Computer Communications 31 (2008) 4138–4148

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Modeling message propagation in random graph networks

Bin Wu, Ajay D. Kshemkalyani *

Computer Science Department, University of Illinois at Chicago, Chicago, IL 60607, United States

ARTICLE INFO

Article history:
Received 27 February 2008
Received in revised form 3 September 2008
Accepted 4 September 2008
Available online 12 September 2008

Keywords:
Message propagation
Random graphs
Node degree
Node coverage
Peer-to-peer network

ABSTRACT

Message propagation is used in a wide range of applications, such as search in unstructured P2P overlays, modeling infection spread in epidemiology, and modeling the spread of gossip in social networks. For example, in a P2P network that has an unstructured overlay, search for a piece of information is conducted by propagating the query message within the network, usually with the desire that as many nodes as possible are covered with as few message forwardings as possible. In this paper, we study the behavior of the message propagation process in random graph networks and give a simple model to describe this process. When applied to a large network with random graph topology, the message propagation process can usually be modeled as a random pick process or the coupon collection problem. We show that these models are less accurate when the number of covered nodes becomes large. We investigate the inaccuracy and then propose refined models which remedy the factors that cause the error. The refined models have been confirmed by our simulations to effectively compensate for the errors, especially under high coverage conditions. Thus, when a large number of messages is expected to be used in the message propagation process, the refined models of higher orders are essential.

1. Introduction

Message propagation is used in a wide range of applications. For example, message propagation is used in unstructured P2P networks to perform keyword and range query searches [14,17]. In such P2P overlay networks, search for a piece of information is conducted by propagating the query message within the network, usually with the desire that as many nodes as possible are covered with as few message forwardings as possible. The overlay of the unstructured P2P network may take various topologies due to various dynamic characteristics such as the arrival and departure of nodes that form the network. Typical topologies include the Gnutella graph [7,15], random graph [5,8], power law random graph [1], or grid [15]. Watt and Strogatz studied the formation and characteristics of a category of random graph networks: the "small world" network [19]. A "small world" network is at some intermediary position between a regular network and a random graph network in terms of randomness. Since a small world network has short network diameter (a property of random graphs) and is highly clustered as well, message propagation can be fast. Much research has been performed on constructing P2P overlays with the "small world" property to balance the search efficiency and the maintenance cost over a partially structured overlay [11,13].

In any of these overlays, to find a specific object without previous knowledge of the object distribution, blind search such as random walk [3,4,9], flooding [7,15,18], and their variations [2,6,10,12,16,21–23] has to be applied to the topology. No matter what approach is chosen, the goal is always to propagate the query message to as many nodes as possible, and the constraint includes the response time or the message overhead. Flooding is the approach that spreads message with the maximum speed and also has the maximum message overhead. Random walk has high message efficiency in terms of the ratio of the node coverage to the message overhead, but the speed can be very slow. Variations that benefit from both of these approaches include flooding with TTL, expanding ring, and multiple random walkers [15].

Some specific phenomena in social activities and epidemiology studies can also be modeled by the message propagation process. For example, disease infection among the population can be caused by contact among people, and the behavior of virus spreading is heavily dependent on the pattern of contacts. Once infected by a disease, a person who has a wider range of social links is more likely to spread the disease to uninfected people, when the number of allowed contacts is fixed. As another example, consider gossip or the spreading of information among the population via using one's social contacts. For such examples, the goal may be to study how many "messages" it takes to reach a certain proportion or all of the population. Or alternately, given a certain number of message hops (that determines the time period) to be used, identify the proportion of the population that is infected by the disease or that receives the news.

A real world social networking example is as follows. In a small town of population 20,000, a new Chinese restaurant is opened. Under normal behavior, the first few customers would randomly

* Corresponding author.
E-mail addresses: bwu@cs.uic.edu (B. Wu), ajayk@cs.uic.edu (A.D. Kshemkalyani).

0140-3664/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2008.09.006

tell this information to one or more of their acquaintances. After a while, when a significant portion of the people (say, 60%) already know about this restaurant, is it still worthwhile for the restaurant–owner to encourage his customers to further spread this information, if the incentive for advertising is the same? What is the possibility that the next effort introduces a new customer, or, ends up being a repetition to a person who already knows it? Using the refined model gives a more accurate estimate of the probability of reaching a new customer. This will help the owner to make a decision.

The random walk approach, with possibly multiple walkers, is of special interest to most research in this area because of its scalable message overhead [3,4,9]. Precisely modeling the random walk is not easy because it is hard to obtain a precise mathematical model for the actual network topologies.

In this paper, we study the behavior of the message propagation process in random graph networks and give a model to describe this process. We give our quantitative analysis models based on the $G(n,p)$ random graph network model [8] and we focus our analytical target on the node coverage and message efficiency. If the number of nodes in the network is large, then at the steps when the node coverage is small, the message propagation process can be modeled by some other random processes, such as the random pick [3,4] or the coupon collector's problem [9]. For example, the study by Gkantsidis et al. [9] shows that the effect of a k-step random walk is statistically similar to that of taking k independent samples (random pick) in a well-connected graph. Our models explain the rationale of such an analogy. We then show that these and similar models are less accurate when the number of covered nodes becomes large. We investigate the inaccuracy and identify the reasons causing it. We then propose refined models which remedy the factors that cause the error. The refined models have been confirmed by our simulations to effectively compensate for the errors, especially under high coverage conditions. Thus, when a large number of messages is expected to be used in the message propagation process, the refined models of higher orders are essential. Note that a large number of messages can be expected to be used in the examples of social contacts and epidemiology studies given above.

1.1. Contributions

1. We study the message propagation process on the random graph topology and formulate a simple mathematical model in terms of node coverage and message overhead. We then draw an analogy to other random processes, i.e., the random pick and the coupon collection problems.
2. More importantly, we investigate the inheritant difference between message propagation and other random processes and discover that the node degree (average number of links per node) plays an important role for the difference: the effect of the limited number of links has a negative effect on the probability of forwarding a message to a new node.
3. By quantitative analysis of such effects, we give a refined model that increases the accuracy, especially when the number of covered nodes has become large. The refinement is introduced by accounting for the "dirty links" of the current node when forwarding a message.
4. As per the model, the probability that a next step message reaches a new node is a function of both the message overhead, x, and the average node degree, d. The order of refinement is expressed in terms of the maximum number of times (k) a node may have been visited before the current arrival. We thus give a family of enhanced models, the k-order refined models. The higher the value k, the more accurate is the refined model. We verify this by simulations.
5. The estimation of "dirty links" is based on the value of k we specify, which is a random variable, whose value may be distributed between 0 and x. We propose two approaches of computing "dirty links": unconditional and conditional estimations, and we also show that these two approaches actually produce the same results but they have different computation complexity.

The addressed analysis models are of special importance in studying the behavior of the querying process in P2P networks. The node coverage and message efficiency are quantitatively modeled via the network and process parameters which can be modulated by an administrator. The n-order ($n \geqslant 1$) refined model should be applied for better accuracy in the cases where the node coverage is high. In these cases an arbitrary node in the network may have been visited multiple times from its different neighbors.

In a typical search in a P2P network, a single message may never be propagated very long, and it is also hard to trace a "single" message in the propagation process. We use the term "hop count" to follow the means of message spreading using random walkers. We extend the hop count to large numbers in order to reach a higher node coverage, where our refined model will give significant difference from the simple model. High node coverage situations are useful when the purpose is to "spread a message to as many nodes as possible" rather than just "finding the desired object".

Note that the Gnutella graph is not a random network. Hence, Gnutella-like P2P networks need a more specific means of analysis, which must also take into account the distribution of node degree within the network. This is fairly complicated and outside the scope of this article. Also, the flooding and partial flooding methods are not modeled differently from random walkers in our investigation of node coverage. However, the clustering of more realistic networks makes the flooding and partial flooding methods even more sophisticated than the random walker method, and the message efficiency could be worse than that for random walk even at low node coverage.

