Hitting Time Analysis for a Class of Random Packet Forwarding Schemes in Ad Hoc Networks

Chih-fan Hsin and Mingyan Liu
Electrical Engineering and Computer Science Department
University of Michigan, Ann Arbor, MI 48109

Abstract

In this paper we study the problem of searching for a node or a piece of data in an ad hoc network using random packet forwarding. In particular, we examine three different methods. The first is a random direction forwarding scheme where the query packet is forwarded along a randomly chosen direction (following an approximate straight line) till it either hits the destination node (the target) or the boundary. It bounces off the boundary in the latter case and the process continues till the target is found. In the second approach, in addition to query packet traversing the network, the target releases an advertisement packet that propagates along a randomly chosen direction so that all nodes visited by the advertisement packet obtain and store the target location information. In the third method the query packet is assumed to follow a random walk type of forwarding. Our primary interest is in comparing the average hitting time under these methods and the memory required to store location information. In particular, we show that under the random direction forwarding the target hitting time is $\Theta(\frac{a^2}{b})$, where a and b denote the size/radii of the network and the target area, assumed to be circular in shape, respectively. The hitting time is $\Theta(a)$ with target advertisement, and $\Theta(a^2 \log \frac{a}{b})$ under the random walk type of forwarding. We further show that the target advertisement method achieves mean hitting time on the same order as greedy forwarding schemes with less memory requirement. We discuss practical implementation issues of these methods and provide simulation results on their performance under more realistic settings.

Index Terms

Wireless sensor networks, ad hoc networks, data search, random forwarding, routing with uncertainty, random walk

This work was supported in part by the Engineering Research Centers Program of the National Science Foundation under NSF Award EEC-9986866, and NSF grant ANI-0238035.

I. INTRODUCTION

In this paper we consider the problem of searching for a node or a desired piece of data in a wireless ad hoc or sensor network. Specifically, a *querier* or source node sends out a query packet in search of a *target* or destination node located somewhere in the network. The query packet has to traverse the network in some way till it reaches the target, which then responds/replies to the source node. This problem arises in and is motivated by a variety of scenarios, including content location [1], service discovery [6], and data query in a sensor network [2], [4], [8].

The primary goal of this paper is to examine a class of query search methods based on random forwarding and attempt to gain a quantitative understanding of their performance in terms of the time it takes to locate the target, as well as the amount of location information required by the network. In particular we are interested in how the hitting time and information storage scale as the network becomes large (both in terms of the size and in terms the number of nodes in it).

The applicability of random forwarding based methods primarily lies with cases where there is no established query or routing infrastructure, e.g., those provided by data caches/replicas, central directory service, or an information gradient field [2], [3], and where the queries are simple and one-shot [4]. They are suited for situations where the data content in the network changes rapidly, thus making it difficult and costly to keep such infrastructure (routing table or gradient field) afresh, and when queries occur relatively infrequently.

These random forwarding schemes can also be applied to navigating a moving vehicle (or robot) in search of a certain target. In this case there will be no packet forwarding, but the vehicle follows successive random directions in its movement. Therefore the results obtained here apply to these problem as well.

There has been extensive study on data query and service discovery in ad hoc networks, and numerous approaches have been investigated. The methods studied here are representative abstractions of a subset of those proposed and studied in the literature. Below we describe these methods within the context of prior work, while noting that our focus in this paper is on the scaling property of hitting time and hitting distance, which is very different from most of the work cited below. Literature on hitting time studies is provided in Section VI.

We start with a scenario where no nodes in the network have the target location information except for the target itself and its neighbors within a certain range. Assuming that nodes have *relative* geographical location information about themselves and their immediate neighbors, the query packet is forwarded along a sequence of approximate straight lines of randomly chosen directions, bouncing off the network boundary, till it reaches the target or its neighbors. This model may be viewed as a special case of the trajectory based forwarding proposed in [5]. We will refer to this as the *random direction forwarding*, more precisely defined in the next section.

We then consider a second scenario similar to the previous one but with the addition that the target sends out an *advertisement* packet that is propagated along an approximate straight line of a randomly chosen direction. Nodes visited by the advertisement packet store the target location information, and when the query packet reaches one of these nodes, the target is considered found. This model may be viewed as a simplified version of those considered in [1], [6], where the target essentially sends out four advertisement packets traveling in four different directions. We shall see that this simplification does not affect our analysis. In [6], a pseudo quorum method was proposed in the context of providing matching service between data producer and subscriber where each procuder/subscriber sends out advertisement/subscription messages along four directions (e.g., north, south, east and west) so that a match can be found at intersecting nodes. Similar idea was used in [1] in the context of content location where both content discovery packets and content advertisement packets are sent along these four directions. In [7] a quorum based location service was proposed where nodes send out position information update along north/south directions with a certain thickness while packets searching for a destination travel along the east/west directions. The same idea of combining query and advertisement, and exploiting the fact that with high probability the two forwarding paths will intersect was proposed in [8] within the context of rumor routing. We will refer to the above method as the *enhanced random direction forwarding*, also defined in the next section.

We compare these two methods with random walk type of forwarding, where the query packet is randomly forwarded to a neighbor. Examples include [9], [10], that studied random walk forwarding on a grid, and [11]–[13], that applied swarm intelligence by sending out multiple query packets each following an independent random walk. This is a method where no location information is stored in the nodes, and no intelligent processing is required of nodes to maintain a consistent direction, as is required in the previous methods.

As a baseline, we will also compare random direction and enhanced random direction forwarding methods with greedy geographic forwarding method, which assumes that nodes already know the target location (thus this is more of a routing problem than searching). Using this method, each intermediate node selects the neighbor closest to the target as the next hop. Examples include greedy forwarding using precise target location information [14]–[16], as well as approximate or probabilistic geographical forwarding based on partial target location information [17].

Our principal results are derived under the following assumptions. We consider n nodes uniformly deployed in a disk of radius a, n may increase with a, and the node density is sufficiently high to ensure connectivity. The target node and nodes surrounding it form a *target area* modeled by a circle of radius b, located at the center of the disk [1]. These nodes do not have to be the target's immediate neighbors; they represent the area within which the target information is known. We will assume that $b \ll a$. Under these conditions, our main results are summarized as follows [2].

- Random direction forwarding achieves a mean target hitting time of $\Theta(\frac{a^2}{b})$ for an arbitrarily located querier/source. Random walk forwarding achieves a mean target hitting time $\Theta(a^2 \log \frac{a}{b})$ when the querier/source is located away (at a distance $\Theta(a)$) from the center of the target area, and $\Theta(a^2)$ when it is located close (at a distance $\Theta(b)$) to the center of target area.
- Enhanced random direction forwarding achieves a mean target hitting time $\Theta(a)$ for arbitrarily located querier/source. This comes at the expense of extra information dissemination and storage of the target location along the advertisement route. The memory requirement (defined as the mean number of nodes required to store target information) is $O(\sqrt{n})$.
- Under the greedy forwarding method, the mean target hitting time is $\Theta(a)$ when the querier/source is located at a distance $\Theta(a)$ from the center of target area, and it has a memory requirement of $\Theta(n)$.

In addition, we note that random direction and random walk forwardings can be viewed as two special cases of the family of methods characterized by the Lévy walk, which has been studied by physicists and biologists. Using Lévy walk we show via simulation that longer query propagation distance between changes of direction is preferred in terms of target hitting time. We also discuss practical implementation issues of using the forwarding methods studied here and examine the validity of our analytical results under different target locations, different network field shapes, as well as in the presence of error in location information.

The rest of the paper is organized as follows. We define the network model, the packet forwarding methods, and performance metrics in Section II. The forwarding delay (target hitting time) under different

[1] In Section V-A we consider other target locations.

[2] Notation: $f(n) = O(g(n))$ means $\limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty$. $f(n) = \Theta(g(n))$ means $f(n) = O(g(n))$ and $g(n) = O(f(n))$. $f(a,b) = \Theta(g(a,b))$ means $0 < \frac{f(a,b)}{g(a,b)} < \infty$ as $\frac{a}{b} \to \infty$.

methods are derived in Section III. These results are compared and discussed in Section IV. Practical implementation issues are presented in Section V. Section VI summarizes related work on hitting time studies, and Section VII concludes the paper.

II. Network Model and Assumptions

A. Network Model

We consider n nodes deployed within a disk of radius a. This circular field is denoted by G, its boundary ∂G. We assume that the destination node, or the *target*, is located at the center of the disk. This assumption is relaxed in Section V-A. The *target area* is a circle of radius b centered at the target, denoted by H, with boundary ∂H, as shown in Figure 1(a). We assume that $b \ll a$. The target area represents the neighborhood surrounding the target node within which nodes have the location information on the target. Thus once a query packet has reached a node within this area, including the boundary, the target is considered found. A *source* node s, or the *querier*, resides between the two concentric circles at a distance r from the center, where $b < r \leq a$. It initiates a query packet to be forwarded in a certain way with the goal of reaching H.

