Flooding Strategy for Target Discovery in Wireless Networks

Zhao Cheng
Department of Electrical and Computer Engineering
University of Rochester
Rochester, NY 14627
zhcheng@ece.rochester.edu

Wendi B. Heinzelman
Department of Electrical and Computer Engineering
University of Rochester
Rochester, NY 14627
wheinzel@ece.rochester.edu

ABSTRACT

In this paper, we address a fundamental problem concerning the best flooding strategy to minimize cost and latency for target discovery in wireless networks. Should we flood the network only once to search for the target, or should we apply a so-called "expansion ring" mechanism to reduce the cost? If the "expansion ring" mechanism is better in terms of the average cost, how many rings should there be and what should be the *radius* of each ring? We separate wireless networks based on network scale and flooding control methods and explore these questions. We prove that when using a geography-based flooding control method, the number of flooding attempts should be one. When using a hop-based flooding control method, we prove that two-tier and three-tier schemes can reduce the cost of flooding compared to a single attempt. We provide a general formula to determine good parameters for the two-tier and three-tier hop-based flooding schemes. Through simulations, we show that choosing flooding parameters according to our techniques gives performance close to that of ideal flooding schemes. Also, we present some preliminary results on flooding strategy with caching and show that a properly chosen searching radius can save much more query overhead than a simple radius selection scheme.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Query formulation

General Terms

Design, Theory, Experimentation

Keywords

Mathematical induction, Expansion ring, Caching

1. INTRODUCTION

Flooding is a basic operation and has extensive applications in target discovery in wireless networks, such as those widely utilized in the route discovery process in several routing protocols [1], [2], those used in wireless sensor networks for sensor discovery [3], or those used in wireless ad hoc networks for service discovery [4]. Query packets are flooded inside the network to search for a certain target node. When the target node receives the query packet, it responds to the source node, not only to inform the source node about its existence, but also to avoid further unnecessary flooding attempts from the source node.

However, even if the flooded packet reaches the target, packets flooded towards other directions continue. Different types of networks introduce different methods to control the flooding. Flooding may be controlled by setting the hop limit, which guarantees a packet will not be transmitted more than the maximum number of hops, or by setting a distance limit from the source, which guarantees a packet will not be transmitted beyond a certain geographical limit.

Using hop limit, the authors of DSR consider a mechanism called "expansion ring" to search for a target [1] (see Section 2 for a description of expansion ring). The authors claim that in this way, a node can explore for the target progressively without flooding the entire network, and the only drawback of this scheme is the increasing latency due to multiple discovery attempts. However, DSR applies a simpler scheme that searches the one-hop neighbors first and then the entire network. The idea of the expansion ring was implemented later in AODV [2]. An interesting question is whether or not using the expansion ring technique always reduces flooding overhead compared with flooding the network just once. If not, when and how should the expansion ring technique be applied?

We can generalize this question as follows. With a certain restriction on the searching scheme, either the maximum hop limit or the longest distance from the source node, how many attempts should we take to achieve minimum cost or latency to find the target? If multiple attempts are better than one, what should be the *radius* for each of these attempts? What if we are looking at small-scale networks in which a node has no restrictions in flooding and is allowed to search the entire network? How much effect will a caching scheme have on the above questions?

In this paper, we explore these questions and provide analytic solutions for each type of network. However, our contributions are more than just solving the problem. First, we propose a general framework to model and analyze flooding schemes in wireless networks. Second, except for some heuristic results to support further analysis, most of our conclusions can be directly applied to existing networks and protocols. Finally, we clarify that the "expansion ring" scheme can reduce the flooding cost only under a relatively strong caching condition with the caching parameters chosen properly[1]. To the best of our knowledge, this is the first formal study undertaken to compare different flooding strategies.

The rest of this paper is organized as follows. Section 2 provides an overview on the previous efforts in reducing discovery overhead and some other related work. Section 3 characterizes wireless networks according to different flooding methods. We also present the analytic framework and procedures we will use throughout this paper. In Section 4, we address the flooding problem in large-scale networks for hop-based flooding control mechanisms. In Section 5, we extend the problem to small-scale networks with no restrictions, that is, nodes may flood the entire network if necessary. Section 6 provides extensive simulations and compares the performance of our schemes with existing schemes and ideal schemes in terms of both cost and latency. Section 7 analyzes the effects of caching on the expansion ring scheme and shows some preliminary results. Section 8 concludes the paper.

2. RELATED WORK

The most common use of target discovery is in routing protocol implementations. Typical examples are DSR [1] and AODV [2]. In both protocols, nodes search for a target gradually in order to avoid flooding the entire network. The procedure in DSR is relatively simple. A node searches its one-hop neighbors first, and if the target is not found, the node then searches the entire network. AODV uses a different approach, where a node increases its searching radius linearly from an initial value until it reaches a predefined threshold. After that, a network-wide search has to be performed. Both of these expansion ring schemes assume that there are route caches residing in the nodes. When the route caching condition is weak and node mobility is high, these schemes will not reduce the search overhead compared with flooding the entire network once as effectively as expected. Instead, an improper utilization of the expansion ring scheme will lead to even more overhead. How to discover a target efficiently under no caching or weak caching condition is the main topic of this paper.

There has also been some work on target discovery in sensor networks, such as ACQUIRE [5] and rumor routing schemes [6]. ACQUIRE avoids flooding and traditional query/response stages by refining the query into sub-queries and resolving each sub-query by means of local searching and random forwarding. Rumor routing manages to find an optimal balance point between query flooding and event notifications flooding. Both these two algorithms show a better performance than the basic flooding search algorithms. However, both of them are confined in certain circumstances and are application-specific. In contrast, the conclusions presented in this paper can be applied to general target discovery scenarios.

There are some other routing protocols that reduce discovery overhead by assuming there are some location-based devices to aid the routing [7]. The cost and the inaccuracy of these devices confine the implementation of these protocols. In our paper, we do not assume the existence of such devices.