Section 2 gives a simple algebraic model for the node coverage analysis of the message propagation process. In Section 3, we give the analogy between the random walk process and random sampling. In Section 4, we refine our algebraic model to factor in the impact of node degree. In Section 5, we perform a comparison and analysis of our refined models. Section 6 gives the conclusions.

2. The algebraic model

When searching an unstructured network using random approaches such as random walk or flooding, *node coverage* is a key concept used to describe the behavior of the search process. It can be defined as the total number of nodes in the network that have already been explored; sometimes we can also use the percentage of such nodes as well [20]. In this paper, we use the first form of definition of node coverage for convenience.

We propose a simple algebraic model that performs a node coverage analysis for the message propagation process but makes no distinction so as to whether the messages are forwarded by flooding or random walkers. Each query message is treated as an independent sample. This model expresses the expected node coverage in terms of the message overhead, x.

In this simple model, an Erdõs–Rényi model random graph network is assumed, in which there are totally N nodes in the network. The probability that there exists a link (undirected) between any two nodes is p ($0 < p < 1$), such that the average node degree is $p(N-1)$. For simplicity, we assume that each node has a degree

of $p(N-1)$. Next, we assume that the graph is well-connected and when a node relays a message, it sends out the message through a different link from the one through which it receives the message, and this outgoing link is selected randomly from all links except the incoming one. Finally, this model makes no distinction between whether or not multiple random walkers are used to spread the message. In the case of multiple (m) random walkers, we assume they are initiated from m randomly chosen nodes and thereafter are forwarded independently.

Suppose at a snapshot of the message propagation process, the current node coverage is u and a specified node is forwarding a message to one of its neighbors, excluding the neighbor who has sent in the message. The probability that a new node receives this message is $\frac{N-u}{N-2}$. Thus, the expected value of node coverage would be $u+\frac{N-u}{N-2}$ after this message is forwarded. As N is large enough, we can use N to replace the term $N-2$ for the sake of convenience.

Let x denote the number of messages so far. Then

$$u(x+1) = u(x) + \frac{N-u(x)}{N} \tag{1}$$

which can be approximated as:

$$u'(x) = 1 - \frac{u(x)}{N} \tag{2}$$

This equation can be solved as

$$u(x) = Ce^{-\frac{x}{N}} + N \tag{3}$$

Here C is a constant determined by the initial condition. This constant has a minor influence on the results of $u(x)$ and it can be computed depending on how we configure our initial condition, for example:

1. If we assume the node coverage is 1 before any message is forwarded, $u(0)=1$, then $C=1-N$.
2. If we assume the node coverage is 2 after a single message has been forwarded, $u(1)=2$, then $C=(2-N)e^{\frac{1}{N}}$.
3. For the special case of multiple random walkers, suppose there are m walkers sent out from a single initiator (which is specific in the cases of P2P search), and also that the first message is required to cover the initiator itself before sending out any walker. Then the initial condition may be

$$u(m+1) = m+1 \tag{4}$$

And the constant C turns out to be

$$C = (m+1-N)e^{\frac{m+1}{N}} \tag{5}$$

Then the node coverage in this case can be expressed as

$$u(x) = \begin{cases} N-(N-m-1)e^{\frac{m+1-x}{N}} & \text{if } x > m+1 \\ x & \text{if } x \leqslant m+1 \end{cases} \tag{6}$$

The choice of the constant C has only a trivial influence on the computation of $u(x)$ and we can determine this constant based on the special case of interest or just convenience.

One of the uses of node coverage is to estimate the success rate in a P2P search: we assume that r copies of a desired object are randomly distributed in the network. The probability of finding a copy by the effort of spreading x messages can be expressed as

$$p_s(x) = 1 - \left(1 - \frac{u(x)}{N}\right)^r \tag{7}$$

The probability $p_s(x)$ is usually referred to as the *success rate* (of finding a desired copy).

3. Random walk modeling and random sampling

3.1. Analogy to random picking

The algebraic model simulates a process of random pick: there is a bag of N balls, at each step, we pick a ball and put it back. Then what is the expected number of distinct balls we will find after x attempts? Our algebraic model approximates this process with a differential equation and the solution to this equation gives the formula for computing the node coverage.

The rationale of this formula can be validated by comparing the formula for success rate using the node coverage above and that given by Bisnik and Abouzeid [3,4].

Recall that the algebraic model gives the node coverage $u(x)$ in Eq. (3). The constant C is determined by the initial condition of a specific search case. If we take the following initial condition:

$$u(x) = 0 \mid x = 0 \tag{8}$$

then we have

$$u(x) = N\left(1 - e^{\frac{-x}{N}}\right) \tag{9}$$

The success rate of finding an object with r copies is then computed using node coverage:

$$p_{\text{succ}} = 1 - \left(1 - \frac{u(x)}{N}\right)^r = 1 - \left(e^{\frac{-x}{N}}\right)^r \tag{10}$$

On the other hand, [3,4] suggest to compute the success rate using the following formula:

$$p_s = 1 - (1-p)^{kT} \tag{11}$$

where $p=\frac{r}{N}$ and $kT=x$, the message overhead when k, T denote the number of walkers and the hop number, respectively. This is the result of treating each walk step as a random pick. To use the notations in our model, we rephrase this formula:

$$p_s = 1 - \left(1 - \frac{r}{N}\right)^x \tag{12}$$

Also, Eq. (10) can be rephrased as

$$p_{\text{succ}} = 1 - \left(e^{\frac{-r}{N}}\right)^x \tag{13}$$

Note that as N is large enough, we have

$$\lim_{N\to+\infty}\left(1-\frac{r}{N}\right)^N = e^{-r} \tag{14}$$

and thus

$$\lim_{N\to+\infty}\left(1-\frac{r}{N}\right) = \lim_{N\to+\infty} e^{\frac{-r}{N}} \tag{15}$$

and this equates the computation of p_{succ} in Eq. (13) and that of p_s in Eq. (12). These two formulas are equivalent as N goes to infinity.

Note that these two approaches use different formulas and result in the same outcome, because they both simulate the same process of "independent random picks", and the proper computation for the success rate will always give the same result.

However, the random walk in random graphs is not actually "independent random picks", a message forwarding is always associated with the status of the current node. The probability of finding a new node by the next message forwarding is decided not only by the current proportion of "undetected" nodes in the network, but also by the current status of the sending node. This impact does not exist in the independent random pick process and it contributes to the differences between our algebraic model and the simulation results. This impact will be studied in detail in Section 4.

In Section 4, we also show how to enhance the algebraic model to account for the node degree in the random graph overlay, so as to improve accuracy. This consideration allows us to account for the fact that the current node may have been explored before.

3.2. Analogy to coupon collector's problem

The coupon collector's problem is another typical random sampling process: "given N types of coupons, a customer buys one coupon at a time. For each time, the probability of getting any specific type is equal: $1/N$. Then what is the expected number of purchases to obtain all the N types?" If this problem is asked this way: "in order to obtain u types among N types of coupons, what is the expected number of purchases x?" then obviously this becomes the reverse problem of the random pick we described in the previous section. In this section, we show that the analysis method also reveals the equivalence of the memoryless random walk and the coupon collector's problem.

It is well known that the expected times to collect all N types is

$$x(N) = N\sum_{k=1}^{N}\frac{1}{k} = NH_N \tag{16}$$

where H_N is the nth harmonic number and its analytical expression is

$$H_N = \ln N + \gamma + O\left(\frac{1}{N}\right) \tag{17}$$

Here γ is the Euler–Mascheroni constant and its value is $0.5772156\ldots$.

