Elastic Ring Search for Ad Hoc Networks

Simon Shamoun[1], David Sarne[1], and Steven Goldfeder[2]

[1] Bar Ilan University
srshamoun@yahoo.com
[2] Columbia University

Abstract. In highly dynamic mobile ad hoc networks, new paths between nodes can become available in a short amount of time. We show how to leverage this property in order to efficiently search for paths between nodes using a technique we call elastic ring search, modeled after the popular expanding ring search. In both techniques, a node searches up to a certain number of hops, waits long enough to know if a path was found, and searches again if no path was found. In elastic ring search, the delays between search attempts are long enough for shorter paths to become available, and therefore the optimal sequence of search extents may increase and even decrease. In this paper, we provide a framework to model this network behavior, define two heuristics for optimizing elastic ring search sequences, and show that elastic ring search can incur significantly lower search costs than expanding ring search.

1 Introduction

An important function in ad hoc networks is the search for nodes, services, and resources in a network by forwarding requests from node to node. Using this function, nodes can search for resources as the need and availability arises. The alternative, maintaining tables of all available resources, requires a significant amount of overhead and may not even be possible given the dynamic changes that characterize ad hoc networks.

Search by forwarding requests is most prominently used in on demand routing protocols [14]. In on demand protocols, one node establishes a route with another as needed by issuing a query for that node and using a path the request was forwarded along until it reached the target node. This is especially useful in mobile ad hoc networks because node movement and wireless communications cause highly dynamic topology changes.

Most on demand routing protocols use broadcast flooding to establish routes either a priori or when other search attempts fail. In broadcast flooding, the search node broadcasts its query, which is then rebroadcast by all nodes that receive the query from the source node or any other node that subsequently broadcasts the query. The obvious drawback of broadcast flooding is that all nodes that receive the query are required to receive, process, and rebroadcast the query. This has a measurable cost to the network, such as power and bandwidth consumption.

One way to ameliorate search costs is by using an expanding ring search [6, 8, 4]. In Expanding ring search (ERS), the search node assigns the query a time-to-live (TTL) value, which limits the number of hops the query traverses along any path from the source, and consequently the cost of flooding. If, however, the number of hops to the target is greater than the TTL value, the search node must repeat the query with a larger TTL value, and repeatedly do so until the target is found. Although the total cost of this technique may sometimes be greater than the cost of full flooding, the expected cost is less when the sequence of TTL values is chosen correctly.

The optimal TTL sequence, the one that minimizes the cost of an expanding ring search, can be derived when the probability distribution of the number of hops to the target node is known. Only enough time is inserted between the queries of an expanding ring search to ensure that no path was found. With each failed query, the searcher updates its assumptions about the hop count probability distribution. Because the time between queries is very short, it is unlikely that new paths will become available at any time during the search, so the searcher assumes that only longer paths are available. Hence it uses *expanding* rings. However, if more time is inserted between queries, then there is a possibility that new paths, possibly shorter than the last TTL value used, will become available. In such a case, when a long time is inserted between queries, the searcher would update its assumptions about the hop count distribution differently, to the point that the assumptions eventually return to the steady-state.

In this paper, we design and analyze a search technique dubbed Elastic Ring Search (ELRS) that leverages changes in the network topology to keep search costs low. ELRS is similar to ERS in that queries are assigned TTL values, but it differs in that enough time is sometimes inserted between queries to allow the probability distribution assumptions to return partially or completely to the steady-state. As such, it may be optimal to search up to increasing *and* decreasing ranges; hence, the rings are *elastic*. We specifically consider the following type of search: At equally spaced time intervals, the searcher issues a query with any TTL value or no query at all until the target is found or the search is terminated. ELRS can be succinctly described as a time-diffused TTL-based search with increasing and decreasing search rings. This is useful when the searcher does not need to immediately establish a connection with the target node, like when transferring files and data. To optimize its performance, one additional network property is needed–the rate at which the hop count probability distribution is assumed to return to the steady-state.

Our main contributions are the model and design of the elastic ring search, the study of the aforementioned network properties, and the design heuristics for cost-efficient elastic ring search sequences. We specifically design a model for understanding the changing assumptions about the hop count probability distribution that is easily verified in simulations and simplifies the derivation of elastic ring search strategies. Our simulation results show that the average costs closely match the expected costs based on the model and analysis, implying that elastic ring search is practically applicable.

The outline of the remainder of the paper is as follows: We discuss related work in Section 2. We formally model the problem in Section 3, and analyze the problem and provide two heuristics to solve it in Section 4. In Section 5, we statistically analyze the steady-state properties, and we evaluate the cost of the two heuristics described in Section 4. Finally, we summarize our results in Section 6.