Caching is a very important factor in reducing the target discovery overhead. Many efforts and studies have been done on caching optimization [1, 9, 10, 11], most of which are in the on-demand routing area. A previous solution on how to efficiently use caching can be found in [8]. However, its expansion ring parameters are arbitrarily chosen from simulation results and its contribution is on whether the expansion ring should be applied or not. In this paper, after the analysis for no caching scenarios, we also study the discovery strategies under caching scenarios. Questions such as what factors determine the caching effects and how to effectively utilize these caches are studied. However, this is work in progress, and we will only provide some preliminary results in this paper.

3. ASSUMPTIONS, TERMINOLOGY AND ANALYSIS METHODOLOGY

In order to provide a general answer to the target discovery questions posed in the introduction, we assume that there is no caching (we will address caching in Section 7).

All of the analysis in Sections 4 and 5 is based on blind flooding. In this basic flooding scheme, nodes forward a packet once and only once, only if they are not the destination of the packet and the nodes are allowed to forward, e.g., the hop limit is not zero, or the nodes are inside the geographical searching region. Also, we ignore the potential increase of the packet length and the cost it brings during packet propagating. For simplicity, we also ignore potential packet collisions, which can be effectively decreased by inserting a random delay time before forwarding. Also, we assume that the first flooded packet that arrives at a certain node follows the shortest path. This assumption is valid since packets following the shortest path usually need less transmission time and delay.

During our analysis, we assume we are studying a snapshot of the network and nodes are static during the analysis. However, even if nodes are mobile, there are several reasons that our analysis is still valid. First, the flooding searching time is short and nodes will not move too far away. Second, we are looking at broadcasting which does not have the problems of unicasting such as link failures. Third, since nodes are moving randomly and independently, the number of nodes in a certain region is stable and will not have adverse effects on our analysis.

The main part of this paper is organized according to the following methodology. First, we classify wireless networks into two categories, large-scale networks and small-scale networks, which provide different assumptions for our analysis. For large-scale networks, we ignore edge effects and assume every node is identical; while for small-scale networks, nodes are no longer the absolute *center* of the flooded area and

[1]By the term *strong caching condition* we mean that the existence of caching among the nodes is very likely. Typically a strong caching condition occurs when there is a large amount of traffic that refreshes the caches in the intermediate nodes before they become stale. Similar terms such as *moderate caching condition* and *weak caching condition* will appear in this paper.

Table 1: Notations used throughout this paper.

C^n	Cost of an n-tier scheme
L^n	Latency of an n-tier scheme
h	Number of hops
M	Maximum hop number
N_T	Total number of nodes
N_i	Number of nodes exactly i hops away
r	Distance between the target and the source
$D_{i,j}$	Cost difference between an i-tier and a j-tier scheme
ρ	Node density
A	Area of a flooded region
R_t	Node transmission range

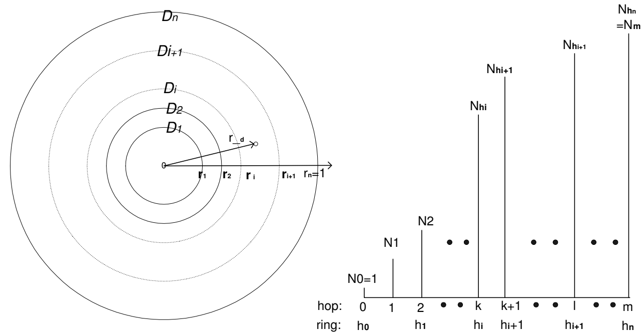

Figure 1: Models of target discovery for large-scale wireless networks. On the left is the geography-based flooding control model, and on the right is the hop-based flooding control model.

edge effects have a significant impact on the desired flooding scheme.

We further subdivide these networks by the type of flooding control employed. Using hop-based flooding control, the source node will set a hop limit that a flooded packet can reach. Upon receiving this packet, intermediate nodes decrement the value and check the value. If the value becomes zero, the node will discard the packet. If not, the node will forward the the packet with the decremented value. Using geography-based flooding control, the source node will set a distance limit for a flooded packet, and only nodes inside the distance limit will forward the packet.

We will focus on two metrics: cost and latency. Cost is defined as the total number of packets transmitted and is closely related to the consumed energy. Latency is defined as the total time that the source node takes to receive a response from the target node. Our analysis concentrates on hop-based networks since hop-based networks are more widely used and do not require nodes to know their location.

Besides theoretical interests, most of our analysis results can be directly employed into realistic wireless network applications and protocols. For example, the analysis results for hop-based flooding in large-scale networks can be employed in DSR route discovery and can be even extended to wired networks. The analysis for geography-based flooding in small-scale networks can be applied in sensor networks for sensor discovery. The results from the hop-based flooding in small-scale networks can be applied to service discovery protocols in ad hoc networks.

For quick reference, we list our terms and notations in this paragraph. *Once-for-all* is a strategy that nodes only flood once to discover a target. *n-tier* is a strategy that nodes attempt at most n times to discover a node. *Two-tier* and *three-tier* are special n-tier strategies and are fully studied. Other notations are listed in Table 1.

4. RESTRICTED FLOODING IN LARGE-SCALE NETWORKS

In this section, we consider the flooding strategy problem in large-scale networks. The most important property of this type of network is that flooding is limited to a small area compared to the whole network. The border of the network is far away from the reaching range of the flooding process, so edge effects are ignored during analysis.

In a general case, nodes adopt a hop limit scheme to control the flooding. Let us model the question under this hop-based assumption as shown in the right part of Fig. 1.

A large number of nodes are placed uniformly and independently in a two-dimensional space $\mathbb{R}^2$ with node density ρ. A node wants to search nodes within the hop limit M. Suppose the number of nodes that are exactly i hops away from the source node is N_i, for $i \in \{1 \dots M\}$.