Our objective is to study a number of query forwarding strategies and to evaluate their effectiveness and timeliness in locating the target within the context of data search. The analysis conducted and results obtained here primarily apply to a static or quasi-static network where nodes' mobility is low with respect to the speed of packet forwarding. As we have mentioned in the previous section, they also apply to the scenario of a moving robot searching for a (static) target or data in a network.

We assume a high node density scenario, under which straight line forwarding is feasible (more precisely defined below), and that the network is connected given the density and a certain transmission range/radius $R(n)$. As the node density increases, the path that the query packet follows becomes increasingly well approximated by a sequence of straight lines. When we say that a path from node A to node B can be approximated by a straight line, we mean that the number of packet forwardings incurred between A and B is $\Theta(\frac{L}{R(n)})$, where L is the Euclidean distance between A and B. Subsequently, forwarding methods that satisfy this requirement will be referred to as *quasi-straight line* forwarding/routing methods.

In this sense, results in [18] showed that quasi-straight line routes can be constructed (which also implies connectivity) if $R(n)$ scales as $\Theta(\sqrt{\frac{\log n}{n}})$, where n nodes are uniformly distributed in a unit disk. This was done in [18] by first partitioning the network into cells each containing a disk of area $\Theta(\frac{\log n}{n})$, and then showing (1) that the sequence of cells crossed by the line segment connecting a source-destination pair each contains at least one (forwarding) node with high probability, and (2) that a node in a given cell can reach any node in a neighboring cell using a transmission range of $\Theta(\sqrt{\frac{\log n}{n}})$. For an expanding network, this result suggests that $R(n)$ has to be such that each node can reach $\Theta(\log n)$ neighbors in order for the network to be connected and for quasi-straight line routing to be feasible. Quantitative scaling relationship between n and $R(n)$ for a network to be asymptotically connected can also be found in [19], [20]. In addition, results in [17] showed that greedy geographic forwarding results in quasi-straight line routes, and [21] studied how to adjust $R(n)$ such that greedy forwarding always successfully finds such a route.

It should be noted that the random forwarding schemes considered here as well as our results do not rely on disk/circular transmission models, so long as quasi-straight line forwarding can be established. However, the connectivity condition we cited above is derived under a disk model.

Using these results, as long as the network density or the transmission range is sufficiently large, it may be assumed that a packet could follow a quasi-straight line route through the network. In our analysis, we will first assume that the query packet follows a perfect straight line. Then in Section V, we examine in more practical and realistic scenarios where the forwarding path is characterized by quasi-straight lines, and show that our results continue to hold.

Note that underlying this class of routing methods is the assumption that a node has relative location information regarding its neighbors. This is to ensure that a packet is forwarded in a consistent direction. This assumption is justified when nodes are equipped with GPS devices through which location information is directed obtained. It is also justified when a node has the ability to measure angle/direction of arrival of an incoming packet as well as the distance between itself and a neighboring node, from which relative location information may be extracted. The effect of error in such information on the forwarding scheme is examined in Section V-C. With such location information a node can also decide whether it is on the network boundary ∂G, by detecting the lack of neighbors in the half plane opposite of the network.

We consider a number of packet forwarding schemes described below.

B. Forwarding Methods

1) Random Direction Forwarding: Under this forwarding method, the querier/source node s randomly (uniformly) selects a direction from $[0, 2\pi)$, and sends the query packet to a neighboring node along that direction (as closely as possible). The same direction is followed by subsequent relaying nodes till

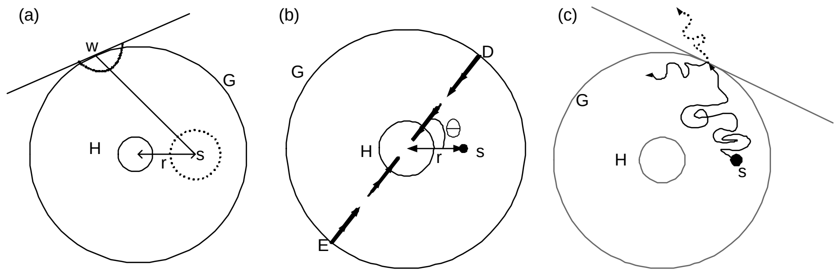

Fig. 1. (a) Random direction forwarding. (b) Enhanced random direction forwarding: target information is stored/advertised along $\overline{DE}$. (c) Boundary reflection under the random walk forwarding.

the packet either reaches the target area boundary ∂H or the network boundary ∂G. This results in an approximate straight line emanating from the source at a randomly chosen angle, illustrated in Figure 1(a). If the packet hits ∂G at node w before it hits ∂H, then node w randomly (uniformly) selects a direction within a half circle ($[0, \pi]$) toward G. This process continues till the packet hits ∂H.

It has been known (for example see [22]) that under the above boundary-reflection model, the query packet is more likely to be near the center of the field than the boundary. This results in a non-uniform search (or sweep) of the field. Uniform sweeping can be achieved by adopting an alternative reflection model. One such method was proposed in [23] that works as follows. Denote by Γ the angle between the random direction of the reflected straight line and the tangent to the boundary at the bouncing point. If, instead of a uniform distribution, Γ has a probability density function of $f_\Gamma(\gamma) = \frac{1}{2}\sin\gamma$, $0 \leq \gamma \leq \pi$, then the resulting search covers the field uniformly. In subsequent sections we will primarily analyze the first model where the reflection angle is uniformly distributed. Then in Section V-A, we will show that this alternative boundary-reflection model gives us the same order results.

2) Enhanced Random Direction Forwarding: Under this scheme the query packet is forwarded in exactly the same way as in the previous method. The difference is that in addition to query forwarding, the destination/target node *a priori* sends out an *advertisement* packet along a quasi-straight line in a randomly selected direction, as shown by the line $\overline{DE}$ in Figure 1 (b). The advertisement packet propagates the target location/data information to the nodes it visits. Such information is stored by nodes along this line, referred to as the *target line*. This effectively extends the target area, such that as soon as the query packet hits the target line, it obtains the target information and can simply follow the target line to reach the target area. Note that the two line segments emanating from the target do not have to be aligned; as we will show in Section III-B, this will not affect our analysis and results. In subsequent sections, we will first assume that the advertisement forwarding follows a perfect straight line in our analysis, and then discuss practical scenarios.

3) Random Walk Forwarding: Under this forwarding method, the query packet initiated by source s is relayed by a randomly chosen neighboring node and the same process repeats till the packet reaches the target area ∂H. This forwarding results in a random walk type of motion. It has been shown in [24] (Chapter II) that a discrete-time random walk on a grid network approaches a standard continuous-time Brownian motion if the distance traveled between two neighboring nodes is sufficiently small. That is, the random walk on a grid of n nodes in a fixed area can be modeled well by a standard Brownian motion for sufficiently large n. Subsequently in our analysis of this forwarding method, we will assume that the query packet follows a standard Brownian motion. Under this assumption, when the packet moves out of G, it is "pushed back" into G, the pushing normal to ∂G, i.e., when the packet moves out of G, the tangent line at the boundary-crossing point reflects the packet back to G, as illustrated in Figure 1 (c).

C. Performance Metric

The performance metric of interest is the *forwarding delay*, or *hitting time*, defined as the number of forwardings (or hops) the query packet takes between leaving the source and hitting the target area. With this definition, the delay due to congestion or collision/retransmission is ignored.

For the random direction forwarding and the enhanced random direction forwarding, we will assume that the transmission range $R(n)$ stays constant as a and n increases, and will subsequently also use the notation R. This allows us to directly relate hitting time to hitting distance [3]. It follows that

$$\text{Hitting time} = \Theta(\text{hitting distance}), \tag{1}$$

where *hitting distance* is the distance that the query packet traveled from the source to the target area. If $R(n)$ also increases with n (e.g., if the node density stays constant then $R(n)$ needs to increase as $\Theta(\sqrt{\log n})$ so satisfy the connectivity condition), then the above needs to be adjusted by dividing the right hand side by $R(n)$. More discussion on this is given in Section III-D.

Under the random walk forwarding scheme, the Brownian motion assumption is a continuous time approximation. In order to make results comparable under different schemes, we need to relate the continuous hitting time to the number of packet forwardings. The number of packet forwardings in a

[3] Note that to satisfy the connectivity and quasi-straight line routing condition mentioned earlier, we need nR^2/a^2, i.e., the number of nodes a transmitter can reach, to be $\Theta(\log n)$. For R to stay constant means that as a increases, n has to increase at a rate such that $a = \Theta(\sqrt{\frac{n}{\log n}})$.

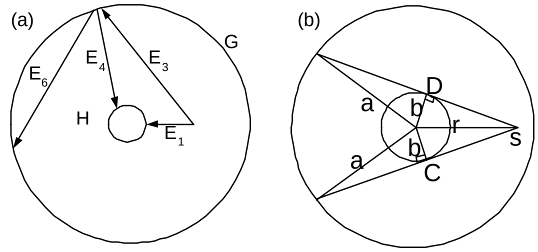

Fig. 2. (a) Illustration of different events. (b) Derivation of P_1.

random walk on a square grid, m, was shown to be $\frac{2t}{\delta^2}$ in approaching Brownian motion [24] (Chapter II), where δ is the grid cell length and t is the continuous time in Brownian motion. Assuming that δ is fixed, the hitting time (t) derived under Brownian motion has the same order as the number of packet forwardings (m). We will subsequently take the order of t to be equivalent of that of m in analyzing the random walk forwarding scheme.