2 Related Work

Expanding ring search has been extensively analyzed in the literature [7, 5, 6, 8, 1, 4]. Most analysis assumes that all nodes are distributed uniformly at random throughout the field [7, 6, 8]. The corresponding hop count probability and cost-per-hop for each TTL value are used to calculate the expected costs of various TTL sequences. Chang and Liu [4] show how to derive the optimal TTL sequence for an arbitrary cost function and hop count distribution when they are known *a priori*. They prove that the optimal randomized strategy when the probability distribution is not known has a tight worst-case approximation factor of e, and that this is the best approximation factor possible by any solution. Baryshnikov, et al. [1], prove that the optimal deterministic strategy in this case is to double the TTL value each round and that it has a tight worst-case approximation factor of 4.

Blocking Expanding Ring Search (BERS) and its variations [13, 12, 11] are designed to reduce redundant transmissions of route requests incurred by expanding ring search. The searcher sends a route request only once, and sends a cancel message when it receives a route reply. Each node waits long enough to receive a cancel message before forwarding the request. Theoretically, BERS only requires two times the number of transmissions needed to reach the extent of the destination, but this depends on the success of the cancel message in reaching all nodes that received the route request and does not necessarily guarantee a lower expected cost than ERS.

Use of the random waypoint mobility model in simulation studies raised numerous questions about its steady-state properties [2, 3, 15]. Yoon, et al. [15] show that typical initial configurations, including node locations and speed, do not conform to the steady-state properties. Bettstetter, et al. [3] show that increasing the maximum speed increases the node concentration in the middle of the field. They additionally show that nodes are distributed uniformly at random throughout the field when moving according to the random walk mobility model. Mukerjee and Avidor [10] provide a theoretical analysis of the hop count distribution when nodes are distributed uniformly at random and paths are subject to fading. More recently, Younes and Thomas [16] analyzed the hop count distribution under the random waypoint model. However, none of these studies consider how these distributions change with new information, as we do in this paper.

3 Problem Model

In this section, we formally model the problem and define the probability model we use in this paper. Throughout the paper, we refer to the node conducting the search as the searcher or the source and the node being searched for as the target or destination. In the model and the analysis, we assume that all nodes in the network are connected.

Let $R = \{x_0 = 0, \ldots, x_m = x_{max}\}$, $x_i < x_{i+1}$ $\forall i$, be the set of m TTL values from which the searcher can choose. x_{max} is the TTL value required to reach all nodes in the network. Rather than setting $x_i = i$ for all $1 \leq i < m$, R is generally defined because sometimes only a few search ranges need to be considered. For example, if the target is known to be a certain distance away, then the searcher does not need to consider anything less than the minimum number of hops required to reach a node in that location. Let $f(x)$ be the steady-state probability mass function that characterizes the distance to the target, and $F(x)$ be its cumulative distribution function. The cost of searching up to range x is denoted $C(x)$. This is the expected number of nodes that are at most x hops away from the searcher. The analysis will sometimes refer to the normalized cost function $C'(x)$, which is defined as $C'(x) = C(x)/C(x_{max})$. The standard assumption in the literature is that these functions can be determined and are known [7, 6, 8, 4].

A search strategy S is a sequence of n search ranges $[u_1, \ldots, u_n]$ chosen from the set R. The search is conducted in rounds, searching up to u_i in round i if $u_i > 0$, or otherwise not searching at all, until the target is found. A non-decreasing strategy is a specific type of strategy in which the non-zero search ranges are non-decreasing. Formally stated, a search strategy $S = [u_1, \ldots]$ is non-decreasing if $\forall_i (u_i = 0 \vee \forall_{j<i} u_j \leq u_i)$. The time between consecutive rounds is fixed and long enough to send a packet to the furthest node and back [4, 9], and even longer to allow the topology to change.

The probability that the distance to the target is x at the beginning of round i, before executing the search up to u_i, depends on the search ranges used in the previous rounds. This is denoted $f(x|u_1,\ldots,u_{i-1})$. Likewise, the cumulative probability is denoted $F(x|u_1,\ldots,u_{i-1})$. The expected cost $J(S)$ of applying the search strategy $S=[u_1,\ldots,u_n]$ is therefore calculated as follows:

$$J(S)=C(u_1)+\sum_{i=2}^{n}\prod_{j=1}^{i-1}(1-F(u_j|u_1,\ldots,u_{j-1}))C(u_i) \tag{1}$$

The conditional probability function $f(x|u_1,\ldots,u_{i-1})$ is potentially different for every combination of values $u_1,\ldots,u_{i-1}$, since each failure to find the target reveals new information about its location. It is infeasible to define a separate function for each combination of values, since there is an exponential number of combinations. Instead, we define a generic model for the conditional probability based on Bayes' Theorem. Not only does this make calculating the conditional probability feasible, it also makes it possible to derive efficient solutions.

