This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2008 proceedings.

Reducing Maximum Stretch in Compact Routing

Mihaela Enachescu
Dept of Computer Science
Stanford University
mihaela@cs.stanford.edu

Mei Wang
Dept of Electrical Engineering
Stanford University
wmei@stanford.edu

Ashish Goel
Dept of Management Science and Engineering
Stanford University
ashishg@stanford.edu

***Abstract*—It is important in communication networks to use routes that are as short as possible (i.e have low stretch) while keeping routing tables small. Recent advances in compact routing show that a stretch of 3 can be achieved while maintaining a sub-linear (in the size of the network) space at each node [14]. It is also known that no routing scheme can achieve stretch less than 3 with sub-linear space for arbitrary networks. In contrast, simulations on real-life networks have indicated that stretch less than 3 can indeed be obtained using sub-linear sized routing tables[6]. In this paper, we further investigate the space-stretch tradeoffs for compact routing by analyzing a specific class of graphs and by presenting an efficient algorithm that (approximately) finds the optimum space-stretch tradeoff for any given network.**

We first study a popular model of random graphs, known as Bernoulli random graphs or Erdős-Renyi graphs, and prove that stretch less than 3 can be obtained in conjunction with sub-linear routing tables. In particular, stretch 2 can be obtained using routing tables that grow roughly as $n^{3/4}$ where n is the number of nodes in the network.

Compact routing schemes often involve the selection of landmarks. We present a simple greedy scheme for landmark selection that takes a desired stretch s and a budget L on the number of landmarks as input, and produces a set of at most $O(L \log n)$ landmarks that achieve stretch s. Our scheme produces routing tables that use no more than $O(\log n)$ more space than the optimum scheme for achieving stretch s with L landmarks. This may be a valuable tool for obtaining near-optimum stretch-space tradeoffs for specific graphs. We simulate this greedy scheme (and other heuristics) on multiple classes of random graphs as well as on Internet like graphs.

I. Introduction

Routing messages is a central functionality of a network. For the efficient use of network resources (link capacities, etc.) we want to build schemes which route along paths that are as short as possible. This is measured by the *stretch factor*, which is the maximum ratio between the length of the path traversed by a message and the length of the shortest path between its source and its destination. On the other hand, as networks grow in size it becomes important to reduce the amount of memory that needs to be maintained at every node for routing purposes.

There is an obvious trade-off between the storage requirement and the stretch factor. On the one hand you can have shortest path routing (i.e. with a stretch factor of 1) while maintaining $O(n \log(n))$ tables at each node (one entry for every other node in the network for a total storage of $O(n^2 \log(n))$. In that scenario upon receiving a message, a node can just look up the entry corresponding to the destination and forward along the shortest path. At the other extreme, you can maintain only an $O(\log(n))$ identifier at each node, and send messages by "flooding" the entire network, in a depth-first search manner for example, with a worst-case stretch factor equal to n for general graphs. Neither of these solutions scales well.

For special types of graphs (e.g. trees [8], [11], outerplanar and planar networks [12], [13], growth-bounded networks [12] [10]) there are routing schemes which achieve optimal or near-optimal stretch with sub-linear memory requirements. Except in the case of special graphs mentioned above, routing schemes with $o(n)$ memory requirement and stretch values less than 3 have been particularly hard to achieve in general graphs. In fact, lower-bound results indicate that *universal routing schemes* cannot achieve stretch less than 3 with $o(n)$ memory at each node [5].

A. Our Results

In this paper, we study how a landmark-based routing approach can lead to achieving stretch values less than 3 with high probability while maintaining $o(n)$ memory requirements for Internet-like graphs and other large distributed networks. We also study the complementary problem: given a maximum stretch value s, find out the minimum routing table necessary to achieve this stretch value.

In Section II, we present the details of the routing scheme we consider. It is similar to the Thorup and Zwick [8] stretch-3 *universal* (i.e. for all graphs) *routing scheme*. The main idea is to select a set of landmarks that act as a post-office/forwarding center for nearby nodes.