When evaluating the enhanced random direction forwarding scheme, we will also consider a *memory requirement* metric, which quantifies the amount of location information that needs to be stored by nodes outside the target area. This requirement is measured by the mean number of nodes required to store such information.

III. Hitting Time under Different Forwarding Methods

A. Random Direction Forwarding

We first present a detailed analysis to precisely compute the mean hitting distance. This is followed by an order analysis to show how the mean hitting distance and time scale with the size of the network and the target area.

Below is a list of notations used; they are also illustrated in Figure 2 (a).

F_1: the event that the query packet reaches H without hitting ∂G given that the initial starting point is at a distance r from the center of H (also the center of the network as the two circles are concentric).

F_2: the event that the query packet reaches H eventually given that the initial starting point is at a distance r from the center of H.

F_3: the event that the query packet reaches ∂G without hitting H given that the initial starting point is at a distance r from the center of H.

F_4: the event that the query packet reaches H without hitting ∂G given that the initial starting point is on ∂G. (Due to symmetry, the exact position on ∂G is irrelevant.)

F_5: the event that the query packet reaches H eventually given that the initial starting point is on ∂G.

F_6: the event that the query packet reaches ∂G without hitting H given that the initial starting point is on ∂G.

L_i: the distance traveled by the query packet under event F_i, $i = 1, \cdots, 6$, respectively. This is a random variable with probability density f_{L_i}.

P_i: the probability of event F_i, $i = 1, \cdots, 6$, respectively. Note that $P_3 = 1 - P_1$ and $P_6 = 1 - P_4$.

Our goal is to derive the mean hitting distance $E[L_2]$. It is straight forward to verify that the following equalities hold:

$$E[L_5] = E[L_4]P_4 + (E[L_6] + E[L_5])(1 - P_4), \tag{2}$$

$$E[L_2] = E[L_1]P_1 + (E[L_3] + E[L_5])(1 - P_1), \tag{3}$$

where $E[L_i]$ denotes the mean value of L_i. Therefore,

$$E[L_2] = P_1E[L_1] + (1 - P_1)(E[L_3] + E[L_4]) + \frac{(1 - P_4)(1 - P_1)E[L_6]}{P_4}. \tag{4}$$

The above equation breaks down the calculation of the mean hitting distance into the mean distance of events F_1, F_3, F_4, and F_6.

Proposition 1:

$$P_1 = \frac{\arcsin(\frac{b}{r})}{\pi} \text{ and } P_4 = \frac{2\arcsin(\frac{b}{a})}{\pi}. \tag{5}$$

Proof: Given that the starting point is r away from the center of H, the packet can reach H if the direction of the straight line is chosen in the range of $\angle DsC$ illustrated in Figure 2 (b). Thus we can derive this probability as follows:

$$P_1 = \frac{\angle DsC}{2\pi} = \frac{2\arcsin(\frac{b}{r})}{2\pi} = \frac{\arcsin(\frac{b}{r})}{\pi}. \tag{6}$$

The derivation of P_4 is similar. The differences are that the starting point is at a distance a from the center and the direction can be chosen from a half circle $[0, \pi)$. ■

Proposition 2: For $l \in [r - b, \sqrt{r^2 - b^2}]$,

$$f_{L_1}(l) = \frac{\left|\frac{\frac{r^2-b^2-l^2}{l}}{\sqrt{2b^2r^2-b^4-r^4+(2r^2+2b^2)l^2-l^4}}\right|}{\arcsin(\frac{b}{r})}, \tag{7}$$

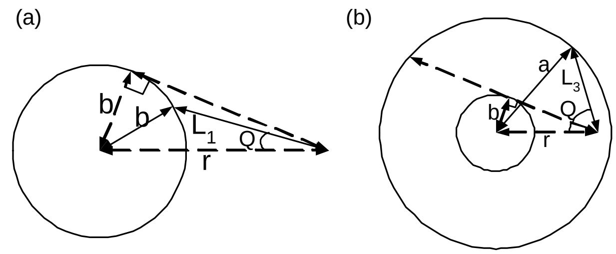

Fig. 3. (a) Illustration of F_1 and F_4 (when $r = a$). The circle is H. (b) Illustration of F_3 and F_6 (when $r = a$).

and for $l \in [a - b, \sqrt{a^2 - b^2}]$,

$$f_{L_4}(l) = \frac{\left|\frac{\frac{a^2-b^2-l^2}{l}}{\sqrt{2a^2b^2-b^4-a^4+(2a^2+2b^2)l^2-l^4}}\right|}{\arcsin(\frac{b}{a})}. \tag{8}$$

Proof: The source is at a distance r away from the center of H. Figure 3 (a) illustrates F_1 and F_4 (when $r = a$). Without loss of generality, we consider only the upper half circle. Given F_1, the angle Q is uniformly distributed in $[0, \arcsin(\frac{b}{r})]$. We have that $L_1^2 + r^2 - 2L_1 r\cos(Q) = b^2$. Thus Q, as a function of L_1, can be written as $Q(L_1) = \arccos(\frac{L_1^2+r^2-b^2}{2L_1 r})$. Therefore, for $l \in [r - b, \sqrt{r^2 - b^2}]$

$$f_{L_1}(l) = \frac{1}{\arcsin(\frac{b}{r})}\left|\frac{\partial Q(l)}{\partial l}\right| = \frac{1}{\arcsin(\frac{b}{r})}\left|\frac{\frac{r^2-b^2-l^2}{l}}{\sqrt{2b^2r^2 - b^4 - r^4 + (2r^2+2b^2)l^2 - l^4}}\right|. \tag{9}$$

Finally, $f_{L_4}(l) = f_{L_1}(l)$ when $r = a$. ■

Proposition 3: For $l \in [a - r, \sqrt{a^2 - b^2} + \sqrt{r^2 - b^2})$,

$$f_{L_3}(l) = \frac{\left|\frac{\frac{r^2-a^2-l^2}{l}}{\sqrt{2a^2r^2-a^4-r^4+(2r^2+2a^2)l^2-l^4}}\right|}{\pi - \arcsin(b)}, \tag{10}$$

and for $l \in [0, 2\sqrt{a^2 - b^2})$,

$$f_{L_6}(l) = \frac{1}{\frac{\pi}{2} - \arcsin(\frac{b}{a})}\left|\frac{l}{\sqrt{4a^2l^2 - l^4}}\right|. \tag{11}$$

Proof: The source is at a distance r away from the center of H. Figure 3(b) illustrates F_3 and F_6 (when $r = a$). Again, without loss of generality we consider only the upper half circle. Given F_3, the angle of Q is uniformly distributed in $(\arcsin(\frac{b}{r}), \pi]$. Given F_6, the angle Q is uniformly distributed in $(\arcsin(\frac{b}{r}), \frac{\pi}{2}]$. We have that $L_3^2 + r^2 - 2L_3 r\cos(Q) = a^2$. Thus Q, as a function of L_3, can be written as $Q(L_3) = \arccos(\frac{L_3^2+r^2-a^2}{2L_3 r})$, where $L_3 \in [a - r, \sqrt{a^2 - b^2} + \sqrt{r^2 - b^2})$. Therefore,

$$\frac{\partial Q(l)}{\partial l} = \frac{\frac{r^2-a^2-l^2}{l}}{\sqrt{2a^2r^2 - a^4 - r^4 + (2r^2+2a^2)l^2 - l^4}}. \tag{12}$$

	a = 2, b = 0.3					a = 2, b = 0.6					a = 3, b = 0.8				
	L_1	L_3	L_4	L_6	L_2	L_1	L_3	L_4	L_6	L_2	L_1	L_3	L_4	L_6	L_2
analytical mean	0.75	1.75	1.7575	2.395	23.625	0.47	1.6026	1.5	2.2125	9.875	0.26	2.525	2.3375	3.375	15
simulation mean	0.7525	1.7525	1.76	2.1025	21.275	0.4725	1.605	1.505	1.8925	8.8	0.265	2.55	2.34	2.9	13.35
simulation s.t.d	0.0275	0.3225	0.03	0.745	10.825	0.04	0.27	0.055	0.715	4.725	0.0375	0.2325	0.0725	1.0875	8.025

Fig. 4. Comparison between simulation and numerical results with initial position $r = 1$.