Suppose we apply an n-tier scheme, $n \geq 1$. For the ith attempt, we set the hop limit to h_i. Thus we have a corresponding hop set $H = \{h_i, i \in \{1 \dots n\}, h_i \in \{1 \dots M\}\}$. The source node will start searching by setting the hop limit to h_1; if it fails for the ith attempt, it will set the searching limit to h_{i+1} and take the $(i+1)$st attempt, until searching for the last time by setting the hop limit to $h_n = M$[2]. The question becomes: what is the optimal number of attempts n, and what is the optimal set H to achieve minimum average cost and minimum average latency, given a specific n?

To aid our analysis, we define $N_0 = 1$ and $h_0 = 0$, which can be understood as the number of nodes that are 0 hops away from the source node is 1, the source node itself.

4.1 Cost

To flood with hop count set to h_k, the average cost equals the number of nodes whose distance is less than h_k hops. Note that nodes that are exactly h_k hops away will not forward the packet and are not taken into account for cost calculation. That is, $co_k = \sum_{j=1}^{h_k} N_{j-1}$.

Suppose the target node is h_d hops from the source node and $h_i < h_d \leq h_{i+1}$, for $i \in \{0 \dots n-1\}$. To find the target, the source node has to fail for the first i attempts and succeed at the $(i+1)$st attempt. Thus the total cost is $C_i = \sum_{k=1}^{i+1} co_k = \sum_{k=1}^{i+1} \sum_{j=1}^{h_k} N_{j-1}$.

If the target is in the searching area and the number of nodes inside the ring i and $i+1$ is $\sum_{m=h_i+1}^{h_{i+1}} N_i$, the probability that the target node hop h_d is within h_i and h_{i+1} is: $\mathcal{P}_i = \mathcal{P}\{h_i < h_d \leq h_{i+1}\} = \frac{\sum_{m=h_i+1}^{h_{i+1}} N_i}{\sum_{m=1}^{M} N_m}$.

The average cost C^n to search for a random node using

[2]Note the difference between h_{i+1} and $h_i + 1$, as they may look similar in the text.

an n-tier scheme is:

$$C^n = \sum_{i=0}^{n-1} \mathcal{P}_i C_i = \sum_{i=0}^{n-1} [\frac{\sum_{m=h_i+1}^{h_{i+1}} N_m}{\sum_{m=1}^{M} N_m} \sum_{k=1}^{i+1} \sum_{j=1}^{h_k} N_{j-1}] \quad (1)$$

N_T is the total number of nodes inside the M-hop searching area, including the source node, and equals $\sum_{m=0}^{M} N_m$. We start by studying the cost when $n = 1$.

1. $n = 1$. In this once-for-all scheme, all the nodes transmit except those that are exactly M hops away. Thus, we have the cost $C^1 = N_T - N_M$.

2. $n = 2$. Suppose we set hop limit $h_1 = k$ for the first attempt, and for the second attempt we set hop limit $h_2 = M$. To simplify the equations, let us define $a_k = \sum_{i=1}^{k} N_i$, which is the total number of nodes within k hops of the source, $k \geq 1$. From equation 1, we have

$$\begin{aligned} C^2 &= \mathcal{P}_1 C_1 + \mathcal{P}_2 C_2 \\ &= \frac{a_k}{N_T - 1}(N_0 + a_k - N_k) \\ &+ \frac{N_T - 1 - a_k}{N_T - 1}[(N_0 + a_k - N_k) + (N_T - N_M)] \quad (2) \\ &= \frac{1}{N_T - 1}(2(N_T - 1) + (N_T - 1)^2 \\ &- N_M(N_T - 1) - (N_T - 1)N_k + a_k N_M - a_k) \end{aligned}$$

Subtracting C^1 from C^2, we have the difference between the two-tier scheme and the once-for-all scheme

$$\begin{aligned} D_{2,1} &= C^2 - C^1 \\ &= \frac{1}{N_T - 1}(N_T - 1 - (N_T - 1)N_k + a_k N_M - a_k) \end{aligned} \quad (3)$$

If $D_{2,1} < 0$, which means $C^2 < C^1$, the two-tier scheme is preferred; otherwise, the once-for-all scheme is better. The only variable of $D_{2,1}$ is k, which is the hop number for the first searching attempt, and all the other parameters such as N_M and N_T are constants for a given network. Now we want to determine if there is some k that enables $D_{2,1} < 0$, and if so, what the optimal k should be to achieve $\min D_{2,1}$.

After placing a large number of nodes in a disk of unit radius and determining the number of the nodes within the first k hops from the center node, we found that before edge effects occur, the number of nodes at a certain hop distance from the source is roughly linear with hop number[3].

Thus, we can estimate $N_i \approx Bi$ for large scale networks, where B is a constant value larger than 1 and is closely related to the network density. Thus we estimate the sequence $N_0, N_1, \cdots, N_M$ as $1, B, 2B, \cdots, MB$. Using this estimation, the total number of nodes inside M hops, N_T, equals $\sum_{i=0}^{M}(Bi)$, and the cost equation 3 finally becomes $D_{2,1} = \frac{B(k-M)(-1-M+k(-1+BM))}{M(M+1)}$.

$D_{2,1}$ is a parabola function with k ranging from 1 to $M - 1$. By analyzing this function, we reach two conclusions. First, When $k_{opt} = \lceil \frac{M}{2} \rceil$[4], $D_{2,1}$ achieves its minimum value, and this value is less than zero. Thus, by applying a two-tier scheme with k_{opt} as the first attempt hop number, we can achieve less cost than the once-for-all scheme. Second, When $k = 1$, $D_{2,1}$ reaches its maximum $\max D_{2,1} = \frac{B(-1+M)(2+M-BM)}{M(M+1)}$. This value is less than zero when $B > 1$.

Based on our estimation, we conclude that the costs of all the two-tier schemes are less than the cost of the once-for-all scheme. Furthermore, the cost of a two-tier scheme reaches its minimum when the first attempt hop limit is set to $\lceil \frac{M}{2} \rceil$. The cost reaches its maximum among all the two-tier schemes when the first attempt hop limit is set to 1.

