Delay Constrained Flooding Search

Nicholas B. Chang and Mingyan Liu
Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI 48109-2122
Email: {changn,mingyan}@eecs.umich.edu

***Abstract*— In this paper we study the problem of using query flooding to find a target (e.g., a node or a piece of data) with unknown location in a network. Specifically, we will consider two types of flooding search methods, one referred to as controlled flooding (CF) and the other suppressed flooding (SF). Under both methods how far the query packet propagates is determined by a time-to-live (TTL) value, decremented for every hop the query traverses. The difference is that under CF, after each failed attempt, the source node times out and initiates a new round of query packet with a larger TTL value, whereas under SF, when the TTL value becomes zero upon reaching certain nodes, those nodes "freeze" or suspend the search and wait for a suppression message from the source node (in case the target is found). They time out if this message does not arrive and resume the propagation of the query packet. We formulate for both methods a constrained optimization problem where the objective is to minimize a worst-case cost measure, subject to a worst-case delay constraint. We derive the solution to this problem under each method and illustrate the cost-delay trade-off inherent in the search problem. These results also highlight the conditions under which one method is preferred over the other.**

***Index Terms*— data query and search, TTL, controlled flooding search, suppressed flooding search, wireless sensor and ad hoc networks, constrained optimization, randomized strategy, competitive analysis**

I. Introduction

Query search is an important functionality for many network applications. Searching for a destination node whose location is unknown is a prime example frequently encountered by ad hoc network routing protocols and services, e.g., [1], [2]. Other examples include the search for certain data of interest in an environmental monitoring sensor network [4], and more broadly, the search for a shared file in a peer-to-peer (P2P) network. A good search mechanism should have a short response time and should incur minimal cost.

There are a variety of mechanisms one may use to conduct search. These include maintaining a centralized directory service, or by sending out a query packet that traverses the network in a certain way [4]. In this paper we focus on two types of search schemes, both within the class of flooding search. The first is the conventional *controlled flooding (CF)* [6], [7]. Under this scheme the node originating the search (also referred to as the *source* node) sends out (broadcasts) a query packet that carries an integer TTL (time-to-live) value. If the search target is found at a neighboring node (or at anytime during the search process), it will reply to the source. Otherwise it decrements the TTL value by one and rebroadcasts the query packet. This continues until the TTL reaches zero. If the target is not found in this search area, the source node will eventually time out and initiate another *round* of search covering a bigger area using a larger TTL value, and presumably setting a larger timeout value for that round of search. This process continues until either the object is found or the source gives up. Hence the performance of a search strategy both in terms of cost and delay is determined by the sequence of TTL values used. Controlled flooding search has previously been studied in [7], [8], [6], [9]. In particular, in [9] we derived optimal CF strategies that minimize a worst-case search cost measure, and in [10] we further considered the search delay involved in CF strategies.

The second type of flooding is referred to as *suppressed flooding (SF)*, also using TTL values. Similar to CF, initially the source node sends out a query packet that is propagated through the network. The difference is that when the TTL is decremented to zero at some node, that node will *freeze* or *suspend* the search rather than discarding the query packet. Such a *frozen* node then waits for a *suppression message* from the source to formally terminate the search process; otherwise it times out and *resumes* the search with a new TTL value. The source only sends out the suppression message if it has received a reply from the target. Note that under SF, timeouts occur at the frozen nodes rather than the source node (as in CF). These timeout values must be set high enough to ensure that a possible suppression message will reach them before the next round of search begins. Thus it is possible that using SF could incur a high delay. On the other hand, SF may have lower cost than CF, because under CF query packets must propagate from the source at the start of search round, whereas under SF the query packets only have to propagate from the previous round's frozen nodes to the new set of frozen nodes for the current round.

The primary goal of this study is to compare controlled and suppressed flooding strategies in terms of the search cost and the search delay. Specifically, we will formulate a constrained optimization problem by minimizing a cost measure subject to a delay constraint. The rest of the paper is organized as follows. In Section II we present the model and relevant assumptions. Section III gives the formulation and main results, as well as a discussion. Section IV concludes the paper.