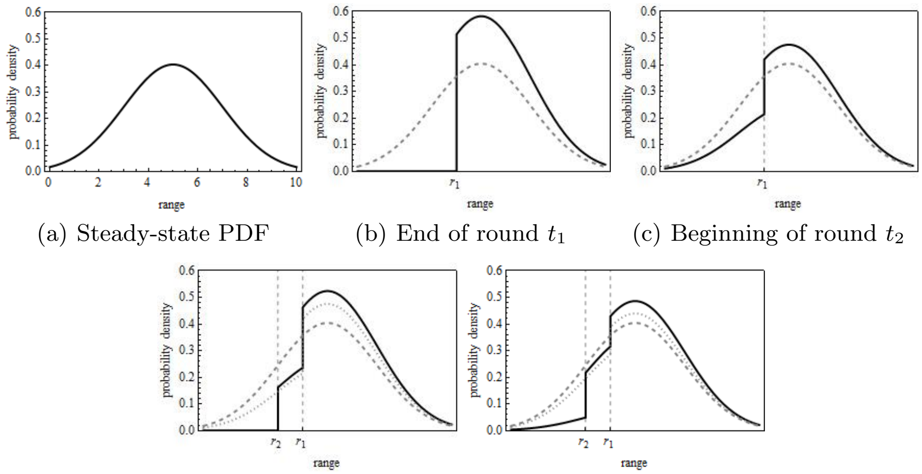
We begin with the following observation: In the instant after a search up to x' fails to find the target, the probability that the distance to the target, x, is less than or equal to x' is 0. Let $g(x,x',t)$ be the probability that the distance to the target is $x\leq x'$ t rounds after a search to x' fails to find the target, and let $G(x,x',t)$ be the cumulative function of $g(x,x',t)$. We will assume that $g(x,x',t)$ monotonically increases with t and that its value never exceeds $f(x')$. We also assume that the probability for all ranges $x\leq x'$ is $g(x,x',t)$, regardless of any searches conducted in previous rounds. That is, knowledge of the probability for all distances $x\leq x'$ is "reset" with a search to x'. In contrast, the probability at all points $x>x'$ is defined by the largest sequence of strictly descending search ranges used until that round.

We first describe our probability model by example before providing a formal description. Throughout the discussion, we will use Figure 1 to visualize the concepts. The figure depicts a continuous probability density function rather than a discrete function to aid in the visualization, but the same principles apply to the discrete case. Figure 1(a) shows the steady-state distribution in this example. In round t_1, the searcher searches up to range r_1. In the instant after the search, the probability that the distance to the target is $x\leq r_1$ is 0. Without any knowledge about the conditional probability for all remaining points, we assume that the probability that the distance to the target is $x>r_1$ is $\frac{1}{1-F(r_1)}f(x)$, according to Bayes' Theorem. See Figure 1(b). At the beginning of some round $t_2>t_1$, the probability at all points $x\leq r_1$ is $g(x,r_1,t_2-t_1)$, as defined above. For all points $x>r_1$, we want to multiply $f(x)$ by some constant C such that $G(r_1,r_1,t_2-t_1)+C(1-F(r_1))=1$, which gives us $C=\frac{1-G(r_1,r_1,t_2-t_1)}{1-F(r_1)}$. We therefore have:

$$f(x|r_1,\underbrace{0,\ldots,0}_{t_2-t_1-1})=\begin{cases}g(x,r_1,t_2-t_1) & x\leq r_1\\ \frac{1-G(r_1,r_1,t_2-t_1)}{1-F(r_1)}f(x) & x>r_1\end{cases} \tag{2}$$

See Figure 1(c). In round t_2, the searcher searches up to range $r_2<r_1$. The probability immediately after searching up to r_2 is updated as before. The probability at all points $x\leq r_2$ is 0, while the probability at all other points is multiplied by $\frac{1}{1-G(r_2,r_1,t_2-t_1)}$ according to Bayes' Theorem (Figure 1(d)).

At the beginning of some round $t_3>t_2$, the probability at all points $x\leq r_2$ is $g(x,r_2,t_3-t_2)$, as before. The probability at all points $x>r_2$ is calculated as



(a) Steady-state PDF (b) End of round t_1 (c) Beginning of round t_2

(d) End of round t_2 (e) Beginning of round t_3

Fig. 1. Changes in probability as elastic ring search progresses. The solid lines represent the beliefs at the time indicated by the caption; the dashed lines indicate the steady-state distribution; and the dotted lines in the last two figures represent what the probability would have been if no search was conducted up to r_2 in round t_2.

in the previous equation, except that it is multiplied by some constant C such that $G(r_2, r_2, t_3 - t_2) + C(1 - G(r_2, r_1, t_3 - t_1)) = 1$, which gives us $C = \frac{1-G(r_2,r_2,t_3-t_2)}{1-G(r_2,r_1,t_3-t_1)}$. We therefore have:

$$f(x|r_1, \underbrace{0,\ldots,0}_{t_2-t_1-1}, r_2, \underbrace{0,\ldots,0}_{t_3-t_2-1}) = \begin{cases} g(x, r_2, t_3 - t_2) & x \le r_2 \\ \frac{1-G(r_2,r_2,t_3-t_2)}{1-G(r_2,r_1,t_3-t_1)} g(x, r_1, t_3 - t_1) & r_2 < x \le r_1 \\ \frac{1-G(r_2,r_2,t_3-t_2)}{1-G(r_2,r_1,t_3-t_1)} \frac{1-G(r_1,r_1,t_3-t_1)}{1-F(r_1)} f(x) & x > r_1 \end{cases} \quad (3)$$

See Figure 1(e).