3. $n = 3$. In a three-tier scheme, there are two parameters to adjust, the first attempt hop limit h_1 and the second attempt hop limit h_2. Let us look at a special scheme in which $h_1 = 1$ first. Using similar procedures as above, we find that the best performance occurs when we choose $h_2 = \lceil \frac{M+1}{2} \rceil$. Also, we find that the cost of this $(1, \lceil \frac{M+1}{2} \rceil, M)$ scheme is even lower than the cost of an $(\lceil \frac{M}{2} \rceil, M)$ scheme when $B > 4$.

In a more general case, h_1 does not equal 1. However, we cannot prove that the special three-tier scheme of $h_1 = 1$ is the best among all the three-tier schemes.

4. $n \geq 2$. Also, we are not able to prove an (n+1)-tier scheme derived from an n-tier scheme is always worse than the n-tier scheme. From the final equation of the cost difference equation between the n-tier and (n+1)-tier schemes, we notice that there is one negative term of the order of h^3, while there are two positive items of the order of h^4. This indicates that there are very few choices for a derived $(n+1)$-tier scheme to be better than an n-tier scheme in terms of cost.

In this part, we have studied hop-based flooding schemes in large scale networks in terms of cost. Let us summarize our conclusions.

1. To obtain a good two-tier scheme, the first hop limit should be set to $\lceil \frac{M}{2} \rceil$ and the second hop limit should be set to M. We have proven that this two-tier scheme has less cost than the once-for-all scheme and is optimal for all the two-tier schemes.

2. To obtain a good three-tier scheme, set the first hop limit to 1, the second hop limit to $\lceil \frac{M+1}{2} \rceil$ and the third hop limit to M. We have proven that this three-tier scheme has even less cost than the optimal two-tier scheme.

3. However, we cannot prove that our three-tier scheme is optimal among all the three-tier schemes. We also cannot prove that it leads to higher cost by splitting an n-tier scheme to an $(n + 1)$-tier scheme. However, we conjecture that it is quite probable that any $(n+1)$-tier scheme will have a larger cost than the n-tier scheme from which it was derived.

4. The scheme that is applied currently in DSR, which is to set the hop limit to 1 for the first attempt and

[3]This can be seen from part (3) and (4) of Fig. 3 when the source node is located at the network center with the distance to the network border $x = 1$. The node distribution shows a linear tendency with small numbers of hops before edge effects occur.

[4]$\lceil x \rceil$ means the smallest integer greater than or equal to x.

M for the second attempt, can be seen as one of the two-tier schemes. Based on our results, we show that this approach leads to the highest cost among all the two-tier schemes.

For geography-based flooding, the model is shown in the left part of Fig. 1. Using similar analytical methods as above, we can easily prove that the once-for-all scheme has the same cost as the two-tier scheme, and this cost is the lowest among all the costs of possible n-tier schemes. The conclusions are valid for both large-scale and small-scale networks.

4.2 Latency

Let us define a constant T as the time interval from when a node receives a packet to when it finishes forwarding the packet. Then the time a flooded packet takes to reach a node l hops away is lT. Suppose the target node is i hops from the source node and $h_k < i \le h_{k+1}$, for $k \in \{0 \dots n-1\}$. To find the target, the source node has to fail for the first k times, which takes $2h_m T$ time for the mth attempt $m \in \{1, \cdots k\}$, and succeed at the $(k+1)$st attempt, which takes $2iT$ time. Note that each attempt time is doubled because the source node has to wait enough time for a potential acknowledgement from the target. Thus we have the total latency to search for a target i-hops away is $L_i = \sum_{m=1}^{k} 2h_m T + 2iT$.

Since N_i is the total number of nodes exactly i hops away from the source, the probability that a node is i hops away from the source node is $\mathcal{P}_i = \frac{N_i}{\sum_{i=1}^{M} N_i} = \frac{N_i}{N_T - 1}$, for $i \in \{1 \dots M\}$. The average latency L to search for a random node is:

$$L = \sum_{i=1}^{M} \mathcal{P}_i L_i = [\sum_{i=1}^{M} 2N_i i + \sum_{k=1}^{n-1} \sum_{i=h_k+1}^{h_n} (2h_k N_i)] \frac{T}{N_T - 1} \quad (4)$$

Applying mathematical induction on equation 4, we can easily prove that for all the $(n+1)$-tier schemes with $n \ge 1$, we can find an n-tier scheme that has a shorter latency. Furthermore, we can prove that the once-for-all scheme has the shortest latency of all the schemes (proof omitted).

5. UNRESTRICTED FLOODING IN SMALL-SCALE NETWORKS

In the previous section, we assumed that during the discovery process, nodes are unwilling to flood the whole network and have certain restrictions on the maximum *region* to be covered. In this section, we extend the model to small-scale networks in which nodes may search the whole network for a target. The main difference between this model and the previous model is that nodes are no longer the absolute center of the flooded region and edge effects must be taken into account during analysis. In other words, the node distribution at certain hops away from the source node no longer shows a linear tendency and is closely related to the source node's position.

In this section, we will confine our discussion to the once-for-all and the two-tier schemes. The average cost is slightly different from that in Section 4. Not knowing what the largest hop number is, in order to guarantee the whole network is covered, the source node has to apply a large enough hop limit number for the last attempt. The direct effect is that nodes at the maximum hops also have to forward the packet; while in Section 4, nodes at M hops do not forward the packet.

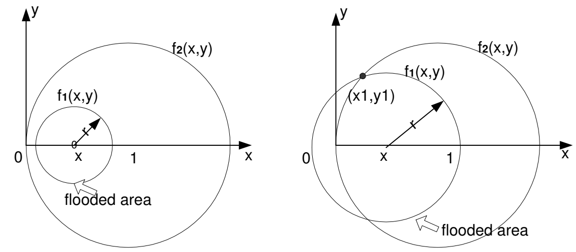

Figure 2: The intersection area is the region covered by flooding. When $r < x_0$, it is $f_1(x, y)$; when $r \ge x_0$, it is the intersection of two areas.

