Revisiting the TTL-based Controlled Flooding Search: Optimality and Randomization

Nicholas Chang
changn@eecs.umich.edu

Mingyan Liu*
mingyan@eecs.umich.edu

Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109-2122

ABSTRACT

In this paper we consider the problem of searching for a node or an object (i.e., piece of data, file, etc.) in a large network. Applications of this problem include searching for a destination node in a mobile ad hoc network, querying for a piece of desired data in a wireless sensor network, and searching for a shared file in an unstructured peer-to-peer network. We limit our attention in this study to the class of controlled flooding search strategies where query/search packets are broadcast and propagated in the network until a preset TTL (time-to-live) value carried in the packet expires. Every unsuccessful search attempt results in an increased TTL value (i.e., larger search area) and the same process is repeated. The primary goal of this study is to derive search strategies (i.e., sequences of TTL values) that will minimize the cost of such searches associated with packet transmissions. The main results of this paper are as follows. When the probability distribution of the location of the object is known *a priori*, we present a dynamic programming formulation with which optimal search strategies can be derived that minimize the expected search cost. We also derive the necessary and sufficient conditions for two very commonly used search strategies to be optimal. When the probability distribution of the location of the object is not known *a priori* and the object is to minimize the worst-case search cost, we show that the best strategies are *randomized* strategies, i.e., successive TTL values are chosen from certain probability distributions rather than deterministic values. We show that given any deterministic TTL sequence, there exists a randomized version that has a lower worst-case expected search cost. We also derive an asymptotically (as the network size increases) optimal strategy within a class of randomized strategies.

*This work is supported by NSF grants ANI-0112801, ANI-0238035 and CCR-0329715.

MobiCom'04, Sept. 26-Oct. 1, 2004, Philadelphia, Pennsylvania, USA.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]

General Terms

Algorithms, Design, Performance, Theory

Keywords

data query, file search, TTL, controlled flooding search, wireless sensor networks, dynamic programming, optimal search strategy, fixed strategy, randomized strategy, best worst-case performance, competitive ratio

1. INTRODUCTION

In this paper we consider the problem of searching for a node or an object (e.g., piece of data, file, etc.) in a large network. The ability to conduct cost effective and fast searches has become an increasingly critical component required by many emerging networks and their applications.

A prime example is data query in a large wireless sensor network, where different data is distributed among a large number of sensor nodes based on different sensor readings. A query may be initiated by any node in search of certain data of interest (e.g., the position coordinates where temperature has exceeded a certain level) [1]. As it is not known *a priori* where the data might be located, or which node has the data, the query has to be somehow advertised to nodes in the network. As the query propagates, a node that has the data that matches the interest will respond to the querying node with the desired data [2]. There may be more than one node in the network (or sometimes none) that has the queried data. Depending on the underlying application, we may need to locate one, some, or all of these nodes. Search has also been extensively used in mobile ad hoc networks. This includes searching for a destination node by a source node in the route establishment procedures of an ad hoc routing protocol (e.g., [3]), searching for a multicast group by a node looking to join the group (e.g., [4]), and locating one or multiple servers by a node requesting distributed services (e.g., [5]). Search is also widely used in peer-to-peer (P2P) networks, marked by the need to locate desired objects/files that are shared among nodes in the network.

A good search mechanism should have a short response time, i.e. the time it takes to find the object/data of interest, and should do so with minimal cost. In ad hoc and sensor networks cost refers to the energy consumption incurred by the search, and will be measured by the amount of packet transmissions involved. For these networks low cost is crucial if due to the stringent energy constraint.

There are a variety of mechanisms one may use to search or locate a node/object in a large network. The first is to maintain a centralized directory service, where nodes issue queries to the central directory to obtain the location of the search target. The central directory needs to be constantly updated as the network topology and data content change. Such systems tend to have very short response time, if the directory information is kept afresh. On the other hand, centralized systems often scale poorly as the network increases in size, and as location information changes more frequently (either due to topology change as a result of mobility or due to the information content change in the network). The latter necessitates a large amount of information update which can cause significant energy consumption overhead, especially when the queries occur less frequent compared to changes in the network.

A second class of methods, which is decentralized, is the random walk based search, where the querier sends out a query packet which is forwarded in a random walk fashion, until it hits the search target. These can be pure random walks or controlled walks such that the propagation of the packet is maintained in an approximately consistent direction. In particular, [2] proposed random walks initiated by both the querier and the node that has data of potential interest (called advertisement). There have been many results on estimating the search cost and response time using such approaches, see for example [6].

In this paper we will take a fresh look at another very widely used search mechanism that uses TTL-based controlled flooding of query packets. This method is widely used in ad hoc routing protocols [7], as well as wired networks [8]. This is also a decentralized approach in that no central directory of information is maintained. Under this scheme the query/search packet is broadcast and propagated in the network. A preset TTL (time-to-live) value is carried in the packet and every time the packet is relayed the TTL value is decremented. This continues until TTL reaches zero when the propagation stops. Therefore the extent/area of the search is controlled by the TTL value. If the target is located within this area, the corresponding node will reply with the queried information. Otherwise, the origin of the search will eventually time out and initiate another round of search covering a bigger area using a larger TTL value. This continues until either the object is found or the querier gives up. Consequently the performance of a search strategy is determined by the sequence of TTL values used. Compared to random walk based approaches, controlled flooding search is much easier to implement, and is likely to result in shorter response times on average.

In this study we will limit our attention to the class of TTL based approaches and will not attempt to make quantitative comparison between this and other mechanisms. Our primary goal is to derive, within this class, search strategies (i.e., sequences of TTL values) that will minimize the cost of such searches, with the intention that they can be applied to wired and wireless networks alike, static or mobile, although decentralized and unstructured searches are more relevant in a wireless scenario, particularly in a wireless sensor network. We will not consider the response time of a search strategy in this paper. One reason is that within the class of controlled flooding search, the fastest search is to flood the entire network. In addition, since the search cost is a function of the number of packet transmissions and receptions, the goal of minimizing cost is generally aligned with the objective of locating the object quickly.

For the rest of our discussion we will use the term *object* to indicate the target of a search, be it a node, a piece of data or a file. We will measure the position of an object by its distance to the source originating the searching. We will use the term *location* of an object to indicate both the actual position of the search target within the network and the minimum TTL value required to locate this object. The terms *search strategy* or simply *strategy* will take on a more limited meaning within the context of controlled flooding search and refer to a TTL sequence.

Main results of this paper are summarized as follows.

1. When the probability distribution of the location of the object is known *a priori*, we present a dynamic programming formulation with which optimal search strategies, i.e., the optimal sequence of TTL values that minimize the expected search cost, can be derived.

2. We derive the necessary and sufficient conditions on the location distribution under which two of the very common search strategies are optimal. The first is the complete flooding of the network (i.e., the use of a single TTL value that covers the entire network); the second is a special case of the expanding ring search where TTL values are incremented by one every time a new search is initiated.

3. When the probability distribution of the location of the object is not known *a priori* and adopting the objective of minimizing the worst-case search cost, we show that the best strategies are *randomized* strategies that consist of sequence of random variables, i.e., successive TTL values are drawn from certain probability distributions rather than being deterministic values. To the best of our knowledge randomized strategies have not been proposed or studied in this context before. We show that given any deterministic TTL sequence, there exists a randomized version that has a lower worst-case expected search cost. The construction of the randomization is presented.

4. For the best worst-case performance measure, we also derive an asymptotically (as the network size increases) optimal strategy from a class of randomized strategies under a linear cost assumption.

The rest of the paper is organized as follows. In Section 2 we present the network model, assumptions on the search cost function, and the two performance objectives. In Section 3 we present the dynamic programming formulation and derive optimal search strategies for the case when distribution of the location is known *a priori*. Section 4 introduces the class of randomized strategies and examines their performance when the distribution of the object is not known *a priori*. Section 5 provides a discussion and some simulation results on a number of search strategies. Section 6 concludes the paper.

2. PROBLEM FORMULATION

In this section, we present the network model and introduce the cost measures and performance objectives of search strategies.

2.1 Network Model and Search Strategy

Within the context of the TTL-based controlled flooding search, the distance between two nodes is measured in number of *hops*, assuming that the network is connected. Two nodes being one hop away means they can reach each other in one transmission. In particular, in a wireless scenario each transmission covers a specific region given the limitation on the transmission power, channel fading, etc. All nodes within that region will be considered one-hop neighbors of the transmitting node.

As described in the previous section, the node originating the search begins by determining an initial positive integer TTL value, and passes this number along with its search query to its neighboring nodes. If the underlying network is wired, this query will be transmitted once along each outgoing link of the originating node. For a wireless network, the originating node can reach all its neighbors in a single broadcast transmission. If the object is found at a neighboring node, then the corresponding node will reply to the originating node. If a neighboring node does not have the desired object, it will decrement the TTL value by one and pass the query to its neighbors in the same fashion. In this way the query packets are duplicated and propagated in the network.

The above process repeats until either the object has been located or the TTL value reaches 0, at which point the query packet is dropped. The originating node starts a timer when the first query packet is sent. If it does not get a response back before the timer expires, it will begin a new *round* of search by selecting a strictly larger TTL value, and the above procedure is repeated. The TTL value is increased in subsequent rounds until the object is located.

In a practical system, a variety of techniques may be used to reduce the number of query packets flowing in the network and to alleviate the *broadcast storm* problem [9]. For example, a node should suppress multiple copies of the same query it receives. In our analysis we will ignore these technical details, and simply assume that a search with a TTL value of k will reach all neighbors that are k hops away from the originating node, and that the cost associated with this search is a function of k, denoted by C_k. This cost may include the total number of transmissions, receptions, etc. Thus C_k is the ultimate abstraction of the nature of the underlying network and the specific broadcast schemes used. For the rest of our discussion we will no longer regard network as wired or wireless, but only discuss in terms of the search cost C_k, since in essence it abstracts the relevant features of lower layers.

We summarize the assumptions underlying our network model as follows.

1. We assume that a single target object exists in the network. Therefore flooding the entire network will for sure locate the object. The approach used and the results derived for this problem can be extended to searching for multiple targets (all or some of them, as is the case with service replication or distributed caching) by considering the joint distribution of these objects.
2. We do not explicitly model the channel interference and packet collision in the network, and simply assume that a TTL value of k will reach all nodes within k hops of the originating node. This essentially assumes that the redundancy inherent in the query broadcast process ensures that a node receives correctly at least one copy of the same query.
3. A search with TTL value of k incurs a cost C_k. The functional form of this cost depends on the properties of the network as well as the underlying broadcast techniques mentioned earlier.
4. We will assume that the timeout values are perfectly set such that when the timer expires for a query with TTL value k, that query has reached all nodes k hops away. Put in another way, we are assuming that there is no excessive delay in the network, thus a timeout event is equivalent to not finding the object in the k-hop neighborhood.

Assumptions (2) and (4) are obvious simplifications, which nevertheless allow us to reveal fundamental features of the problem and obtain insights. We will discuss relaxing these assumptions in Section 5.

We denote by L the minimum TTL value required to search every node within the network, and will also refer to L as the *dimension* or *size* of the network. Since we have assumed that the object exists, using a TTL value of L will locate the object with probability 1.

We will use X to denote the minimum TTL value required to locate the object. We will also loosely refer to X as the object "location". Note that X is an integer-valued random variable taking values between 1 and L such that $Pr(X \in \{1, 2, \cdots, L-1, L\}) = 1$. We denote the cumulative distribution of X by $F(k)$. By definition $F(k) = Pr(X \leq k)$. Similarly, the tail distribution of X is denoted by $\bar{F}(k)$, so that $\bar{F}(k) = 1 - F(k) = Pr(X > k)$. Note that $F(L) = 1$ and $\bar{F}(L) = 0$ for any X.