This work is supported by NSF award ANI-0238035, ONR grant N00014-03-1-0232, and through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011.

II. Network Model

A. Model and Assumptions

We will limit our analysis to the case of searching for a single target, which is assumed to exist in the network. 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 measure the position of an object by its distance to the source initiating the search, measured in *hops*. We will use the term *object location* to indicate the minimum TTL value needed to locate the object, denoted by X. The term *network dimension* refers to the minimum TTL required to reach every node in the network, denoted by L. Also, $\bar{F}_X(u) = P(X > u)$ denotes the tail distribution of the random variable X.

We will assume that when the source times out, a TTL value of u will have reached all nodes within u hops of the source and will find the object with probability 1 if it is located within u hops. This assumption implies that (1) the query propagation process is reliable and that (2) the timeout values are set perfectly, such that a timeout event is equivalent to not finding the object in the u-hop neighborhood. This assumption is a simplification if the network is lossy. In addition, random delay experienced in the network may cause timer to expire prematurely. It nevertheless allows us to reveal some very interesting fundamental features of the problem and obtain valuable insights.

B. Search Strategies

For a CF scheme, a search strategy can be described by a sequence of TTL values, denoted by $\mathbf{u} = [u_1, u_2, \cdots, u_N]$ of certain length N. It can be either fixed/deterministic or random. For a fixed strategy we assume that $\mathbf{u}$ is an increasing sequence. For randomized strategies, we assume all realizations are increasing sequences. Note that in a specific search experiment we may not need to use the entire sequence; the search stops whenever the object is found.

For an SF scheme, the search strategy is uniquely defined by the distance of the frozen nodes from the source. It can be similarly described by a sequence $\mathbf{u}$: in the first round the query reaches all nodes within u_1 hops of the source. Nodes that are exactly u_1 hops away from the source freeze at the end of this round. If the object is not found, then a timeout will occur at these frozen nodes. They then begin a new round of search by propagating the query to all nodes that are u_2 hops away from the source (and thus $u_2 - u_1$ hops away from these frozen nodes). We will also refer to an SF strategy $\mathbf{u}$ as a TTL sequence, and refer to element u_k as the kth TTL value, because this is the TTL value used by the source node on the kth round if it sends out a suppression message. Note the frozen nodes set the TTL to $u_k - u_{k-1}$ for the kth round. It can be seen that using the same TTL sequence for both CF and SF will successively search the same regions, although the underlying methods are quite different.

In practice, for both schemes it is natural to only consider integer-valued (*discrete*) policies. However, considering real-valued sequences proves to be helpful in deriving optimal integer-valued strategies. For this reason we will also consider *continuous* (real-valued) strategies, denoted by $\mathbf{v}$, where $\mathbf{v} = [v_1, v_2, \cdots, v_N]$, and v_i is either fixed or a continuous random variable. When considering discrete strategies, TTL values are integers and the object location X is assumed to be a positive integer taking values between 1 and L. In analyzing continuous strategies, X is assumed to be a real number in the interval $[1, L]$.

A strategy is *admissible* if it locates any object of finite location with probability 1. For a fixed strategy this implies $u_N = L$. For a random strategy, this implies $Pr\,(u_i = L \text{ for some } 1 \le i \le N) = 1$. In the asymptotic case as $L \to \infty$, a strategy $\mathbf{u}$ is admissible if $\forall\ x \ge 1,\ Pr\,(u_n \ge x \text{ for some } n \in \mathbb{Z}^+) = 1$, implying that $\mathbf{u}$ is an infinite-length vector in the asymptotic case. We let U and V denote the set of all real-valued and integer-valued admissible strategies (random and fixed), respectively.

C. Search Cost and Delay

For both CF and SF, we will let $C(u)$ denote the cost of sending the query from the source to all nodes within u hops of the source. The functional form of this cost will depend on the properties of the network as well as the underlying broadcast techniques used. Similarly, $C_s(u)$ denotes the cost of sending the suppression message to all nodes within u hops of the source. This cost may be smaller than $C(u)$ if the suppression message is in the form of a much smaller signaling packet.