We now formally define the model. The probability at the beginning of any round t is defined by the largest sequence of strictly descending search ranges used until that round. Let $t_1, \ldots, t_l$ be the sequence of the rounds in which these ranges were used. t_i can be recursively defined as follows:

$$t_1 = \operatorname*{argmin}_{1 \le j < t}(\forall_{k>j} u_j > u_k) \qquad t_{i+1} = \operatorname*{argmin}_{t_i < j < t}(\forall_{k>j} u_j > u_k) \quad (4)$$

Let u'_i be the search range used at time t_i. The conditional probability density function is defined as follows:

$$f(x|u_1, \ldots, u_t)$$
$$= \begin{cases} g(x, u'_l, t - t_l) & \text{when } x \le u'_l \\ \prod_{j=i}^{l-1} \frac{1-G(u'_{j+1},u'_{j+1},t-t_{j+1})}{1-G(u'_{j+1},u'_j,t-t_j)} g(x, u'_i, t - t_i) & \text{when } u'_{i+1} < x \le u'_i,\ 1 \le i < l \\ \prod_{j=1}^{l-1} \frac{1-G(u'_{j+1},u'_{j+1},t-t_{j+1})}{1-G(u'_{j+1},u'_j,t-t_j)} \frac{1-G(u'_1,u'_1,t-t_1)}{1-F(u'_1)} f(x) & \text{when } x > u'_1 \end{cases} \quad (5)$$

4 Analysis

Based on the how the distribution changes and the cost associated with flooding the network to different extents, our goal is to find a sequence that minimizes (1), the expected cost of searching with an elastic ring search. We begin by establishing a necessary condition for a multi-round search to have a lower expected cost than full flooding. We then describe solutions for the best-case and worst-case scenarios, and finally provide two heuristics for the general case.

Theorem 1 establishes the intuitive notion that a search up to range x reduces the total cost only if the probability of success exceeds $C'(x)$. This extends a similar result in expanding ring search (Theorem 1 in [4]).

Theorem 1. *Given a search sequence $S = [u_1, \ldots, u_n]$ and any round $i < n$, let K_1 be the expected cost of searching with $u_{i+1}, \ldots, u_n$ from round $i+1$ onwards if the sequence $u_1, \ldots, u_i$ was used in the first i rounds, and K_2 be the expected cost if $u_1, \ldots, u_{i-1}, 0$ was used instead. Then $C'(u_i) + (1 - F(u_i|u_1, \ldots, u_{i-1}))K_1 < K_2$ implies $C'(u_i) < F(u_i)$.*

Proof. $K_1 \geq K_2$, since the probability of a successful search in round $i+1$ is no less than the probability of success if no search was conducted in round i. Assume that $C'(u_i) + (1 - F(u_i|u_1, \ldots, u_{i-1}))K_1 < K_2$ and $C'(u_i) \geq F(u_i)$. Then $C'(u_i) + (1 - F(u_i|u_1, \ldots, u_{i-1}))K_1 \geq F(u_i) + K_1 - F(u_i)K_1 = (1 - K_1)F(u_i) + K_1 > K_1 \geq K_2$, contradicting the assumption.

The following corollary establishes a necessary condition for a multi-round search to improve search costs.

Corollary 1. *If $C'(x) \geq F(x)$ for all x, then the optimal strategy is to search the entire space in the first round.*

Ideally, the searcher can assume that $f(x_i|u_1, \ldots, u_{j-1}) = f(x_i)$, for all i, meaning that the probability of finding a node up to range x_i is unconditional. The next result establishes that the optimal strategy in this case is an infinite sequence when there is no time constraint to the search.

Theorem 2. *If the probability distribution is the same each round (i.e., $f(x|u_1, \ldots, u_i) = f(x)$, $\forall i$) and there is no constraint on the number of rounds, then the infinite sequence $[x, x, \ldots]$, for some x, is an optimal strategy.*

Proof. Let $V(S)$ be the expected cost of applying strategy S. Assume that the optimal strategy of least length is the finite sequence $S_1 = [u_1, \ldots, u_n]$. Let $S_1' = [u_2, \ldots, u_n]$. By definition, $V(S_1) < V(S_1')$. If the target is not found in round one, then the the optimal strategy from round two onwards is $S_2 = [u_2, \ldots, u_n]$. Let $V'(S)$ be the cost of applying some strategy S after a search with u_1 failed to find the target in the first round. Let $S_2' = [u_1, \ldots, u_n]$, such that $V'(S_2) \leq V'(S_2')$. Since the probability distribution does not change, then the expected cost of applying any strategy in round two is unaffected by the search range used in round one. Therefore, $V(S_1) < V(S_1') = V'(S_2) \leq V'(S_2') = V(S_1)$, contradicting the assumption that S_1 is the optimal strategy. Rather, the optimal strategy is an infinite sequence S_{opt}. Since the searcher is faced with the same decision each round, the searcher can always choose the same search range.

