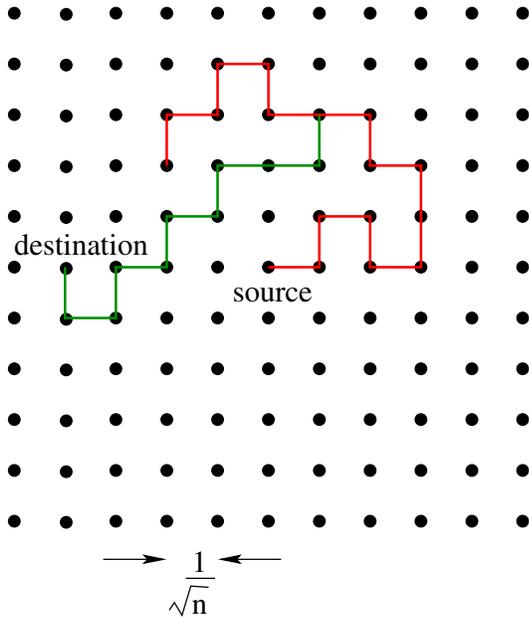
Fig. 2. A source and destination based "sticky" random walk on a sensor grid
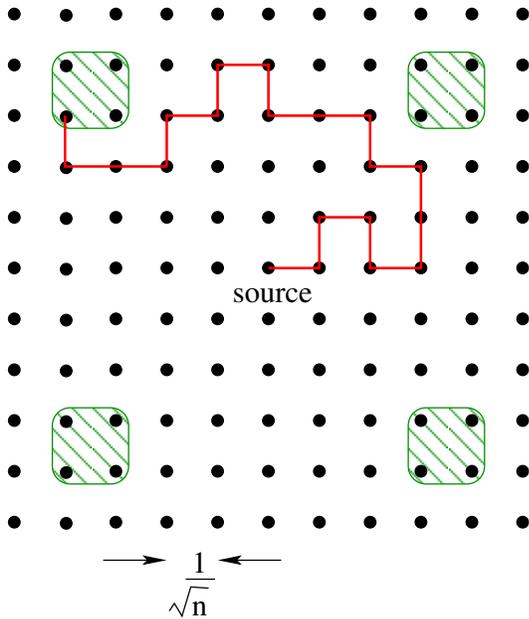
Fig. 3. Periodic caching over a sensor grid

walk with a geometrically distributed time-out. However, in this case, we assume that the destination information is assumed to be periodically cached along a regular grid as shown in Figure 3. Thus, we assume that there is some infrastructure which allows the dispersion of the information to each of the caches. Further, we assume that each of the caches is of radius $0 < \epsilon < 1$. Thus, as in the source-only search strategy, *each* cache of radius ϵ could either be a distributed memory structure, or else one of the nodes in each cache could have the data, and let all the other nodes within the cache point toward it.

A. A Brownian Motion Based Model

In the previous section, we considered a random walk on a grid, with step size $1/\sqrt{n}$ and a geometrically distributed time-out. In this section, we will informally show that for n large enough, this model can be approximated by a two dimensional planar model, where the source and destination probes propagate as a Brownian motion with an exponentially distributed time-out. Thus, in the rest of the paper, we will consider Brownian motion based models for querying over a sensor network.

Let us rotate the coordinate axes by 45 degrees, and let $(X(i), Y(i))$ be the random variables which correspond to the discrete step taken at time i, with respect to the new coordinate system. Thus, $(X(i), Y(i))$ can take one of four values: $(1/\sqrt{2}, 1/\sqrt{2})$, $(-1/\sqrt{2}, 1/\sqrt{2})$, $(1/\sqrt{2}, -1/\sqrt{2})$, or $(-1/\sqrt{2}, -1/\sqrt{2})$, each with equal probability, depending on which direction the random walk propagates.

Thus the location of the random walk $(L_x(j), L_y(j))$ at time j is given by

$$
\begin{aligned}
L_x(j) &= \frac{1}{\sqrt{n}} \sum_{i=0}^{j} X(i) \\
L_y(j) &= \frac{1}{\sqrt{n}} \sum_{i=0}^{j} Y(i)
\end{aligned}
$$

Now, let us define the continuous time processes

$$
B_x^n(s) = \frac{1}{\sqrt{n}} \sum_{i=0}^{\lfloor 2ns \rfloor} X(i) \tag{1}
$$

$$
B_y^n(s) = \frac{1}{\sqrt{n}} \sum_{i=0}^{\lfloor 2ns \rfloor} Y(i). \tag{2}
$$

Thus, at any continuous time s, $B_x^n(s)$ is the x-coordinate of the random walk after $\lfloor 2ns \rfloor$ time-steps (and correspondingly $B_y^n(s)$), and $\lfloor \cdot \rfloor$ is the integer floor function (i.e., the integer part of the argument). From standard theory for Brownian motion [11], we can show that the processes described in (1) and (2) converges (in a suitable sense) to a two dimensional Brownian motion $B(s) = (B_x(s), B_y(s))$ where $B_x(s)$ and $B_y(s)$ are independent one-dimensional Brownian motions. This follows from a central limit theorem type of an argument for functions[1].

In addition, we can show that under the time scaling that has been considered here (the mean time-out of the geometric random variable is scaled as $2nt$), the geometrically distributed time-out converges to an exponentially distributed time-out. We will denote the random variable corresponding to the time-out by τ, where $\tau \sim \exp(1/t)$. In other words, τ is exponentially distributed, with mean duration $E(\tau) = t$.

Thus, in the rest of this paper, we will study unit variance Brownian motion based models for querying, and the details of the models will be described in the appropriate sections of this paper.

[1]We scale time by $2n$ in (1) and (2) to ensure that the limiting Brownian motion has unit variance.