Note that in general, a node receiving the search query will be unaware whether the object is found at another node in the same round. Thus this node will continue the process by decrementing the TTL value. Therefore the query search cost for each round is determined by the TTL value and not by whether the object is located in that round. Similarly, a suppression message will propagate till the TTL reaches 0, thus its cost is also completely determined by the TTL value.

For CF, we denote by $D_t(u)$ the source timeout value used when searching with TTL u. This is the delay incurred when the object is not found using u, i.e., when $u < X$. On the other hand, if $u \ge X$, then the delay incurred in this case is the amount of time it takes for the query to propagate X hops and for the reply to reach back to the source. We will denote this delay by $D_r(X)$ for object location X. Thus the search delay of using TTL value u under CF can be written as: $I(u < X)D_t\,(u) + I(X \le u)D_r\,(X)$, where I is the indicator function: $I(A) = 1$ if A is true and 0 otherwise.

For SF, we let $d_t(u, \tilde{u})$ denote the amount of time it takes for the query to travel from nodes at distance u from the source to nodes that are at distance $\tilde{u}$. We let $u = 0$ indicate the source. Similarly, $d_r(u, 0)$ denotes the amount of time it takes for the reply message to travel from a node at distance u back to the source. Thus the delay $D_r(.)$ defined earlier can be written as $D_r(u) = d_t(0, u) + d_r(u, 0)$.

In the kth round of SF, if $u_{k-1} < X \le u_k$ then the query travels from frozen nodes to the target, which sends back a reply to the source. This takes a total time of $d_t(u_{k-1}, X) + d_r(X, 0)$. Otherwise, $X > u_k$ and a timeout occurs at the frozen nodes. Note that if we assume that the suppression message travels at the same speed as the query, then the timeout at the frozen nodes should be set to $D_t(u_k)$, the same as that set by the source node under CF. The time it takes for the query to reach these frozen nodes plus the timeout associated with the suppression message is thus equal to $d_t(u_{k-1}, u_k) + D_t(u_k)$. For the rest of our analysis we will assume that the suppression message travels at the same speed as the query packet, and that $d_t(u_{k-1}, u_k) = d_t(0, u_k) - d_t(0, u_{k-1})$.

For real-valued sequences, we require that the above cost and delay functions be defined for all $v \in [1, \infty)$, while for integer-valued sequences we only require that these functions be defined for positive integers. When the cost function is invertible, we write $C^{-1}(\cdot)$ to denote its inverse. We denote by $\mathbb{C}$ the class of cost functions $C : [1, \infty) \to [C(1), \infty)$, that are increasing, differentiable, and have the property $\lim_{v\to\infty} C(v) = \infty$. Finally, let $\mathbb{C}_1$ denote the set of cost functions $C(\cdot) \in \mathbb{C}$ such that $\lim_{x\to\infty} \frac{C(x+1)}{C(x)} = 1$. Note that this subset contains all polynomial cost functions.

III. Problem Formulation and Main Results

A. Problem Formulation

We will consider the search performance in the asymptotic regime as $L \to \infty$. This is because it is difficult if at all possible to obtain a general strategy that is optimal for all finite-dimension networks as the optimal TTL sequence often depends on the specific value of L. In this sense, an asymptotically optimal strategy may provide much more insight into the intrinsic structure of the problem.

Let $J_X^{\mathbf{u}}$ denote the expected search cost of using CF strategy $\mathbf{u}$ when the object location is X. This quantity can be calculated as follows:

$$J_X^{\mathbf{u}} = E_{\mathbf{u}} E_X \left[\sum_{k=1}^{\infty} I\left(X > u_{k-1}\right) C(u_k) \right] \tag{1}$$

$$= E_{\mathbf{u}} \left[\sum_{k=1}^{\infty} \bar{F}_X(u_{k-1}) C(u_k) \right] , \tag{2}$$

where $u_0 = 0$. We will drop the subscript when it is clear which variable the expectation is taken with respect to. Similarly, let $H_X^{\mathbf{u}}$ denote the expected search cost of using SF strategy $\mathbf{u}$ when the object location is X. This can be calculated as follows:

$$\begin{aligned} H_X^{\mathbf{u}} &= E_{\mathbf{u}} E_X \left[\sum_{k=1}^{\infty} I\left(u_{k-1} < X \le u_k\right) \left(C(u_k) + C_s(u_k)\right) \right] \\ &= E_{\mathbf{u}} \left[\sum_{k=1}^{\infty} \left(\bar{F}_X(u_{k-1}) - \bar{F}_X(u_k)\right) \left(C(u_k) + C_s(u_k)\right) \right] , \end{aligned} \tag{3}$$