Similarly, we have

$$\begin{aligned} x(u) &= 1 + \frac{N}{N-1} + \frac{N}{N-2} + \cdots + \frac{N}{N-(u-1)} \\ &= N\left(\sum_{k=1}^{N}\frac{1}{k} - \sum_{k=1}^{N-u}\frac{1}{k}\right) \\ &= N(H_N - H_{N-u}) \\ &= N\left(\ln N - \ln\,(N-u) + O\left(\frac{1}{N}\right) - O\left(\frac{1}{N-u}\right)\right) \end{aligned}$$

As N and u are large enough, we can omit the term $O(\frac{1}{N}) - O(\frac{1}{N-u})$ and have

$$x(u) = N\ln\frac{N}{N-u} \tag{18}$$

If we express u in terms of x, we are answering this question: "what is the expected types of coupons we can collect, u, after x purchases?":

$$u(x) = N\left(1 - e^{-\frac{x}{N}}\right) \tag{19}$$

which is identical to Eq. (9), when the initial condition $u(x) = 0$ is applied to Eq. (3) in the algebraic model. This identity illustrates that the simplified random walk process on a regular random graph which is described by our algebraic model, is equivalent to the coupon collector's problem when the sampling space, N, is large.

4. Refinements of the algebraic model

The algebraic model is intended to capture the random process of message forwarding in normal random graph overlays. The method we applied is somehow over-simplified as we ignored the influence of node degree in our models and this ignorance leads to overestimate of node coverage. We now refine the algebraic model by studying and factoring in the impact of node degree.

4.1. Impact of node degree

As assumed in Section 2, at each step of message forwarding, our analytical model assumes the probability of visiting a new node to be the same as the ratio of the number of unvisited nodes to the total number of nodes: $\frac{N-u(x)}{N}$. This seems reasonable at first glance since the next message forwarding is a random visit. The above assumption is true if the current node itself is first visited by the current message. In this case, all the remaining links of this node are "fresh" (not probed yet) and whether one such link leads to a visited node or a new node is random. Hence, the probability of reaching a new node via that link is thus reasonably determined by the proportion of new nodes currently in the network. However, if the current node has been visited before, then this probability should be lowered because the current node may forward the current message via a link that has been traversed before and thus to a visited node. Consider an extreme condition: if every link of the current node has been visited before, then there is no way that the next forwarding visits a new node. The impact of node degree thus can be expressed in this way: considering the scenario that the current node has been visited before, the higher the node degree, the higher the possibility that the next forwarding traverses a "fresh" link, and thus the closer the probability of finding a new node approaches the proportion of unvisited nodes within the network. This is confirmed in our simulation, see Section 5.

4.2. Conditional estimate of dirty links

Recall that in our algebraic model, the iterative relationship of node coverage is expressed in Eq. (1). We now modify this formula to take into account the possibility that the current node has been visited before

$$u(x+1) = u(x) + \frac{N-u(x)}{N}p(x) \tag{20}$$

Hence, $p(x)$ is the probability that the next forwarding takes a "fresh" link. As the network topology is fixed, this probability is determined only by the proportion of "fresh links" of the current node, which in turn is a random function of the message overhead x. Hence, we can also express this probability as a function of x. The core challenge of refining the algebraic analysis model is to compute the probability $p(x)$. Let us branch into different preconditions in terms of the number of previous visits to the current node. Note that this number excludes the arrival of the "current" message. The notation used is summarized in Table 1.

- *0 previous visits:* As explained in the previous section, $p_0 = 1$.
- *1 previous visit:* $p_1 = \frac{d-3}{d-1}\cdot\frac{d-2}{d} + \frac{d-2}{d-1}\cdot\frac{2}{d}$.
  The number of dirty links after the first visit, DL_1, is 2 – one incoming, one outgoing. Hence, $\frac{d-2}{d}$ denotes the probability that the current message has come from a link other than the two links made "dirty" by the previous visit. In this case, the probability of probing a "fresh" link is $\frac{d-3}{d-1}$. On the other hand, if the current message has come though a "dirty" link, the probability of traversing a "fresh" link is $\frac{d-2}{d-1}$.
- *2 previous visits:* After 2 visits have been paid to this node, the probabilities that 2, 3, or 4 links have been "contaminated", denoted $w_2(2)$, $w_2(3)$, $w_2(4)$, respectively, are:
  - $w_2(2) = \frac{2}{d}\cdot\frac{1}{d-1}$
  - $w_2(3) = \frac{2}{d}\cdot\frac{\max[d-2,0]}{d-1} + \frac{\max[d-2,0]}{d}\cdot\frac{2}{d-1}$
  - $w_2(4) = \frac{\max[d-2,0]}{d}\cdot\frac{\max[d-3,0]}{d-1}$

Table 1
Notations used in the analysis

DL_i	Expected number of dirty links after i visits
CDL_i	Expected number of dirty links after i visits (conditional estimate)
UDL_i	Expected number of dirty links after i visits (unconditional estimate)
$w_i(j)$	Probability that after i visits, j links are dirty
$w(a \mid b)$	Probability of a dirty links after current forwarding, assuming the node has b dirty links before the current visit
p_i	Probability that the next forwarding after i previous visits takes a fresh link
$q_i(x)$	Probability that a node is visited i times after x messages have been used
$p(x)$	Probability that the next forwarding takes a fresh link

After two visits to a node, the expected number of dirty links can be expressed as

$$DL_2 = 2 \cdot w_2(2) + 3 \cdot w_2(3) + 4 \cdot w_2(4) \tag{21}$$

Similarly, the probability that the next message forwarding takes a "fresh" link, p_2, can be computed as follows:

$$p_2 = \frac{d - DL_2}{d} \cdot \frac{d - 1 - DL_2}{d - 1} + \frac{DL_2}{d} \cdot \frac{d - DL_2}{d - 1} \tag{22}$$

- *3 previous visits:* For simplicity, we assume that DL_2 links have been dirty before the third visit to the current node. In other words, we use the conditional probabilities with the assumption that exactly DL_2 (usually a non-integer number as the result of estimation) of the links are known to be dirty. Thus the resulting number of dirty links after the third visit, can be DL_2, $DL_2 + 1$, or $DL_2 + 2$, respectively; the corresponding conditional probabilities, with the given DL_2 as the pre-condition, are denoted as $w(DL_2 \mid DL_2)$, $w(DL_2 + 1 \mid DL_2)$, and $w(DL_2 + 2 \mid DL_2)$, respectively. These can be computed as follows:
  - $w(DL_2|DL_2) = \frac{DL_2}{d} \cdot \frac{DL_2 - 1}{d-1}$
  - $w(DL_2 + 1|DL_2) = \frac{DL_2}{d} \cdot \frac{\max[d - DL_2, 0]}{d-1} \cdot 2$
  - $w(DL_2 + 2|DL_2) = \frac{\max[d - DL_2, 0]}{d} \cdot \frac{\max[d - DL_2 - 1, 0]}{d-1}$

  Now DL_3 can be computed as

$$CDL_3 = DL_2 \cdot w(DL_2|DL_2) + (DL_2 + 1) \cdot w(DL_2 + 1|DL_2) + (DL_2 + 2) \cdot w(DL_2 + 2|DL_2) \tag{23}$$

  p_3 can be computed as

$$p_3 = \frac{d - DL_3}{d} \cdot \frac{d - 1 - DL_3}{d - 1} + \frac{DL_3}{d} \cdot \frac{d - DL_3}{d - 1} \tag{24}$$

- *i previous visits:* Similarly, for $i \geqslant 3$ previous visits, we obtain DL_i and p_i as follows:
  - $w(DL_{i-1}|DL_{i-1}) = \frac{DL_{i-1}}{d} \cdot \frac{DL_{i-1} - 1}{d-1}$
  - $w(DL_{i-1} + 1|DL_{i-1}) = \frac{DL_{i-1}}{d} \cdot \frac{\max[d - DL_{i-1}, 0]}{d-1} \cdot 2$
  - $w(DL_{i-1} + 2|DL_{i-1}) = \frac{\max[d - DL_{i-1}, 0]}{d} \cdot \frac{\max[d - DL_{i-1} - 1, 0]}{d-1}$