Thus $f_{L_3}(l) = \frac{1}{\pi - \arcsin(\frac{b}{r})} \left| \frac{\partial g(l)}{\partial l} \right|$ and similarly $f_{L_6}(l) = \frac{1}{\frac{\pi}{2} - \arcsin(\frac{b}{a})} \left| \frac{\partial g(l)}{\partial l} \right|_{r=a}$. ∎

With Propositions 1, 2, and 3, we can obtain the individual terms in Equation (4) and numerically evalutate $E[L_2]$. Figure 4 compares the simulation and the numerical results under different pairs of (a, b) values and the initial position $r = 1$. In the simulation results, each point is the average of 40000 random runs. In each run an object, starting from s as shown in Figure 2 (b), moves along a straight line with uniformly-selected direction and tries to reach H. We see that the numerical means of L_1, L_3, and L_4 match well with the simulation means. Small difference exists between the numerical and the simulation means for L_6 and L_2. This is partially attributed to the numerical integration approximation in Matlab.

The previous analysis provides us with a numerical method to compute the hitting distance. We next derive the scaling behavior of the hitting distance and time with respect to the size of the network.

Proposition 4: Suppose that the initial source position is at a distance r ($b < r \le a$) from the center of the field and $b \ll a$. Under random direction forwarding, the mean hitting time is $\Theta(\frac{a^2}{b})$.

Proof: From the previous propositions we know that $L_1 \in [r-b, \sqrt{r^2 - b^2}]$, $L_3 \in [a-r, \sqrt{a^2 - b^2} + \sqrt{r^2 - b^2}]$, $L_4 \in [a-b, \sqrt{a^2 - b^2}]$, and $b < r \le a$. Thus we have $E[L_1] = E[L_3] = O(a)$ and $E[L_4] = \Theta(a)$. In addition, for $l < 2\sqrt{a^2 - b^2}$,

$$E[L_6] = \frac{1}{\frac{\pi}{2} - \arcsin(\frac{b}{a})} \int_0^{2\sqrt{a^2-b^2}} l \cdot \left| \frac{l}{\sqrt{4a^2 l^2 - l^4}} \right| dl = \frac{1}{\frac{\pi}{2} - \arcsin(\frac{b}{a})} \int_0^{2\sqrt{a^2-b^2}} \frac{l}{\sqrt{4a^2 - l^2}} dl$$
$$= \frac{2a - 2b}{\frac{\pi}{2} - \arcsin(\frac{b}{a})} = \Theta(a). \quad (13)$$

From Proposition 1 we have $P_1 = \Theta(\arcsin(\frac{b}{r})) = \Theta(\frac{b}{r})$ and $P_4 = \Theta(\arcsin(\frac{b}{a})) = \Theta(\frac{b}{a})$. Thus from Equation (4), we have that

$$E[L_2] = \Theta(\frac{b}{r})O(a) + (1 - \Theta(\frac{b}{r}))(O(a) + \Theta(a)) + \frac{(1 - \Theta(\frac{b}{a}))(1 - \Theta(\frac{b}{r}))\Theta(a)}{\Theta(\frac{b}{a})} .$$

Since $b < r$, there are two possible cases, either b and r are on the same order, i.e., $\frac{b}{r} = \Theta(1)$, or $\frac{b}{r} \to 0$ as $\frac{a}{b} \to \infty$. In either case we see that the last term of the above equation dominates and has an order of

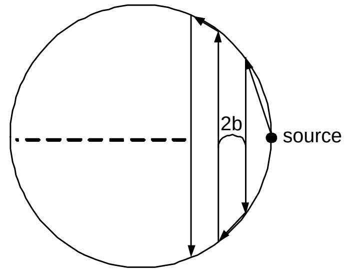

Fig. 5. An illustration of the query packet following a completely controlled regular sweep.

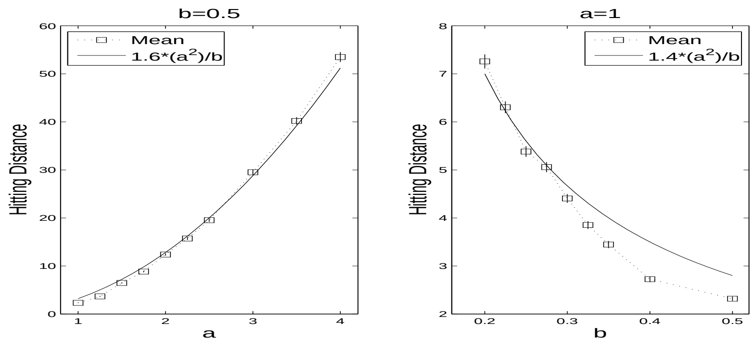

Fig. 6. Mean hitting distance of random direction forwarding: comparison between the order result and simulation (illustrated by curves with markers). Small vertical line segments are 95.4% confidence intervals.

$\Theta(\frac{a^2}{b})$. That is, we have

$$E[L_2] = \Theta(\frac{a^2}{b}). \tag{14}$$

The proposition is proven by noting that the mean hitting time has the same order as the mean hitting distance. ■

Interestingly, this result is the same as that can be achieved when the query packet is forwarded in a completely controlled regular sweep, as illustrated in Figure 5. However, in this regular sweep, the query packet needs to be much more precisely controlled, compared to the random direction forwarding.

Figure 6 compares the simulated mean hitting distance with function $\Theta(\frac{a^2}{b})$. Each point is the average of 10000 random runs. In each run an object, starting from position $(a, 0)$ (center of the disk has coordinates $(0, 0)$), moves along a straight line with uniformly-selected direction and tries to reach H. In the left plot, we fix b and vary a. We see that the mean varies closely to $\frac{1.6a^2}{b}$. In the right plot, we fix a and vary b. We see that the mean varies closely to $\frac{1.4a^2}{b}$ except for $b > 0.3$. This is because our order result is derived under the assumption $a \gg b$.

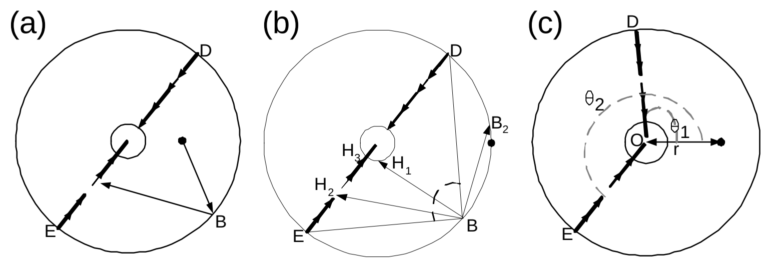

Fig. 7. (a) A possible sample path. (b) $\overline{BH_1} = \Theta(a)$, $\overline{BH_2} + \overline{H_2H_3} = \Theta(a)$, and $\overline{BB_2} = O(a)$. (c) General enhanced random direction forwarding. $\overline{DO}$ and $\overline{OE}$ are the target lines. $\theta_1, \theta_2 \in (0, 2\pi)$.

B. Enhanced Random Direction Forwarding

In this section we analyze the performance of the enhanced random direction forwarding, as described in Section II. Figure 7 (a) illustrates a possible sample path. Suppose that the query packet hits the boundary at the bouncing point B, and that the random direction is chosen such that the next straight line can hit the target area or the target line. The length of this hitting straight line and the probability of such an event depend on the position of B on the boundary. Thus results similar to Equations (2) and (3) are much harder to obtain. However, the derivation can be greatly simplified if we are only interested in the scaling behavior.

Proposition 5: Suppose that the initial source position is at a distance r ($b < r \leq a$) from the center of the field and $a \gg b$. Under the enhanced random direction forwarding, the mean hitting time is $\Theta(a)$.

Proof: As illustrated in Figure 7 (b), regardless of the bouncing point B, the straight-line distance to the target area, $\overline{BH_1}$, is $\Theta(a)$. Similarly, the summation of the straight-line distance to the target line $\overline{BH_2}$ and line $\overline{H_2H_3}$ is between a and $3a$, thus on the order $\Theta(a)$. Since the distances under these two hitting events (the target area hit and the target line hit) are on the same order, we can combine them and regard them as a single event, denoted by F_4. This is analogous to the F_4 in the derivation of Equation (4). Thus, $E[L_4] = \Theta(a)$ and $P_4 = \frac{\pi}{2} = \Theta(1)$.

Therefore, we can reuse Equation (4) with the difference that the target now is the combination of H and the target line. Following the same notation we can easily obtain $E[L_1] = E[L_3] = E[L_6] = O(a)$ and $P_1 \in (0, 1)$. The result then follows from Equation (4) and noting that the mean hitting time has the same order as the mean hitting distance. ■

A more general enhanced random direction forwarding is illustrated in Figure 7 (c), where $\overline{DO}$ and $\overline{OE}$ are the target lines, but may not be aligned. We have the following corollary. The proof is essentially the same as that of Proposition 5 and is therefore omitted.

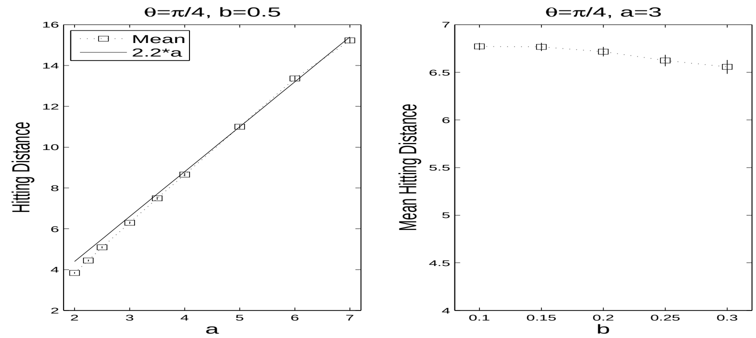

Fig. 8. Simulation results of the mean hitting distance of enhanced random direction forwarding. Small vertical line segments are 95.4% confidence intervals.