As before, we look at the cost from $n = 1$.

1. $n = 1$. All the nodes have to forward and the cost $C^1 = N_T$ (note the difference with that in large-scale networks).

2. $n = 2$. The cost of a two-tier scheme with k as the first attempt hop number is

$$\begin{aligned} C^2 = &\frac{1}{N_T - 1}[a_k(1 + a_k - N_k) \\ &+ (N_T - a_k - 1)(1 + a_k - N_k + N_T)] \end{aligned} \quad (5)$$

   where $a_k = \sum_{i=1}^{k} N_i$, which is the total number of the first k hops nodes, as defined earlier.

Whether to use the once-for-all scheme or a two-tier scheme depends on $D_{2,1}$, the cost difference between these two schemes.

$$D_{2,1}(k) = C^2 - C^1 = [(N_T - 1) - a_k - N_k(N_T - 1)] \frac{1}{N_T - 1} \quad (6)$$

If we apply $k + 1$ for the first attempt hop limit instead of k, the difference becomes

$$D_{2,1}(k+1) = \frac{(N_T - 1) - a_k - N_{k+1} - N_{k+1}(N_T - 1)}{N_T - 1} \quad (7)$$

If $D_{2,1}(k+1) < D_{2,1}(k)$, a two-tier scheme applying $k+1$ is preferred over a two-tier scheme applying k. Subtracting equations 7 and 6, we have

$$\begin{aligned} &D_{2,1}(k+1) - D_{2,1}(k) \\ &= [N_k(N_T - 1) - N_{k+1}N_T] \frac{1}{N_T - 1} \approx N_k - N_{k+1} \end{aligned} \quad (8)$$

As can be seen, as long as $N_k < N_{k+1}$, we should apply a two-tier scheme using $k + 1$ as the first attempt hop limit instead of k. This trend continues until N_k starts to become larger than N_{k+1}. To determine this turning point, an estimation of the sequence N_i is necessary.

Just as we estimated N_i in Section 4, we provide a general algorithm for the sequence estimation of nodes at different locations. We set up two-dimensional coordinates as shown in Fig. 2. The node is located at x_0 away from the network border. When the flooding radius is large, part of the potential flooding area exceeds the edge of the network. In Fig. 3, we show the flooding area $A(r)$ using geography

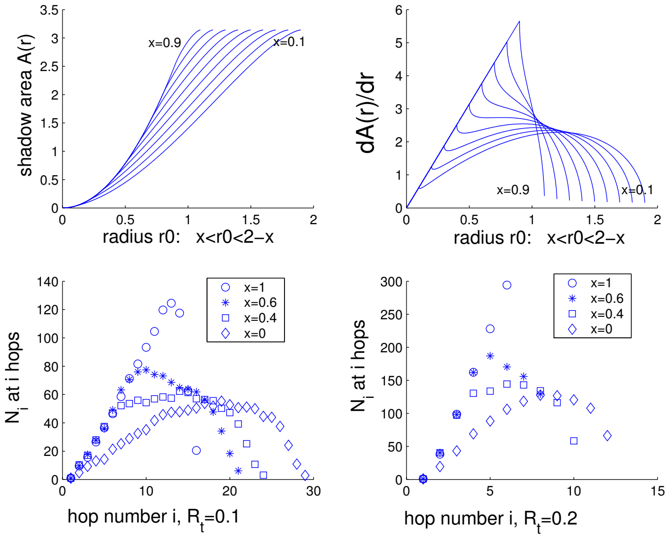

Figure 3: Area $A(r)$ of flooded region and the relation between the number of nodes N_i exactly i hops away with $\frac{dA(r)}{dr}$. Part (1) is the flooded area $A(r)$; part (2) is $\frac{dA(r)}{dr}$; part (3) is the N_i in a network of 1000 nodes and transmission range 0.1; part (4) is the N_i in a network of 2000 nodes and transmission range 0.2. Part (3) and part (4) look like a sampling of part (2).

control methods and the number of nodes at different hop numbers when the source node is located at different distances x from the network border. We notice that the node distribution in the lower part is just like the sampling of the derivative of the continuous geography flooding area shown in the upper part. Thus, we only need to estimate the maximum hop number and do proper samplings at each hop number. Due to space constraints, we omit the details here. We suppose that given the node's distance from the border x_0 and the network parameters of total number of nodes N_T and the node transmission range R_t, we have the estimated sequence $\tilde{N}_{i,x_0}$ of N_i for a node at location x_0.

5.1 Self-location aware

First, suppose a node knows its distance to the border of the network x_0 and can adjust its own hop limit k for the first attempt. Here is our proposed two-tier scheme for this case: the node first estimates $\tilde{N}_{i,x_0}$ based on its location x_0. Then it finds out from the estimated sequence $\tilde{N}_{i,x_0}$ the value of i where $\tilde{N}_{i,x_0} \approx \tilde{N}_{i+1,x_0}$ and sets k to this value for the first attempt. If this fails, the node must pick a large enough number for the second attempt to ensure that the flooded packet reaches all nodes in the network.

5.2 Self-location unaware

More realistically, nodes do not have knowledge of their location in the network. Every node has to apply the same flooding strategy and set the same predetermined values for each attempt. Now we consider how to minimize the cost from the system view.