III. Main Results and Discussion

The main results in this paper are the following:

(i) For a source-only search, in Section IV, we show that the probability that a query is unsuccessful decays as $\frac{1}{\log(t)}$, where t is the mean time-out interval.

(ii) Next, in Section V, we consider the case where both the source and the destination send "sticky" probes (thus, memory in the network is utilized).
We show that the probability that a query is unsuccessful decays as $\frac{1}{t^{5/8}}$. Further, with k probes (i.e., the source sends multiple queries, and the destination sends multiple advertisements), the probability that a query is unsuccessful decays *at least* as fast as $t^{-5k/8}$.

(iii) Finally, in Section VI, we consider a spatially periodic caching strategy. We show that the probability that a query is unsuccessful decays *no faster than* $\frac{1}{t}$. We also provide an order computation which argues that spatial caching in fact leads to an order $\frac{1}{t}$ decay.

The results in Section IV shows that the probability that a query is unsuccessful decays as $(\log t)^{-1}$ for a single source-only search. By using multiple source probes, it is easy to see that the probability decays as $(\log t)^{-k}$, where k is the number of probes. On the other hand, the source and receiver driven search, as well as spatial caching utilize memory in the network. *This enables the decay probability to change from logarithmic decay to polynomial decay.* Thus, the implication is that *no matter how many finite number of source queries are used, we cannot match query techniques which utilize memory that is spatially distributed over the network.* In Section VIII, we will see that there is a marked difference in performance between the source-only search and the performance of query/search strategies which use memory.

Next, for the source and receiver driven "sticky" search, the probability that a query is unsuccessful decay *at least* as fast as $t^{-k5/8}$, where k is the number of probes. However, from the analysis in Section VI for spatially periodic caching, the probability that a query is unsuccessful decays *at most* as fast as t^{-k}, where k is the number of probes.

Thus, as both strategies have polynomial decay laws, by choosing the *number of probes* appropriately, it is possible to make the decay asymptotics for both the source and receiver driven search, as well as that due to spatial caching to be the same.

Finally, in Section VII, we present an argument to show that the memory requirement for caching is larger (in an order-wise sense) than that for "sticky" searches. While the spatial caching strategy requires memory that is proportional to the node density n, the "sticky" search strategy requires memory of the order[2] of $o(n)$.

Further, a caching strategy requires a higher degree of co-operation in the network than to simply send advertisements. These arguments indicate that the appropriate strategy for querying over large sensor networks would be to use multiple probes with a "sticky" strategy.

---

[2] A function $f(n)$ is said to be $o(g(n))$ if $\lim_{n \to \infty} f(n)/g(n) = 0$.
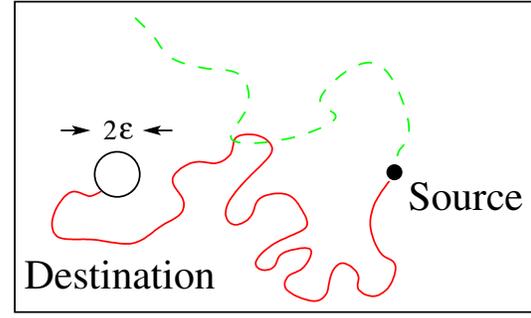
Fig. 4. A Source driven search

IV. Source-only Search

As described in Section II-A, we consider a query which propagates as a two-dimensional planar Brownian motion, denoted by $B(s)$, and ceases after an exponentially distributed time-out τ, with $E(\tau) = t$.

We assume that the destination is at a (normalized) distance '1' from the source. Without loss of generality, we assume that it is located at $(-1, 0)$ on the x-y plane (see Figure 4). The destination advertises that it has the required information to a small neighborhood (of radius ϵ) around itself. Thus, the query successfully finds the destination if the Brownian motion trajectory enters the circular region of radius ϵ about $(-1, 0)$.

Let us define

$$\hat{B}(s) = \begin{cases} B(s) & s \leq \tau \\ B(\tau) & s > \tau \end{cases}$$

Thus, $\hat{B}(s)$ is a Brownian motion that has been "stopped" at a random time τ. Next, we fix some small value of $\epsilon > 0$, ($\epsilon < 1$) and let $y = (-1, 0)$. Let $\mathcal{A} = \{x \in \Re^2 \mid d(x, y) \leq \epsilon\}$ be a ball of radius ϵ centered at $(-1, 0)$. Define

$$p_{sd}(t) = \Pr\left(\hat{B}(s) \text{ does not intersect } \mathcal{A}\right)$$

Thus, $p_{sd}(t)$ is the probability that the query does not locate the information before time-out.

Proposition IV.1: Fix any $\epsilon \in (0, 1)$. We have

$$\lim_{t \to \infty} (\log t) p_{sd}(t) = -2 \log \epsilon \quad (3)$$

Proof: Let us translate the axes such that the receiver is centered at origin. To do so, let us define

$$W(s) = B(s) - (-1, 0),$$
$$\hat{\mathcal{A}} = \{x \in \Re^2 \mid \mid x \mid \leq \epsilon\}$$

Thus, $W(s)$ is the shifted Brownian motion which starts at $(1, 0)$, and $\hat{\mathcal{A}}$ is a small circular region about origin. Then, from the translation invariance property of Brownian motion [11], we have

$$p_{sd}(t) = \Pr\left(\hat{W}(s) \text{ does not intersect } \hat{\mathcal{A}}\right)$$

Let $R(t) = \mid W(t) \mid$ be the Euclidean distance of $W(t)$ from the origin. $R(t)$ is known as the Bessel process of order zero.