Corollary 1: Suppose that the initial source position is at a distance r ($b < r \leq a$) from the center of the field, $a \gg b$, and $\theta_1, \theta_2 \in (0, 2\pi)$. For the general enhanced random direction forwarding illustrated in Figure 7(c), the mean hitting time is $\Theta(a)$.

Figure 8 shows the simulated mean hitting distance. Each point is the average of 20000 random runs. In each run an object, starting from position $(a, 0)$ (center of the disk has coordinates $(0, 0)$), moves along a straight line with uniformly-selected direction and tries to reach H or $\overline{DE}$. In the left plot b remains constant, and we see the mean hitting length increases linearly with a. In the right plot a remains constant, and the mean hitting length stays relatively constant over small values of b.

C. Random Walk Forwarding

In this subsection we study random walk forwarding modeled by a standard Brownian motion.

Proposition 6: The mean hitting time under the random walk forwarding with initial distance r ($b < r \leq a$) from the center is given by

$$m(r) = a^2 \log(\frac{r}{b}) + \frac{b^2 - r^2}{2}. \quad (15)$$

Therefore, when $a \gg b$ and $r = \Theta(a)$, the mean hitting time is $\Theta(a^2 \log(\frac{a}{b}))$. When $a \gg b$ and $r = \Theta(b)$ (e.g., $r = 2b$), the mean hitting time is $\Theta(a^2)$.

Proof: The mean hitting time is a function of the initial position, denoted by $m(r, \theta)$, where (r, θ) is the 2-dimensional polar coordinate. The Brownian motion moves in a ring of outer radius a and inner radius b. From [24], for such a bounded domain, the mean hitting time satisfies the Poisson equation:

$$\nabla^2 m = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial m}{\partial r}\right) + \frac{1}{r^2}\left(\frac{\partial^2 m}{\partial \theta^2}\right) = -2. \quad (16)$$

Note that the mean hitting time does not depend on θ due to symmetry. Therefore,

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial m}{\partial r}\right) = -2. \tag{17}$$

To solve this differential equation, we need certain boundary conditions. We know that our Brownian motion stops at the inner circle of radius b. Thus $m(r = b) = 0$. It reflects at the outer circle of radius a. Thus we have the Neumann condition $\left[\frac{\partial m(r)}{\partial r}\right]_{r=a} = 0$. The solution to $m(r)$ then follows. ■

D. Discussion

The preceding results essentially show how the hitting time/distance scales as $\frac{a}{b} \to \infty$. This limiting regime can potentially describe a number of scenarios, e.g., when a and b both increase, with a increasing faster. A particularly relevant case is when b remains constant while a increases. In this case the target location information is limited to a constant sized region, and our results reveal how the hitting time scales when the network expands.

These results directly apply to hitting distance, provided that quasi-straight line routing is feasible. Applying them to hitting time relies on the assumption that the transmission range $R(n)$, or the advance made by the query packet in each hop, is constant. As discussed earlier, as the network increases in size, in order for the transmission range to remain constant while satisfying the connectivity and quasi-straight line routing conditions, the node density also needs to increase, i.e., the total number of nodes, n, needs to increase faster than a^2. If we wish to maintain constant node density, then the transmission range $R(n)$ will need to increase as well. In this case the scaling results derived in this section will need to be modified as follows. Note that $R(n)$ is on the order of $\sqrt{\log n}$ assuming constant node density. Since n scales as a^2, we have that R scales as $\sqrt{\log a}$. In the case of random direction forwarding, this means that the target hitting time is on the order of $\frac{a^2}{b\sqrt{\log a}}$.

IV. Hitting Time and Storage Comparison

With the preceding analysis, in this section we compare the hitting time of these schemes under different target location information assumptions. By doing so we also show the tradeoff between hitting time and location information storage required by these schemes. We will consider three cases discussed in the following.

A. Zero Target Location Information

We first examine the random direction forwarding and the random walk forwarding methods, both of which do not require nodes to have any target location information. From Propositions 4 and 6, we see that in terms of mean target hitting time, random direction forwarding is superior to random walk forwarding when the source is far away (with initial distance $\Theta(a)$) from the center of the target area. Nevertheless, when the source is very close (with distance $\Theta(b)$) to the center, random walk forwarding results in less mean hitting time ($\Theta(a^2)$) than random direction forwarding ($\Theta(\frac{a^2}{b})$). This can be intuitively explained by the fact that the query packet in random walk forwarding tends to move within a local neighborhood for a long time, while in the random direction forwarding it quickly leaves the neighborhood.

It is worth noting that these two forwarding methods may be viewed as two special cases of the same family characterized by the L$\acute{e}$vy distribution. Specifically, suppose that we can perfectly control the query packet's trajectory, and select for the j-th stage/trip a uniformly distributed random direction and a random trip distance l_j. The packet subsequently moves along the chosen direction for distance l_j, and the same process is repeated for the next, $(j+1)$-th stage/trip. This process specifies a family of forwarding/movement patterns determined by the distribution of l_j. This process is known as the L$\acute{e}$vy walk [25] if l_j has a L$\acute{e}$vy distribution given by

$$P(l_j) \sim l_j^{-\mu}, \tag{18}$$

where $1 < \mu \leq 3$.

L$\acute{e}$vy walk has been used by physicists and biologists in studying the movement of particles or animals, see for example [26]. It is known [25] that L$\acute{e}$vy walk approaches straight-line motion with random direction (i.e., random direction forwarding in our case) when μ approaches 1, and it becomes Brownian motion when μ is greater than or equal to 3 and the number of trips is large enough. Figure 9 shows the simulated mean hitting distance under L$\acute{e}$vy walk with different μ. Each point is the average of 1000 random runs. In each run, an object, starting from position $(a, 0)$, moves along a straight line with uniformly-selected direction until it reaches a trip distance drawn from the distribution $P(l_j)$ with the corresponding μ, or hits the boundary. It then re-selects a random direction and a random trip distance. The simulation ends when the object reaches H.

The functions $\frac{1.6a^2}{b}$ and $\frac{1.4a^2}{b}$ are provided in the figure as references of the mean hitting distance under random direction forwarding as shown earlier in Section III-A. We see that the mean hitting distance

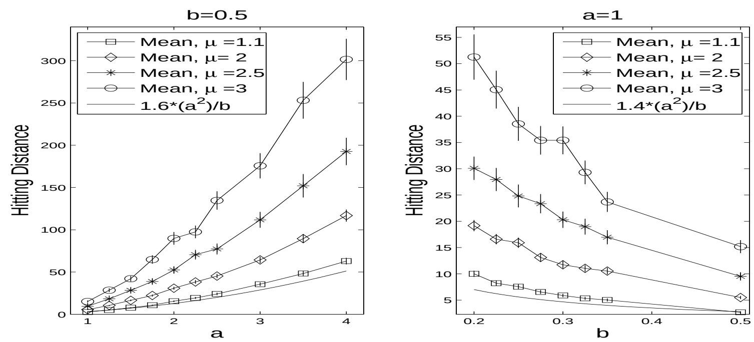

Fig. 9. Simulation results of the mean hitting distance of Lévy walk. Vertical line segments denote the the 95.4% confidence intervals.

decreases as μ decreases. Note that smaller μ means higher probability of longer trip lengths. Therefore, under our network model and performance metric, longer trip lengths are preferred. Note that the random direction forwarding and enhanced random direction forwarding schemes studied in this paper maximize the trip lengths in that they follow a straight line till the boundary is hit. By contrast, it was shown in [25] that, if there are multiple targets randomly distributed in the network and each can be multiply visited, then in order to hit the most number of targets per unit of time, the optimal value for μ is 2.

B. Partial Target Location Information

Under enhanced random direction forwarding, the target sends out a controlled advertisement to a small set of nodes, and the query packet only needs to reach this small set of nodes or the target itself. This can be regarded as a scenario where the network has partial target information. From Propositions 4 and 5, we see that enhanced random direction forwarding (with mean forwarding delay $\Theta(a)$) has less mean target hitting time than random direction forwarding (with mean forwarding delay $\Theta(\frac{a^2}{b})$). However, this comes at the expense of dissemination and storage of target location information along the target line. Specifically, an advertisement packet needs to be forwarded by nodes along a certain direction, and thus the number of nodes storing the target location is $O(\sqrt{n})$.