  Now, with the precondition that DL_{i-1} dirty links are present before the ith visit, DL_i is computed in the following formula:

$$CDL_i = DL_{i-1} \cdot w(DL_{i-1}|DL_{i-1}) + (DL_{i-1} + 1) \cdot w(DL_{i-1} + 1|DL_{i-1}) + (DL_{i-1} + 2) \cdot w(DL_{i-1} + 2|DL_{i-1}) \tag{25}$$

  and

$$p_i = \frac{d - DL_i}{d} \cdot \frac{d - 1 - DL_i}{d - 1} + \frac{DL_i}{d} \cdot \frac{d - DL_i}{d - 1} \tag{26}$$

Note that in the computation of DL_2, the notations $w_2(2)$, $w_2(3)$, and $w_2(4)$ denote the probabilities that after two visits, 2, 3, and 4 links are dirty, respectively. These terms describe the unconditional probabilities associated with those values after 2 visits. Since $DL_1 = 2$ is a fact known to us, we have $w_2(2) = w(2 \mid 2)$, $w_2(3) = w(3 \mid 2)$, and $w_2(4) = w(4 \mid 2)$. The conditional and unconditional probabilities are equal when $i = 2$. In the conditional estimates of dirty links, we use the terms DL_1 and DL_2, while we use CDL_i for $i \geqslant 3$ because the preconditions for estimating DL_1 and DL_2, $DL_0 = 0$ and $DL_1 = 2$, are deterministic. Also note that the $w_i(j)$ is a unconditional probability that is associated with j, an integer number of dirty links and i, an integer number of times of visits; whereas $w(p \mid q)$, without subscript, denotes a conditional probability that is independent of the times of visits, and the values p and q are usually non-integer numbers.

4.2.1. Complexity analysis

Observe from Eq. (25) that CDL_i can be computed in $O(i)$ steps because each step to compute $CDL_j (j = 0, \ldots, i)$ takes constant time. Hence, the conditional estimate of the number of dirty links can be computed in linear time.

4.3. Unconditional estimate of dirty links

Since the expected number of "dirty links" after i visits, DL_i, is a random variable when $i > 1$, the proper estimate of DL_i should have listed all the possible number of "dirty links" after i visits and then sum them up weighted by their corresponding probabilities. For example, for DL_3, the possible number of "dirty links" after three visits is 2, 3, 4, 5, or 6, and the formula for computing DL_3 should be

$$DL_3 = 2 \cdot w_3(2) + 3 \cdot w_3(3) + 4 \cdot w_3(4) + 5 \cdot w_3(5) + 6 \cdot w_3(6) \tag{27}$$

The probabilities are given as follows:

- $w_3(2) = w_2(2) \cdot w(2 \mid 2)$
- $w_3(3) = w_2(2) \cdot w(3 \mid 2) + w_2(3) \cdot w(3 \mid 3)$
- $w_3(4) = w_2(2) \cdot w(4 \mid 2) + w_2(3) \cdot w(4 \mid 3) + w_2(4) \cdot w(4 \mid 4)$
- $w_3(5) = w_2(3) \cdot w(5 \mid 3) + w_2(4) \cdot w(5 \mid 4)$
- $w_3(6) = w_2(4) \cdot w(6 \mid 4)$

The computation of these $w_i(j)$ that are required to compute the unconditional estimate of DL_3 is illustrated in Fig. 1.

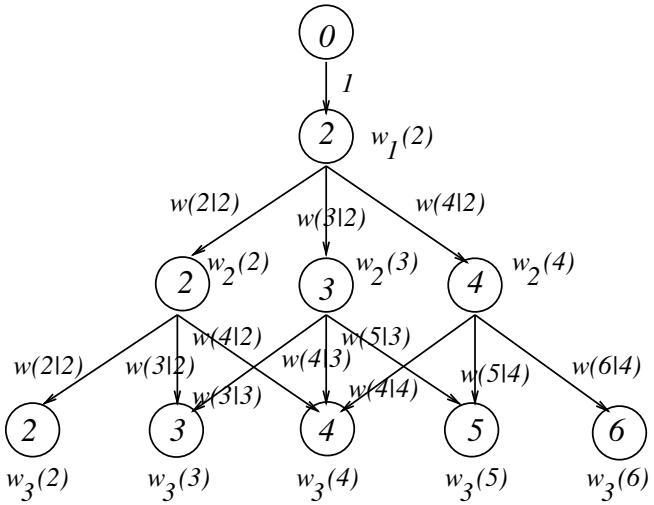

Fig. 1. Unconditional estimate of DL_3.

Generalizing the above derivation, we can show that

$$w_i(j) = \begin{cases} w_{i-1}(j)\cdot w(j|j) & \text{if } j = 2 \\ w_{i-1}(j)\cdot w(j|j) + w_{i-1}(j-1)\cdot w(j|j-1) & \text{if } j = 3 \\ w_{i-1}(j)\cdot w(j|j) + w_{i-1}(j-1)\cdot w(j|j-1) + w_{i-1}(j-2)\cdot w(j|j-2) & \text{if } 3 < j < 2i-1 \\ w_{i-1}(j-1)\cdot w(j|j-1) + w_{i-1}(j-2)\cdot w(j|j-2) & \text{if } j = 2i-1 \\ w_{i-1}(j-2)\cdot w(j|j-2) & \text{if } j = 2i \end{cases} \quad (28)$$

As mentioned in Section 4.2, the term with the subscript is called "unconditional probability". For example, $w_i(j)$ means the probability that exactly j links are dirty after the node has been visited i times. The term without subscript is "conditional probability" or "single step probability". For example, $w(a \mid b)$ denotes the conditional probability that a links are dirty after the next visit, given that the current number of dirty links is b ($b+2 \geqslant a \geqslant b$). The computation of DL_i in Section 4.2 is an estimation that uses only the "single step probabilities" with the assumption that the number of dirty links is given by DL_{i-1} before the ith visit. A seemingly more accurate formula is given by the following equation:

$$DL_i = \sum_{j=2}^{2i} j \cdot w_i(j), \text{ where } w_i(j) \text{ is given in Eq. (28)} \quad (29)$$

$j = 2, \ldots, 2i$ enumerates all possible number of dirty links after i visits. For now, we assume that the node degree is sufficiently large so that we can reasonably ignore the case that all links become "dirty". $j-2$, $j-1$, and j are the only possible number of dirty links prior to the ith visit that may result in j dirty links after the ith visit.

Fig. 1 illustrates on the bottom layer all the possibilities (of the number of dirty links) after three visits. The numbers in the circles represent the possible numbers of dirty links. The term attached to a circle represents the unconditional probability associated with the circled number (dirty links) and the layer index (number of previous visits). The terms labeled on the edges are conditional probabilities which indicate the transition from one state to another after one more visit.

When the number of visits to the current node is 0, the number of dirty links can only be 0, so we define $w_0(0) = 1$, and at layer 0 there is only one circle. At layer 1, which describes the resulting possibilities after only one visit, we also have only one circle because the number of dirty links can only be 2. The algorithm for computing $w_i(j)$ is to find all paths from the top circle to the circle enclosing j at layer i. For each of the paths, the probability corresponding to this path is computed by multiplying all the conditional probabilities $w(a \mid b)$ along that path. By summing up the path probabilities for all such paths leading to that circle from the top circle, we obtain the unconditional probability $w_i(j)$.