Note that we are not making any explicit assumptions on the distribution of the node deployed in the network (e.g., uniformly distributed). This is because such information is implied in $F(k)$.

For a given search strategy, we will denote by u_i the TTL value used during the i-th round, and let $\mathbf{u} = [u_1, u_2, \cdots, u_N]$ be a vector denoting the increasing sequence of N TTL (integer) values. The N-tuple $\mathbf{u}$ represents a specific search strategy. For any sensible strategy, we must have $u_i < u_{i+1}$, for all $1 \leq i \leq N-1$. Note that in a specific search experiment we may not need to use the entire sequence. However, in order to guarantee that the strategy $\mathbf{u}$ will locate the object with probability 1, it must be true that $u_N = L$. Also note that the value of N can vary between different policies.

2.2 Search Costs

As discussed earlier we will associate a round of search with TTL value k with a search cost C_k. It is important to note that in general, a node receiving the search query on the same round that the object is found in another node will be unaware that the desired object has just been located. (The only exception is perhaps a linear network where each transmission reaches only one neighboring node.) Consequently, this node will continue decrementing its TTL value and passing the search query to more neighbors. We can therefore assume that *search costs are paid in advance*; that is, the search cost for each round is determined by the TTL value and not by whether the object is located in that round.

We will adopt the natural assumption that $C_j > C_l$ if $j > l$, i.e., the cost increases as the search covers a bigger region. For the analysis in Section 3 this is the only assumption we need.

In Section 4 we will further investigate two specific types of cost functions, a linear cost and a quadratic cost. The first refers to a cost of the type

$$C_k = \alpha \cdot k \ ,$$

for some constant $\alpha > 0$, i.e., the cost is proportional to the TTL values used. This is a good model in a network where the number of transmissions incurred by the search query is proportional to the TTL value used, e.g., in a linear network with constant node density.

The quadratic cost refers to the type

$$C_k = \alpha \cdot k^2$$

for some constant $\alpha > 0$. This is a more reasonable assumption for a two dimensional network, as the number of nodes reached in k hops is on the order of k^2. In particular, in a wireless network with uniformly distributed nodes (thus the number of nodes in an area is proportional to the size of the area), one can show the number of transmissions incurred by the search is on the same order. Changing the retransmission/rebroadcast scheme may result in a different constant, but with the same order. In Section 5 we present simulated cost as a function of k. It was mentioned in [8] that the number of transmissions incurred with a TTL value of k is roughly $k+\beta k^2$, where β is some constant depending on the network parameters. Here we ignore the linear term and concentrate only on the quadratic part.

In Section 4.6 we will establish a mapping between the simple linear cost and more general cost functions.

2.3 Performance Measures

We will adopt two performance measures in this study. When the distribution of the object location X is known *a priori*, our goal is to find search strategies that will minimize the *expected search cost*, given that distribution. We will refer to this as the *average cost* measure or performance objective, and the corresponding strategies *optimal average cost strategies*. These will be precisely defined in Section 3.

When the distribution of the object location X is not known *a priori*, our goal is to find search strategies that will minimize the search cost in the worst case scenario, i.e., min-max strategies. We will refer to this as the *best worst-case* measure, and the corresponding strategies *best worst-case strategies*. This measure as well as the definition of a worst-case scenario will be precisely defined in Section 4.

All proofs are provided either in the text or in the appendix.

3. OPTIMAL AVERAGE COST STRATEGIES

In this section we consider strategies that minimize the expected search cost when the probability distribution of the object location X is known a priori. We first present a dynamic programming formulation with which optimal strategies may be obtained. We then derive conditions on the distribution of X under which two very commonly used TTL sequences are optimal.

3.1 A Dynamic Programming Formulation

Consider object location X with a tail distribution $\bar{F}(k)$, where $1 \leq k \leq L$, and a search strategy with TTL values $\mathbf{u} = [u_1, u_2, \cdots, u_N]$. The total expected search cost using strategy $\mathbf{u}$ is given by

$$J_X^{\mathbf{u}} = \sum_{i=1}^{N_u} C_{u_i} Pr(X > u_{i-1}) = \sum_{i=1}^{N_u} C_{u_i} \bar{F}(u_{i-1}) \ , \quad (1)$$

where N_u is the number of elements in the vector $\mathbf{u}$, C_{u_i} is the cost of searching with TTL value u_i, and $u_0 = 0$ is assumed. The search policy that minimizes this cost, denoted by $\mathbf{u}^*$, is thus

$$\mathbf{u}^* = \operatorname*{argmin}_{\mathbf{u}\in U} J_X^{\mathbf{u}} = \operatorname*{argmin}_{\mathbf{u}\in U} \sum_{i=1}^{N_u} C_{u_i} \bar{F}(u_{i-1}) \ , \quad (2)$$

where U denotes the set of all admissible search strategies (TTL sequences), i.e., all vectors $\mathbf{u}$ such that $u_i < u_{i+1}$ for all 1 ≤i≤ $N-1$ and $u_N = L$ as explained in Section 2.

This minimization can be solved backward in time using standard dynamic programming techniques [10]. Specifically, we use the most recently used TTL value, denoted by n, as the information state. For convenience we will denote by $\bar{F}(j|n)$ the conditional tail distribution of the object given that the most recently used TTL value n did not locate the object, i.e.,

$$\begin{aligned} \bar{F}(j|n) &= Pr(X > j | X > n) \quad (3) \\ &= \begin{cases} 1 & 1 \leq j \leq n \\ \bar{F}(j)/\bar{F}(n) & n+1 \leq j \leq L \end{cases} \end{aligned}$$

We then obtain the following dynamic programming equations that can be solved recursively for $0 \leq n \leq L-1$:

$$\begin{aligned} V(L) &= 0 \quad (4) \\ V(n) &= \min_{n+1 \leq l \leq L} \left\{ C_l + \bar{F}(l|n) V(l) \right\} \ , \quad (5) \end{aligned}$$

where the value function $V(n)$ is the minimum expected cost-to-go (over all choices of TTL values), given that the most recently used TTL value n did not locate the object.

The initial condition (4) reflects the fact that using a TTL value of L ensures finding the object and thus there would be no more remaining cost. Equation (5) follows from the fact that after unsuccessfully searching with a TTL value of n, the remaining choices for TTL values are the integers from $n+1$ to L. Any such choice l incurs an immediate search cost C_l plus an expected future cost if the object is not located using l. Note that because $\bar{F}(L|n) = 0$ for any value of n, $V(L-1) = C_L$ for any search strategy from (5). This agrees with the fact that if searching with a TTL value of $L-1$ is unsuccessful, then the only remaining option is to search with a TTL value of L.

Solving this set of equations backward we can obtain $V(n)$ for all n and determine the optimal TTL sequence $\mathbf{u}^*$. Finally $V(0)$ is the optimal (minimum) total expected search cost $\min_{\mathbf{u}\in U} J_X^{\mathbf{u}}$.

As an example, consider the special case where X is uniformly (discrete) distributed between 1 and L on a linear network. Therefore, $\bar{F}(l) = \frac{L-l}{L}$ for $1 \leq l \leq L-1$, and

$\bar{F}(l|n) = \frac{L-l}{L-n}$ for $n \le l \le L-1$. Further assume that the search cost is linear, i.e., $C_k = \alpha k$ for TTL value k and some constant α. We can then calculate $V(L-2)$ as follows (noting $L \ge 2$):

$$\begin{aligned} V(L-2) &= \min_{L-1 \le l \le L} \left\{ C_l + \bar{F}(l|L-2)V(l) \right\} \\ &= \min \left\{ C_{L-1} + \frac{C_L}{2}, C_L \right\} \\ &= C_L \cdot \min \left\{ \frac{L-1}{L} + \frac{1}{2}, 1 \right\} = C_L \end{aligned} \quad (6)$$

Repeating the above calculation, we can easily show that $V(n) = C_L$ for $1 \le n \le L-1$, meaning that if using a TTL value of n fails to locate the object, then it is optimal to next use a TTL value of L. Consequently the minimum total expected cost is

$$\begin{aligned} V(0) &= \min_{1 \le l \le L} \left\{ C_l + \bar{F}(l)V(l) \right\} \\ &= \min_{1 \le l \le L} \left\{ C_l + \frac{L-l}{L} C_L \right\} \\ &= C_L \cdot \min_{1 \le l \le L} \left\{ \frac{l}{L} + \frac{L-l}{L} \right\} = C_L \end{aligned} \quad (7)$$

Therefore, the optimal search cost when X is uniformly distributed with linear search cost is C_L. From Eqn. (7) we see that this minimum can be obtained by either using an initial TTL value of L so that $\mathbf{u} = [L]$, or by using $\mathbf{u} = [u_1, L]$ where u_1 is any integer such that $1 \le u_1 \le L-1$ (they are equally optimal).

This set of equations can be similarly solved for general cost functions and location distributions to obtain the optimal search strategy as well as the optimal total expected search cost. Unfortunately, optimal strategies cannot in general be qualitatively described without referring to specific numerical computation and therefore will not be discussed further in this paper (the above linear network with uniformly distributed location example is a nice exception).

Instead we will examine two very commonly used search strategies and consider the reverse problem, i.e., under what conditions are these strategies optimal.

3.2 Preliminaries

We derive some basic properties of the value function $V(n)$ that will be useful in later sections.

Proposition 1. *For any object location X with tail distribution $\bar{F}$, the value function $V(n)$ is a nondecreasing function of the information state n, i.e.,*

$$V(l) \le V(k) \quad \text{if} \quad l < k \ . \quad (8)$$

Proof: From (3) we have that $l < k$ implies $\bar{F}(j|l) \le \bar{F}(j|k)$ for all j. Thus

$$\begin{aligned} V(l) &= \min_{l+1 \le j \le L} \left\{ C_j + \bar{F}(j|l)V(j) \right\} \\ &\le \min_{l+1 \le j \le L} \left\{ C_j + \bar{F}(j|k)V(j)) \right\} \\ &\le \min_{k+1 \le j \le L} \left\{ C_j + \bar{F}(j|k)V(j)) \right\} = V(k) \end{aligned} \quad (9)$$

proving the proposition. □

This result implies that after each unsuccessful round of search, the expected cost-to-go increases. The increase is due to the fact that after each unsuccessful search, it is more likely that the object is located far away, thus requiring a larger search cost. Using Proposition 1 we can obtain the following result.

Proposition 2. *If $V(l) = C_L$ for some value of l, where $1 \le l \le L-2$, then $V(k) = C_L$ for all $l+1 \le k \le L-1$.*

Proof: If $V(l) = C_L$, then from Proposition 1 we have that $V(k) \ge V(l) = C_L$ for $k > l$. However, from equation (5) we also have

$$\begin{aligned} V(k) &= \min_{k+1 \le j \le L} \left\{ C_l + \bar{F}(j|k)V(j) \right\} \\ &= \min \left\{ \min_{k+1 \le j \le L-1} \left\{ C_l + \bar{F}(j|k)V(j) \right\}, C_L \right\} \\ &\le C_L \ . \end{aligned} \quad (10)$$

Thus $V(k) \ge C_L$ and $V(k) \le C_L$, implying $V(k) = C_L$. □

This result implies that if it is optimal to use a TTL value of L after having searched unsuccessfully with a TTL value of j, then it is also optimal to use a TTL value of L after an unsuccessful search with a TTL value of k, for any $k > j$.

In the next subsection we consider two specific search strategies, namely the broadcast flooding and the expanding ring search.

3.3 Broadcast Flooding and Expanding Ring Search

Under the *broadcast flooding* search scheme, all nodes are searched on the first round, i.e., $u_1 = L$. Broadcast flooding always locates the object on the first search and thus incurs a fixed search cost of C_L.