According to this theorem, an optimal strategy under immediate convergence is to use the value of x that minimizes $\sum_{i=0}^{\infty}(1-F(x))^i C(x) = \frac{C(x)}{F(x)}$. When the search is limited to n rounds, the optimal strategy can be derived by solving the following recursive formula for the optimal expected cost $V(n)$ of an n round strategy using backward induction on $1 \leq i \leq n$:

$$V(n) = C(x_{max}) \qquad V(i) = \min_{1 \leq j \leq m} \{C(x_j) + (1 - F(x_j))V(i+1)\} \tag{6}$$

The worst-case scenario is when there is no return to the steady-state, such as when the network is static. This is precisely the condition when expanding ring search is optimal. The optimal TTL sequence for expanding ring search can be derived by solving the following dynamic programming equations for $0 \leq i \leq m$ using backwards induction [4]:

$$V(x_m) = 0 \qquad V(x_i) = \min_{i+1 \leq j \leq m} \{C(x_j) + (1 - F(x_j|x_i))V(x_j)\} \tag{7}$$

Here, $V(x_i)$ is the minimum expected cost-to-go, which is the cost of continuing the search, when a search using x_i fails to find the target. The first condition reflects the fact the search ends when searching up to x_m. The second condition reflects the fact that failing to find the target using x_i requires continuing the search with some TTL value $x_{j>i}$ that is larger than x_i. The cost-to-go in this case includes the cost $C(x_j)$ of searching up to x_j, plus the expected cost if the target is beyond x_j. This second value is the cost-to-go after searching with x_j multiplied by the probability $(1 - F(x_j|x_i))$ that the target is not within range x_j when it is already known that it is not within range x_i. The value of x_j that minimizes the cost-to-go for x_i is the one that is used to continue the search. $V(x_i)$ can be solved for backwards, for all $0 \leq i < m$. $V(x_0)$ reflects the expected cost of the optimal strategy. By recording the x_j chosen for each x_i, the optimal strategy can be extracted by following these links forwards from x_0.

It is not feasible to derive a similar solution for the general case because the cost-to-go at any round i depends on the entire subsequence $u_1 \dots u_{i-1}$, which has an exponential number of combinations. Instead, we define two heuristics that use nondecreasing rings. The first heuristic is based on the optimal strategy for optimal conditions. The idea is to search using the same TTL value at regular intervals until the last search round, at which time the entire network is flooded.

Heuristic 1 Choose the values of x and i, over all values of $x \in R$ and $i < n$, for which the strategy that uses x every i rounds has minimal cost.

Heuristic 1 takes advantage of node movement by not increasing the search extent, but does not guarantee a lower expected cost than expanding ring search. The second heuristic, the optimal nondecreasing ring search, is guaranteed to have an expected cost no greater than that of expanding ring search. While this is still not necessarily the optimal elastic ring search, it can be solved for in $O(n^2m^2)$ time and $O(nm)$ space using a dynamic programming formulation modeled after the formulation for expanding ring search.

Heuristic 2 Derive the optimal nondecreasing ring search with the following dynamic programming formulation, where $1 \leq \{j,k\} \leq m-1$, $0 \leq i \leq n$, and $0 \leq s < n$.

$$V(n,j) = \infty \qquad ; \qquad V(i,m) = 0$$

$$V(s,k) = \min_{\substack{s<t\leq n \\ k \leq l \leq m}} \left\{ C(x_l) + (1 - F(x_l|x_k, \underbrace{0, \dots, 0}_{t-s-1}))V(t,l) \right\} \tag{8}$$

In this formulation, $V(s,k)$ is the minimum expected cost-to-go when search range x_k is applied in round s, which is the expected remaining costs after round s. The cost of the optimal strategy is defined by $V(0,0)$. In the derived strategy, the probability of success using search range x_l in round t is affected by the use of x_k in round s and only by x_k, as indicated by the expression $F(x_l|x_k,\underbrace{0,\ldots,0}_{t-s-1}))$.

5 Evaluation

We chose the number of nodes that would transmit a route request to be the cost-per-hop in evaluating the two elastic ring search heuristics. For this purpose, it is only necessary to know the connectivity graph in each round of the search procedure. We used Java to simulate node movement, construct the connectivity graph at regular time steps, and construct the breadth-first search (BFS) tree of the connectivity graphs. The BFS tree is used to determine the cost-per-hop of a route request and the hop count to the destination count. For each TTL value less than or equal to the depth of the tree, the cost-per-hop is the number of nodes at all depths up to but not including that TTL value. This is because the last nodes to receive the query do not retransmit it. For any TTL value larger than the depth of the tree, the cost-per-hop is the number of nodes in the tree. Note that this is not necessarily equal to all nodes in the network, since in practice, not all nodes are necessarily connected to the source. The hop count to the destination is its depth in the BFS tree.