4.3.1. Complexity analysis

To compute DL_i ($i \geqslant 1$), we need to compute $w_i(j)$ for $j = 2, \ldots, 2i$. The computation of $w_i(j)$ in turn, depends on $w_{i-1}(k)$ for $k = j-2$, $j-1$, and j. We use dynamic programming to compute and record this unconditional probability $w_i(j)$ for all the layers of circles as illustrated in Fig. 1. Observe that we have i^2 such values for any given i. The single step probabilities $w(a \mid b)$ along the edges are determined by a, b, and the node degree d. Note that the value of a can only be b, $b+1$, or $b+2$. The probabilities are

$$\left.\begin{aligned} w(b|b) &= \tfrac{b}{d}\cdot\tfrac{b-1}{d-1} \\ w(b+1|b) &= \tfrac{b}{d}\cdot\tfrac{\max[d-b,0]}{d-1}\cdot 2 \\ w(b+2|b) &= \tfrac{\max[d-b,0]}{d}\cdot\tfrac{\max[d-b-1,0]}{d-1} \end{aligned}\right\} \quad (30)$$

These are computed the same way we compute the conditional probabilities in Section 4.2. It can be seen from Fig. 1 that, to compute DL_i, we must compute the values for all edges that lead to layer i from layer 1 (or layer 0), the number of which is $3\cdot(i-1)^2$. We must also compute the i^2 unconditional probability values. The time complexity for evaluating $w_i(j)$ is constant when the unconditional probability values of the upper layer and all the associated edge values are known, as given in Eq. (28). Thus, the complexity for computing DL_i is $O(i^2)$ and the space requirement is also $O(i^2)$.

4.4. Comparison of estimation approaches

We claim that the unconditional estimate of the number of dirty links UDL_i is an accurate estimate because it takes into account all the possible values of the number of dirty links after i visits. However, this method takes time complexity and storage requirement both of $O(i^2)$. In Section 4.2, we introduced the conditional estimate that bases the next step estimation on the result of the previous step, which has time complexity of $O(i)$ for computing DL_i. The differences between these two methods may serve as a justification for the simplified estimation of dirty links using the conditional approach.

For simplicity, assume that d is large enough so that we do not have to bother with the marginal cases of having all links dirty. We also denote the conditional estimate of DL_i as CDL_i and the unconditional estimate as UDL_i. It is obvious that $CDL_2 = UDL_2$.

Assume that at step $i-1$, the number of dirty links is m. We then compute DL_{i+1} using the unconditional method:

$$\begin{aligned} UDL_{i+1} = {} & m w_{i+1}(m) + (m+1) w_{i+1}(m+1) + (m+2) w_{i+1}(m+2) \\ & + (m+3) w_{i+1}(m+3) + (m+4) w_{i+1}(m+4) \end{aligned} \quad (31)$$

We then compare this result with that from the conditional estimation:

$$\begin{aligned} CDL_{i+1} = {} & DL_i w(DL_i|DL_i) + (DL_i+1) w(DL_i+1|DL_i) \\ & + (DL_i+2) w(DL_i+2|DL_i) \end{aligned} \quad (32)$$

The value of UDL_{i+1} can be computed as follows. There are only three possible values for DL_i:

- if $DL_i = m$, what is the expected value for DL_{i+1}?
- if $DL_i = m+1$, what is the expected value for DL_{i+1}?
- if $DL_i = m+2$, what is the expected value for DL_{i+1}?

Then we average these three values weighted by $w(m \mid m)$, $w(m+1 \mid m)$, and $w(m+2 \mid m)$ (computed using Eq. (30)), which are the possibilities that DL_i should be m, $m+1$, or $m+2$, respectively. This is functionally equivalent to the computation described in Section 4.3.

- if $DL_i = m$, then at the next step:

$$DL_{i+1} = m w(m|m) + (m+1) w(m+1|m) + (m+2) w(m+2|m) \quad (33)$$

$$= m\frac{m(m-1)}{d(d-1)} + (m+1)\frac{2m(d-m)}{d(d-1)} + (m+2)\frac{(d-m)(d-m-1)}{d(d-1)} \quad (34)$$

$$= m\frac{d-2}{d} + 2 \quad (35)$$

We can see that DL_{i+1} depends only on d and m. Similarly, we have the following:

- if $DL_i = m+1$, then

$$DL_{i+1} = (m+1)\frac{d-2}{d} + 2 \quad (36)$$

- if $DL_i = m+2$, then

$$DL_{i+1} = (m+2)\frac{d-2}{d} + 2 \quad (37)$$

Now, to compute DL_{i+1} concerning all possible values of DL_i $(m, m+1, m+2)$, we make the weighted average:

$$UDL_{i+1} = w(m|m)\left[m\frac{d-2}{d}\right] + w(m+1|m)\left[(m+1)\frac{d-2}{d}\right] + w(m+2|m)\left[(m+2)\frac{d-2}{d}\right] \quad (38)$$

Note that $w(m|m) = \frac{m(m-1)}{d(d-1)}$, $w(m+1|m) = \frac{2m(d-m)}{d(d-1)}$, and $w(m+2|m) = \frac{(d-m)(d-m-1)}{d(d-1)}$ according to Eq. (30). Finally, it turns out:

$$UDL_{i+1} = m\left(\frac{d-2}{d}\right)^2 + 4\frac{d-1}{d} \quad (39)$$

Now compute the conditional estimate of DL_{i+1}, according to Eq. (35):

$$CDL_{i+1} = DL_i\frac{d-2}{d} + 2 = m\left(\frac{d-2}{d}\right)^2 + 4\frac{d-1}{d} \quad (40)$$

Comparing Eqs. (39) and (40), we have

$$UDL_{i+1} = CDL_{i+1} \quad (41)$$

Based on the above analysis, we have the following conclusion: *Conditional estimate and unconditional estimate of dirty links give the same result*.

This result seems counter-intuitive at first glance. However, the conditional method uses a generalization at each step, and this is a linear generalization of the state at the previous visit. The final result of the unconditional method is also a linear computation from the previous steps, which implies that the conditional method does not lose any precision against the unconditional method.

The unification of the two approaches of estimation gives us the convenience of computing the number of dirty links with linear time and space complexity. As per Eq. (35), the expected numbers of dirty links, $DL_i (i \geqslant 2)$, can be expressed in terms of the node degree d only. For example, $DL_2 = DL_1 \cdot \frac{d-2}{d} + 2 = 4(d-1)/d$ and $DL_3 = DL_2 \cdot \frac{d-2}{d} + 2 = 2(\frac{d-2}{d})^2 + 4\frac{d-1}{d}$.

4.5. Influence of previous visits

In the last section, we formulated the concept DL_i, which denotes the expected number of "dirty" links after i visits have been paid to the current node. Using DL_i, we can then compute the corresponding p_i, which denotes the probability that the next message forwarding takes a "fresh" link. In this case, however, the current node may have been visited 0, 1, 2, or more times before, which means that the value of $p(x)$ in Eq. (20) is also a random function of the message overhead, x. So, we need to figure out the expected value of $p(x)$, which depends on the probabilities of how many times the current node has been visited before.

Given that after x messages have been forwarded, $u(x)$ nodes are covered, we now focus on the computation of such probabilities. Let us denote the probability that an arbitrary node in the network was visited i times after x messages as $q_i(x)$.