First, we can prove that for a uniformly distributed network, the Probability Distribution Function (pdf) of a random node location X away from the border is

$$f_X(x) = 2(1-x) \text{ for } 0 < x \leq 1 \tag{9}$$

For a node located at x, after the estimation of its node number sequence, we can correspondingly calculate the cost $C(x, k)$ of a two-tier scheme with first attempt k from equation 5. Since a consistent strategy has to be applied, suppose every node sets k_{sys} as its first attempt. We find the overall cost for the whole system $C_{sys}(k_{sys})$ as

$$C_{sys}(k_{sys}) = \int_0^1 f_X(x) C(x, k_{sys}) dx = \int_0^1 2(1-x) C(x, k_{sys}) dx \tag{10}$$

Based on equations 5 and 10, we propose our two-tier scheme here. First, gather enough samples of x and estimate the sequences $\tilde{N}_{i,x}$ for each sample of x. Then from equation 5, calculate the two-tier cost sequence $C^2(x, k)$ of each sample x. Put each $C^2(x, k)$ into equation 10 and calculate the system cost $C_{sys}(k)$. Finally, determine the global point k_{opt} as the point where the minimum C_{sys} is achieved. The calculation of this good k_{sys} can be implemented during the network design phase and input to each node as a system parameter—nodes do not need to calculate this value on the fly.

Overall, we propose a good two-tier scheme to reduce the cost for hop-based small-scale networks. For the first attempt, if a node has knowledge of its current position with regard to the network boundary, it can set its first hop limit from the estimation of $\tilde{N}_{i,x}$ to minimize cost. If it has no such information, it should set the number to the pre-calculated good value to minimize overall system cost. For the second attempt, nodes just need to set a large enough hop limit to cover the entire network.

6. SIMULATIONS

6.1 Goals, metrics and simulation models

We have drawn several conclusions in Sections 4 and 5. The first goal of our simulations is to verify our conclusions and conjectures. Second, we have proposed some good schemes based on the analysis of estimated N_i. Another goal of our simulations is to verify the accuracy of the estimation and the performance of these schemes by determining how far away our proposed schemes are from the ideal schemes. The ideal n-tier scheme is found by thoroughly testing all the possible n-tier schemes through simulations on randomly generated scenarios.

We only investigate the behavior of networks that used hop-based flooding control. In each simulation section, we compare the cost savings per target search and the latency performance of different schemes. We measured the cost savings of each scheme compared to the basic once-for-all scheme whose cost is constant for a determined scenario. This metrics indicates how much improvement we can achieve by replacing the once-for-all scheme with the investigated scheme. Every test is repeated on 50 different random scenarios, and the results are averaged.

The schemes that are tested are: the once-for-all scheme, the expansion ring scheme, the DSR scheme, our two-tier scheme, our three-tier scheme, the ideal two-tier scheme, the ideal three-tier scheme and the ideal four-tier scheme. The expansion ring scheme is an n-tier scheme, which sets the first attempt hop limit to 1 and doubles the hop limit upon

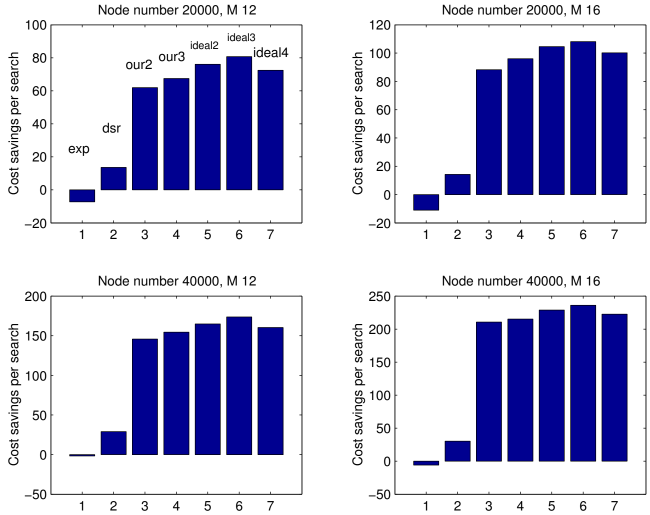

Figure 4: Cost savings per search for each scheme. From left to right labeled as 1 to 7: (1) expansion ring, (2) DSR, (3) our two-tier scheme, (4) our three-tier scheme, (5) ideal two-tier scheme, (6) ideal three-tier scheme and (7) ideal four-tier scheme. The y-axis indicates the number of packets saved per search.

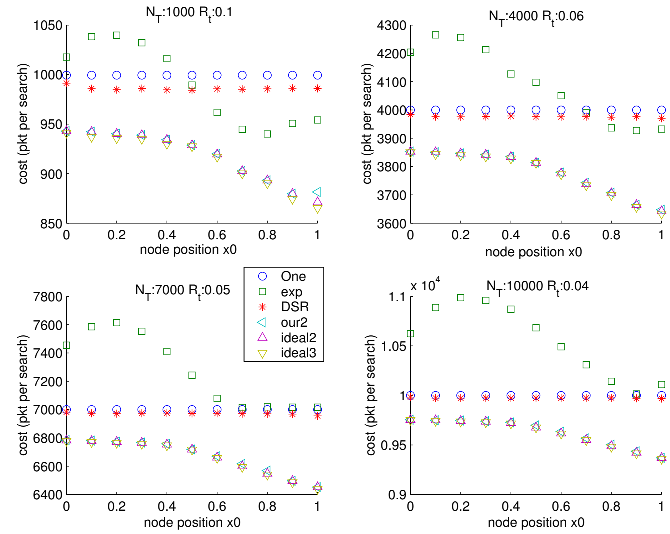

Figure 5: Cost of different schemes for nodes at different locations in small-scale networks. The X-axis indicates the location of the investigated nodes. The Y-axis indicates the cost of different schemes. Different network sizes of 1000, 4000, 7000 and 10000 nodes are simulated. The tested schemes are: the once-for-all scheme, the DSR scheme, the expansion ring scheme, our two-tier scheme, the ideal two-tier scheme and the ideal three-tier scheme.

each failure until the maximum restriction M is reached. The current DSR scheme can be seen as a two-tier scheme with the first attempt hop limit set to 1. The choice of our schemes varies for different types of networks and will be specified in each individual simulation part.

6.2 Performance comparison in large-scale networks

In this section, we compare the cost and latency performance of different schemes in hop-based large-scale networks. In this section, 20000 or 40000 nodes with transmission range $R_t = 0.03$ are placed in a unit radius disk. The center node searches for a random target from nodes within M hops away. We test for $M = 12$ and $M = 16$; with these values of M, the flooding area is far away from the border of the whole large network and no edge effects need to be taken into account.