Note that if broadcast flooding is the optimal average-cost strategy, then $V(0) = C_L$. In Section 3.1, we showed that broadcast flooding is the optimal strategy to employ when the object location is uniformly distributed on a linear network. A more general result is as follows, by defining the *normalized cost* $\hat{C}_k$ as

$$\hat{C}_k = \frac{C_k}{C_L}, \quad 1 \le k \le L \ . \quad (11)$$

Theorem 1. *For any value of n where $0 \le n \le L-2$, $V(n) = C_L$* iff *$\bar{F}(k|n) \ge 1 - \hat{C}_k = \frac{C_L - C_k}{C_L}$ for all k such that $1 \le k \le L-1$.*

Theorem 1 implies the following corollary.

Corollary 1. *Broadcast flooding is the optimal strategy* iff *$\bar{F}(k) \ge 1 - \hat{C}_k$ for all $1 \le k \le L-1$. Equivalently, broadcast flooding is optimal* iff *$F(k) \le \hat{C}_k$ for all $1 \le k \le L-1$.*

Corollary 1 is a special case of Theorem 1 with $n = 0$. These results reflect the following intuition: broadcast flooding search is optimal if and only if the "return" (chances of finding the object) is at least the "cost" (the normalized additional cost incurred by flooding).

If we construct a random variable Y with tail distribution given by the function $1 - \hat{C}_k$, then we can see from Corollary 1 that broadcast flooding is optimal if and only if X is stochastically larger than Y. Figure 1 depicts an example of a tail distribution under which broadcast flooding is the optimal search strategy, where $L = 100$ and the cost is linear. In fact, broadcast flooding is optimal for any object location X with tail distribution that lies "above" or at the dashed line in the figure.

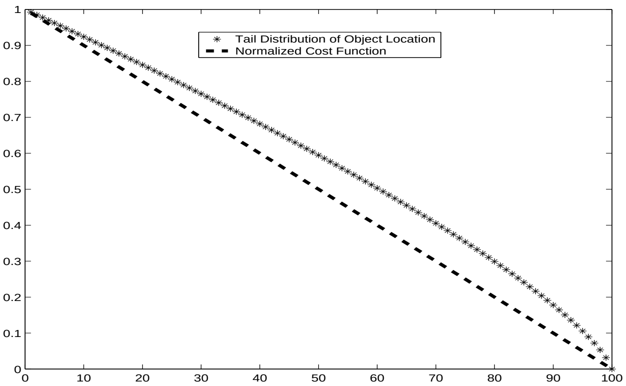

Figure 1: Example of a tail distribution $\bar{F}(k)$ (values given by asterisks) for X under which broadcast flooding is optimal, when cost is linear and $L = 100$. Normalized cost function $\hat{C}_k$ given by dashed lines. Note that for this example, broadcast flooding is optimal for any object location X with tail distribution that lies "above" or at the dashed line in the figure.

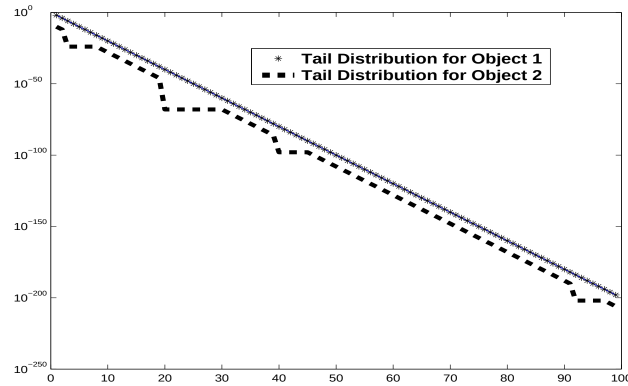

Figure 2: Logarithmic plot of tail distribution for two objects when $L = 100$; cost is a linear function of TTL value. For object 1 (denoted by asterisks), expanding ring search is optimal strategy. However, expanding ring search is not optimal for tail distribution of object 2 (dashed line) even though its tail distribution is stochastically smaller.

The second commonly used search strategy is the *expanding ring* search scheme. Here we consider a special case of this scheme, where $u_i = i$ for $1 \le i \le L$, i.e., the TTL value is incremented by 1 after each unsuccessful round of search. Note that the expected search cost for expanding ring search is given by $\sum_{l=1}^{L} C_l \bar{F}(l-1)$.

Theorem 2. *Given that a TTL value n on the m-th search failed to locate the object, it is optimal to use TTL values of $u_{m+l} = n + l$ for all l, $1 \le l \le L - n$ (i.e., increment the TTL by 1 after each additional unsuccessful search)* iff

$$\bar{F}(m+1|m) \quad \le \quad \frac{C_{m+2} - C_{m+1}}{C_{m+2}} = 1 - \frac{C_{m+1}}{C_{m+2}} \qquad \text{for} \quad n \le m \le L-2$$

This condition is equivalent to

$$Pr\,(X = m | X \ge m) \ge \frac{C_m}{C_{m+1}}, \quad n+1 \le m \le L-1 \; . \tag{12}$$

The proof of Theorem 2 is similar to the proof of Theorem 1 and therefore omitted from the text; for a complete proof, see [11]. Setting $n = 0$, we obtain the following sufficient and necessary condition on the object tail distribution in order for an expanding ring search scheme to be the optimal strategy.

Corollary 2. *Expanding ring search (i.e., setting $u_i = i$) is the optimal average cost search strategy* iff

$$\frac{\bar{F}(m+1)}{\bar{F}(m)} \le 1 - \frac{C_{m+1}}{C_{m+2}}, \quad 0 \le m \le L-2 \; .$$

This condition is equivalent to

$$Pr\,(X = m | X \ge m) \ge \frac{C_m}{C_{m+1}}, \quad 1 \le m \le L-1 \; . \tag{13}$$

The intuition behind these results is that in order for incremental TTL values to be optimal, the conditional probability of finding the object on the next search needs to be sufficiently large compared to the relative costs of searching at the incremental value and searching further, as clearly indicated by (12) and (13).

Figure 2 depicts an example of a tail distribution that satisfies the condition of Corollary 2, where the object location is exponentially distributed with $\bar{F}(k) = e^{-\lambda k}$ for $0 \le k \le L-1$, the cost is linear, and $\lambda \ge \ln(L)$. Note that the tail distribution decreases rapidly. This decrease is a necessary consequence of Corollary 2, and can be described quantitatively as follows:

Corollary 3. *If expanding ring search is the optimal strategy then the following is true for all $1 \le m \le L-1$:*

$$\bar{F}(m) \le \prod_{j=1}^{m} \left(1 - \frac{C_j}{C_{j+1}}\right) = \frac{\prod_{j=1}^{m} (C_j - C_{j-1})}{\prod_{j=1}^{m+1} C_j} \tag{14}$$

where $C_0 = 0$.

As an example, in a network where C_k is a linear function of k, (14) then becomes the following inequality for all m such that $1 \le m \le L-1$:

$$\bar{F}(m) \le \prod_{j=1}^{m} \left(1 - \frac{j}{j+1}\right) = \frac{1}{(m+1)!} \tag{15}$$

Equation (15) implies that the tail distribution must have a very sharp decrease. For example, $\bar{F}(5) \le \frac{1}{6!} = 0.0014$ must be true in order for expanding ring search to be optimal; in other words, the probability that the object can be located with a TTL value of 5 must be at least .9986. This requirement holds regardless of the network dimension L. This result essentially reveals the observation that the expanding ring search may only be used when the object is expected to be in a very close neighborhood.

Earlier, we saw that when broadcast flooding is the optimal strategy for an object location with tail distribution X, then it is the optimal strategy for object with tail distribution Y if Y is stochastically larger than X. However, an analogous result does not hold for the expanding ring search, as shown in the Figure 2.

4. BEST WORST-CASE PERFORMANCE STRATEGIES

In the previous section we examined search strategies that minimize the expected search cost for a given object location distribution. We now turn to the problem of finding good search strategies when this probability distribution is not known a priori. In this case, a natural performance criterion is the worst-case performance. That is, we would like to find a search strategy that has the lowest worst-case cost.

It has been shown in [8] that under a linear cost, the California Split rule achieves asymptotically (as the network size increases) the minimum worst-case cost. However, we will show in Sections 4.2 and 4.3 that this is true when only fixed or deterministic search strategies are considered. It is in fact always possible to find a *random* TTL sequence for any given *nonrandom* sequence that performs better in the worst case. Therefore, under this criterion the best search strategies are *randomized strategies.*

4.1 Randomized Strategies and Worst-Case Performance Measure

With a slight abuse of notation, we introduce randomized search strategies as follows:

Definition 1. *A randomized search strategy* $\mathbf{u}$ *is a TTL sequence that consists of random variables of certain probability distributions (discrete), i.e.,* $\mathbf{u} = [u_1, u_2, \cdots, u_N]$ *where* u_i *is a random variable,* $1 \leq i \leq L$*, and their distribution can be independently or jointly defined.*

This is a generalization of the previous definition of a (nonrandom) search strategy that consists of deterministic TTL values. Accordingly, we will now expand the set of all possible search strategies U to include both nonrandom and random search strategies. Note that with this inclusion, we are now faced with a much bigger set of strategies.

We will use the same notation $J_X^{\mathbf{u}}$ to denote the expected search cost of using strategy $\mathbf{u}$ for object location X, noting that when $\mathbf{u}$ is a deterministic vector $J_X^{\mathbf{u}}$ is the average over all realizations of X as defined in (1), while $\mathbf{u}$ being a random vector means $J_X^{\mathbf{u}}$ is the average over all realizations of X as well as all realizations of $\mathbf{u}$. We only consider the case where the random vector $\mathbf{u}$ and X are mutually independent since the distribution of X is not known *a priori.*

On the other hand, the expected search cost using an ideal (omniscient) observer who knows precisely the location (realization of X) is $E[C_X]$, where $C_X = C(X)$ is the cost of searching with TTL value X (linear or quadratic as given earlier). This is also the best (minimum cost) obtainable. We will use the ratio between these two costs $\frac{J_X^{\mathbf{u}}}{E[C_X]}$, also referred to as the *cost ratio*, to evaluate the performance of strategy $\mathbf{u}$. This is a slight generalization of the criterion used in [8] where only linear search cost for deterministic TTL sequences is considered.

The worst-case performance (cost ratio) for any strategy $\mathbf{u}$ is therefore

$$\rho^{\mathbf{u}} = \sup_{\{p_X(x)\}} \frac{J_X^{\mathbf{u}}}{E[C_X]} , \qquad (16)$$

where the supremum is taken over all possible probability distributions $\{p_X(x)\}$ for the object location. The quantity $\rho^{\mathbf{u}}$ is also known as the *competitive ratio* with respect to an *oblivious adversary* [12] who knows the search strategy $\mathbf{u}$ in advance.

The best worst-case strategy, denoted by $\mathbf{u}^*$, is one that achieves the minimum over all admissible search strategies, denoted by ρ^*:

$$\rho^* = \inf_{\mathbf{u}\in U} \rho^{\mathbf{u}} = \inf_{\mathbf{u}\in U} \sup_{\{p_X(x)\}} \frac{J_X^{\mathbf{u}}}{E[C_X]} . \qquad (17)$$

We require that any probability distribution within the set $\{p_X(x)\}$ must satisfy $E[C_X] < \infty$, so that the object can be located with finite average cost.

The following lemma will be critical in our subsequent analysis.