Then, we have

$$p_{sd}(t) = \Pr\left(\hat{W}(s) \text{ does not intersect } \hat{\mathcal{A}}\right)$$

$$= \Pr\left(\inf_{0 \le s \le \tau} R(s) > \epsilon\right),$$

with $R(0) = 1$. From standard formulas for a Bessel process [1, page 373, 1.2.2], we have that

$$\Pr\left(\inf_{0 \le s \le \tau} R(s) > \epsilon\right) = 1 - \frac{K_0(\sqrt{2/t})}{K_0(\epsilon\sqrt{2/t})} \quad (4)$$

where $K_0(x)$ is the modified Bessel function of order zero, and is given by

$$K_0(x) = I_0(x)\log\left(\frac{2}{x}\right) - \mathcal{C}$$
$$+ \sum_{k=1}^{\infty} \frac{x^{2k}}{2^{2k}(k!)^2}\psi(k+1), \quad (5)$$

$$\psi(k+1) = -\mathcal{C} + \sum_{i=1}^{k}\frac{1}{i}, \quad (6)$$

$$I_0(x) = \sum_{k=0}^{\infty} \frac{1}{(k!)^2}\left(\frac{x}{2}\right)^{2k}, \quad (7)$$

where $\mathcal{C} = 0.5772$ is the Euler's constant. Let us denote

$$g(x) = \sum_{k=1}^{\infty} \frac{x^{2k}}{2^{2k}(k!)^2}\psi(k+1)$$

To show (3), we first prove the following.

$$\lim_{x \to 0} |g(x)| = 0 \quad (8)$$
$$\lim_{x \to 0} |g(x) - g(\epsilon x)| = 0 \quad (9)$$

These follow directly from the definition of $g(\cdot)$. To see that the limits are well defined, we observe that

$$\psi(k+1) < -\mathcal{C} + k < k!.$$

Thus, we have

$$|g(x)| < \sum_{k=1}^{\infty} \frac{1}{2^{2k}(k!)^2}x^{2k}(k!)$$
$$= \sum_{k=1}^{\infty} \frac{1}{k!}\left(\frac{x}{2}\right)^{2k}$$
$$= e^{x^2/4} - 1$$

Thus, (8) follows. The proof of (9) follows from the triangle inequality. Similarly, we have

$$\lim_{x \to 0} I_0(x) = 1 \quad (10)$$
$$\lim_{x \to 0} |I_0(x) - I_0(\epsilon x)|\log(1/x) = 0 \quad (11)$$

Equation (10) follows directly from (7) by substitution. To show (11), for any fixed $0 < \epsilon < 1$, and for any $x \ge 0$, we have from (7), we have

$$|I_0(x) - I_0(\epsilon x)| = \sum_{k=1}^{\infty} \frac{(1-\epsilon^{2k})}{2^{2k}(k!)^2}x^{2k}$$
$$< \sum_{k=1}^{\infty} \frac{1}{2^{2k}(k!)^2}x^{2k}$$
$$< \sum_{k=1}^{\infty} \frac{1}{2^{2k}k!}x^{2k}$$
$$= e^{x^2/4} - 1$$
$$\le \frac{x^2}{4},$$

Thus, we have

$$0 \le \lim_{x \to 0} |I_0(x) - I_0(\epsilon x)|\log(1/x)$$
$$\le \lim_{x \to 0} \frac{x^2}{4}\log(1/x)$$
$$= \lim_{y \to \infty} \frac{\log y}{4y^2}$$
$$= 0$$

Next, from (4), we have

$$\log(t)p_{sd}(t) = \log(t)\left(1 - \frac{K_0(\sqrt{2/t})}{K_0(\epsilon\sqrt{2/t})}\right)$$

$$= \log(t)\left[\frac{\left(\begin{array}{c}I_0(\epsilon\sqrt{2/t})\log(2/(\epsilon\sqrt{2/t}))\\-I_0(\sqrt{2/t})\log(2/\sqrt{2/t})\\-g(\sqrt{2/t}) + g(\epsilon\sqrt{2/t})\end{array}\right)}{\left(\begin{array}{c}I_0(\epsilon\sqrt{2/t})\log(2/(\epsilon\sqrt{2/t}))\\-\mathcal{C} + g(\epsilon\sqrt{2/t})\end{array}\right)}\right]$$

$$= \log(t)\left[\frac{\left(\begin{array}{c}\frac{1}{2}\log(2t)(I_0(\epsilon\sqrt{2/t}) - I_0(\sqrt{2/t}))\\-I_0(\epsilon\sqrt{2/t})\log(\epsilon)\\+(g(\epsilon\sqrt{2/t}) - g(\sqrt{2/t}))\end{array}\right)}{\left(\begin{array}{c}\frac{1}{2}I_0(\epsilon\sqrt{2/t})\log(2t)\\-\mathcal{C} + g(\epsilon\sqrt{2/t})\\-I_0(\epsilon\sqrt{2/t})\log(\epsilon)\end{array}\right)}\right]$$

Now, from (10) and (8), it follows that

$$\lim_{t \to \infty} \frac{\log(t)}{\left(\begin{array}{c}\frac{1}{2}I_0(\epsilon\sqrt{2/t})\log(2t)\\-\mathcal{C} + g(\epsilon\sqrt{2/t})\\-I_0(\epsilon\sqrt{2/t})\log(\epsilon)\end{array}\right)} = 2 \quad (12)$$

Finally, combining the various limits, from (12), (10), (11) and (9), we have

$$\lim_{t \to \infty} \log(t)p_{sd}(t) = -2\log(\epsilon).$$

■
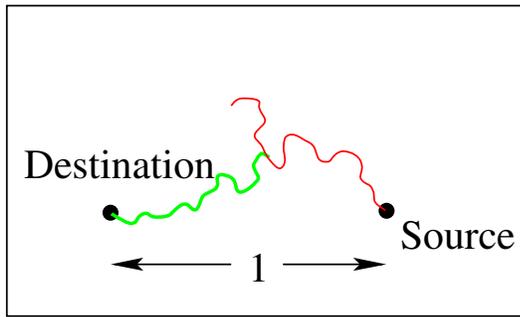
Fig. 5. Source-destination "sticky" search, where the query and the advertisement both leave trails in the network

We finally comment that by using multiple independent queries, it is clear that Proposition IV.1 implies

$$\lim_{t\to\infty} (\log t)^k p_{sd} = (-2\log(\epsilon))^k,$$

where k is the number of queries.