C. Complete Target Location Information

When every node in the network knows the target location, one can use geographical/greedy forwarding to get to the target, where each node chooses the next hop to be the neighbor closest to the target. If this path is a straight line and all forwarding step distances are R, then the hitting time is simply $\frac{r}{R}$, where r is the initial distance between the source and the target. When the forwarding path cannot be modeled as a perfect straight line, results of the same order may be obtained if the forwarding satisfies certain conditions. For example, *imperfect* greedy forwarding was studied in [17] where an intermediate node randomly selects as next hop a neighbor within a sector of its communication area toward the target, and the results showed that the mean hitting time is $\Theta(\frac{r}{R})$ where r is the initial distance (normalized to 1 and R scales as $\Theta(\sqrt{\frac{\log n}{n}})$. The same order result is also obtained in [17] when a randomly selected fraction of nodes have the target location information. Specifically, a node knows the target's quadrant with probability $p \in (0,1)$, and with probability $(1-p)$ it randomly forwards the packet. Subsequently, this model will be called greedy forwarding with partial information.

From Proposition 5, we know that the mean hitting time under enhanced random direction forwarding is $\Theta(a)$. Therefore, when the query source is at distance $\Theta(a)$ from the target, enhanced random direction forwarding achieves comparable mean hitting time (in terms of order) as greedy forwarding and greedy forwarding with partial information.

On the other hand, the enhanced random direction forwarding method has less memory requirement. In particular, the number of nodes along the target line is at most on the order of $\sqrt{n}$, n being the total number of nodes in the network. This results in a memory requirement of $O(\sqrt{n})$, while greedy forwarding has a memory requirement of n, and greedy forwarding with partial information has memory requirement $np = \Theta(n)$ since $p \in (0,1)$. This comparison points to the fact that a controlled scheme (i.e., nodes storing the location information are selected along a line) is more effective in this context.

V. DISCUSSION

A. Different Boundary Reflection Models, Target Locations, and Field Shapes

Our discussion so far has centered on the specific scenario of a circular field and target area, with the target at the center of the field. It would be desirable to obtain more general results for arbitrary convex shaped network fields and arbitrary target locations. This appears to be a difficult problem at least in the random walk forwarding case, for which most if not all of the existing results are limited to symmetric field shapes. When the forwarding paths are assumed to be perfect straight lines, it is possible to obtain more general results, see for example [23] where the field is assumed to be a general convex shape. On the other hand, the method used in [23] does not immediately apply to the enhanced random direction forwarding. More discussion on this is given in Section VI.

The following proposition generalizes our results of random direction forwarding and enhanced random direction forwarding when the target is centered at an arbitrary location while the field remains circular.

This result applies to both the non-uniform search method (boundary reflection according to a uniformly selected angle) and the uniform search method (boundary reflection according to $f_\Gamma(\gamma) = \frac{1}{2}\sin\gamma$, $0 \leq \gamma \leq \pi$). The proof is given in the appendix.

Proposition 7: Consider the same scenario defined in Section II, except that now the source is on the boundary and the target is centered at an arbitrary location (with the target area completely within the disk field) subject to the condition that its distance from the nearest boundary is $\Theta(a)$ [4]. Then under the two boundary-reflection models outlined earlier, the random direction forwarding has mean hitting time $\Theta(\frac{a^2}{b})$, and the enhanced random direction forwarding has mean hitting time $\Theta(a)$.

B. Random Direction Forwarding in a More Realistic Scenario

In our analysis so far, we have assumed that the query packet can follow a perfect straight line. Results obtained are compared with that of simulation which also assumes perfect straight line forwarding paths. Below we consider more realistic scenarios when packets follow quasi-straight line paths, and use simulation to examine the applicability of the results obtained under ideal assumptions.

We employ the following simple algorithm to realize the random direction forwarding in the simulation; the algorithm is similar to those found in [1], [6]. We assume that the initial position of a forwarding path (either the location of the querier or the location of the hitting point on the boundary) and the randomly chosen direction α with respect to a known reference zero degree are carried in the header of the query packet. An intermediate node first selects a set of neighbors within the same quadrant as the direction α [5], and then among these neighbors selects a relay that is the closest to direction α and provides the largest advance in terms of distance made forward along α.

In the simulation, 4000 nodes are uniformly deployed in a $[-a, a] \times [-a, a]$ square. Each node has transmission radius $R = 0.3$. The field G and the target area H are two concentric circles centered at $(0, 0)$, with radius a and b, respectively. The querier/source is chosen to be the node closest to $(a, 0)$. When the query packet reaches the node outside G or inside H, a boundary hit or a target hit occurs, respectively. Time is measured in terms of number of hops. Each data point is the average over 10 instances of random deployment, each with 20 runs, resulting in a total of 200 runs. The left plot of Figure 10 verifies that the mathematical result in Section III-A matches the simulation results well, i.e., the mean hitting time

[4] This means that while the target center can be very close to the boundary, e.g., with distance 0.001a, it scales with a.

[5] Under the assumption of sufficiently high node density, this set is non-empty with high probability. Otherwise, this scheme or in general greedy forwarding schemes will not work without additional enhancement.

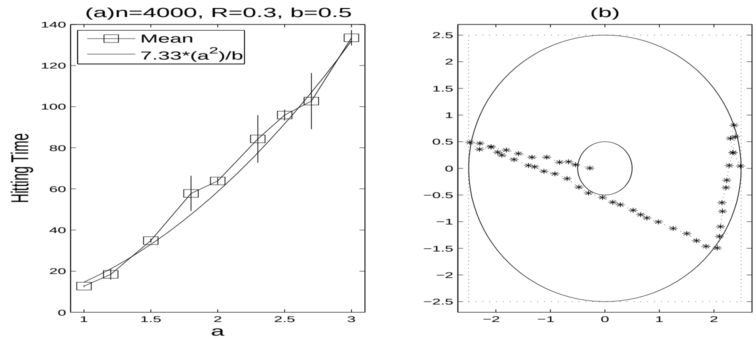

Fig. 10. (a) Mean hitting time (number of hops) of the practical packet forwarding scenario under random direction forwarding. Vertical line segments denote the 95.4% confidence intervals. (b) A sample path of random direction forwarding. Here $b = 0.5$ and $a = 2.5$. The source is located at approximately $(2.5, 0)$. The dotted line is the forwarding path, and the dots are the forwarding nodes.

scales as $\Theta(\frac{a^2}{b})$. The right plot of Figure 10 shows a sample path of random direction forwarding when $a = 2.5$ and $b = 0.5$.

The above result can be compared with that shown in Figure 6 under an idealized scenario. From Figure 6, we see that the mean hitting distance equals $1.6\frac{a^2}{b}$ when the packet follows a perfect straight line. If the maximum forwarding distance can be achieved in each hop, the mean hitting time under this idealized scenario is $\frac{1.6}{R}\frac{a^2}{b} = 5.33\frac{a^2}{b}$ for $R = 0.3$. On the other hand, Figure 10(a) gives a larger mean hitting time of $7.33\frac{a^2}{b}$. The difference is essentially due to the fact that the latter is obtained using quasi-straight line forwarding and less-than-maximum (but more realistic) forwarding distance per hop.

C. Effect of Location Errors

For the (enhanced) random direction forwarding, we have assumed that each node knows the relative location of its neighbors, and that the location information is error-free. In practice, such location information may contain error. Figure 11(a) illustrates an example of such, where we are trying to establish a quasi-straight line forwarding path from node A to C, and B is an intermediate node. Suppose that node B's knowledge about its neighbors' locations is erroneous, and B thinks that its neighbor N is located at where the white node is, while in reality N's true location is represented by the grey node. Subsequently node B selects N to be the next hop relay node and forwards the packet to N. As a result, the forwarding path goes backward instead of forward. If this situation occurs frequently, the quasi-straight line forwarding may not be possible.

Figure 11(b) illustrates how we can prevent this from happening when the location error is upper bounded. Suppose that the maximum location error (defined as the maximum displacement between the

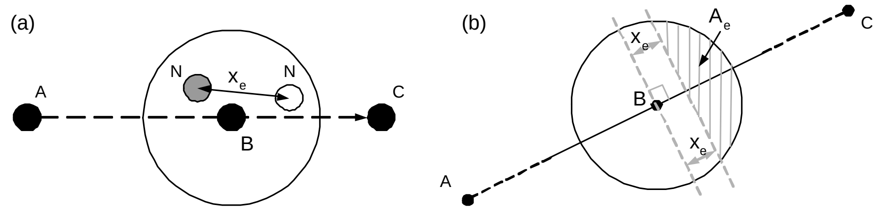

Fig. 11. Illustration of the effect of location errors on quasi-straight line route. Node B is an intermediate node along the route from A to C. (a) The gray node is the real position of neighbor N. The white node is the fake position of neighbor N due to location error. (b) A simple method to solve the problem caused by location error.

erroneous location and the actual location [6]) is x_e, and $x_e < R$. From Figure 11 (b), we can see that if node B selects any neighbor (based on its possibly erroneous location information) in the shaded area A_e, the true location of that selected neighbor will still lie within the right half of the circle, resulting in positive advance toward the target. This observation is formalized in the following proposition.