Throughout the discussion, we refer to the time steps as rounds. In an elastic ring search, each consecutive search round is conducted in consecutive rounds of movement. In an expanding ring search, all search rounds are conducted in the same movement round.

We simulated movement according to two settings, which we refer to as Scenario 1 and Scenario 2. In both settings, there are 250 nodes; the transmission range of each node is 100 meters; the field size is 1200x1200 meters; the source and destination are stationary throughout the simulation; and all other nodes move according to the random waypoint model with speeds randomly selected from the range 5 m/s to 8 m/s and a 2 second pause time between waypoints. The only differences between the settings are the (x,y) coordinates of the source and destination: In Scenario 1, their coordinates are $(200,600)$ and $(200,460)$, respectively, and in Scenario 2, their locations are $(200,600)$ and $(200,340)$, respectively. The time steps were set to 100ms. The simulations ran for a total of 50 seconds (500 rounds).

Next we show how we determined the steady-state properties, and then we provide the results of our evaluation.

5.1 Steady-State Properties

The steady-state properties of the random waypoint model have been well studied [2, 15], including the steady-state spatial node distribution [3] and hop count distribution [16]. Studies show that the nodes are concentrated in the center of the field in the steady-state node distribution, so the system is not immediately in the steady-state when nodes are initially assigned uniformly random locations. Our goal, therefore, is to establish when there is a steady-state to the hop count distribution and to derive the conditional probability distribution.

To demonstrate the existence of a steady-state, we used two statistical tests: the Kolmogorov-Smirnov test and the Wilcoxon Rank-sum test. Both are nonparametric tests, meaning they make no assumptions about the distribution of the data. They are used to test the null hypothesis that two sets of samples come from the identical distribution. That is, the data that we enter are samples drawn from larger distributions, and we are trying to determine whether the generating distributions are identical. The tests return a p-value, which is a number between 0 and 1 that answers the following question: if our null hypothesis was correct and these distributions are identical, what is the probability that we would draw samples that differ this greatly? Thus, a low p-value means that it would be very unlikely to obtain these two samples from one distribution. A high p-value is consistent with the assertion that the samples were drawn from identical distributions. A high p-value is not a proof that the distributions are identical, but rather it tells us that we cannot reject the null hypothesis that they are identical. A sufficiently low p-value, on the other hand, would serve as a basis for rejecting the null-hypothesis and concluding with relative certainty that the samples were not drawn from identical distributions.

We use these tests to show the progression of the p-values when comparing the distribution from each round to the steady-state distribution. We took the hop counts from rounds 3900-4000 over 1000 different simulations as representatives of the steady-state. We compared the samples at rounds $10, 20, \ldots$ from 500 simulations (unique to each round) to the steady-state using the two tests and checked if and by what round they attain and sustain consistently high p-values. Our results showed that after round 1000, the p-values are mostly over 0.3. It is noted that typically the assumption of having the two samples derived from the same distribution is rejected for p=0.05. This leads us to believe that the system enters a steady-state after round 1000 in this case.

We derived the conditional probability function for Scenario 1 as follows. We recorded the hop count every round from round 1000 to round 1499 for 5,000 randomly initialized simulations. The hop counts were in the range $[2, 28]$, but the largest portion of samples were either 2 or 3. We then selected the data from the simulations for which the hop count was > 3 in round 1000, which is the case when a search in round 1000 with a TTL=3 would have failed to find the target. There were 632 such instances.

Figure 2 shows the number of samples equal to 2, 3, 4, and 5 hops each round. Each plot is close to a horizontal line, supporting the claim that the hop count distribution is in a steady-state. Figure 3 shows the number of samples equal to 2, 3, 4, or 5 hops each round in these instances. Here it is clear that the distribution converges back to the steady-state, in about 180 rounds, with the number of samples equal to 2 and 3 increasing each round and the number of samples equal to 4 and 5 decreasing.

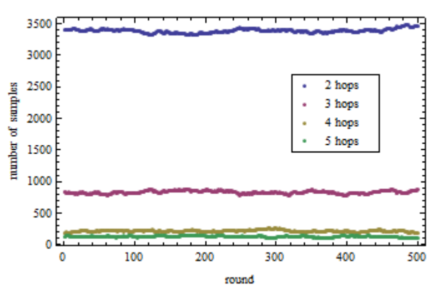


Fig. 2. Number of samples each round (Scenario 1)

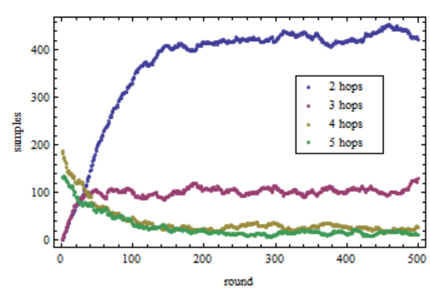


Fig. 3. Number of samples each round when the hop count > 3 in the first search round (Scenario 1)