Lemma 1. *For any search strategy* $\mathbf{u} \in U$ *(random and nonrandom),*

$$\rho^{\mathbf{u}} = \sup_{\{p_X(x)\}} \frac{J_X^{\mathbf{u}}}{E[C_X]} = \sup_{x\in\mathbb{Z}^+} \frac{J_x^{\mathbf{u}}}{C_x} , \qquad (18)$$

where J_x^u *denotes the expected search cost using TTL sequence* $\mathbf{u}$ *when* $Pr(X = x) = 1$*,* C_x *is the search cost using TTL value* x*, and* $\mathbb{Z}^+$ *denotes the set of natural numbers and represents all possible singleton object locations.*

In words, this lemma says that for any TTL sequence, the worst case scenario is when the object location is a constant, i.e., with a singleton probability distribution. We will also subsequently refer to such a single-valued location as a *point.* Note that this constant (i.e., worst case) may not be unique. This result allows us to limit our attention to singleton-valued X and equivalently redefine the minimum cost ratio ρ^* in equation (17) as:

$$\rho = \inf_{\mathbf{u}\in U} \rho^{\mathbf{u}} = \inf_{\mathbf{u}\in U} \sup_{x\in\mathbb{Z}^+} \frac{J_x^{\mathbf{u}}}{C_x} . \qquad (19)$$

4.2 Constructing a Randomized Strategy

In this section we show that for any nonrandom finite TTL sequence, there exists a random search strategy that achieves a lower worst-case cost.

We begin by defining a class of search cost functions denoted by $\mathbb{C}$ as follows:

Definition 2. *An increasing cost function* C_k *of using a TTL value of* k *is said to belong to the class* $\mathbb{C}$ *if for all* $3 \leq k \leq L$*,*

$$C_k < C_{k-1} + \frac{C_{k-1}^2}{\sum_{i=1}^{k-2} C_i} . \qquad (20)$$

Note that the linear and quadratic costs, given by $C_k = \alpha k$ and $C_k = \alpha k^2$ for some $\alpha > 0$ respectively, both satisfy (20). This constraint limits the amount of increase $C_k - C_{k-1}$ in the cost function, and is introduced for technical reasons (in proving Theorem 3). The randomization scheme presented next can be modified accordingly for different cost functions.

Consider any nonrandom TTL sequence given by the finite length vector $\mathbf{g} = [g_1, g_2, \cdots, g_N]$, where $g_N = L$ and $g_1 < g_2 < ... < g_{N-1} < g_N$. Also define $g_0 = 0$. We have the following result.

Lemma 2. *Let an integer* x^**,* $1 \leq x^* \leq L$*, be such that*

$$\rho^{\mathbf{g}} = \frac{J_{x^*}^{\mathbf{g}}}{C_{x^*}} = \max_{1\leq x\leq L} \frac{J_x^{\mathbf{g}}}{C_x} . \qquad (21)$$

If the cost $C_x \in \mathbb{C}$*, then either* $x^* = g_n + 1 < g_{n+1}$ *for some* $0 \leq n \leq N-1$ *or* $x^* = g_N = L$*.*

What this lemma says is that under $C_x \in \mathbb{C}$, the worst-case location is either immediately following one of the TTL values in the sequence $\mathbf{g}$, or at the boundary L for any given deterministic TTL sequence $\mathbf{g} = [g_1, g_2, \cdots, g_N]$. This result is intuitively clear in that the worst location is the closest point that is outside some searched area g_n.

Define a set S as follows:

$$S = \left\{ 1 \leq x^* \leq L : \frac{J^{\mathbf{g}}_{x^*}}{C_{x^*}} = \max_{1 \leq x \leq L} \frac{J^{\mathbf{g}}_x}{C_x} \right\} . \tag{22}$$

This is essentially the set of all the worst-case location values. It follows from Lemma 2 that the number of elements in S is between 1 and N.

We now construct a randomized strategy $\hat{\mathbf{g}}$ from the fixed sequence $\mathbf{g} = [g_1, g_2, \cdots, g_N]$ as follows. Note that the length of $\hat{\mathbf{g}}$ will vary depending on the values within the set S.

(C.1) For all g_m such that $g_m + 1 \notin S$ and $1 \leq m \leq N-1$, define $\tilde{g}_m = g_m$. That is, we will keep the same TTL values when they are not next to a worst case point. In addition, set $\tilde{g}_N = g_N = L$ regardless of whether it belongs to set S.

(C.2) For any g_m such that $g_m + 1 \in S$ and $1 \leq m \leq N-1$, define $\tilde{g}_m = g_m + 1$.

(C.3) Define the following quantities:

$$0 < p < \min \left\{ 1, \min_{x \notin S} \left\{ \left(\rho^{\mathbf{g}} - \frac{J^{\mathbf{g}}_x}{C_x} \right) \frac{C_x}{M_L} \right\} \right\} , \tag{23}$$

and

$$M_j = \sum_{i \in S, i \leq j} (C_i - C_{i-1}) , \quad 1 \leq j \leq L ,$$

where $C_0 = 0$. Note that M_j is a positive nondecreasing function of j because C_x is an increasing function, and that p is strictly positive. The quantity M_j will be used in our analysis of the cost ratio for the new TTL sequence $\hat{\mathbf{g}}$.

(C.4) For the search strategy, we employ the following construction. With probability $1-p$, use the original unmodified TTL sequence $\mathbf{g}$. With probability p, use a modified sequence whose elements will depend on whether the worst-case point occurs at object location of 1 or $L = g_N$. All four possible cases for this modified sequence are described in $(C.5)$ through $(C.8)$.

(C.5) If $g_N \notin S$ and $1 \notin S$, use the TTL sequence $[\tilde{g}_1, \tilde{g}_2, ...\tilde{g}_N]$ with probability p.

(C.6) When $g_N \notin S$ and $1 \in S$, it follows from Lemma 2 that $\tilde{g}_1 \geq g_1 > 1$. So we can use the increasing TTL sequence $[1, \tilde{g}_1, \tilde{g}_2, ...\tilde{g}_N]$ with probability p. Note that this step only differs from (C.5) due to the insertion of a "1" before $\tilde{g}_1$.

(C.7) When $g_N \in S$ and $1 \in S$, note that $g_N \in S$ implies that $g_{N-1} = L - 1 = g_N - 1$. This is because if we assume otherwise, then $J^{\mathbf{g}}_{g_N - 1} = J^{\mathbf{g}}_{g_N}$ and therefore $\frac{J^{\mathbf{g}}_{g_N-1}}{C_{g_N-1}} > \frac{J^{\mathbf{g}}_{g_N}}{C_{g_N}}$, which contradicts the fact that g_N belongs to the set S and achieves the maximum cost ratio. Therefore, use the TTL sequence $[1, \tilde{g}_1, \tilde{g}_2, ...\tilde{g}_{N-2}, \tilde{g}_N]$ with probability p. Note that this step only differs from (C.6) due to the removal of $\tilde{g}_{N-1}$.

(C.8) Finally, when $g_N \in S$ and $1 \notin S$ then with probability p use the TTL sequence $[\tilde{g}_1, \tilde{g}_2, ...\tilde{g}_{N-2}, \tilde{g}_N]$. Note that this step only differs from (C.7) due to the removal of the "1" because $1 \notin S$.

The random TTL sequence $\hat{\mathbf{g}} = [\hat{g}_1, \hat{g}_2, \cdots]$ generated by the above steps can be viewed as taking one of two possible realizations: with probability $1-p$, we will employ the original TTL sequence $\mathbf{g}$. With probability p, we will employ a sequence of modified TTL values $\{\tilde{g}_k\}$, where the modified values will depend on the elements of the set S. This modified sequence could be shorter or longer than $\mathbf{g}$ depending on whether 1 or $g_N = L$ are members of the set S. The advantage of using such a construction is that the expected search cost will decrease at points which obtain the maximum cost ratio, as described by the following theorem.

Theorem 3. *Consider any nonrandom TTL seq. given by the integer-valued finite length vector $\mathbf{g} = [g_1, g_2, ...g_N]$, where $g_N = L$ and $g_1 < g_2 < ... < g_{N-1} < g_N$. Construct a new random TTL sequence $\hat{\mathbf{g}}$ using the method outlined above in (C.1) through (C.8). If the cost function belongs to the class $\mathbb{C}$, then:*

$$\rho^{\hat{\mathbf{g}}} = \max_{1 \leq x \leq L} \frac{J^{\hat{\mathbf{g}}}_x}{C_x} < \max_{1 \leq x \leq L} \frac{J^{\mathbf{g}}_x}{C_x} = \rho^{\mathbf{g}} . \tag{24}$$

Therefore there exists at least one random TTL sequence given by $\hat{\mathbf{g}}$ that achieves a lower worst-case cost than that using the nonrandom sequence $\mathbf{g}$.

The key result here is thus that given any fixed nonrandom TTL sequence, one can always construct a random TTL sequence that performs better in the worst case. There is a very nice intuition behind this construction/randomization, which is to "spread" the cost at the worst-case point to its neighboring points, and therefore bring down the worst-case cost. This will be elaborated further via examples at the end of this section.

Given this result, it is thus clear that under this performance criterion, the optimal search strategy must be a randomized strategy. This particular construction obviously does not guarantee optimality. We next derive a randomized search strategy that is asymptotically optimal within a class of randomized strategies.

4.3 Uniform Randomization

In this subsection we introduce a class of *uniformly randomized strategies* and derive the asymptotically optimal strategy within this class in the next subsection. In particular, we will be interested in the performance of a search strategy when the network increases in size, and thus will consider an infinitely large network and infinitely long TTL sequences. We will also limit our attention to linear search cost for simplicity and discuss the case of quadratic cost at the end.

Definition 3. *For any infinite, increasing sequence $\mathbf{g} = [g_1, g_2, ...]$ in which the elements g_k are positive integers and $g_j > g_k$ for all $j > k$, a* uniformly randomized *TTL sequence $\hat{\mathbf{g}} = [\hat{g}_1, \hat{g}_2, ...]$ is created by assigning the following* *probability distribution to each TTL random variable* $\hat{g}_k$*:*

$$Pr(\hat{g}_k = l) = \begin{cases} \frac{1}{g_{k+1}-g_k} & \textit{if} \quad g_k \le l \le g_{k+1}-1 \\ 0 & \textit{otherwise} \end{cases} \tag{25}$$

where l *is any positive integer.*

Essentially the elements in the nonrandom sequence $\mathbf{g} = [g_1, g_2, ...]$ serve as the boundaries of a sequence of non-overlapping ranges over which each random variable $\hat{g}_k$ is uniformly distributed. These ranges collectively cover all positive integers. Following this definition, for each nonrandom TTL sequence, there exists a corresponding uniformly randomized version.

As the constant α in the linear cost $C_k = \alpha k$ gets cancelled out in the computation of the cost ratio, we will simply assume that the cost is $C_k = k$ which does not affect our discussion. Then the worst-case performance measure given by (16) reduces to for any $\alpha > 0$

$$\rho^{\mathbf{u}} = \sup_{x \in \mathbb{Z}^+} \frac{J_x^{\mathbf{u}}}{x} \ , \tag{26}$$

We then have the following result.

Lemma 3. *Under a uniformly randomized search strategy* $\hat{\mathbf{g}}$ *with boundaries defined by the fixed sequence* $\mathbf{g}$*, the worst-case cost ratio is given by:*

$$\begin{aligned} \rho^{\hat{g}} &= \sup_{x \in \mathbb{Z}^+} \frac{J_x^{\hat{g}}}{x} = \sup_{m \in \mathbb{Z}^+} \frac{J_{g_m}^{\hat{g}}}{g_m} \\ &= \sup_{m \in \mathbb{Z}^+} \frac{\sum_{k=1}^{m} g_k + \frac{g_{m+1}-g_1}{2} - \frac{m}{2}}{g_m} \end{aligned} \tag{27}$$

Lemma 3 reveals two things. Most important is that the worst-case object location for a uniformly randomized strategy must be on a boundary g_m for some m (this is the lower boundary of one of the uniform distributions), rather than an arbitrary positive integer. This greatly simplifies the process of finding the worst-case cost ratio. It also gives the expression of this cost ratio in terms of the boundary sequence.

It was shown in [8] that the California Split algorithm, which uses fixed TTL values of $u_i = 2^{i-1}$, achieves a worst-case cost ratio of $\rho^{\mathbf{u}} = 4$. It was also shown that this is asymptotically the best (lowest) over all nonrandom TTL sequences, and thus the California Split algorithm is asymptotically optimal (best worst-case performance) of all nonrandom strategies and $\rho = 4$ is asymptotically the best worst-case cost ratio.