- $q_0(x)$: After x messages were arbitrarily forwarded, the node was not visited. This probability can be expressed as [9]:

$$q_0(x) = \left(1 - \frac{1}{N}\right)^x \quad (42)$$

- $q_i(x)$: The probability that the node was visited exactly i times $(i = 0, \ldots, x)$ is:

$$q_i(x) = \left(1 - \frac{1}{N}\right)^{x-i} \cdot \left(\frac{1}{N}\right)^i \cdot \binom{x}{i} \quad (43)$$

Now we come back to the computation of $p(x)$ in Eq. (20). Note that $p(x)$ is the probability that the next message takes a "fresh" link out, and this probability depends on how many times the current node has been visited before, which is also a random variable depending on x. The expected value of $p(x)$ thus can be computed by taking the average of the probabilities of "fresh links out", weighted by the probabilities of the corresponding number of previous visits:

$$p(x) = \sum_{i=0}^{x} p_i(x) \cdot q_i(x) \quad (44)$$

5. Analysis and simulations

Eq. (20) proposed a refined model to compute node coverage, $u(x)$. The refinement introduces the influence of the (potential) previous visits to the current node on the probability of visiting a new node at each forwarding step. This consideration was prompted by the deviation between the model results and the simulation results. The deviation also reflects the difference between the message forwarding process and a random sampling. In a random sampling process, a sampling is independent of previous steps. Message forwarding, on the other hand, is dependent on the status of the current node. Consider an extreme condition: if all links of the current node have been explored, then the next forwarding will not find a new node at all, while with random sampling, this probability is equal to the current proportion of new nodes in the network.

Note from Eq. (43) that the probability $q_i(x)$ decreases dramatically with i when N, the total number of nodes, is reasonably large and x is not overly large. As an example, let $N = 20,000$, $x = 10,000$, we have:

- $q_0(x) = 0.6065$
- $q_1(x) = 0.3033$
- $q_2(x) = 0.076$
- $q_3(x) = 0.01263$
- $q_4(x) = 0.00158$

From this example, it is observed that for x values that are not too large, say, less than N, the probabilities of multiple visits more than two times are so small that we can reasonably ignore them. When we use k terms for the summation in Eq. (44), the refined model is termed as the "*k-order refined model*". k-order refinement considers the influence of 0, 1, ..., k previous visits. The higher the order of refinement, the more precise results the model will produce.

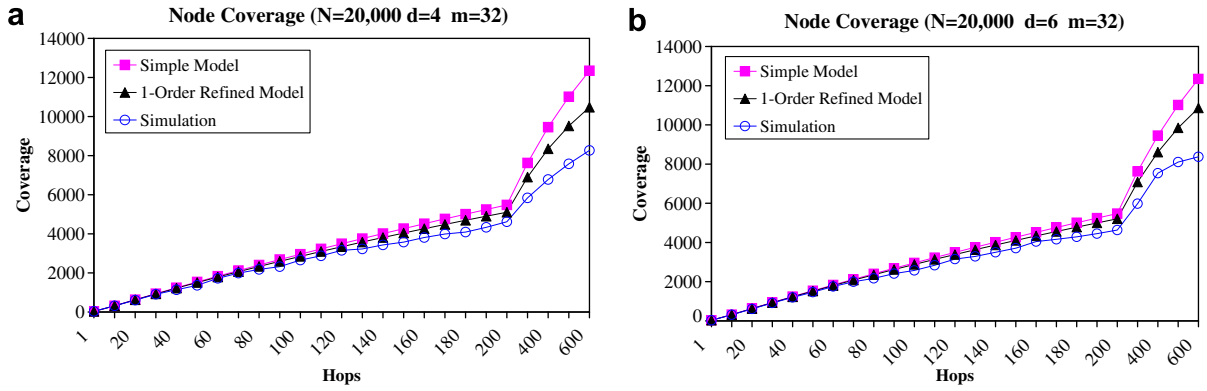

Fig. 2. 1-Order refinement on node coverage. (a) Average node degree = 4. (b) Average node degree = 6.

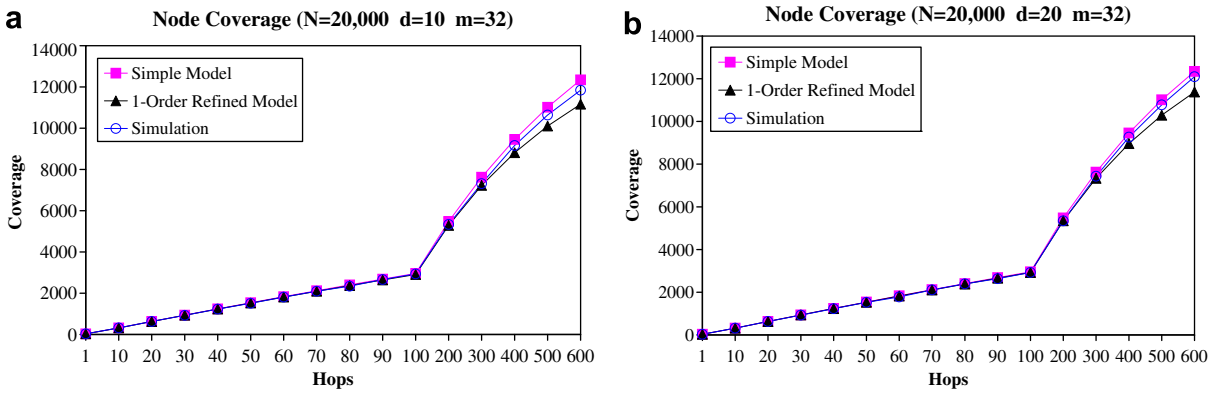

Fig. 3. 1-Order refinement on node coverage. (a) Average node degree = 10. (b) Average node degree = 20.

In Section 5.1, we apply the 1-order refinement of the algebraic model (Section 2) to compute the node coverage with various average node degrees. A comparison of the results from the simple algebraic model, from the 1-order refined algebraic model, and from the simulations is presented in Figs. 2 and 3. In Section 5.2, we apply the 2-order refinement and perform a similar comparison. The distinction between these two refined models is also analyzed in that section. Note that in the graphs, there is an abrupt change in the shape of the curves at a certain number of hops because we change the scale of hops (X-axis) to accommodate both small hop count and large hop count in the same graphs.

5.1. 1-Order refined model

It is observed from Figs. 2 and 3 that the results yielded by the refined model fall below those for the simple model, which is expected since the refinement term, $p(x)$, decreases the probability of visiting a new node at each step.

We also noticed that in the cases where average node degree d equals 4 or 6, the 1-order refinement still results in node coverage values greater than those in the simulation results. This is somehow unexpected at first glance, since the 1-order refinement underestimates the probability. (The 2-order term and higher order terms are neglected from the computation of $p(x)$). Two reasons are possible for this scenario:

- *Precision of simulation:* The precision of simulation is significantly affected by average node degree. For smaller node degree setups, the results oscillate wildly among multiple runs. Even when we use the average of results from multiple runs, the error is still large and unstable. This can be illustrated from the fluctuations (though not visibly prominent) on the curves for the simulation results in Fig. 2.
- *Possible graph separation at low node degree:* When the average node degree is small, it has been shown by Erdõs and Rényi [8] that is likely that the graph is not connected: some parts of the network can never be reached from the starting node. Specifically, consider the $G(n,p)$ model, where p is the probability of a link. One of the properties of the random graph is as follows:
  - If $pn < ln\ n$, then a graph in $G(n,p)$ will almost surely not be connected.
  - If $pn > ln\ n$, then a graph in $G(n,p)$ will almost surely be connected

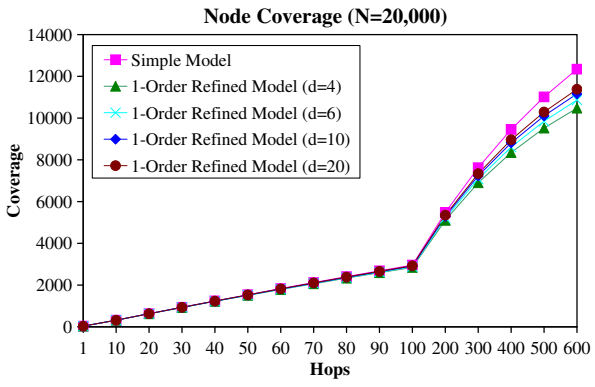

Fig. 4. Impact of node degree on 1-order refined model.