We normalized the histogram of hop counts from this subset each round to 1, and then divided each bin by the corresponding bin in the normalized histogram of all samples. This gives us the coefficient by which $f(x)$, $x \in 2, 3$, would need to be multiplied at round $1000 + t$ to give us the conditional probability $g(x, 3, t)$ if the two histograms mentioned above were to correspond to $g(x, 3, t)$ and $f(x)$, respectively. We found the least-squares fit of each sequence of fractions to the function $\sum_{i=0}^{5} x^i$. Let $g'(t)$ be this function. We define $g(x, x', t) = g'(t)f(x)$ for all combinations of (x, x') in $g(x, x', t)$. Figure 4 shows the histogram of all samples normalized to 1. Figure 5 shows the sequence of fractions and the fitted curves that were used to define $g(x, x', t)$.

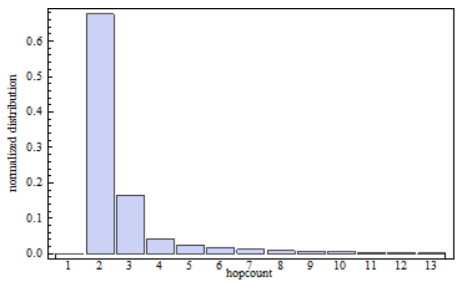


Fig. 4. Normalized histogram of hop count samples (Scenario 1)

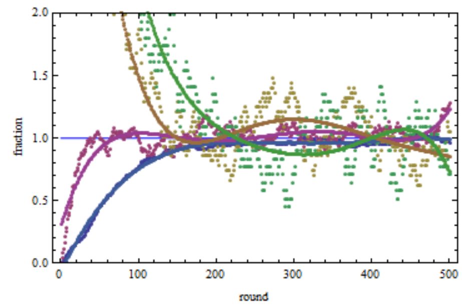


Fig. 5. Functions fitted to fractional conditional distributions (Scenario 1)

We followed the same procedure for Scenario 2. In this case, the range of hop counts in the distribution is $[3, 13]$, with most samples falling the range $[3, 7]$. We considered the conditional probability when the hop count was > 4 and set $g(x, x', t)$ to the best fit of conditional probability of hop count 4 to the function $1 + \log t$. The time to converge in this case was longer than the previous case, about about 300 rounds.

5.2 Results

For both Scenarios 1 and 2, we constructed the hop count distribution using samples from round 1000 from 500 simulations. We constructed the cost function in a similar manner, by counting the number of nodes at each level of the BFS tree in round 1000 over 500 simulations and taking the average. Using these distributions and cost functions, we derived the optimal expanding ring search, the expected cost of an infinite elastic ring search under optimal conditions, and the $50, 100, \ldots, 500$ round strategies using Heuristics 1 and 2. We simulated the use of these heuristics in 500 simulations for each sequence length, beginning from round 1000 in Scenario 1 and round 1500 in Scenario 2.

Figure 6 is a plot of the average cost of using expanding ring search, Heuristic 1, and Heuristic 2. Additionally, the expected cost under optimal conditions, according to Theorem 2, is included as a lower bound on expected costs. We can see from the figure that Heuristic 1 performs better than expanding ring search when it is allowed more than 50 rounds; Heuristic 2 always performs better than expanding ring search; Heuristic 2 performs slightly better than Heuristic 1; and both converge to the lower bound. Additionally, the average costs are very close to the expected costs (not shown), which shows that the estimation of $g(x, x', t)$ was adequate despite the fact that it did not account for all cases.

Figure 7 is a plot of the average costs of the different strategies. Here, Heuristics 1 and 2 are close in performance, but both are significantly better than expanding ring search, even when limited to 50 rounds. In this case, the potential difference between using expanding ring search and elastic ring search is significant. The average cost of expanding ring search is about 95, while the lower bound is about 55, which is about an additional 16% reduction in costs. Both heuristics approach this lower bound when allowed 500 rounds.

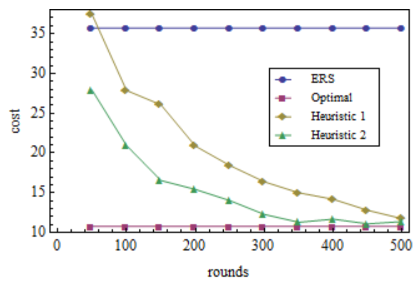


Fig. 6. Average costs of different search strategies in Scenario 1 (1 round=100ms)

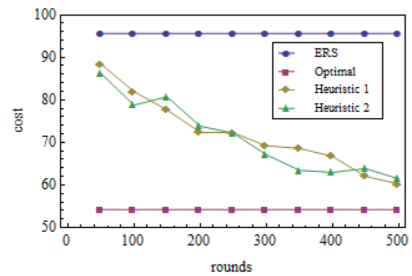


Fig. 7. Average costs of different search strategies in Scenario 2 (1 round=100ms)

6 Conclusion

As illustrated throughout the paper, when considering dynamic settings, the searcher can benefit substantially by deviating from the optimal expanding ring search to an elastic sequence. The frequent changes in the network topology are inherent in the problem definition, and the tolerance of delay characterizes several important applications (e.g., text messaging, emails, data collection). The new method takes advantage of the potential formation of new, potentially shorter, routes along time. In such cases, it is useful to delay search for some time (e.g., not to search in some of the rounds) as well as to search in reduced extents from time to time.