In the next section we will show that by including randomized search strategies we can do much better (a lower ρ). In particular, we will derive the asymptotically optimal search strategy within the class of uniformly randomized strategies.

4.4 Optimal Uniform Randomization

Consider the following sequence $\mathbf{g} = \{g_k\}$, $g_k = \lfloor r^{k-1} \rfloor$ for some positive real number r, $k = 1, 2, \cdots$. Define as in (25) a uniformly randomized search strategy $\hat{\mathbf{g}}$ using the boundary sequence $\mathbf{g}$. Note that each $g_k = r^{k-1} - \delta_{k-1}$ for some $0 \le \delta_{k-1} < 1$. Taking this boundary value into (27),

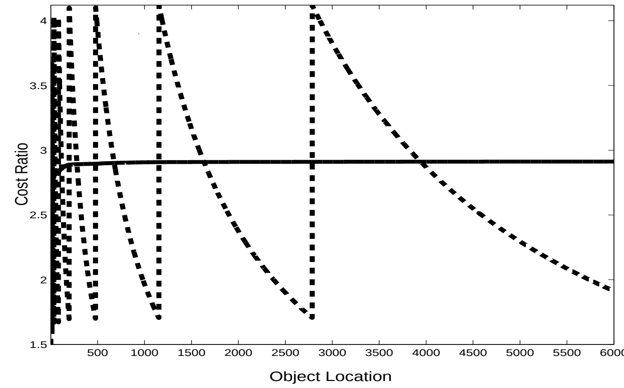

Figure 3: Cost ratio as a function of object location for a nonrandom TTL sequence (dotted line) with $g_k = \lfloor r^{k-1} \rfloor$**,** $r = \sqrt{2}+1$ **and the cost ratio for its uniformly randomized version (solid line). Cost is assumed to be a linear function of TTL values.**

we obtain the cost ratio for the randomized sequence:

$$\begin{aligned} \frac{J_{g_m}^{\hat{\mathbf{g}}}}{g_m} &= \frac{\sum_{k=1}^{m}(r^{k-1} - \delta_{k-1}) + \frac{r^m - \delta_m - r^0 + \delta_0}{2} - \frac{m}{2}}{r^{m-1} - \delta_{m-1}} \\ &= \frac{r^{m-1}}{r^{m-1} - \delta_{m-1}} \left(\frac{r}{r-1} + \frac{r}{2} - \frac{1}{2r^{m-1}} \right) \\ &- \frac{r^{m-1}}{r^{m-1} - \delta_{m-1}} \left(\frac{\sum_{k=1}^{m} \delta_{k-1} + \frac{\delta_m + \delta_0}{2} + \frac{m}{2}}{r^{m-1}} \right) \end{aligned}$$

It can be seen from this result that for m large enough, $\frac{J_{g_m}^{\hat{\mathbf{g}}}}{g_m}$ is an increasing function of m, and that we can obtain the supremum by taking the asymptotic limit:

$$\rho^{\hat{\mathbf{g}}} = \sup_{m \in \mathbb{Z}^+} \frac{J_{g_m}^{\hat{\mathbf{g}}}}{g_m} = \lim_{m \to \infty} \frac{J_{g_m}^{\hat{\mathbf{g}}}}{g_m} = \frac{r}{r-1} + \frac{r}{2} \tag{28}$$

Differentiating (28) and noting convexity, we find that the value of r that minimizes $\rho^{\hat{\mathbf{g}}}$ is $r = \sqrt{2}+1 \approx 2.4142$, which achieves a worst-case cost ratio of $\frac{3}{2} + \sqrt{2} \approx 2.9142$. This ratio represents a 27% improvement over the worst-case cost ratio of 4 for the nonrandom California Split algorithm. The resulting uniformly randomized TTL sequence is defined by the boundary sequence $[1, 2, 5, 14, 33, \cdots]$ by taking the optimal value r into the power series.

The next theorem establishes the optimality of this uniformly randomized search strategy.

Theorem 4. *Let* U' *denote the set of all nonrandom and uniformly randomized TTL sequences. Then:*

$$\inf_{\mathbf{u} \in U'} \rho^{\mathbf{u}} = \inf_{\mathbf{u} \in U'} \sup_{x \in \mathbb{Z}^+} \frac{J_x^{\mathbf{u}}}{x} = \frac{3}{2} + \sqrt{2} \approx 2.9142 \ . \tag{29}$$

That is, the uniformly randomized sequence given by the boundary sequence $g_k = \lfloor r^{k-1} \rfloor$ *with* $r = \sqrt{2}+1$ *is asymptotically optimal within the set* U'*.*

Figure 3 depicts the cost ratio of using this random TTL sequence, along with the cost ratio of using its nonrandom boundary sequence g_k as TTL values. Most notably, using the nonrandom TTL sequence results in oscillation of the cost ratio, while the uniformly randomized search sequence results in a smooth cost ratio curve and approaches the maximum value 2.9142 asymptotically from below as the network dimension grows to infinity. This figure reveals the fundamental difference between a fixed TTL sequence and a random TTL sequence and why the latter performs better. This is elaborated in the next subsection.

4.5 How Randomization Works

We have shown in the previous two subsections that the minimum worst-case cost ratio ρ is obtained by a random TTL sequence regardless of whether the network is finite or approaches infinity. It remains to explain why a randomized sequence performs better in the worst-case than any nonrandom TTL strategy.

The randomized strategy $\hat{\mathbf{g}}$ constructed in Section 4.2 shows that randomizing some of the TTL values of $\mathbf{g}$ increases the cost ratio at points $x \notin S$ (but not sufficient to exceed the worst-case cost ratio $\rho^{\mathbf{g}}$), and at the same time lowering the cost ratio at $x \in S$.

The same effect is exhibited in Figure 3, where the uniform randomization effectively balances the oscillating high and low cost ratios under the nonrandom TTL sequence and achieves a much lower maximum cost ratio. This is the fundamental reason why random TTL sequences result in lower worst-case cost ratio, and it possibly generalizes to other types of randomizations.

In hindsight this reason is intuitively clear: For fixed nonrandom TTL sequences the worst-case cost ratio is determined by certain singleton-valued locations and moving away from these locations may cause significant changes in the cost ratio. Randomization essentially has the *averaging* effect that "smoothes out" the cost ratio across neighboring locations/points. In fact the optimal uniformly randomized TTL sequence (under the linear cost assumption) has a cost ratio curve that does not have local minima or maxima as depicted in Figure 3. One may also view this as the built-in *robustness* of a randomized policy for the underlying criterion of worst-case performance.

Note that for a fixed object location, a fixed TTL sequence results in a fixed deterministic search cost, whereas a randomized TTL sequence results in different realizations of the search strategy and hence different search cost. Shown in Figure 3 (solid line) is the average over all realizations. For a detailed disussion on the variance and best/worst case realizations of the search cost, see [11].

4.6 General Cost Functions

The uniformly randomized search strategy derived in Section 4.4 is shown to be optimal using a linear cost function. What if the cost is quadratic or more generally any increasing function? Then the uniform probability distribution used in Definition 3 must be adjusted to account for the new cost function if we would like to obtain the same averaging effect discussed in Section 4.5.

In fact, it can be shown that if we allow TTL values to be any real value in $[1, \infty)$, then we have the following result: for any (random or nonrandom) TTL sequence under a continuous and increasing cost function, there exists a TTL sequence that attains the same expected search cost under the linear cost function. This equivalence relationship allows us to limit our attention to finding optimal search strategies for a linear cost function. The real-valued TTL sequences can be discretized to obtain an approximate mapping. Due to space limitations, the reader is referred to [11] for the

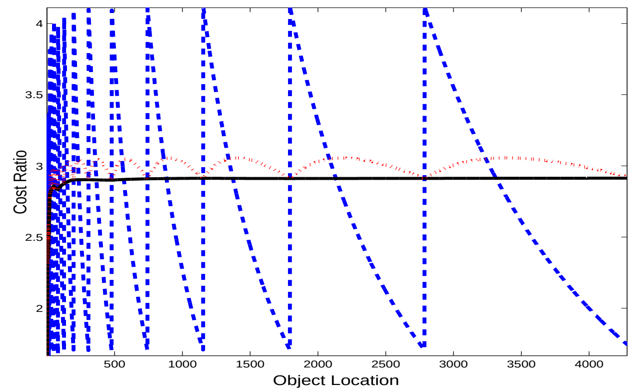

Figure 4: Under a quadratic cost function, the cost ratio as a function of object location for a nonrandom TTL sequence (dashed line) with $g_k = \lfloor r^{\frac{k-1}{2}} \rfloor$, $r = \sqrt{2}+1$, its uniformly randomized version (dotted line) corresponding to distribution given in (25), and its randomized version (solid line) corresponding to the distribution given in (30). Note the distribution given by (30) produces cost ratio curve that is similar to Figure 3.

complete derivation of such a mapping between sequences under different cost functions. As an example, suppose the cost is quadratic, i.e. $C_k = k^2$. Then from any nonrandom TTL sequence $\mathbf{g}$, we can create a TTL sequence $\hat{\mathbf{g}}$ by assigning the following distribution to each TTL value $\hat{g}_k$:

$$Pr(\hat{g}_k = l) = \begin{cases} \frac{2l+1}{{g_{k+1}}^2 - {g_k}^2} & \text{if} \quad g_k \leq l \leq g_{k+1} - 1 \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

Figure 4 depicts the cost ratio for the sequence $\mathbf{g}$ (dashed line) defined by $g_k = \lfloor r^{\frac{k-1}{2}} \rfloor$, $r = \sqrt{2} + 1$, its uniformly randomized version (dotted line) corresponding to distribution given in (25), and its randomized version (solid line) corresponding to the distribution given in (30). Note that under the quadratic cost, the uniform randomization results in oscillation of the cost ratio, although much less significant than its nonrandom counterpart, and achieves a maximum cost ratio of approximately 3.06. This phenomenon originates from a mismatch between the TTL random variables being uniformly distributed over each interval, and the search cost being quadratic. However, by using the distribution given in (30) and the modified boundary sequence $\mathbf{g}$, we obtain a cost ratio curve similar to the optimal uniformly randomized sequence under a linear cost function (Figure 3). In both of these similar curves, the randomized sequences obtain an asymptotic maximum worst-case cost of approximately 2.9142.

Hence, the randomization of (30) has the effect of smoothing out the cost ratio curve. Similarly, the distribution can be adjusted for other cost functions in order to obtain a smooth cost ratio curve with worst-case value of 2.9142.

5. DISCUSSION AND PRACTICAL IMPLICATIONS

Most of the ideas and results derived in this paper are directly applicable to the design of practical networking mechanisms. In particular, given known object location distributions and search costs, the optimization framework presented in Section 3 can be used to derive optimal search strategies, which can be further incorporated into networking protocols that require a search functionality, e.g., an ad hoc routing protocol. Similarly, strategies introduced in Section 4 can also be incorporated into a variety of networking mechanisms.

In this section we discuss how some of the assumptions made earlier may be relaxed. We then compare the performance of a number of search strategies in a wireless network scenario via numerical simulation.

5.1 Weaker Assumptions

We have adopted two fairly strong assumptions (Assumptions 2 and 4 in Section 2.1 on collision-free communication and perfect timeout value). If using a TTL value of k does not reach all nodes within k hops, either due to inteference, packet collision or imperfect timeout values, the problem formulation changes significantly and perhaps a different approach is needed. In this case, there is essentially a probability that a TTL value k misses the object even if the object location is within k hops.