Let $D_X^{\mathbf{u}}$ denote the expected search delay induced by CF strategy $\mathbf{u}$ for X. This quantity can be calculated as follows:

$$\begin{aligned} D_X^{\mathbf{u}} &= E_{\mathbf{u}} E_X \left[\sum_{k=1}^{\infty} I(X > u_k) D_t(u_k) \right] \\ &\quad + E_{\mathbf{u}} E_X \left[\sum_{k=1}^{\infty} I(u_k \ge X > u_{k-1}) D_r(X) \right] \\ &= E_{\mathbf{u}} \left[\sum_{k=1}^{\infty} \bar{F}_X(u_k) D_t(u_k) \right] + E_X\left[D_r(X)\right] \ . \end{aligned}$$

Similarly, for SF the expected delay $T_X^{\mathbf{u}}$ is given by:

$$\begin{aligned} T_X^{\mathbf{u}} &= E_{\mathbf{u}} E_X \left[\sum_{k=1}^{\infty} I(X > u_k)(d_t(u_{k-1}, u_k) + D_t(u_k)) \right] \\ &\quad + E_{\mathbf{u}} E_X \left[\sum_{k=1}^{\infty} I(u_k \ge X > u_{k-1})(d_t(u_{k-1}, X) + d_r(X, 0)) \right] \\ &= E_{\mathbf{u}} \left[\sum_{k=1}^{\infty} \bar{F}_X(u_k) D_t(u_k) \right] + E_X\left[D_r(X)\right] \ . \end{aligned}$$

which turns out to be identical to $D_X^{\mathbf{u}}$.

In this study we adopt a worst-case performance measure. Specifically we can measure the performance of a CF strategy $\mathbf{u}$ by the following competitive ratio (or worst-case cost ratio):

$$\rho^{\mathbf{u}} = \sup_{\{p_X(x)\}} \frac{J_X^{\mathbf{u}}}{E[C(X)]} , \tag{4}$$

where $\{p_X(x)\}$ denotes the set of all probability distributions for X such that $E[C(X)] < \infty$, and $E[C(X)]$ denotes the expected cost of an omniscient observer. The worst-case cost ratio of an SF strategy is similarly defined.

Similarly, the *worst-case delay ratio* of a CF strategy $\mathbf{u}$ is given by:

$$\tau^{\mathbf{u}} = \sup_{\{p_X(x)\}} \frac{D_X^{\mathbf{u}}}{E[D_r(X)]} , \tag{5}$$

where we note in this case $\{p_X(x)\}$ is the set of all location distributions such that $E[D_r(X)] < \infty$. Again the worst-case delay ratio for SF is similarly defined.

With the above cost and delay measures, we formulate the constrained optimization problem as follows. Consider the set of CF strategies:

$$U_d = \left\{ \mathbf{u} \in U : \sup_{\{p_X(x)\}} \frac{D_X^{\mathbf{u}}}{E[D_r(X)]} \le d \right\} , \tag{6}$$

for some constant $d > 1$. This is the set of all strategies whose delay is always within a factor d of the delay of the omniscient observer, regardless of X. We will call d the *delay constraint*. Note that as $d \to \infty$, the delay constraint becomes less restrictive and the set U_d approaches U.

We seek a CF strategy that satisfies this delay constraint d and has the smallest worst-case cost ratio:

$$\rho_d^* = \inf_{\mathbf{u} \in U_d} \sup_{\{p_X(x)\}} \frac{J_X^{\mathbf{u}}}{E[C(X)]} . \tag{7}$$

This constitutes our constrained optimization problem (P). Note that the two supremums in (P), one in the objective function and the other in the constraint (6), are in general *not* achieved under the same distribution $p_X(x)$. A similar optimization problem can be formulated for SF strategies in a straightforward manner.