Proposition 8: Consider the scenario shown in Figure 11 (a) where a packet originating from node A needs to be forwarded to node C (or location C) which is assumed to be perfectly known. If all nodes along the forwarding path select the next hop from their corresponding shaded area A_e shown in Figure 11 (b) and $x_e > 0$, then the hitting time for the packet to go from A to C is $\Theta(\frac{L}{R})$, where L is the distance between A and C, and R is the transmission range. Thus quasi-straight line forwarding can be achieved.

Proof: If an intermediate node selects the next hop in the shaded area A_e, then the progress made toward the end position C is positive; denoted this by ξ_i where i indexes the i-th intermediate forwarding node. Thus there exist $\alpha, \beta > 0$ such that $\max_i \{\xi_i\} = \alpha \cdot R$ and $\min_i \{\xi_i\} = \beta \cdot R$. Then we have

$$\frac{L}{\alpha R} \leq \text{hitting time} \leq \frac{L}{\beta R}. \tag{19}$$

Since $\alpha, \beta \in (0, 1]$, the hitting time = $\Theta(\frac{L}{R})$. ■

This proposition essentially shows that, if the location error is upper bounded, then restricting the selection of the next hop to the area A_e can ensure (sufficient but not necessary) quasi-straight line forwarding.

Next we examine the effect of x_e on the probability of finding a next hop node in the area A_e. Basic geometric calculation gives the following:

$$A_e = R^2 \left[\frac{\pi}{2} - \arcsin(\frac{x_e}{R})\right] - x_e\sqrt{R^2 - x_e^2}. \tag{20}$$

[6]This displacement can either be the Euclidean distance between the true and erroneous positions, or the projection of this distance onto the direction $\overline{AC}$. If we take the latter then the sufficient condition given by Proposition 8 is tighter.

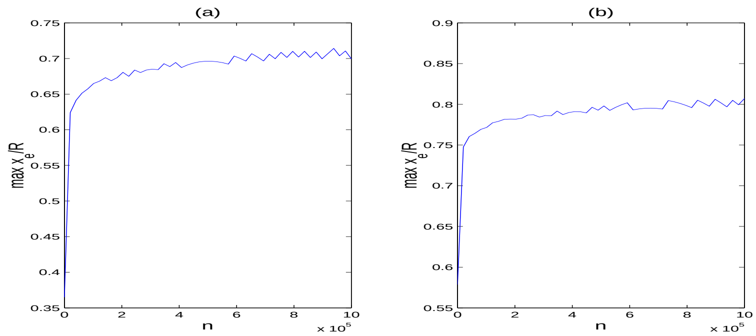

Fig. 12. The maximum location error in terms of percentage of R (i.e. $\max \frac{x_e}{R}$) to achieve $P_s = 0.9999$ when (a) $A = 1$ and $R = 1.5\sqrt{\frac{\log n}{n}}$ and (b) $A = 1$ and $R = 2\sqrt{\frac{\log n}{n}}$.

Thus the larger the location error x_e, the smaller the area A_e from which the next hop can be selected. Assume that the n nodes are uniformly deployed in a field of area A. Further assume that the noisy version of the locations are also uniformly distributed. Then the probability of finding at least one neighbor in A_e, denoted by P_s, is given by

$$P_s = 1 - \left(1 - \frac{A_e}{A}\right)^{n-1}. \tag{21}$$

With Equations (20) and (21), P_s is given as a function of x_e.

From Proposition 8, we know that, to enable quasi-straight line forwarding, it would be desirable to have P_s close to 1. Using Equations (20) and (21), we can find the condition under which P_s approaches 1. For example, Figure 12 shows the maximum location error in terms of percentage of R (i.e. $\max \frac{x_e}{R}$) to achieve $P_s \approx 1$ when $A = 1$ and $R = c\sqrt{\frac{\log n}{n}}$, where c equals 1.5 and 2, respectively. When $R = 1.5\sqrt{\frac{\log n}{n}}$ and n is large enough, this maximum location error x_e is around 70% of the transmission radius. When $R = 2\sqrt{\frac{\log n}{n}}$, it is around 80% of the transmission radius.

This analysis shows that the maximum location error allowed to achieve quasi-straight line forwarding depends on the parameter R. It is worth pointing out that if an intermediate node selects neighbors in its entire communication region instead of only in the shaded area A_e, then [27] has shown that the quasi-straight line routing is not significantly affected when the location errors are less than 40% of the transmission radius.

VI. Related Works

Search problems and movement patterns have been studied by physicists and biologists, within the context of food scavenging and foraging, particle movement, etc. For example, [25] studied how to efficiently search for multiple uniformly-deployed targets. The searcher follows the L$\acute{e}$vy walk described in Section IV-A, and the performance metric studied is the search efficiency, defined as the number of targets visited per unit distance traveled (or unit time). It was shown that, if a target can only be visited once, then the optimal μ in the L$\acute{e}$vy walk approaches 1, while if a target can be visited multiple times, then the optimal μ equals 2. It was also mentioned in [25] that the L$\acute{e}$vy walk with $\mu = 2$ has been used by bumble bees, deer, and wandering albatross in searching for food. It remains an open problem to relate search efficiency to the mean hitting time.

Stochastic search problems have also been studied by mathematicians. As mentioned earlier, [23] studied the mean hitting time under a model similar to the random direction model (the selection of the direction is different in [23]) in a general convex field with a circular target. The order result obtained in [23] is similar and comparable to the one derived in this paper, but the method used in [23] is very different from ours in this paper. In particular it does not immediately apply to methods like the enhanced random direction forwarding scheme. In this sense the method employed in the current paper lends itself to the analysis of enhanced random direction forwarding, but is otherwise not as general as that used in [23].

The idea of using a randomly-moving searcher to model the random forwarding/routing of packets has also been explored in wireless networks, including those cited earlier in Section I. In [10] search failure probability as a function of a preset timeout value was studied by modeling the query packet's movement as a standard Brownian motion. In [28] the minimum mean hitting time of a query packet searching for a target located at the center of a field was studied, where the packet's random route is modeled by Brownian motion in a constrained drift field.

VII. Conclusion

In this paper we studied the problem of searching for a node in a network using random packet forwarding. We derived scaling properties of target hitting times as funtions of the network and target dimensions for a number of commonly used random forwarding methods. We showed that when no node has target location information, random direction forwarding achieves smaller (order wise) mean hitting time than random walk forwarding, the additional requirement being slightly more intelligent forwarding in maintaining a consistent direction. With extra memory storing target information, enhanced random direction forwarding further reduces the mean hitting time. It also achieves comparable hitting time and less storage than the commonly used greedy forwarding scheme.

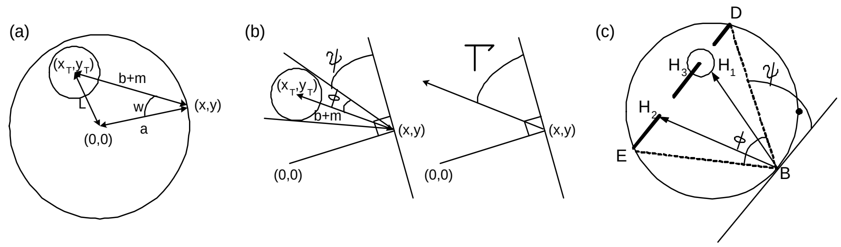

Fig. 13. (a) The illustration of the field G and the target area. (b) (x, y) is an arbitrary bouncing point, and the circle is the target area. The distance between the point and the target center is $b + m$. The random direction chosen by the bouncing point is Γ. (c)A possible sample path under enhanced random direction forwarding.

APPENDIX

PROOF OF PROPOSITION 7

Recall that the field G is a circle of radius a, and the target area is a circle of radius b, where $a \gg b$. The target center is located at an arbitrary location (x_T, y_T). The source is located at an arbitrary boundary point.

We first prove the mean hitting time under random direction forwarding. Figure 13(a) shows the field and the target area. In (a), the distance between the target center and an arbitrary boundary is $b+m$, and the distance between the target center and field center is L. Since the target area has to be inside the field, we know $L < a - b$. From the equation $L^2 = a^2 + (b+m)^2 - 2a(b+m)\cos\omega$, we can obtain

$$\cos\omega = \frac{a^2 + (b+m)^2 - L^2}{2a(b+m)} > \frac{m^2 + 2ab + 2mb}{2ab + 2ma}. \tag{22}$$

Consider the two boundary-reflection models. In the first model, the random direction Γ chosen by the bouncing node is uniformly distributed in the range of $[0, \pi]$. In the second model, Γ has probability density function $f_\Gamma(\gamma) = \frac{1}{2}\sin\gamma$, $0 \le \gamma \le \pi$. Suppose that the bouncing point is at (x, y), as shown in Figure 13(b), and the distance between the point and the target center is $b+m$. As we can see in Figure 13(b), if the packet is to hit the target area, Γ has to be in the range of $[\psi, \psi + 2\phi]$. Consider now the first boundary-reflection model. The probability P of hitting the target area from this arbitrary bouncing point is

$$P = \frac{2\phi}{\pi} = \frac{2}{\pi}\arcsin(\frac{b}{b+m}) = \Theta(\frac{b}{a}), \tag{23}$$

since we assume that $b + m = \Theta(a)$. It is easy to see that $1 - P = \Theta(1)$.