V. SOURCE AND RECEIVER DRIVEN "STICKY" SEARCH

In the previous section, we assumed that the destination advertises in a small neighborhood about itself. Instead, in this section, we consider the case where *both* the source and destination send probes into the network. As before, the source sends (one or more) queries, which are described by means of Brownian motions with exponential time-outs. The destination also sends probes into the network to advertise itself. These advertisements propagate over the network also as Brownian motions with exponential time-outs. In addition, as discussed in Section II, the probes each leave "sticky" trails as they traverse the networks (see Figure 5), where intermediate nodes in the network remember the query or advertisement that passed through it.

Let the source initiate m queries, each of which is described by a Brownian motion $B_i^{src}(t)$, $i = 1, 2, \ldots, m$. Over any interval of time $[0, T]$, we denote $B_i^{src}[0, T]$, $i = 1, 2, \ldots, m$ to be the corresponding trajectories. Similarly, let the destination initiate n advertisements, each of which is described by a Brownian motion $B_j^{dst}(t)$, $j = 1, 2, \ldots, n$, with the corresponding trajectories over $[0, T]$ denoted by $B_j^{dst}[0, T]$, $j = 1, 2, \ldots, n$.

Then, in recent work by Lawler et. al. [15], [16], [17] on intersection exponents for Brownian motion, the authors show the following result. Let us define

$$q_{mn}(T) =$$

$$\Pr\left[\left(\bigcup_{i=1}^{m} B_i^{src}[0, T]\right) \bigcap \left(\bigcup_{i=1}^{n} B_i^{dst}[0, T]\right) = \phi\right]$$

Thus, $q_{mn}(t)$ is the probability that the two "bundles" of Brownian motions will never intersect over the time interval $[0, T]$.

Theorem V.1: [Lawler, Schramm and Werner 2000] There exist $c > 0$ such that for any $T \geq 1$,

$$\frac{1}{c}T^{-\eta(m,n)} \leq q_{mn}(T) \leq cT^{-\eta(m,n)}$$

where

$$\eta(m, n) = \frac{1}{96}\left[\left(\sqrt{24m + 1} + \sqrt{24n + 1} - 2\right)^2 - 4\right]$$

In the problem we consider on source and receiver driven search, each Brownian motion has an independent time-out. Let τ_i^{src}, $i = 1, 2, \ldots, m$, and τ_j^{dst}, $j = 1, 2, \ldots, n$, be independent, identically distributed exponential random variables, with $E(\tau_i^{src}) = E(\tau_j^{dst}) = t$. Let us denote

$$p_{mn}(t) =$$

$$\Pr\left[\left(\bigcup_{i=1}^{m} B_i^{src}[0, \tau_i^{src}]\right) \bigcap \left(\bigcup_{j=1}^{n} B_j^{dst}[0, \tau_j^{dst}]\right) = \phi\right]$$

Thus, $p_{mn}(t)$ is the probability that the two "bundles" of Brownian motions[3], each with an independent time-out will never intersect. In our case, we have randomness due to two sources: the path of the Brownian motions, and the exponential time-outs. Interestingly, we observe that for $n + m \leq 3$, the decay function of $p_{mn}(t)$ depends only on the randomness due to the path of the Brownian motions. Using Theorem V.1, we will derive the exact asymptote. However, for larger numbers of Brownian motions, the randomness due to the exponential time-outs could also become a factor, and we will derive an upper bound on the non-intersection probability.

Proposition V.1: There exist finite, strictly positive constants c_1, c_2 such that

$$
\begin{aligned}
c_1 &\leq \liminf_{t\to\infty} t^{5/8} p_{11}(t) \\
&\leq \limsup_{t\to\infty} t^{5/8} p_{11}(t) \leq c_2 \quad (13) \\
c_1 &\leq \liminf_{t\to\infty} t p_{21}(t) \\
&\leq \limsup_{t\to\infty} t p_{21}(t) \leq c_2 \quad (14)
\end{aligned}
$$

Further, for any $k \geq 1$, there exists a finite, positive constant, $c_3(k) > 0$, such that

$$\limsup_{t\to\infty} t^{5k/8} p_{kk}(t) \leq c_3(k) \quad (15)$$

Proof: We first consider (13). Let us define $\bar{\tau} = \max\{\tau_1^{src}, \tau_1^{dst}\}$, and $\underline{\tau} = \min\{\tau_1^{src}, \tau_1^{dst}\}$. Then, we have

$$
\begin{aligned}
p_{mn}(t) &= \Pr\left[B_1^{src}[0, \tau_1^{src}] \bigcap B_1^{dst}[0, \tau_1^{dst}] = \phi\right] \\
&\leq \Pr\left[B_1^{src}[0, \underline{\tau}] \bigcap B_1^{dst}[0, \underline{\tau}] = \phi\right] \\
&= \int_0^\infty q_{11}(s)\frac{2}{t}e^{-2s/t}ds,
\end{aligned}
$$

where the last step follows from the definition of $q_{mn}(T)$, and the fact that $\min$ of two independent exponential random vari-

---

[3]From symmetry, we have $p_{mn}(t) = p_{nm}(t)$.

ables is also an exponential random variable. Thus, we have

$$
\begin{aligned}
p_{nm}(t) &= \int_0^1 q_{11}(s)\left(\frac{2}{t}\right)e^{-2s/t}ds \\
&\quad + \int_1^\infty q_{11}(s)\left(\frac{2}{t}\right)e^{-2s/t}ds \\
&\leq \int_0^1 \frac{2}{t}e^{-2s/t}ds \\
&\quad + c\int_1^\infty t^{-5/8}\left(\frac{2}{t}\right)e^{-2s/t}ds \qquad (16) \\
&= (1 - e^{-2/t}) + \frac{c\Gamma(3/8, 2/t)}{t^{5/8}}, \qquad (17)
\end{aligned}
$$

where $\Gamma(\cdot, \cdot)$ is the incomplete Gamma function. Equation (16) follows from Theorem V.1, and the fact that $q_{11}(s) \leq 1$. Equation (17) follows from expressions for exponential integrals [5, Section 3.381].