The above definitions also hold analogously for continuous strategies. We will thus denote $\rho^{\mathbf{v}}$, $\tau^{\mathbf{v}}$ and V_d as the continuous versions of (4), (5), (6), respectively.

We have shown [10] that for problem (P), there is no loss in generality in assuming that $D_t(\cdot) = D_r(\cdot)$. We thus let $D(u) = D_t(u) = D_r(u)$ for all u. It follows that using a TTL value u for object location X will incur a delay of $D(\min\{X, u\})$. We will also use the same notation $D_X^{\mathbf{u}}$ to denote the expected delay for both CF and SF strategies $\mathbf{u}$, since they are the same. This implies that we can use the same notation U_d and V_d to describe the same classes of CF and SF strategies.

B. Main Results

We define the following class of continuous strategies:

Definition 1: Assume that the cost function $C(\cdot) \in \mathbb{C}$. Let $\mathbf{v}[r, F_{v_1}(\cdot)]$ denote a jointly defined sequence $\mathbf{v} = [v_1, v_2, ...]$ generated as follows:

(J.1) The first TTL value v_1 is a continuous random variable taking values in the interval $\left[1, C^{-1}(r \cdot C(1))\right)$, with its cdf given by some nondecreasing, right-continuous function $F_{v_1}(x) = Pr(v_1 \leq x)$. Note that this means $F_{v_1}(1) = 0$ and $F_{v_1}\left(C^{-1}(r \cdot C(1))\right) = 1$.

(J.2) The k-th TTL value v_k is defined by $v_k = C^{-1}\left(r^{k-1} C(v_1)\right)$ for all positive integers k.

We see that that given the selection of v_1, the cost of successive TTL values essentially form a geometric sequence of base r.

Our main theorems regarding the class of continuous CF and SF strategies are as follows.

Theorem 1: When $C(\cdot) \in \mathbb{C}$ and $C(\cdot) = \beta D(\cdot)^m$ for some $m, \beta > 0$, we have:

(1) For any fixed $1 < d < m+1$,

$$\inf_{\mathbf{v} \in V_d} \sup_{\{f_X(x)\}} \frac{J_X^{\mathbf{v}}}{E[C(X)]} = \frac{(d-1)}{m} e^{\frac{m}{d-1}} . \quad (8)$$

Moreover, this minimum worst-case ratio is achieved by using the CF strategy $\mathbf{v}[r, \frac{1}{\ln r} \ln \frac{C(\cdot)}{C(1)}]$ with $r = e^{\frac{m}{d-1}}$.

(2) For $d \geq m+1$, we have:

$$\inf_{\mathbf{v} \in V_d} \sup_{\{f_X(x)\}} \frac{J_X^{\mathbf{v}}}{E[C(X)]} = e . \quad (9)$$

Moreover, this minimum worst-case ratio is achieved by using the strategy $\mathbf{v}[r, \frac{1}{\ln r} \ln \frac{C(\cdot)}{C(1)}]$ with $r = e$.

Theorem 2: When $C(\cdot) \in \mathbb{C}$, $C(\cdot) = \beta D(\cdot)^m$ for some $m, \beta > 0$, and $C_s(\cdot) = \alpha C(\cdot)$ for some $\alpha > 0$, we have for any fixed $d > 1$,

$$\inf_{\mathbf{v} \in V_d} \sup_{\{f_X(x)\}} \frac{H_X^{\mathbf{v}}}{E[C(X)]} = (1+\alpha)\frac{(d-1)}{m} \left[e^{\frac{m}{d-1}} - 1\right] . \quad (10)$$

This minimum worst-case ratio is achieved by using the SF strategy $\mathbf{v}[r, \frac{1}{\ln r} \ln \frac{C(\cdot)}{C(1)}]$ with $r = e^{\frac{m}{d-1}}$.