While the computational complexity of the dynamic programming approach presented in this paper for extracting the optimal ELRS sequence is substantial, efficient sequences can be extracted in polynomial time. Two such heuristics are given in this paper. The first repeats the same TTL value at regular intervals, while the second uses a non-decreasing sequence. These heuristics can be derived in polynomial run-time and their performance, as illustrated experimentally, converges to a lower bound on optimal costs. We believe that these solutions are ideal for ad-hoc networks terminals that are inherently bounded by their power supply.

The key barrier to implementing elastic ring search in practice is learning the steady state on the device. Notice that most of the work that we performed on the distribution was not on getting a candidate steady state. To do that, we ran 1000 simulations of the system, which is not computationally intensive. All that is needed to generate a candidate steady state are a few parameters: the area of the search, the number of nodes, and the broadcast capabilities of the devices being used. With this information, each participant can generate the steady-state independently on their own device, or it can be generated when setting up the network. Another approach is to create a database of pre-generated steady-states and choose a best match according to the parameters above. As a data-driven algorithm, increasing the size of the database will increase the precision of the matches and thus increase the performance of ELRS using that data.

With the constantly decreasing size and price of storage, it is certainly practical to store a very large database on the devices.

There are a number of possible ways to practically implement ELRS, and further research will likely result in better heuristics that are faster to compute and incur a lower expected cost. The key point though is that there is fortunately a large discrepancy between the amount of work necessary to convincingly demonstrate that the network is in a steady-state and the amount of work necessary to generate a candidate steady state. The latter is computationally inexpensive and is all that is necessary to implement ELRS in practice.

References

1. Y. M. Baryshnikov, E. G. Coffman Jr., P. R. Jelenkovic, P. Momcilovic, and D. Rubenstein. Flood search under the california split rule. *Oper. Res. Lett.*, 32(3):199–206, 2004.
2. C. Bettstetter, H. Hartenstein, and X. P. Costa. Stochastic properties of the random waypoint mobility model. *Wireless Networks*, 10(5):555–567, 2004.
3. C. Bettstetter, G. Resta, and P. Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Trans. Mob. Comput.*, 2(3):257–269, 2003.
4. N. B. Chang and M. Liu. Revisiting the TTL-based controlled flooding search: optimality and randomization. In Z. J. Haas, S. R. Das, and R. Jain, editors, *MOBICOM*, pages 85–99. ACM, 2004.
5. N. B. Chang and M. Liu. Controlled flooding search in a large network. *IEEE/ACM Trans. Netw.*, 15(2):436–449, 2007.
6. Z. Cheng and W. B. Heinzelman. Searching strategies for target discovery in wireless networks. *Ad Hoc Networks*, 5(4):413–428, 2007.
7. J. Deng and S. A. Zuyev. On search sets of expanding ring search in wireless networks. *Ad Hoc Networks*, 6(7):1168–1181, 2008.
8. J. Hassan and S. Jha. On the optimization trade-offs of expanding ring search. In N. Das, A. Sen, S. K. Das, and B. P. Sinha, editors, *IWDC*, volume 3326 of *Lecture Notes in Computer Science*, pages 489–494. Springer, 2004.
9. S.-J. Lee, E. M. Belding-Royer, and C. E. Perkins. Scalability study of the ad hoc on-demand distance vector routing protocol. *Int. Journal of Network Management*, 13(2):97–114, 2003.
10. S. Mukherjee and D. Avidor. On the probability distribution of the minimal number of hops between any pair of nodes in a bounded wireless ad-hoc network subject to fading. In *IWWAN*, 2005.
11. I. Park and I. Pu. Energy efficient expanding ring search. In *Asia International Conference on Modelling and Simulation*, pages 198–199. IEEE Computer Society, 2007.
12. I. Pu and Y. Shen. Enhanced blocking expanding ring search in mobile ad hoc networks. In *New Technologies, Mobility and Security (NTMS), 2009 3rd International Conference on*, pages 1 –5, dec. 2009.
13. I. Pu and Y. Shen. Analytical studies of energytime efficiency of blocking expanding ring search. *Mathematics in Computer Science*, 3:443–456, 2010.
14. E. M. Royer and C. k Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6:46–55, 1999.
15. J. Yoon, M. Liu, and B. Noble. Sound mobility models. In *MOBICOM*, pages 205–216, 2003.
16. O. Younes and N. Thomas. Analysis of the expected number of hops in mobile ad hoc networks with random waypoint mobility. *Electr. Notes Theor. Comput. Sci.*, 275:143–158, 2011.