Now, observe that $(1 - e^{-2/t}) = 2/t + o(1/t)$. Further, it follows from the definition of the incomplete Gamma function that $\lim_{t \to \infty} \Gamma(3/8, 2/t) = \Gamma(3/8) \approx 2.27$. Thus, the upper bound in (13) follows. Similarly, the lower bound in (13) can be derived using the definition of $\bar{\tau}$. The proof of (14) is identical. We skip the details for brevity.

To show (15), observe that

$$
p_{kk}(t) \leq [p_{11}(t)]^k \qquad (18)
$$

To see this, recall that we have k independent Brownian motions originating from the source, which are indexed by $i = 1, 2, \ldots, k$; and k independent Brownian motions originating from the destination, which are indexed by $j = 1, 2, \ldots, k$. By considering intersections *only between equal indices* (i.e., $i = j$), and neglecting all intersections with indices $i \neq j$, the bound in (18) follows from the independence of the paths and the time-outs. Thus, (15) follows. ∎

Remark V.1: In Proposition V.1, we have the exact asymptotics when $n+m \leq 3$. However, when $n+m \geq 4$, we provide only an upper bound.

This is explained by observing that when $n+m \leq 3$, the upper bound based on the minimum of the time-outs is dominated by the randomness in the Brownian paths, and not the randomness in the time-out. This can be seen from (17). The first term in the summation corresponds to the randomness in the time-out, and the second term corresponds to the randomness in the paths. The first term is of order $1/t$, while the second term decays slower. Thus, the asymptote is dominated by the randomness in the Brownian path. However, for $n + m \geq 4$, the asymptote due to the Brownian path decays as $1/t^a$ for some $a > 1$. Thus, a bound using the minimum of time-outs will not provide a tight bound. In general, it is possible that the exact asymptote will depend on both the randomness due to the paths as well as the randomness due to the time-outs.

VI. SPATIALLY PERIODIC CACHING

Unlike in the previous sections where the target information was localized, in this section, we consider asymptotics
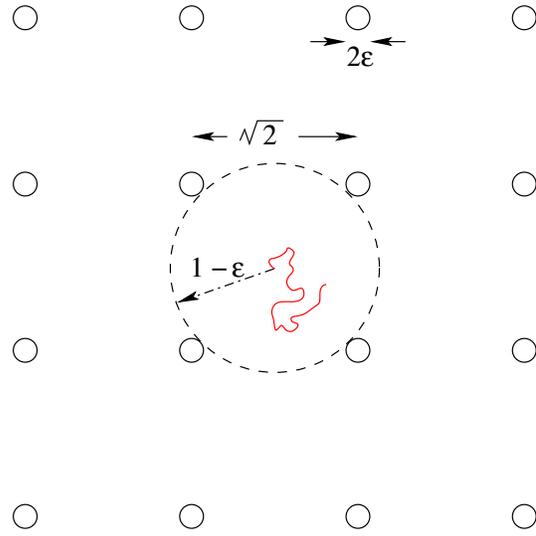
Fig. 6. A spatially periodic cache

of caching (see Section II). Assume that the source initiates a query from origin. The destination information is assumed to be periodically cached along a grid (see Figure 6), with the distance from the origin to the nearest cache being normalized to '1'. A small ϵ is chosen, and each cache is assumed to be of radius ϵ.

Let us define $p_{spc}(t)$ to be the probability that the query times-out before it reaches any cache (such as for the trajectory in Figure 6).

Proposition VI.1: Fix any $\epsilon \in (0, 1)$. We have

$$
\lim_{t \to \infty} t p_{spc}(t) \geq \frac{(1 - \epsilon)^2}{2} \qquad (19)
$$

Proof: The result follows from asymptotics of a killed Bessel process. As in Proposition IV.1, let $R(t) = |B(t)|$ be the Euclidean distance of $B(t)$ from the origin.

Suppose that the Bessel process $R(t)$, (the "radius'" process) does not exceed $1 - \epsilon$ before time-out (see Figure 6). Then, this clearly implies that the Brownian motion does not hit a cache before time-out. Thus, we have

$$
p_{spc}(t) \geq \Pr\left(\sup_{0 \leq s \leq \tau} R(s) < 1 - \epsilon\right),
$$

with $R(0) = 0$. From standard formulas for a Bessel process [1, page 373, 1.1.2], we have that

$$
\Pr\left(\sup_{0 \leq s \leq \tau} R(s) < 1 - \epsilon\right) = 1 - \frac{1}{I_0((1 - \epsilon)\sqrt{2/t})}
$$

where $I_0(x)$ is defined in (7). Thus, we have

$$
t p_{spc}(t) \geq \frac{t(I_0((1 - \epsilon)\sqrt{2/t}) - 1)}{I_0((1 - \epsilon)\sqrt{2/t})} \qquad (20)
$$