Now consider the second boundary-reflection model. Under this model, the hitting probability P is

$$P = \int_{\psi}^{\psi+2\phi} \frac{1}{2} \sin\gamma d\gamma = \frac{1}{2}[\cos\psi - \cos(\psi + 2\phi)]. \tag{24}$$

It is easy to check that $P \leq \sin\phi = \frac{b}{b+m} = \Theta(\frac{b}{a})$, where the equality holds when $\psi = \frac{\pi}{2} - \phi$. Furthermore, from Figure 13, $\psi = \frac{\pi}{2} - \omega - \phi$. Therefore, $P = \frac{1}{2}[\cos(\frac{\pi}{2} - \omega - \phi) - \cos(\frac{\pi}{2} - \omega + \phi)] = \cos\omega \cdot \sin\phi > \frac{m^2b+2ab^2+2mb^2}{2m^2a+2ab^2+4mab}$, where the inequality is from Equation (22). Since $b + m = \Theta(a)$, we know $P \geq \Theta(\frac{b}{a})$. Thus, under the second boundary-reflection model, $P = \Theta(\frac{b}{a})$.

Using the notation defined in Section III, since the source is on the field boundary, events $E_1 = E_4$, $E_3 = E_6$, and $E_2 = E_5$. Thus Equation (4) becomes

$$E[L_5] = E[L_4] + \frac{1 - P_4}{P_4} E[L_6]. \tag{25}$$

It is easy to see that $L_4 \in (0, 2a)$. Therefore, $E[L_4] = O(a)$. We also know that $P_4 = P = \Theta(\frac{b}{a})$ and $1 - P_4 = 1 - P = \Theta(1)$.

Consider $E[L_6]$ under the first boundary-reflection model.

$$\begin{aligned} E[L_6] &= \left(\int_0^{\psi} + \int_{\psi+2\phi}^{\pi}\right) 2a\sin\gamma \cdot \frac{1/\pi}{1-P} d\gamma, \text{denote } 1-P \text{ by a constant } c \\ &= \frac{2a}{c\pi}\left(\int_0^{\psi} + \int_{\psi+2\phi}^{\pi}\right) \sin\gamma d\gamma = \frac{a}{c\pi}[2 + \cos(\psi + 2\phi) - \cos\psi] \\ &\geq \frac{a}{c\pi}(2 - 2\sin\phi), \text{ since } \cos\psi - \cos(\psi + 2\phi) \leq 2\sin\phi. \end{aligned} \tag{26}$$

We know that $\sin\phi = \frac{b}{b+m} = \Theta(\frac{b}{a})$; thus, $E[L_6] = \Theta(a)$.

Consider $E[L_6]$ under the second boundary-reflection model.

$$\begin{aligned} E[L_6] &= \left(\int_0^{\psi} + \int_{\psi+2\phi}^{\pi}\right) 2a\sin\gamma \cdot \frac{\sin\gamma/2}{1-P} d\gamma, \text{denote } 1-P \text{ by a constant } c \\ &= \frac{a}{c}\left[\frac{\pi - 2\phi}{2} - \frac{\sin(2\psi) - \sin(2\psi + 4\phi)}{4}\right]. \end{aligned} \tag{27}$$

We can find that $\frac{a}{c}\left[\frac{\pi-2\phi}{2} - \frac{1}{2}\right] \leq$ Equation (27) $\leq \frac{a}{c}\left[\frac{\pi-2\phi}{2} + \frac{1}{2}\right]$. Since $\phi = \arcsin(\frac{b}{b+m}) = \Theta(\frac{b}{a})$, $E[L_6] = \Theta(a)$.

Therefore, under both boundary-reflection models, the mean hitting distance $E[L_5] = \Theta(\frac{a^2}{b})$. The proposition is proven by noting that the mean hitting time has the same order as the mean hitting distance.

The proof of the mean hitting time under enhanced random forwarding is similar to Proposition 5, but we need to use Equation (25) instead of Equation (4). In Figure 13(c), the straight-line length to the target area, $\overline{BH_1}$, is $\Theta(a)$ since the shortest distance between the target center and the boundary is $\Theta(a)$. Similarly, the summation of the straight-line length to the target line, $\overline{BH_2}$, and line $\overline{H_2H_3}$ is $\Theta(a)$. Again, we will combine the target area hit and the target line hit as a single event, and denote it by E_4. Thus, $E[L_4] = \Theta(a)$.

For P_4, under the first reflection model, $P_4 = \frac{\angle DBE}{\pi} = \frac{\phi}{\pi}$. Since the shortest distance between the target center and the field boundary is $\Theta(a)$, we know $\overline{DE} = \Theta(a)$. Furthermore, at least one of $\overline{DB}$ and $\overline{EB}$ is $\Theta(a)$. Thus, we can find $0 \ll \phi < \pi$ regardless of the value of a, which means $\phi = \Theta(1)$. As a result, $P_4 = \Theta(1)$. On the other hand, under the second reflection model, $P_4 = \frac{1}{2}[\cos\psi - \cos(\psi + \phi)]$. Since we know that $\phi = \Theta(1)$ and $\phi \gg 0$, $P_4 = \Theta(1)$.

It is easy to see that $E[L_6] = O(a)$. From Equation (25), the mean hitting distance $E[L_5] = \Theta(a)$. The proposition is proven by noting that the mean hitting time has the same order as the mean hitting distance.

REFERENCES

[1] J. Tchakarov and N. Vaidya, "Efficient content location in wireless ad hoc networks," in *IEEE International Conference on Mobile Data Management (MDM)*, January 2004.

[2] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, 2000.

[3] J. Faruque, K. Psounis, and A. Helmy, "Analysis of gradient-based routing protocols in sensor networks," *Lecture Notes in Computer Science*, July 2005, springer-Verlag.

[4] N. Sadagopan, B. Krishnamachari, and A. Helmy, "Active query forwarding in sensor networks," *Journal of Ad Hoc Networks*, vol. 3, pp. 91–113, 2005.

[5] B. Nath and D. Niculescu, "Routing on a curve," in *First Workshop on Hot Topics in Networks (HotNets-I)*, October 2002, princeton, NJ.

[6] I. Aydin and C. C. Shen, "Facilitating match-making service in ad hoc and sensor networks using pseudo quorum," in *11th International Conference on Computer Communications and Networks (ICCCN)*, 2002.

[7] D. Liu, I. Stojmenovic, and X. Jia, "A scalable quorum based location service in ad hoc and sensor networks," in *IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, October 2006, vancouver, Canada.

[8] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," in *First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002.

[9] D. Kim and N. Maxemchuk, "A comparison of flooding and random routing in mobile ad hoc networks," in *3rd New York Metro Area Networking Workshop (NYMAN)*, 2003.

[10] S. Shakkottai, "Asymptotics of query strategies over a sensor network," in *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2004.

[11] C. Shen and C. Jaikaeo, "Ad hoc multicast routing algorithm with swarm intelligence," *Mobile Networks and Applications*, vol. 10, 2005.

[12] J. S. Baras and H. Mehta, "A probabilistic emergent routing algorithm for mobile ad hoc networks," in *International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, 2003.

[13] G. D. Caro and M. Dorigo, "Antnet: Distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, vol. 9, 1998.

[14] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-case optimal and average-case efficient geometric ad-hoc routing," in *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, 2003.

[15] B. Karp and H. T. Kung, "Gpsr: greedy perimeter stateless routing for wireless networks," in *ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, 2000.

[16] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)*, August 1999, seattle, WA.

[17] S. Subramanian and S. Shakkottai, "Geographic routing with limited information," in *International Symposium on Information Processing in Sensor Networks (IPSN)*, 2005.

[18] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, 2000.

[19] ——, "Critical power for asymptotic connectivity in wireless networks," *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*, 1998.

[20] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: Coverage, connectivity and diameter," in *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2003.

[21] A. A. Jiang and J. Bruck, "Monotone percolation and the topology control of wireless networks," in *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2005.

[22] D. W. Gage, "Many-robot mcm search systems," in *Autonomous Vehicles in Mine Countermeasures Symposium*, April 1995.

[23] S. Lally and H. Robbins, "Stochastic search in a convex region," *Probability Theory and Related Fields*, vol. 77, pp. 99–116, 1988.

[24] E. Dynkin and A. A. Yushkevich, *Markov Processes: Theorems and Problems*. Prenum Press, 1969.

[25] G. M. Viswanathan, V. Afanasyev, S. V. buldyrev, and S. Havlin, "Statistical physics of random searches," *Brazillian Journal of Physics*, vol. 31, no. 1, 2001.

[26] M. F. Shlesinger, G. M. Zaslavsky, and J. Klafter, "Lévy flights and related topics in physics," *Nature*, vol. 363, no. 31, 1993.

[27] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, 2003.

[28] B. Hajek, "Minimum mean hitting times of brownian motion with constrained drift," in *27th Conference on Stochastic Processes and their Applications (SPA)*, 2001.