One potential approach to tackling this problem could be the following: let γ_k denote the conditional probability that a TTL value of k successfully locates the object, given that the object is within k hops. Note that if $\gamma_k = 1$, the problem reduces to the one presented in this paper. The value of γ_k will depend on the broadcast techniques used, as well as the likelihood that the timeout value is imperfectly set. Various broadcast techniques have been proposed and studied in the literature, such as in [9], [13], and [14]. Selecting an appropriate timeout value is dependent on the system properties; this factor can be incorporated into the study by considering the probability that using a TTL value of k reaches a particular node that is j hops away from the source, for $j \leq k$. Hence, it can be seen that the broadcast and timeout factors can be incorporated into the conditional probability γ_k. Developing a new performance measure with these modified assumptions and determining the corresponding optimal strategies for general γ_k is part of our future work.

5.2 Simulation Results

We present simulation results that compare the average search cost of a number of search strategies in the following problem scenario. All results are obtained using Matlab. The network consists of 4,000 static nodes uniformly distributed in a circle of unit radius, with the source node located at the center. Each node retransmits a received query exactly once, and the query reaches every other node within a transmission radius. Nodes disregard multiple copies of the same query. Packet transmission times are jittered to avoid simultaneous transmission. We subsequently assume all transmissions are correctly received. Thus using a TTL value k will reach every node within k hops of the source. In addition, we assume that the single target exists in the network and is equally likely to be located in one of the non-source nodes. Search cost is measured by the total number of transmissions.

As an example of the type of cost function generated under these assumptions, consider when each node has transmission radius of 0.115. In this case, the average network dimension is 10. The cost, averaged over 20 randomly generated networks, of using a TTL value of k, for $1 \leq k \leq 10$

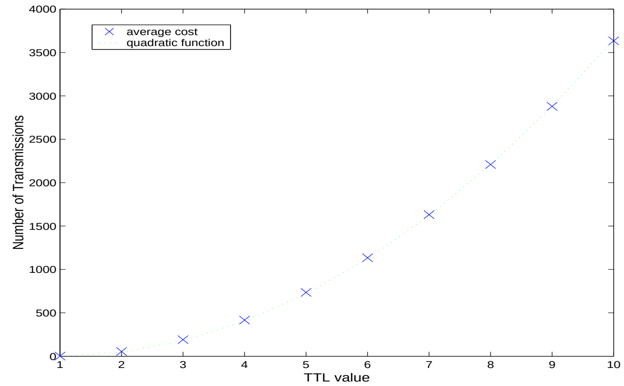

Figure 5: Average search cost as a function of TTL value under conditions described in Section 5.2, and function $45(k-1)^2$.

is depicted in Figure 5. Note that the cost is very well approximated by the function $45(k-1)^2$, a quadratic function of the TTL value, as was mentioned earlier in Section 2.2 [1].

We examined the average-cost performance of the following strategies: broadcast flooding (BF), expanding ring (ER), california split (CS), and the optimal strategy computed using the dynamic programming formulation presented in Section 3.1. The value function is computed by using the cost function obtained in Figure 5, and the fact that the object is equally likely to be located in any of the non-source nodes. The results are depicted in Figure 6.

The horizontal axis is labeled with increasing network dimension L. Since we are fixing the area and controlling the transmission radius to obtain different values of L, the horizontal axis can also be viewed as the transmission radius in decreasing values. Specifically, the performance of the aformentioned 4 strategies are evaluated with L ranging from 5 to 17, and equivalently with transmission radius ranging from .075 to .225 by increments of .01. Each point on a curve represents the average search cost of the corresponding search strategy over 10^6 randomly chosen object locations in each of 20 different randomly generated networks, under a particular transmission radius value.

We see that using the optimal strategy can significantly decrease the search cost compared to some of the commonly used strategies. Expanding ring search is far from being optimal in most instances while flooding and the California split search give reasonable performance in most of the instances in this particular example. Note that in this example nodes are randomly placed and the object is uniformly distributed among all nodes. In addition, only the number of transmissions factors into the search cost. If any of these parameters change, the relative performance of BF, ER and CS is expected to change as well. However, one should always be able to compute the optimal search strategy using the method presented here.

Simulation results on randomized strategies are not included due to space limitations. We note however, that the normalized average search costs of two randomized strategies are essentially given in Figure 4 for every possibile location of the object (using a quadratic cost function).

[1]In addition, note that the cost function also belongs to the class $\mathbb{C}$ as given by Definition 2.

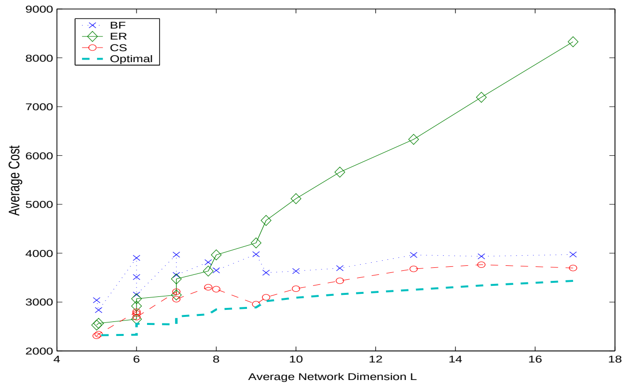

Figure 6: Under the conditions outlined in Section 5.2, the average-cost performance as a function of network dimension, of broadcast flooding (BF), expanding ring (ER), California split (CS), and the optimal strategy computed using the dynamic programming formulation presented in Section 3.1.

6. CONCLUSION AND FUTURE WORK

In this paper we studied the class of TTL-based controlled flooding search methods used to locate an object/node in a large network. The objective is to derive search strategies that minimize the expected search cost. We presented a dynamic programming approach with which optimal search strategies can be derived when the probability distribution of the object location is known a priori. More interestingly, when the object location distribution is not known we discovered that *randomized* search strategies outperform fixed nonrandom strategies. We provided a randomization construction and also derived the asymptotically optimal strategy within the class of uniformly randomized strategies for linear search cost. These results are directly applicable in designing practical algorithms.

7. REFERENCES

[1] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in the sensor networks," *Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (MobiCOM'99)*, August 1999, Seattle, Washington.

[2] D. Braginsky and D.Estrin, "Rumor routing algorithm for sensor networks," *Proc. International Conference on Distributed Computing Systems (ICDCS-22)*, 2002.

[3] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, pp. 153–181, 1999.

[4] J. Xie, R. Talpade, T. McAuley, and M. Liu, "AMRoute: Ad Hoc Multicast Routing Protocol," *ACM Mobile Networks and Applications (MONET) Special Issue on Mobility of Systems, Users, Data and Computing*, vol. 7, no. 6, pp. 429–439, December 2002.

[5] C. Carter, S. Yi, P. Ratanchandani, and R. Kravets., "Manycast: Exploring the space between anycast and multicast in ad hoc networks," *Proceedings of the Ninth Annual International Conference on Mobile Computing and Networks (MobiCOM'03)*, September 2003, San Diego, California.

[6] S. Shakkottai, "Asymptotics of query strategies over a sensor network," *Proceedings of IEEE Infocom*, March 2004, Hong Kong.

[7] Z. Cheng and W. Heinzelman, "Flooding strategy for target discovery in wireless networks," *Proceedings of the Sixth ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2003)*, Sept 2003.

[8] Y. Baryshinikov, E. Coffman, P. Jelenkovic, P. Momcilovic, and D. Rubenstein, "Flood search under the california split rule," *Operations Research Letters*, vol. 32, no. 3, pp. 199–206, May 2004.

[9] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (MobiCOM'99)*, August 1999, Seattle, Washington.

[10] P.R. Kumar and P. Karaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control*, Prentice-Hall, Inc, 1986, Englewood Cliffs, NJ.

[11] N. Chang and M. Liu, "Revisiting the TTL-based controlled flooding search: Optimality and randomization," *EECS Technical Report CGR 04-06*, 2004, University of Michigan, Ann Arbor.

[12] A. Borodin and R. El-Yaniv, *Online Comptutation and Competitive Analysis*, Cambridge University Press, 1998, Cambridge, UK.

[13] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," *Proceedings of Mobihoc*, pp. 194–205, June 2002, Lausanne, Switzerland.

[14] T-J.Kwon Y. Yi, M. Gerla, "Efficient flooding in ad hoc networks using on-demand (passive) cluster formation," *Proceedings of Mobihoc*, June 2003, Annapolis, USA.

APPENDIX

A. PROOFS

As mentioned earlier, included in this appendix are proofs of all theorems and lemmas except for Theorem 2. See [11] for this proof.

A.1 Proof of Theorem 1

We begin by proving the necessary condition of Theorem 1, i.e. that the following holds for all n where $0 \le n \le L-2$:

$$\text{If } V(n) = C_L, \text{ then } \bar{F}(k|n) \ge 1 - \hat{C}_k \quad \text{for } \forall\, k \text{ such that } 1 \le k \le L-1 \tag{31}$$

Note that (31) is true for $1 \le k \le n$, because $\bar{F}(k|n) = 1 \ge 1 - \hat{C}_k$ for these values of k. Next, from Proposition 2 we have that if $V(n) = C_L$, then $V(j) = C_L$ for all $n+1 \le j \le L-1$. From the definition of $V(n)$ and the fact that $V(j) = C_L$ for all $n \le j \le L-1$, we have:

$$\begin{aligned} V(n) &= \min_{n+1 \le l \le L} \left\{ C_l + \bar{F}(l|n) V(l) \right\} \\ &= \min_{n+1 \le l \le L} \left\{ C_l + \bar{F}(l|n) C_L \right\} = C_L \end{aligned} \tag{32}$$

Equation (32) means that $C_l + \bar{F}(l|n) C_L \ge C_L$ for all l such that $n+1 \le l \le L$. Rearranging gives us: $\bar{F}(l|n) \ge \frac{C_L - C_l}{C_L} = 1 - \hat{C}_l$. Therefore, we have proven (31).

Now, we prove the sufficient condition of Theorem 1, i.e. that the following holds for some n such that $0 \le n \le L-2$,

$$\begin{aligned} &\text{If } \bar{F}(k|n) \ge 1 - \hat{C}_k \text{ for } \forall\, k, \;\; 1 \le k \le L-1 \\ &\quad \text{then } V(n) = C_L. \end{aligned} \tag{33}$$

From the given information that $\bar{F}(k|n) \ge 1 - \hat{C}_k$ for all k in $1 \le k \le L-1$, we will prove by backward induction on m that $V(m) = C_L$ for all $n \le m \le L-2$.

Induction Basis: This step requires proof that $V(L-2) = C_L$. We are given that $\bar{F}(k|n) \ge 1 - \hat{C}_k$, for all $1 \le k \le L-1$ and for some $n \le L-2$. From the fact that $n \le L-2$, it must be true that $\bar{F}(L-1|L-2) \ge \bar{F}(L-1|n) \ge 1 - \hat{C}_{L-1}$. This fact can be rearranged as $C_{L-1} + C_L\bar{F}(L-1|L-2) \ge C_L$. By definition, $V(L-2) = \min\{C_{L-1} + C_L\bar{F}(L-1|L-2), C_L\}$, which is clearly C_L from the above inequality. Therefore, $V(L-2) = C_L$ and we have established the induction basis.

Induction Step: From the given information $\bar{F}(k|n) \ge 1 - \hat{C}_k$ for all $1 \le k \le L-1$, we need to prove that if $V(m+1) = C_L$ for some $n \le m \le L-2$ then $V(m) = C_L$. Note that by Proposition 2, because $V(m+1) = C_L$, then $V(l) = C_L$ for all $m+1 \le l \le L-1$. As before, let $g_l = C_l + \bar{F}(l|m)C_L$ for $m+1 \le l \le L$. Because $n \le m$, then $\bar{F}(l|m) \ge \bar{F}(l|n) \ge 1 - \hat{C}_l = \frac{C_L - C_l}{C_L}$ for all $0 \le l \le L-1$. Rearranging this fact gives us: $g_l = C_l + \bar{F}(l|m)C_L \ge C_L$. Hence, from the definition of $V(m)$, we obtain:

$$\begin{aligned} V(m) &= \min_{m+1 \le l \le L} \left\{ C_l + \bar{F}(l|m)V(l) \right\} \\ &= \min_{m+1 \le l \le L} \{g_l\} = C_L \end{aligned} \tag{34}$$

Equation (34) proves the induction step and thus we have proven the sufficient condition of Theorem 1 by induction.