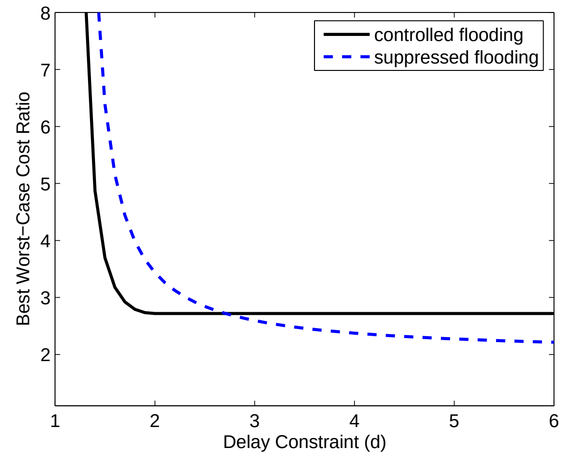

Fig. 1. When $C(\cdot) = \beta D(\cdot)$ and $\alpha = 1$, plot of the minimum worst-case cost ratio as a function of the delay constraint d for both CF and SF schemes.

Note the optimal strategies of Theorems 1 and 2 can be adjusted for different delay constraints by varying the parameter r. For discrete strategies we have the following.

Theorem 3: When $C(\cdot) \in \mathbb{C}$ and $C(\cdot) = \beta D(\cdot)^m$ for some $m, \beta > 0$, we have:

(1) For $1 < d < m+1$,

$$\inf_{\mathbf{u} \in U_d} \sup_{\{p_X(x)\}} \frac{J_X^{\mathbf{u}}}{E[C(X)]} \leq \frac{(d-1)}{m} e^{\frac{m}{d-1}} . \quad (11)$$

(2) For $d \geq m+1$,

$$\inf_{\mathbf{u} \in U_d} \sup_{\{p_X(x)\}} \frac{J_X^{\mathbf{u}}}{E[C(X)]} \leq e . \quad (12)$$

Theorem 4: When $C(\cdot) \in \mathbb{C}$, $C(\cdot) = \beta D(\cdot)^m$ for some $m, \beta > 0$, and $C_s(\cdot) = \alpha C(\cdot)$ for some $\alpha > 0$, we have for any fixed $d > 1$:

$$\inf_{\mathbf{u} \in U_d} \sup_{\{p_X(x)\}} \frac{H_X^{\mathbf{u}}}{E[C(X)]} \leq (1+\alpha)\frac{(d-1)}{m} \left[e^{\frac{m}{d-1}} - 1\right] . \quad (13)$$

Whether the upper bounds in Theorems 3 and 4 become equalities appears to depend on the specific cost function $C(\cdot)$. By restricting our attention to cost functions $C(\cdot) \in \mathbb{C}_1$, the inequalities in the previous two theorems can be achieved via discrete strategies $\mathbf{u}^* = \lfloor \mathbf{v}^* \rfloor$, by taking the floor (operated on each element of the vector) of the corresponding optimal continuous strategies for CF and SF, respectively.

C. Discussion

The differentiation between the two cases, $1 < d < m+1$ vs. $d \geq m+1$, in all Theorems 1 and 3 is due to the fact that the optimization problem (P) for CF has an active/binding constraint in the former, and an inactive/non-binding constraint in the latter.

The main results rely on the relationship $C(\cdot) = \beta D(\cdot)^m$ for some $m, \beta > 0$, where the factor m describes the relative rate at which the cost and delay functions grow with respect

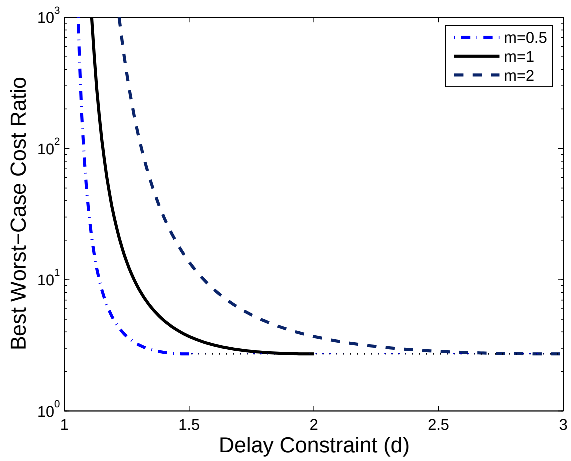

Fig. 2. When $C(\cdot) = \beta D(\cdot)^m$, a logarithmic plot of the minimum worst-case cost ratio as a function of the delay constraint d. Dotted portions indicate when the delay constraint is not binding and hence the unconstrained strategy of Theorem 1, part (2) is optimal. For $d \geq 3$, the best worst-case cost ratio is e for all three curves.