In our simulation where random graphs of 20,000 nodes are generated, for the cases where the average node degree is 4 and 6, the generated graphs are almost surely not connected, and in the cases where the node degrees are 10 and 20, the graphs are almost surely connected. This phenomena significantly reduces the resulting node coverage generated from the simulation in which a disconnected overlay *may* exist, i.e., for $d = 4$ and 6. Note that in our analysis models, we assume the network is connected.

Fig. 4 illustrates how the refined model is affected by the node degree. Note that the simple model takes no account for this effect, thus resulting in over-estimated node coverage, which is plotted as the top curve. Compared to the simple model, the refined model lowers the node coverage in all cases with respect to the different node degree values. It is observed from the figure that the smaller the node degree, the more the deviation of the refined model from the simple model. In other words, the simple model produces larger errors for a lower node degree topology than for higher node degree topologies. This is because for a smaller node degree, the effect of node revisits becomes more important. If the current node has been visited before, then the chance that the next message forwarded from this node visits a new node is smaller for a lower degree node than for a higher degree node (as there are more fresh links for higher degree nodes).

It is also worth noting that for all node degree values, the influence of the node revisits becomes significant only when the hop

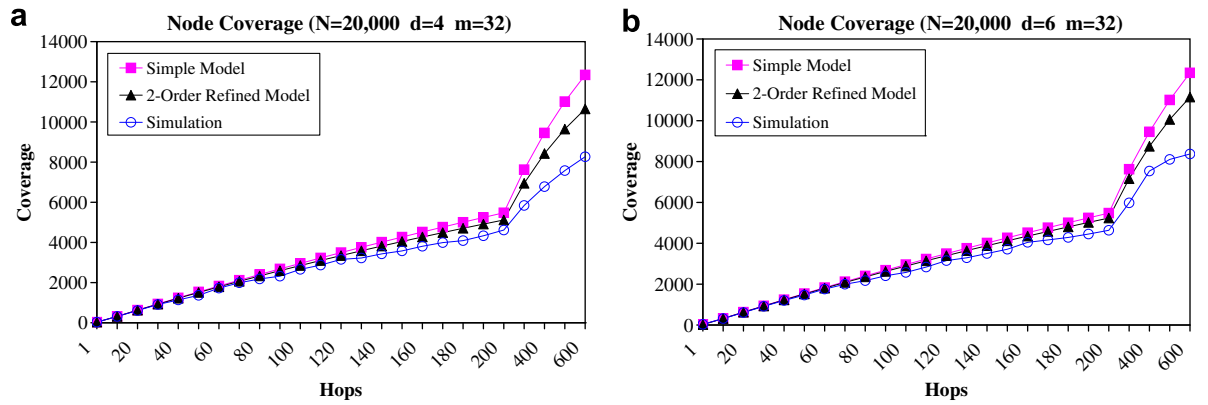

Fig. 5. 2-Order refinement on node coverage. (a) Average node degree = 4. (b) Average node degree = 6.

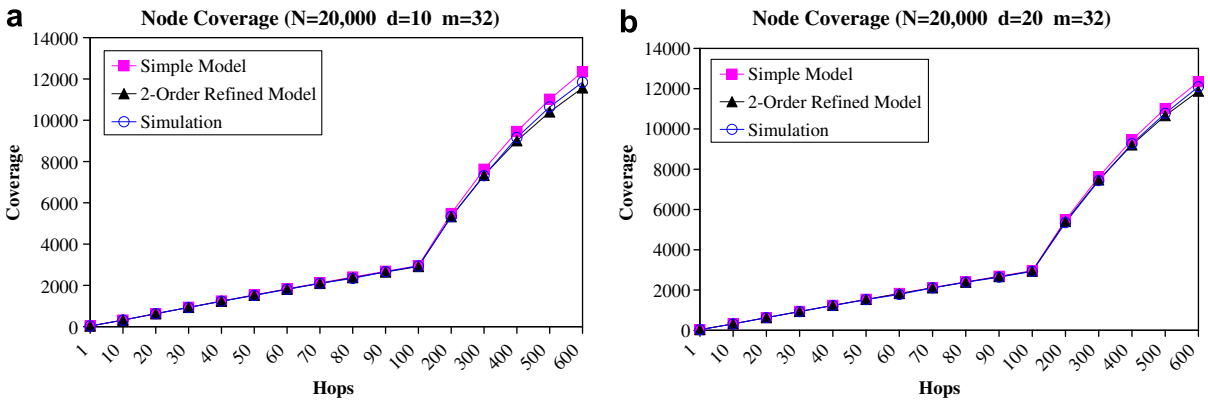

Fig. 6. 2-Order refinement on node coverage. (a) Average node degree = 10. (b) Average node degree = 20.

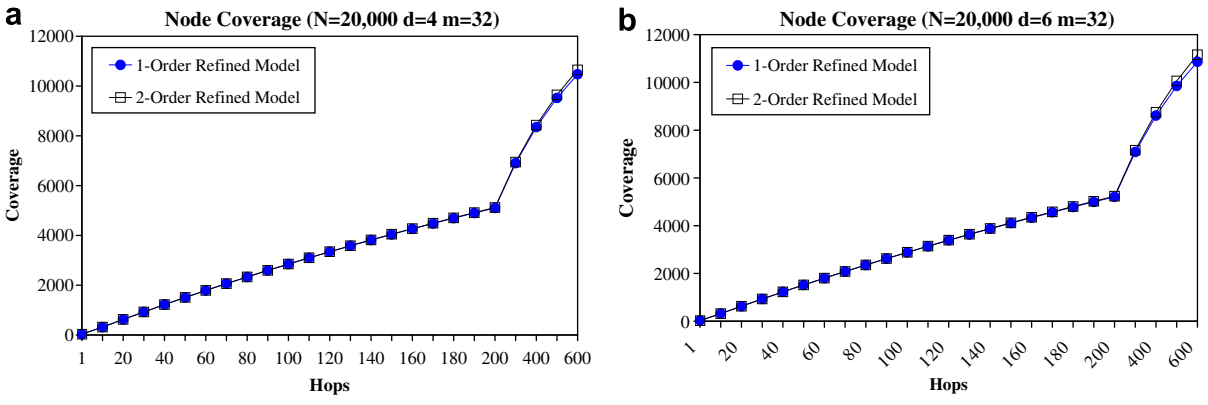

Fig. 7. Comparing 1-order and 2-order refined models. (a) Average node degree = 4. (b) Average node degree = 6.

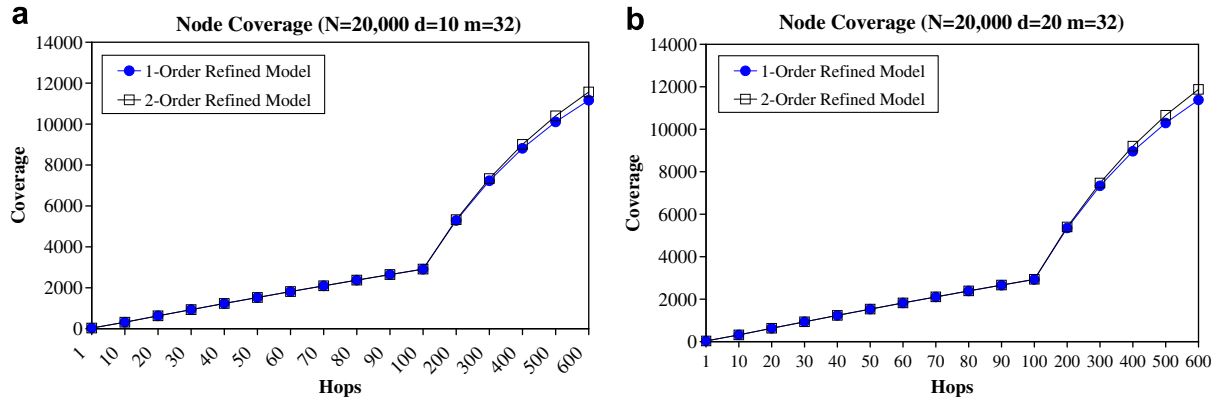

Fig. 8. Comparing 1-order and 2-order Refined Models. (a) Average node degree = 10. (b) Average node degree = 20.

number, or, message overhead, becomes considerably large. The deviation of the refined model from the simple model reflects the impact of node revisits. As the hop number increases, the number of nodes that have been visited before and the expected number of visits also increase, which in turn increase the impact of node links.