Equations (31) and (33) collectively prove Theorem 1. □

A.2 Proof of Lemma 1

We begin by noting that for every $x \in \mathbb{Z}^+$, there corresponds a singleton probability distribution $p_X(x)$ with $Pr(X = x) = 1$. We thus have the following inequality

$$\sup_{\{p_X(x)\}} \frac{J_X^{\mathbf{u}}}{E[C_X]} \ge \sup_{x \in \mathbb{Z}^+} \frac{J_x^{\mathbf{u}}}{C_x}\,, \tag{35}$$

since the left-hand side is a supremum over a larger set.

On the other hand, setting $A = \sup_{x \in \mathbb{Z}^+} \frac{J_x^{\mathbf{u}}}{C_x}$ we have $\frac{J_x^{\mathbf{u}}}{C_x} \le A$ for all $x \in \mathbb{Z}^+$. Thus $J_x^{\mathbf{u}} \le AC_x$. Then for any random variable X denoting object location, we can use this inequality along with the independence between $\mathbf{u}$ and X to obtain:

$$\begin{aligned} \frac{J_X^{\mathbf{u}}}{E[C_X]} &= \frac{\sum_{x \in \mathbb{Z}^+} J_x^{\mathbf{u}} Pr(X = x)}{\sum_{x \in \mathbb{Z}^+} C_x Pr(X = x)} \\ &\le \frac{\sum_{x \in \mathbb{Z}^+} AC_x Pr(X = x)}{\sum_{x \in \mathbb{Z}^+} C_x Pr(X = x)} = A \end{aligned} \tag{36}$$

Equation (36) tell us that $\frac{J_X^{\mathbf{u}}}{E[C_X]} \le A = \sup_{x \in \mathbb{Z}^+} \frac{J_x^{\mathbf{u}}}{C_x}$. Since this inequality holds for all possible random variables X denoting object location, we obtain:

$$\sup_{\{p_X(x)\}} \frac{J_X^{\mathbf{u}}}{E[C_X]} \le \sup_{x \in \mathbb{Z}^+} \frac{J_x^{\mathbf{u}}}{C_x} \tag{37}$$

Equations (35) and (37) collectively imply the equality in equation (18), and we have proven Lemma 1. □

A.3 Proof of Lemma 2

We prove Lemma 2 by contradiction. Suppose for some x^* satisfying equation (21) that the claim is not true, which means that either $x^* = g_n$ or $x^* = g_n + a < g_{n+1}$ for some $a \ge 2$ and for some $0 \le n \le N-1$. We will prove the contradiction for both cases.

Case 1: Suppose $x^* = g_n$ for $1 \le n \le N-1$. Then the corresponding search cost $J_{x^*}^{\mathbf{g}} = \sum_{l=1}^{n} C_{g_l}$. This can be rearranged as:

$$\frac{J_{g_n}^{\mathbf{g}}}{C_{g_n}} = \frac{\sum_{l=1}^{n} C_{g_l}}{C_{g_n}} = \frac{\frac{C_{g_n+1}}{C_{g_n}} \left(\sum_{l=1}^{n-1} C_{g_l} + C_{g_n} \right)}{C_{g_n+1}} \tag{38}$$

However, using $m = g_n + 1$ in the constraint of equation (20), along with the fact that $\sum_{l=1}^{n-1} C_{g_l} \le \sum_{i=1}^{n-1} C_i$, and then rearranging gives us:

$$\frac{C_{g_n+1}}{C_{g_n}} \sum_{l=1}^{n-1} C_{g_l} < C_{g_n} + \sum_{l=1}^{n-1} C_{g_l} = \sum_{l=1}^{n} C_{g_l} \tag{39}$$

In addition, because $g_n + 1 \le g_{n+1}$, then it follows that $J_{g_n+1}^{\mathbf{g}} = \sum_{l=1}^{n+1} C_{g_l} \ge \sum_{l=1}^{n} C_{g_l} + C_{g_n+1}$. Combining equations (38) and (39) and using this inequality gives us:

$$\frac{J_{g_n}^{\mathbf{g}}}{C_{g_n}} < \frac{\sum_{l=1}^{n+1} C_{g_l}}{C_{g_n+1}} \le \frac{J_{g_n+1}^{\mathbf{g}}}{C_{g_n+1}} \tag{40}$$

However, this contradicts the assumption that $x^* = g_n$ satisfies $\frac{J_{x^*}^{\mathbf{g}}}{C_{x^*}} = \max_{1 \le x \le L} \frac{J_x^{\mathbf{g}}}{C_x}$. Therefore it cannot be true that $x^* = g_n$ for some $1 \le n \le N-1$.

Case 2: Now we can consider the second case of $x^* = g_n + a < g_{n+1}$ for $1 \le n \le N-1$ and some $a \ge 2$. Then $J_{x^*}^{\mathbf{g}} = \sum_{l=1}^{n+1} C_{g_l}$. However, we also have $J_{g_n+1}^{\mathbf{g}} = \sum_{l=1}^{n+1} C_{g_l}$. This gives us:

$$\frac{J_{x^*}^{\mathbf{g}}}{C_{x^*}} = \frac{\sum_{l=1}^{n+1} C_{g_l}}{C_{g_n+a}} < \frac{\sum_{l=1}^{n+1} C_{g_l}}{C_{g_n+1}} = \frac{J_{g_n+1}^{\mathbf{g}}}{C_{g_n+1}} \tag{41}$$

Again, this contradicts the assumption that x^* satisfies $\frac{J_{x^*}^{\mathbf{g}}}{C_{x^*}} = \max_{1 \le x \le L} \frac{J_x^{\mathbf{g}}}{C_x}$. Therefore, it cannot be true that $x^* = g_n + a < g_{n+1}$ for $0 \le n \le N-1$ and $a \ge 2$ □

A.4 Proof of Theorem 3

To begin, we will use the notation that m is in the set R if $g_m + 1 \in S$, for any $0 \le m \le N-1$. Hence R has at most N members, and each member is less than N. Now we will prove Theorem 3 for two separate cases.

Case 1: $g_N \notin S$.
This case corresponds to the sequence generated by (C.5) or (C.6). We will prove that $\frac{J_x^{\hat{\mathbf{g}}}}{x} < \rho^{\mathbf{g}}$ for all x. First, let's consider all $x \notin S$. Note that for any $1 \le x \le L$, there must exist a corresponding positive integer m such that $g_{m-1} < x \le g_m$ (because the TTL sequence is strictly increasing and $g_N = L$). Then for $x \notin S$, the corresponding cost of the randomized sequence is given by $J_x^{\hat{\mathbf{g}}} = J_x^{\mathbf{g}} + pM_{g_m+1}$, where we define $M_{L+1} = M_L$ for notational reasons. This

statement is true because for such x:

$$J_x^{\hat{\mathbf{g}}} = p\left(\sum_{k\in R,\ 0\le k\le m} C_{g_k+1} + \sum_{k\notin R,\ 1\le k\le m} C_{g_k}\right) + (1-p)\sum_{k=1}^{m} C_{g_k} = \sum_{k=1}^{m} C_{g_k} + pM_{g_m+1} = J_x^{\mathbf{g}} + pM_{g_m+1} \quad (42)$$

We then have from inequality (23) defining our chosen p:

$$\frac{J_x^{\hat{\mathbf{g}}}}{C_x} = \frac{J_x^{\mathbf{g}} + pM_{g_m+1}}{C_x} \le \frac{J_x^{\mathbf{g}}}{C_x} + \frac{pM_L}{C_x} < \frac{J_x^{\mathbf{g}}}{C_x} - \frac{J_x^{\mathbf{g}}}{C_x} + \rho^{\mathbf{g}} = \rho^{\mathbf{g}} \quad (43)$$

Therefore, $\frac{J_x^{\hat{\mathbf{g}}}}{C_x} < \rho^{\mathbf{g}}$ for all $x \notin S$.

If $1 \in S$, then it must be true from Lemma 2 that $g_1 > 1$. In addition, $J_1^{\hat{\mathbf{g}}} = pC_1 + (1-p)C_{g_1} < C_{g_1} = J_1^{\mathbf{g}}$, where the strict inequality holds because the cost function is strictly increasing and $g_1 > 1$. Therefore the cost ratio has decreased at location 1.

Next consider the case $x = g_m + 1 \in S$ for some positive integer m. This means that the expected search cost is given by:

$$J_x^{\hat{\mathbf{g}}} = p\left(\sum_{k\in R,\ 0\le k\le m} C_{g_k+1} + \sum_{k\notin R,\ 1\le k\le m} C_{g_k}\right) + (1-p)\sum_{k=1}^{m+1} C_{g_k} = \sum_{k=1}^{m+1} C_{g_k} + pM_{g_m} - pC_{g_{m+1}} < \sum_{k=1}^{m+1} C_{g_k} = J_x^{\mathbf{g}} \quad (44)$$

where the last inequality in (44) follows from the fact that $M_{g_m} \le C_{g_m} < C_{g_{m+1}}$, which follows from the definition of M_j. Equation (44) implies that $\frac{J_x^{\hat{\mathbf{g}}}}{C_x} < \frac{J_x^{\mathbf{g}}}{C_x} = \rho^{\mathbf{g}}$ because $x \in S$ and achieves the maximum cost ratio for $\mathbf{g}$.

Combining the above, we have that $\frac{J_x^{\hat{\mathbf{g}}}}{C_x} < \rho^{\mathbf{g}}$ for all $1 \le x \le L$ when $g_N \notin S$.

Case 2: $g_N \in S$.

This case corresponds to (C.7) and (C.8). We first consider $1 \le x \le g_{N-2}+1$. For these values of x, we have that $\frac{J_x^{\hat{\mathbf{g}}}}{C_x} < \rho^{\mathbf{g}}$ by following similar steps to those used in the first part (case 1) of this proof. As discussed earlier, if $g_N \in S$ then this means that $g_{N-1} = g_N - 1$. In addition, from Lemma 2 we know that $g_{N-1} \notin S$. Therefore, for all $g_{N-2}+2 \le x \le g_{N-1}$, we have:

$$J_x^{\hat{\mathbf{g}}} = (1-p)\sum_{k=1}^{N-1} C_{g_k} + p\left(\sum_{k\in R,\ 0\le k\le N-2} C_{g_k+1} + \sum_{k\notin R,\ 1\le k\le N-2} C_{g_k} + C_N\right) = J_x^{\mathbf{g}} + pM_L$$

which gives:

$$\frac{J_x^{\hat{\mathbf{g}}}}{C_x} = \frac{J_x^{\mathbf{g}} + pM_L}{C_x} < \frac{J_x^{\mathbf{g}}}{C_x} - \frac{J_x^{\mathbf{g}}}{C_x} + \rho^{\mathbf{g}} = \rho^{\mathbf{g}}, \quad (45)$$

where the last inequality follows from inequality (23) defining our chosen p. Since $g_N = g_{N-1}+1$ is the only value of x such that $x > g_{N-1}$, it only remains to prove that $\frac{J_{g_N}^{\hat{\mathbf{g}}}}{C_{g_N}} < \rho^{\mathbf{g}}$. When $x = g_N$, we have the following expected search cost:

$$J_L^{\hat{\mathbf{g}}} = \sum_{k=1}^{N-2} C_{g_k} + pM_{L-2} + (1-p)C_{g_{N-1}} + C_{g_N} = \sum_{k=1}^{N} C_{g_k} + p(M_{L-2} - C_{g_N-1}) < \sum_{k=1}^{N} C_{g_k} = J_L^{\mathbf{g}}, \quad (46)$$

where the last inequality follows from the fact that $C_{g_{N-1}} > M_{L-2}$ using the definition of M_j and fact that $g_{N-1} = L-1$.