to TTL. First note that the constant positive factor β cancels out in the cost or delay ratio calculated in (4) and (5). Hence we can assume $\beta = 1$ without loss of generality. Secondly, the relationship $C(\cdot) = D(\cdot)^m$ holds, for example, in a very representative case of searching in a 2-dimensional network with search cost proportional to the number of transmissions incurred. In this case $C(v)$ is well approximated by a quadratic function (see e.g., [7], [8]) and $D(v)$ can be chosen to be a linear function of v (implying $m = 2$), or quadratic (implying $m = 1$).

Finally, the condition $C_s(\cdot) = \alpha C(\cdot)$, for $\alpha > 0$, describes the cost of sending the suppression message relative to that of sending the query. In general, $\alpha < 1$ is desired, indicating sending the suppression message is cheaper (e.g., smaller packet size).

Figure 1 depicts the minimum cost ratio for the CF and SF schemes when $C_s(\cdot) = C(\cdot) = \beta D(\cdot)$ (so $\alpha = m = 1$). Note that for large d, the suppressed flooding scheme performs better. This is because if delay is not a factor, then the minimum cost strategy is to increase TTL by the smallest possible increment after every round. Such a strategy incurs high delay as the frozen nodes must wait for possible suppression message, but only a minimal cost is committed for every round. Thus for larger delay tolerance, the suppressed flooding scheme performs better because more low-cost strategies are admissible. On the other hand, when d is small then the opposite is true. Since waiting for the suppression message can incur a high delay, under a more stringent delay requirement it is harder for SF to achieve low-cost.

Figure 2 depicts the trade-off between optimal worst-case cost ratio under CF as given by Theorem 1 and the delay constraint d when $C(\cdot) = \beta D(\cdot)^m$. The dotted portion of each curve indicates when the delay constraint becomes non-binding, i.e., for $d \geq m + 1 = 1.5, 2, 3$, respectively. In these cases the optimal unconstrained strategy (using $r = e$) has a minimum worst-case cost ratio of e. Note that the plot is logarithmic. As d approaches 1 from above, the best worst-case cost ratio approaches ∞ for all m. Hence, as the constraint on delay becomes tighter, the minimum worst-case cost increases unboundedly.

IV. Conclusion

In this paper we studied the cost and delay performance of a type of TTL-based controlled flooding search methods. In particular, we analyzed and compared two such methods, the controlled flooding (CF) and the suppressed flooding (SF). We presented a constrained optimization framework in order to derive strategies that minimize a worst-case search cost measure subject to a worst-case search delay constraint. We derived the solution to this problem under each method and illustrated the inherent cost-delay trade-offs. These results also highlight the conditions under which one search method is preferred over the other.

References

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

[2] 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.

[3] 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.

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

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

[6] 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.

[7] 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.

[8] N. Chang and M. Liu, "Revisiting the ttl-based controlled flooding search: Optimality and randomization," *Proceedings of the Tenth Annual International Conference on Mobile Computing and Networks (MobiCom'04)*, pp. 85–99, September 2004, Philadelphia, PA.

[9] N.B.Chang and M. Liu, "Optimal controlled flooding search in a large wireless network," *Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'05)*, pp. 229–237, April 2005, Riva Del Garda, Italy.

[10] N.Chang and M.Liu, "Controlled flooding search with delay constraints," *Proceedings of IEEE Infocom*, April 2006, Barcelona, Spain.

[11] N. Chang and M. Liu, "Controlled flooding search with delay constraints," *EECS Technical Report CGR 05-08*, 2005, University of Michigan, Ann Arbor, http://www.eecs.umich.edu/~mingyan/pub.html.