Consider the numerator in (20). We have

$$
t(I_0((1 - \epsilon)\sqrt{2/t}) - 1) = t\sum_{k=1}^\infty \frac{1}{(k!)^2}\left(\frac{1 - \epsilon}{2}\sqrt{\frac{2}{t}}\right)^{2k}
$$

$$= t\sum_{k=1}^{\infty}\frac{1}{(k!)^2}(1-\epsilon)^{2k}\left(\frac{1}{2t}\right)^k$$

$$= \frac{(1-\epsilon)^2}{2}\sum_{k=1}^{\infty}\frac{1}{k^2}\frac{1}{((k-1)!)^2}(1-\epsilon)^{2(k-1)}\left(\frac{1}{2t}\right)^{k-1}$$

$$= \frac{(1-\epsilon)^2}{2}\sum_{l=0}^{\infty}\frac{1}{(l+1)^2}\frac{1}{((l)!)^2}(1-\epsilon)^{2l}\left(\frac{1}{2t}\right)^l$$

$$= \frac{(1-\epsilon)^2}{2}$$
$$+\sum_{l=1}^{\infty}\frac{(1-\epsilon)^2}{2(l+1)^2}\frac{1}{((l)!)^2}(1-\epsilon)^{2l}\left(\frac{1}{2t}\right)^l \qquad (21)$$

Substituting (21) in (20), and taking the limit as $t \to \infty$, the result follows. ∎

The bound derived in this section is optimistic, in the sense that it under-estimates the probability that the query does not locate a cache (note that for a "perfect" search strategy, this probability should be zero). Thus, the result says that the "miss" probability is no better than order $1/t$. In the next section, we present an approximate computation which suggests that the bound captures the correct order of decay with respect to t.

Remark VI.1: We comment that the computation used in Proposition VI.1 could also be used to model a sensor network with multiple exit points to a wired Internet. Another example could consist of a large number of sensor nodes which measure some physical quantity, and report this data to one of many "fusion centers." If the location of the fusion centers are not known, the analysis in this section and that in Section VI-A could be used to determine the time-out that is needed for the message to be transfered to any one fusion center with a low probability of failure.

A. A Computation for Correctness of Order

In this section, we present an approximate computation that indicates that the optimistic bound in Proposition VI.1 is of correct order (with respect to t). As in the previous section, consider a spatially periodic cache as shown in Figure 7, with distance between consecutive caches being $\sqrt{2}$. In addition, we consider a boundary (indicated by the dotted curve in Figure 7), which consists of a circle with four protuberances. For small ϵ, the curve is approximately a circle of radius '1'.

Now consider a Brownian motion which starts at origin. There are three possible cases to consider: **(i)** due to time-out, the Brownian motion terminates before it hits the dotted curve in Figure 7 (i.e., before it travels a distance (approximately) '1' from origin) which is illustrated by the thick trajectory in Figure 7, **(ii)** the Brownian motion hits the dotted curve, but *does not* hit a cache, and **(iii)** the Brownian motion hits the dotted curve, and hits a cache.

Consider Case (ii), where the Brownian motion hits the dotted curve, but *does not* hit a cache. This is illustrated in Figure 7 by the trajectory which hits some point 'x' on the dotted curve. From the independent increments property of the path of the Brownian motion [11], and the memoryless property of the exponential time-out, we can 'restart' the Brownian motion from
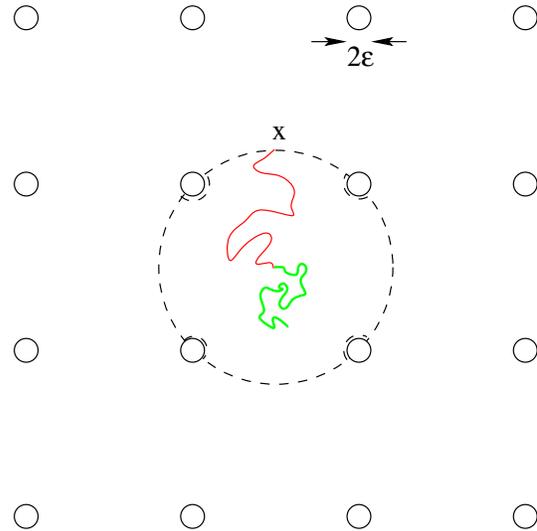
Fig. 7. Approximation by periodic "restarting"

location 'x' independent of both the path and the time it took to get there. In other words, we can reset the time to zero and consider a new exponential time-out as well as a new Brownian motion with initial value of 'x', independent of the past history.

As an approximation, instead of restarting at 'x', we will restart the Brownian motion at origin. This seems to be a conservative approximation. To see this, suppose that 'x' is very close to a cache. Then, restarting at origin will lead to a smaller cache hitting probability than restarting at 'x'. However, for a small cache radius ϵ, the probability that 'x' is close to a cache is small. For a small value of ϵ, given that the Brownian motion trajectory hits the dotted curve, 'x' is (approximately) uniformly distributed over the dotted curve, and the probability that it hits a cache is approximately $(4\times 2\epsilon)/(2\pi)$. Thus, with large probability, 'x' is not very close to a cache.

Now, let $q(t)=1-p_{spc}(t)$, be the probability that the Brownian motion trajectory hits a cache before time-out. Next, let $f(\epsilon)$ to be the probability that the trajectory hits a cache on the boundary of the dotted curve given that the Brownian motion trajectory hits the dotted curve. From the discussion in the previous paragraph, for a small value of ϵ, we have

$$f(\epsilon) \approx \frac{4\epsilon}{\pi}$$

Finally, let $g(\epsilon,t)$ be the probability that the trajectory hits the dotted curve (irrespective of whether it hits a cache or not) before time-out. For small ϵ and large t, from the discussion in Proposition VI.1, we have

$$g(\epsilon,t) \approx 1-\frac{(1-\epsilon)^2}{2t}$$

With the above definitions, and our approximation (of restarting the Brownian motion at origin), we have

$$q(t) \approx [g(\epsilon,t)f(\epsilon)] + [g(\epsilon,t)(1-f(\epsilon))][g(\epsilon,t)f(\epsilon)]$$
$$+[g(\epsilon,t)(1-f(\epsilon))]^2[g(\epsilon,t)f(\epsilon)]+\dots$$