Our proposed two-tier scheme is to set the first hop limit to $\lceil \frac{M}{2} \rceil$ and the second hop limit to M. Our proposed three-tier scheme is to set the first hop limit to 1, the second hop limit to $\lceil \frac{M+1}{2} \rceil$ and the third hop limit to M. We also measure the performance of all the two-tier, three-tier and four-tier schemes and pick the minimum of each scheme as the ideal value for these n-tier schemes.

Fig. 4 shows the results. As can be seen from this figure, the expansion ring's savings are less than zero, that is, it costs even more than the basic once-for-all scheme. The DSR scheme, as a member of the two-tier scheme family, does have some savings over the once-for-all scheme. However, the savings are low. The reason is that it is the worst of all the two-tier schemes, as proven in Section 4. Our two-tier scheme has less cost saving than our three-tier scheme. However, their difference is small, and both schemes' performance approaches that of the ideal schemes. From simulation, the performance of the best four-tier scheme is less than that of the best three-tier scheme, which matches our conjecture that $(n+1)$-tier schemes may not be better than n-tier schemes. The network density and the maximum number of hops M have an effect on the amount of the cost savings. However, they do not affect our above conclusions.

Earlier, we have claimed that all other schemes' latency is larger than the once-for-all scheme. Through simulations, we find that our two-tier and three-tier schemes have close performance and they have a higher latency than the once-for-all scheme with a percentage around $50\% - 60\%$. The expansion ring scheme has around 120% higher latency, while the DSR scheme has only around 10% higher latency. The latency is not related to the network density. When the network density changes, N_i changes with the same scale, and from equation 4, L remains the same.

6.3 Performance comparison for small-scale networks with location knowledge

In this part, we compare the performance of different schemes in a small-scale network in which nodes have knowledge of their own locations x_0. The total number of nodes N_T varies from 1000, 4000, 7000 to 10000. Nodes have different costs for target searching based on their locations. In Fig. 5, the x-axis is the different location x_0 of the investigated nodes. The y-axis is the cost of nodes at location x_0 applying different schemes of once-for-all, expansion ring, DSR, our two-tier scheme, the ideal two-tier scheme and the ideal three-tier scheme. Our scheme is to estimate N_i with the given location x_0 and find the point where $N_k = N_{k+1}$ as the first attempt hop number. The second attempt hop number is chosen large enough to cover the entire network.

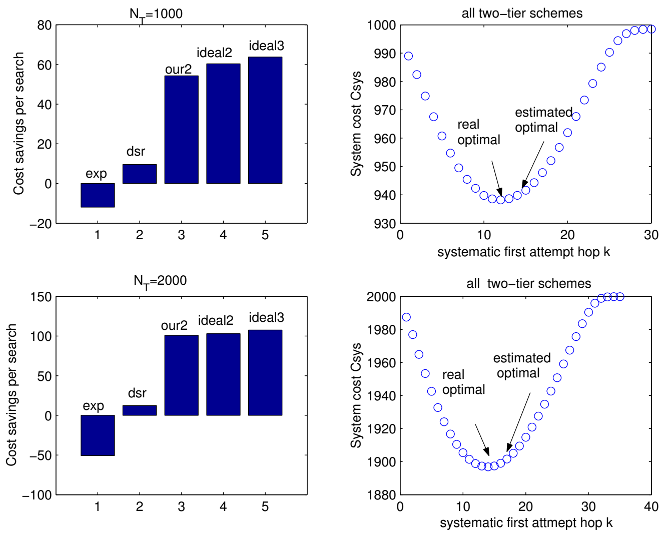

Figure 6: The cost savings of different schemes for small-scale networks. From left to right labeled as 1 to 5: (1) expansion ring, (2) DSR, (3) our two-tier scheme, (4) the ideal two-tier scheme, (5) the ideal three-tier scheme. The graph on the right shows the system cost for different k_{sys} of the two-tier scheme. Our estimated points are 13 for 1000 nodes and 16 for 2000 nodes, which is close to reality.

As can be seen from Fig. 5, our scheme performs consistently close to that of the ideal schemes, which indicates that our estimation matches reality quite well. DSR also performs consistently and is close to the once-for-all scheme. Nodes at different locations have much different costs when applying the expansion ring schemes. Some nodes may achieve less cost while other nodes may achieve more cost. Overall, the expansion ring's cost tends to be larger than that of the once-for-all scheme. When $x_0 = 1$, which means nodes are close to the center, the average cost decreases since more nodes are a relatively smaller number of hops away from the center.

As for latency, similar results can be achieved. Nodes that are close to the center usually need less time to cover the whole network and the resulting latency in finding the target is smaller.

6.4 Performance comparison for small-scale networks without location information

In this part, we compare the system cost savings of different schemes in a small-scale network without location information. Nodes have to apply a consistent parameter for a searching scheme instead of choosing different values for the first attempt.

In the right part of Fig. 6, we find that our estimated optimal point through equation 10, which is 13 for the network of 1000 nodes and 16 for the network of 2000 nodes, is quite close to the real optimal point. For this reason, the cost of our scheme is also close to that of the optimal scheme. Again, the expansion ring is the worst scheme and the DSR scheme achieves little cost savings.

7. TARGET DISCOVERY WITH CACHING

First, let us conclude the study on expansion ring schemes for target discovery without caching. Based on our analysis for different flooding control methods and network scales, we conclude with the results shown in Table 2.

The existence of caching makes the expansion ring schemes more promising for target discovery, as the potential discovery of caches for the target in the local area reduces the need for a network-wide query to a significant extent. The biggest concern about caching is its correctness and optimality, especially when the caching is for routing purpose. Many optimizations like negative caches and route error wider notifications are proposed to solve these problems [10, 11].