Combining these two cases, we have that $\frac{J_x^{\hat{\mathbf{g}}}}{C_x} < \rho^{\mathbf{g}}$ for all $1 \le x \le L$ and have proven this theorem. □

A.5 Proof of Lemma 3

Consider any uniformly randomized TTL sequence $\hat{\mathbf{g}}$. In order to prove Lemma 3, we will first determine the possible values of x such that $\frac{J_x^{\hat{\mathbf{g}}}}{x} = \rho^{\hat{\mathbf{g}}}$.

From (25), each expected TTL value can be calculated as:

$$E[\hat{g}_k] = \frac{g_k + g_{k+1} - 1}{2} \quad (47)$$

Now we can calculate $\rho^{\hat{\mathbf{g}}}$. Because $\mathbf{g}$ is an increasing sequence of positive integers, any positive integer x must lie between two consecutive elements of $\mathbf{g}$ such that $g_n \le x \le g_{n+1}$. Let's rewrite x as $x = g_n + \Delta$, where $0 \le \Delta \le g_{n+1} - g_n$. Then the expected cost $J_x^{\hat{\mathbf{g}}}$ of using a TTL sequence $\hat{\mathbf{g}}$ when the object location is x is given by:

$$J_x^{\hat{\mathbf{g}}} = \sum_{k=1}^{n} E[\hat{g}_k] + P(\hat{g}_n < x)E[\hat{g}_{n+1}] = \sum_{k=1}^{n} E[\hat{g}_k] + \frac{\Delta}{g_{n+1} - g_n}E[\hat{g}_{n+1}] \quad (48)$$

We will show that the ratio $\frac{J_x^{\hat{\mathbf{g}}}}{x}$ is either nonincreasing or nondecreasing for all values of x within $g_n \le x \le g_{n+1}$, and therefore the maximum value of this cost ratio within this range occurs at either $x = g_n$ or $x = g_{n+1}$.

We have for all $g_n \le x \le g_{n+1} - 1$:

$$\begin{aligned}(x+1)J_x^{\hat{\mathbf{g}}} - xJ_{x+1}^{\hat{\mathbf{g}}} &= (g_n + \Delta + 1)\left(\sum_{k=1}^{n} E[\hat{g}_k] + \frac{\Delta E[\hat{g}_{n+1}]}{g_{n+1} - g_n}\right) \\ &\quad - (g_n + \Delta)\left(\sum_{k=1}^{n} E[\hat{g}_k] + \frac{(\Delta+1)E[\hat{g}_{n+1}]}{g_{n+1} - g_n}\right) \\ &= \sum_{k=1}^{n} E[\hat{g}_k] + \frac{\Delta - (g_n + \Delta)}{g_{n+1} - g_n}E[\hat{g}_{n+1}] \\ &= \sum_{k=1}^{n} E[\hat{g}_k] - \frac{g_n}{g_{n+1} - g_n}E[\hat{g}_{n+1}] \quad (49)\end{aligned}$$

Therefore, the sign of the difference between two consecutive cost ratio terms, $\frac{J_x^{\hat{\mathbf{g}}}}{x} - \frac{J_{x+1}^{\hat{\mathbf{g}}}}{x+1} = \frac{(x+1)J_x^{\hat{\mathbf{g}}} - xJ_{x+1}^{\hat{\mathbf{g}}}}{x(x+1)}$, does not change for x in $g_n \le x \le g_{n+1} - 1$ because the numerator of this difference is constant (does not depend on Δ) as given by equation (49) and the denominator is always positive.

Therefore, the cost ratio is either nonincreasing or nondecreasing for x in $g_n \leq x \leq g_{n+1}$, so the maximum cost ratio in this region occurs at either $x = g_n$ or $x = g_{n+1}$. Therefore, the maximum value of the ratio $\frac{J_x^{\hat{\mathbf{g}}}}{x}$ must be obtained at $x = g_m$ for some positive integer m. In other words,

$$\rho^{\hat{\mathbf{g}}} = \sup_{x\in\mathbb{Z}^+} \frac{J_x^{\hat{\mathbf{g}}}}{x} = \sup_{m\in\mathbb{Z}^+} \frac{J_{g_m}^{\hat{\mathbf{g}}}}{g_m} = \sup_{m\in\mathbb{Z}^+} \frac{\sum_{k=1}^{m} E\left[\hat{g}_k\right]}{g_m}$$
$$= \sup_{m\in\mathbb{Z}^+} \frac{\sum_{k=1}^{m} g_k + \frac{g_{m+1}-g_1}{2} - \frac{m}{2}}{g_m} \tag{50}$$

Therefore, we have proven Lemma 3 for any uniformly randomized strategy. □

A.6 Proof of Theorem 4

It has been shown in [8] that the maximum cost ratio for any nonrandom TTL strategy is bounded below by 4, and therefore to calculate the infimum given in (29), we need to only consider uniformly randomized strategies. We will prove Theorem 4 by showing that $\frac{3}{2} + \sqrt{2}$ is both a lower bound and an upper bound on $\inf_{\mathbf{u}\in U'} \rho^{\mathbf{u}}$.

We begin by showing that $\inf_{\mathbf{u}\in U'} \rho^{\mathbf{u}} \geq \frac{3}{2} + \sqrt{2}$. We will proceed using proof by contradiction via a similar method to the one presented in [8] to establish the lower bound on the maximum cost ratio for any nonrandom TTL strategy. Assume that the maximum cost ratio for a uniformly randomized sequence $\hat{\mathbf{g}}$, defined by the boundary values $\mathbf{g} = [g_1, g_2, ...]$, is some constant $C < \frac{3}{2} + \sqrt{2}$. We have already shown that the worst-case ratio for $\hat{\mathbf{g}}$ takes the form given in (50). Therefore, by this equation and the assumption that the maximum ratio is C, then the following must be true for all $m \in \mathbb{Z}^+$:

$$\sum_{k=1}^{m} g_k + \frac{g_{m+1} - g_1}{2} - \frac{m}{2} \leq C g_m$$
$$\Longrightarrow \sum_{k=1}^{m} g_k + \frac{g_{m+1}}{2} \leq C g_m + B_m$$

where $B_m = \frac{g_1}{2} + \frac{m}{2}$. Now define $\tilde{y}_n = \sum_{k=1}^{n} g_k$, so the above equation becomes:

$$\tilde{y}_m + \frac{1}{2}(\tilde{y}_{m+1} - \tilde{y}_m) \leq C(\tilde{y}_m - \tilde{y}_{m-1}) + B_m$$
$$\Longrightarrow \tilde{y}_{m+1} + (1-2C)\tilde{y}_m + 2C\tilde{y}_{m-1} \leq 2B_m$$

Now, because $\mathbf{g}$ is an increasing sequence of positive integers, $\tilde{y}_m$ is increasing faster than B_m. This fact means that for some $N \geq 0$, we have: $\tilde{y}_{N+1} > B_{N+1} + \frac{C}{2} - \frac{1}{4}$. Let $y_k = \tilde{y}_{N+k} - B_{N+k} - \frac{C}{2} + \frac{1}{4}$, so that the y_k are increasing and positive on $\mathbb{Z}^+$. Our above equation then becomes under this new variable with $m = N + k$:

$$y_{k+1} + B_{N+k+1} + \frac{C}{2} - \frac{1}{4}$$
$$+ (1-2C)\left(y_k + B_{N+k} + \frac{C}{2} - \frac{1}{4}\right)$$
$$+ 2C\left(y_{k-1} + B_{N+k-1} + \frac{C}{2} - \frac{1}{4}\right) \leq 2B_{N+k}$$

Rearranging, we obtain:

$$y_{k+1} + (1-2C)y_k + 2Cy_{k-1}$$
$$\leq (2C+1)B_{N+k} - (1-2C)B_{N+k+1} - C + \frac{1}{2}$$

Using the definition of $B_m = \frac{g_1}{2} + \frac{m}{2}$, we obtain:

$$y_{k+1} + (1-2C)y_k + 2Cy_{k-1}$$
$$\leq \frac{N+k}{2}(2C+1) - \frac{N+k+1}{2}$$
$$- \frac{N+k-1}{2}2C - C + \frac{1}{2}$$

Cancelling out terms, we obtain:

$$y_{k+1} + (1-2C)y_k + 2Cy_{k-1} \leq 0 \tag{51}$$

Now, we will prove that (51) cannot hold for all k if $C < \frac{3}{2} + \sqrt{2}$. Form a sequence $...\xi_{-1}, \xi_0 = 0, \xi_1 = 1, \xi_2, ...$ which satisfies the equation $\xi_{l-1} + (1-2C)\xi_l + 2C\xi_{l+1} = 0$. Note that this sequence is uniquely defined by its values $\xi_0 = 0$ and $\xi_1 = 1$. Then the corresponding characteristic equation for this sequence is:

$$1 + (1-2C)\lambda + 2C\lambda^2 = 0 \tag{52}$$

The nature of the roots of this characteristic equation can be determined by calculating $(1-2C)^2 - 4(2C) = 4C^2 - 12C + 1$. Notice that for $C = \frac{3}{2} + \sqrt{2}$, then $4C^2 - 12C + 1 = 0$ and that for $1 \leq C < \frac{3}{2} + \sqrt{2}$, then $4C^2 - 12C + 1 < 0$. In the latter case, the characteristic equation has complex conjugate roots which means that the solution to $\xi_{l-1} + (1-2C)\xi_l + 2C\xi_{l+1} = 0$ has a sinusoidal form. Therefore, there exists some $M \geq 1$ such that $\xi_i > 0$ for $0 < i < M+1$ and that $\xi_{M+1} \leq 0$. Also we know that $\xi_{-1} < 0$ from the recursion defining our sequence. So from equation (51), we have:

$$\sum_{i=1}^{M} \left(y_{i+2} + (1-2C)y_{i+1} + 2Cy_i\right)\xi_i \leq 0 \tag{53}$$

This equation can be arranged into the following:

$$\sum_{i=1}^{M+1} y_i\left(\xi_{i-2} + (1-2C)\xi_{i-1} + 2C\xi_i\right) \tag{54}$$
$$+ \left[-2Cy_{m+1}\xi_{m+1} - (1-2C)y_1\right] \tag{55}$$
$$+ \left[y_{m+2}\xi_m - y_2\xi_0 - y_1\xi_{-1}\right] \leq 0 \tag{56}$$

However, the term in line (54) is zero by the recursion equation for our ξ_i, and the terms in line (55) and on the left-hand side of (56) are both positive due to the fact that $\xi_{-1} < 0, \xi_0 = 0, \xi_1 = 1, \xi_m > 0, \xi_{m+1} < 0$ and $y_i > 0$ for all i. Therefore, we have arrived at a contradiction and it cannot be possible that $C < \frac{3}{2} + \sqrt{2}$. Hence, $\inf_{\mathbf{u}\in U'} \rho^{\mathbf{u}} \geq \frac{3}{2} + \sqrt{2}$.

However, we have already shown that for the uniformly randomized sequence $\hat{\mathbf{g}}$ defined by the boundary values $g_k = \lfloor(\sqrt{2}+1)^{k-1}\rfloor$, the worst-case cost ratio $\rho^{\hat{\mathbf{g}}}$ is $\frac{3}{2} + \sqrt{2}$. It thus follows that $\inf_{\mathbf{u}\in U'} \rho^{\mathbf{u}} \leq \frac{3}{2} + \sqrt{2}$.

Combining these two results, we see that $\inf_{\mathbf{u}\in U'} \rho^{\mathbf{u}} = \frac{3}{2} + \sqrt{2}$. □