The first term corresponds to the probability that the Brownian motion trajectory hits a cache when it first hits the dotted curve

(Case (iii) we considered earlier). The second term corresponds to the probability that initially Brownian motion hits the dotted curve, but *does not* hit a cache. However, the restarted Brownian motion hits a cache. A similar reasoning holds for the third term, and so on. Thus, summing the geometric series, we have

$$q(t) \approx \frac{g(\epsilon, t) f(\epsilon)}{1 - g(\epsilon, t)(1 - f(\epsilon))}$$

From the definition of $q(t)$, we now have

$$
\begin{aligned}
p_{spc}(t) &= 1 - q(t) \\
&\approx \frac{1 - g(\epsilon, t)}{1 - g(\epsilon, t)(1 - f(\epsilon))}
\end{aligned}
$$

Simplifying, and using the approximate expression for $g(\epsilon, t)$, we have

$$p_{spc} \approx \frac{(1-\epsilon)^2}{2} \frac{1}{f(\epsilon)t + \frac{(1-\epsilon)^2}{2}(1 - f(\epsilon))}$$

Thus, for large t, and fixed $\epsilon > 0$, this indicates that

$$p_{spc}(t) \sim \frac{1}{t},$$

i.e., this suggests that the bound derived in Proposition VI.1 is of the correct order with respect to t.

VII. MEMORY REQUIREMENT AND COMPARISON

As we have discussed in Section III, for a source only search, the results in Section IV shows that the probability decays to zero only logarithmically fast (i.e., $(\log t)^{-k}$, where k is the number of probes).

On the other hand, the "sticky" search as well as caching utilize memory in the network, and *this enables the decay probability to change from logarithmic decay to polynomial decay.* Thus, as both are polynomial decay laws, by choosing the number of probes appropriately, it is possible to make the decay asymptotics for the "sticky" search and due to spatial caching to be the same.

Now, let us consider the memory requirements for the spatially periodic caching strategy. From properties of Brownian motion, over an interval of time $[0, T]$, the Brownian motion traverses a distance $\sqrt{T}$. Thus, given that the probability that a query fails is at most δ, the search time-out should be set to have a mean of (order) $(\delta)^{-1}$ (because $p_{spc}(t) \sim 1/t$). Over this interval of time, the Brownian motion will traverse a distance of order $\sqrt{\delta^{-1}}$ from origin (see Figure 8). Thus, when spatially periodic caching is used, we need to make sure that caching is present at least over the region of radius $\sqrt{\delta^{-1}}$ from origin.

As there is one cache per unit area (a unit square about origin contains four "quarter" caches), we need a total of $\pi/\delta - 1$ caches[4], where we subtract '1' in the expression to discount for the "original" destination.

On the other hand, the Lebesgue measure of the trajectory of Brownian motion (i.e., the "area occupied" by the Brownian

---

[4]The area of a circle with radius $\sqrt{\delta^{-1}}$.
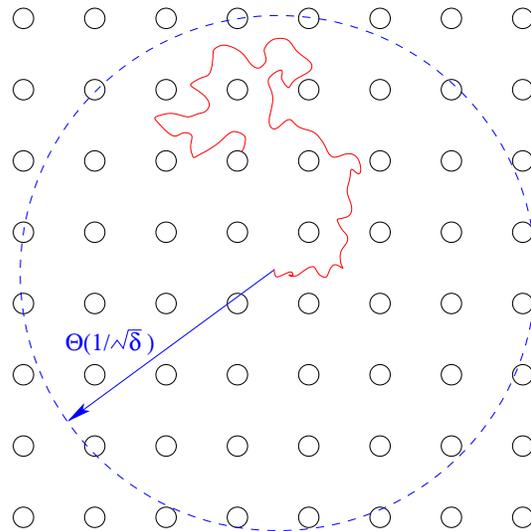
Fig. 8. Order computation for memory with periodic caching

motion) is zero (though its Hausdorff dimension is 2 which indicates that is not far from having positive Lebesgue measure [4]).

The analysis in this paper is a continuum approximation of a large sensor network. Let us now consider the discrete grid model with n nodes per unit area. As the caching strategy requires memory corresponding to a strictly positive area in the limiting regime, it follows that for large n, the memory requirement will need to scale *linearly* with the number of nodes per unit area. In other words, memory in $\Theta(n)$ nodes[5] will be used to aid the query.

However, a "sticky" search uses memory only along the path of the Brownian motion. Thus, this strategy will require memory to scale *more slowly than the number of nodes per unit area (i.e., memory in $o(n)$ nodes will be used to aid the query),* as the Lebesgue measure of the Brownian trajectory is zero.

Thus, the above argument indicates that the memory requirement for caching is larger (in an order-wise sense) than that for "sticky" searches. Further, a caching strategy requires a higher degree of cooperation in the network than to simply send "sticky" probes. These arguments indicate that the appropriate strategy for querying over large sensor networks would be to use multiple "sticky" probes.

VIII. SIMULATION RESULTS OVER A DISCRETE GRID

In this section, we present simulation results for a grid network. We consider a regular grid with $n = 100$ nodes per unit area (see Figure 1), and the spacing between neighboring nodes is $1/\sqrt{n} = 0.1$. Thus, in this section, we study simulation results for the random walk models described in Section II and the scaling described in Section II-A.

From the scaling described in Section II-A, an interval of time T will correspond to $2nT$ discrete time-steps. Thus, exponentially distributed time-out with mean t will correspond to a geometrically distributed random variable with mean $2nt$.