Route caches must be combined with expansion ring schemes to reach the goal of saving routing overhead. The expansion ring scheme applied by DSR, which searches only the one-hop neighbors first and then the entire network, can only achieve good results under strong caching conditions in which the neighbors are very likely to provide some cached information. The expansion ring scheme from AODV may have too many excessive rings and lead to unnecessarily high latency. Compared to once-for-all searching, a two-tier scheme with caching, using a first ring search of k hops, can achieve a cost saving ratio CSR as follows:

$$CSR(k) = (1 - \frac{N_l(k)}{N})P(k) \tag{11}$$

In this equation, $N_l(k)$ denotes the total number of nodes in the local area within k hops away from the source node, N is the total number of nodes in the network, and $P(k)$ denotes the probability of finding a valid route cache. Intuitively, the above equation indicates that the routing query saving occurs when the target is non-local but a valid cache with information about the target is found locally.

Increasing k, while decreasing the first factor $(1 - \frac{N_l(k)}{N})$, increases the chance to find a valid route cache $P(k)$. The study of the selection of the first tier hop k to maximize the saving ratio CSR comes to the same result as without caching. When $k = \frac{M}{2}$, the saving reaches its maximum under moderate caching availability. When the caching condition is stronger, a smaller k or $k = 1$ may have an optimal performance. However, a k larger than $\frac{M}{2}$ will never be optimal. This is due to the fact that it is very probable to find the target itself with a large k and thus caching provides little or no benefit. Correspondingly, the gain from query overhead is decreased.

Simulation results show that choosing $k = \frac{M}{2}$ reduces the routing query overhead by about 50% compared to the once-for-all scheme and by about 25% compared to choosing k as 1 under moderate caching conditions. Our future work on target discovery with caching is to design a measurement scheme for the caching availability and adjust the first tier searching radius according to the actual caching conditions.

8. CONCLUSIONS AND DISCUSSIONS

We can be certain about three things. First, the cost saving percentage using an optimal n-tier scheme rather than the once-for-all scheme is less than 10% for most of the cases. Second, the cost saving percentage decreases when the number of involved nodes increases[5]. Third, the latency

[5] We conjecture the percentage to be the order of $\sqrt{\frac{\log(N)}{N}}$.

Table 2: **Overall conclusions for networks with no caching.**

Network Scale	Flooding Control	Metrics	Proposal	Parameters	Performance level
Large	geography-based	cost	one-tier	maximum distance	best
	hop-based	cost	two-tier	$\lceil \frac{M}{2} \rceil$,M	good
			three-tier	1,$\lceil \frac{M}{2} \rceil$,M	good
		latency	one-tier	M	best
Small	geography-based	cost	one-tier	r_{max}	best
	hop-based, location aware	cost	two-tier	$N_k = N_{k+1}$	good
	location unaware	cost	two-tier	valley of C_{sys}	good

increases significantly when using an n-tier scheme instead of the basic once-for-all scheme.

Thus, our conclusions based on the results of our analysis and simulations of different flooding mechanisms are as follows. For fast discovery, the best strategy is to flood just once. By applying the schemes presented in this paper, less than 10% of cost (lower in most cases) can be saved. Thus, even the optimal scheme may not be worth using, as there will be a substantial sacrifice in latency in exchange for insignificant gain in cost savings. Furthermore, when caching availability is weak, the existing scheme currently used in DSR is not a good scheme for target discovery, and the arbitrary "expansion ring" scheme is a bad scheme due to the performance degradation in both cost and latency.

When caching is available, expansion ring schemes are much more promising in reducing the query flooding overhead. We have shown that a significant saving gain can be achieved by choosing a proper first tier radius. Resolutions on reducing the route response storms and adjusting routing protocols towards optimal performance are are of interest to us and will be studied in our future work.

9. REFERENCES

[1] D.B.Johnson and D.A.Maltz. *Mobile Computing*, Chapter Dynamic source routing in ad hoc wireless networks, pages 153-181. Kluwer Academic Publishers, Imielinski and Korth edition, 1996.

[2] C.Perkins and E.M.Royer. "Ad hoc on-demand distance vector routing" *Proceedings of IEEE WMCSA'99*, pp. 90–100, Feb. 1999.

[3] Chalermek Intanagonwiwat and Ramesh Govindan and Deborah Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," *Mobile Computing and Networking*, pp. 56–67, 2000.

[4] E. Woodrow and W. Heinzelman, "SPIN-IT: A Data Centric Routing Protocol for Image Retrieval in Wireless Networks," *Proc. International Conference on Image Processing (ICIP '02)*, Sep 2002.

[5] Narayanan Sadagopan and Bhaskar Krishnamachari and Ahmed Helmy, "ACtive QUery forwarding In sensoR nEtworks (ACQUIRE)",*Computer Engineering Technical Report CENG 02-11*, 2002.

[6] D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks", *Proc. International Conference on Distributed Computing Systems (ICDCS-22)*, 2002.

[7] Ya Xu, John Heidemann, and Deborah Estrin, "Geography-informed energy conservation for ad hoc routing", *Proc. Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001.

[8] J. Sucec and I. Marsic, "An Application of Parameter Estimation to Route Discovery by On-Demand Routing Protocols", *Proc. ICDCS 2001*, pp. 207–218.

[9] Y.-C. Hu and D. B. Johnson, "Caching strategies in on-demand routing protocols for wireless networks," *Proc. ACM/IEEE MobiCom*, August 2000, pp. 231–242

[10] Yih-Chun Hu and David B. Johnson, "Ensuring Cache Freshness in On-Demand Ad Hoc Network Routing Protocols," *Proc. POMC 2002 Workshop on Principles of Mobile Computing*, pp. 25–30, ACM, Toulouse, France, October 2002.

[11] M. K. Marina and S. R. Das, "Performance of Route Caching Strategies in Dynamic Source Routing," *Proc. the Int'l Workshop on Wireless Networks and Mobile Computing (WNMC) in conjunction with Int'l Conf. on Distributed Computing Systems (ICDCS)*, pp. 425–432, 2001.