5.2. 2-Order refined model

Observe from Fig. 3 that although the simple model over-estimates the node coverage, the 1-order model tends to under-estimate node coverage as the hop number becomes large. The curve for our refined model falls below that for the simulation results, which is more obvious as the hop number grows beyond 200 in this figure. This is due to the order of refinement we used for the model.

The impact factor of node revisits, $p(x)$, as defined in Eq. (44), neglects the second and higher order terms in the 1-order refined model. These terms represent the scenario of multiple previous visits to the node. The probability for revisits more than once with small hop number is very small and negligible; this is the reason for using 1-order model for simplicity. However, when the hop number is large enough, the probability will become significant and the corresponding higher order terms are no longer negligible. The truncation of those higher order terms results in a reduced value of node coverage. The greater the hop number, the more significant is this effect, as shown at the right end in Fig. 3.

To account for this effect, we include the second order term in computing $p(x)$ and the 2-order refined model. We perform the same comparison as in Section 5.1, however, using this model to replace the 1-order model. The results are displayed in Figs. 5 and 6. We base our analysis primarily on Fig. 6 because the simulation results in Fig. 5 have unreasonably large variations and instability due to small node degrees, as explained in Section 5.1.

Comparing Figs. 6 and 3, the curves for the 2-order refined model turn out to be much closer to the simulation curve than those for the 1-order model when the hop number is larger. This confirms our analysis. The slight difference between this model and the simulation is still due to term truncation in computing $p(x)$: even higher order terms will begin to take effect as hop number grows even larger.

The comparison between Figs. 6 and 3 further confirms the validity of our refined models and provides a hint for model design: *if a large number of messages is expected in the message propagation process, we should use the refined models with higher order for more accuracy*.

Finally, Figs. 7 and 8 compare the 1-order and the 2-order refined models under different node degrees. As predicted from the earlier analysis in this section, the curves for the 2-order refined model always lie above that for the 1-order refined model and their difference becomes more significant as the hop number grows.

6. Discussion and conclusions

Message propagation on random graphs is used in a wide range of applications such as search in unstructured P2P networks, modeling the spread of infection in epidemiology, and propagation of gossip in social networks. Until now, message propagation processes were generally modeled only as random sampling processes. As the node coverage and the number of messages increase, we showed that random sampling no longer serves as an accurate model. In this paper, we studied the distinction between the message propagation process and random sampling over the $G(N,p)$ random graphs. We investigated the effect of "node degree" in the message propagation upon the efficiency of covering distinct nodes. Compared to the "pure" random sampling model, this factor has a negative effect: the actual node coverage is less than that given by the simplified models using random sampling. This influence is more significant when the node coverage becomes high and when the average node degree is small. The difference was also quantitatively studied. We then introduced refined models that account for "dirty links", which is the reason for the reduced probability of forwarding a message to a new node. The number of "dirty links" is a random variable and the expectation of this number is dependent upon the number of times that the current node has been visited before. The number of (previous) visits to a node is also a random variable whose distribution depends on x, the message overhead. Under normal conditions, the probability that a node has been visited many times is usually small and these cases can be omitted depending on how precise we want our refined model to be.

We presented our quantitative analysis based on the random graph topology, whereas in real world applications, the network structures are more close to "small world" networks or power law networks. The quantitative analysis of the message propagation process in small world networks and power law networks still remains a challenge.

References

[1] L. Adamic, R. Lukose, A. Puniyani, B. Huberman, Search in power-law networks, Physical Review E 64 (2001).
[2] X. Bao, B. Fang, M. Hu, Cocktail search in unstructured P2P networks, Proceedings Grid and Cooperative Computing Workshops, LNCS 3252, Springer, 2004. pp. 286–293.
[3] N. Bisnik, A. Abouzeid, Optimizing random walk search algorithms in P2P networks, Computer Networks 51 (6) (2007) 1499–1514.

[4] N. Bisnik, A. Abouzeid, Modeling and analysis of random walker search algorithm in P2P networks, IEEE Workshop on HOT-P2P, 2005.
[5] B. Bollobas, Random Graphs, Academic Press, London, 1985.
[6] V. Cholvi, P. Felber, E. Biersack, Efficient search in unstructured peer-to-peer networks, European Transactions on Telecommunications 15 (6) (2004).
[7] E. Cohen, S. Shenker, Replication strategies in unstructured peer-to-peer networks, ACM SIGCOMM (2002) 177–190.
[8] P. Erdõs, A. Rényi, Random graphs. Publications Mathematics (Debrecen) 6 (1959) 290.
[9] C. Gkantsidis, M. Mihail, A. Saberi, Random walks in peer-to-peer networks: algorithms and evaluation, Performance Evaluation 63 (2006) 241–263.
[10] C. Gkantsidis, M. Mihail, A. Saberi, Hybrid search schemes for unstructured peer-to-peer networks, Proceedings of IEEE Infocom, 2005.
[11] K.Y.K. Hui, J.C.S. Lui, D.K.Y. Yau, Small world overlay P2P networks, Proceedings of the 12th International Workshop on Quality of Service, IWQoS, 2004.
[12] J. Kim, G. Fox, A hybrid keyword search across peer-to-peer federated databases, ADBIS (Local Proceedings) 2004.
[13] M. Li, W.-C. Lee, A. Sivasubramaniam, Semantic small world: an overlay network for peer-to-peer search, Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP), 2004.
[14] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, S. Lim, A survey and comparison of peer-to-peer overlay network schemes, IEEE Communications Survey and Tutorial, March 2004.
[15] Q. Lv, P. Cao, E. Cohen, K. Li, S. Shenker, Search and replication in unstructured peer-to-peer networks, International Conference on Supercomputing (ICS) (2002) 84–95.
[16] F. Otto, S. Ouyang, Improving search in unstructured P2P systems: intelligent walks (I-Walks), Intelligent Data Engineering and Automated Learning (IDEAL 2006), LNCS 4224, Springer, 2006. pp. 1312–1319.
[17] J. Risson, T. Moors, Survey of research towards robust peer-to-peer networks: search methods, Computer Networks 50 (17) (2006) 3485–3521.
[18] S. Tiwari, L. Kleinrock, Analysis of search and replication in unstructured peer-to-peer networks, in Proceedings of ACM SIGMETRICS 2005, Banff, Canada, June 2005, Full version appears as UCLA Technical Report, 2005, ftp://ftp.cs.ucla.edu/tech-report/2005-reports/050006.pdf.
[19] D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (1998) 440–442.
[20] B. Wu, A.D. Kshemkalyani, Analysis models for blind search in unstructured overlays, Proceedings of the Fifth IEEE Symposium on Network Computing and Applications (NCA) (2006) 223–226.
[21] B. Yang, H. Garcia-Molina, Efficient search in peer-to-peer networks, IEEE International Conference on Distributed Computing Systems (ICDCS), 2002.
[22] D. Zeinalipour-Yazti, V. Kalogeraki, D. Gunopulos, On constructing internet-scale P2P information retrieval systems, DBISP2P, 2004, pp. 136–150.
[23] H. Zhuge, X. Chen, X. Sun, Preferential walk: towards efficient and scalable search in unstructured peer-to-peer networks, ACM WWW Conference, 2005.