---

[5]A function $f(n)$ is said to be $\Theta(n)$ if there exist positive constants $c_2 > c_1 > 0$ such that for all n large enough, $c_1 \leq f(n)/n \leq c_2$.

Mean Time (t)	S-O S	S-R S	S-P C
1	0.975	0.515	0.659
10	0.826	0.132	0.125
100	0.646	0.016	0.014

TABLE I

COMPARISON OF QUERY STRATEGIES OVER A DISCRETE GRID. WE CONSIDER A GRID A WITH DENSITY OF 100 NODES PER UNIT AREA. THE QUERIES AND ADVERTISEMENTS PROPAGATE AS A RANDOM WALK WITH A GEOMETRICALLY DISTRIBUTED TIME-OUT. THE NUMBERS IN THE TABLE REPRESENT PROBABILITIES OF UNSUCCESSFUL QUERY. ABBREVIATIONS KEY: SOURCE-ONLY SEARCH (S-O S), SOURCE-RECEIVER "STICKY" SEARCH (S-R S), SPATIALLY-PERIODIC CACHING (S-P C)

In Table I, we present simulation results for the mean time-out $t = 1, 10$ and 100 respectively, with the cache radius, $\epsilon = 0.15$.

We see that for the source-only search, the probability that a query is unsuccessful decays extremely slowly with t. However, for the source and receiver driven search as well as spatial caching, the probability that a query is unsuccessful decays much faster. This agrees with the asymptotes which we have earlier discussed, where-in, we have shown that using spatial memory considerably increases the decay rate with respect to t, the mean time-out.

On the other-hand, the probability that the destination is not found is approximately the same for the source and receiver driven search, as well as with spatial caching. It is seen that for large t, spatial caching only slightly out-performs a "sticky" strategy. As both are polynomial decays, this simulation result is not very surprising, given the analytical results in the previous sections. Thus, the numerical results validate the asymptotic results in this paper.

IX. CONCLUSION

In this paper, we have considered the the problem of a user querying for information over a sensor network, where the user does not have prior knowledge of the location of the information. We have shown that for a source-only search, the probability that a query is unsuccessful decays to zero only logarithmically fast (i.e., $(\log t)^{-1}$). However, for schemes that utilize spatial memory in the network (the source and receiver driven search as well as spatial caching), this probability decays polynomially fast (i.e., $t^{-\alpha}$, for some $\alpha > 0$ depending on the scheme).

On the other, we have argued that the memory consumption with the spatial caching is much larger than that with a source and receiver driven search. Thus, the results in this paper make a strong case for utilizing a source and receiver driven "sticky" search in a sensor network.

REFERENCES

[1] A. N. Borodin and P. Salminen. *Handbook of Brownian Motion - Facts and Formulae*. Birkhauser, 2002.
[2] D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. In *First Workshop on Sensor Networks and Applications (WSNA)*, September 2002.
[3] M. Chu, H. Haussecker, and F. Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. Technical Report P2001-10113, Xerox PARC, 2001.
[4] J. F. Le Gall. *Some properties of planar Brownian motion*. Ecole d' Ete de Probabilites XX-1990, Lecture Notes in Mathematics 1527. Springer-Verlag, 1990.
[5] I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series and Products*. Academic Press, 2000.
[6] M. Grossglauser and M. Vetterli. Locating nodes with EASE: Last encounter routing in Ad Hoc networks through mobility diffusion. In *Proceedings of IEEE Infocom*, San Francisco, CA, June 2003.
[7] B. Hajek. Minimum mean hitting times of Brownian motion with constrained drift. In *Proceedeings of the 27th Conference on Stochastic Processes and Their Applications*, July 2000.
[8] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of ACM Mobicom*, Boston, MA, August 2000.
[9] R. Jain, A. Puri, and R. Sengupta. Geographical routing using partial information for wireless ad hoc networks. *IEEE Personal Communications*, 8(1):48–57, February 2001.
[10] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Mobile networking for smart dust. In *Proceedings of ACM Mobicom*, Seattle, WA, August 1999.
[11] I. Karatzas and S. Shreve. *Brownian Motion and Stochastic Calculus*. Springer, New York, NY, 1996.
[12] B. Karp and H. T. Kung. GSPR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 243–254, Boston, MA, August 2000.
[13] A.-M. Kermarrec, L. Massoulie, and A.J. Ganesh. Scamp: Peer-to-peer lightweight membership service for large-scale group communication. In *Proceedings of the Third International Workshop on Networked Group Communications (NGC 2001)*, London, UK, November 2001.
[14] G. Kesidis, T. Konstantopoulos, and S. Phoha. Surveillance coverage and communication connectivity properties of ad hoc sensor networks under a random mobility strategy. preprint, 2002.
[15] G. F. Lawler, O. Schramm, and W. Werner. Values of Brownian intersection exponents I: half-plane exponents. *Acta Mathematica*, 187:237–273, 2001.
[16] G. F. Lawler, O. Schramm, and W. Werner. Values of Brownian intersection exponents II: plane exponents. *Acta Mathematica*, 187:275–309, 2001.
[17] G. F. Lawler, O. Schramm, and W. Werner. Values of Brownian intersection exponents III: two-sided exponents. *Ann. Inst. Henri Poincare*, 38:109–123, 2002.
[18] M. J. Lin, K. Marzullo, and S. Masini. Gossip versus deterministic flooding: Low message overhead and high reliability for broadcasting on small networks. Technical Report CS1999-0637, University of California at San Diego, 18 1999.
[19] M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad-hoc networks. *IEEE Network Magazine*, 15, November 2001.
[20] N. Sadagopan, B. Krishnamachari, and A. Helmy. The ACQUIRE mechanism for efficient querying in sensor networks. In *IEEE International Workshop on Sensor Network Protocols and Applications (SNPA'03)*, May 2003.
[21] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5):16–27, October 2000.