In Section III, we show that with high probability such a routing scheme can achieve stretch less than 3 with $o(n)$ space requirement in Bernoulli random graphs. Our method relies on neighborhood results extended from the work of Chung and Lu [9] for Bernoulli random graphs. We show that stretch $s = 2$ can be achieved with $\tilde{O}(n^{3/4})$ memory at each node in Bernoulli random graphs (as opposed to $\Omega(n)$ memory for general graphs. In general, stretch s can be achieved while using $\tilde{O}(n^{\frac{2}{s+1}+\epsilon})$ memory at each node. The proof of this generalized result is omitted due to space constraints. We hope our results for Bernoulli random graphs can be extended to power-law graphs via a similar neighborhood expansion analysis.

Research supported in part by NSF grant CCR-0126347, NSF Career grant No. 0339262, and an Alfred P. Sloan Faculty Fellowship.

In Section IV, we present and analyze a simple compact routing scheme that achieves an approximately optimum stretch-space tradeoff. The core of our scheme is a simple (and efficient) greedy algorithm for landmark selection. Our scheme takes a desired stretch s and a budget L on the number of landmarks as input, and produces a set of at most $O(L \log n)$ landmarks that achieve stretch s. Our scheme produces routing tables that use no more total space than the optimum landmark-based scheme for achieving stretch s with L landmarks, ignoring the space used to route to the landmarks. This may be a valuable tool for obtaining near-optimum stretch-space tradeoffs for specific graphs.

In Section V, we perform simulations of our routing schemes on Bernoulli random graphs, on power law graphs, and on the Internet AS graph. We observe that in practice stretch less than 3 can be achieved with $o(n)$ memory requirement at each node. We also explore a highest-degree landmark selection heuristic as an alternative to the random landmark selection explored in Section III.

B. Related Work

Much of the work in compact routing has been incremental, and some of the initial algorithms and ideas proved to be so flexible as to allow subsequent layers of requirements to be added to the existing framework. One such example is the first $o(n)$ algorithm of stretch 3, proposed by Cowen [4], which required $\tilde{\mathrm{O}}(n^{2/3})$ maximum space at every node. This algorithm is based on a very natural idea: to designate a set of "landmarks" which cover all other nodes in the network. The non-landmark nodes are then re-labeled to include the name of the closest landmark. Each node needs to remember routing information regarding its immediate neighbors and the landmarks. Expanding on this idea, Thorup and Zwick [7], [8] created a scheme that achieves $\tilde{\mathrm{O}}(\sqrt{n})$ memory for stretch 3, matching the lower bound for general graphs (up to $\log$-factors). Furthermore they provide a generalized scheme that achieves $2k-1$ stretch while using $\tilde{\mathrm{O}}(n^{1/k})$ bits of memory at each node, for any integer $k \geq 2$. They do not explore stretches less than 3, since lower bounds indicate it cannot be achieved in universal compact routing schemes.

To the best of our knowledge there are no lower-bounds for random graphs and power-law graphs known other than those of *universal routing schemes* for which the following is known: in order to use $o(n)$ memory at each node (or $o(n^2)$ total memory), we need to route with a stretch factor of at least 3, as shown by Gavoille and Gengler [5]. More generally a girth conjecture of Erdos and others, implies that $\Omega(n^{1+1/k})$ bits of total storage are needed to give distances with stretch strictly less than $2k+1$, for any integer $k \geq 1$. This conjecture is proved for $k = 1, 2, 3, 5$ as documented in [15]. We should note that these lower bounds are proved under strict naming constraints (names are usually restricted to $\log n$ size for a network with n nodes).

Previous empirical studies document that simple routing scheme have excellent performance on Internet-like graphs [6]. Under certain assumptions regarding the distance distribution in these graphs, Krioukov et. al [6] show an average stretch of 1.1 and table size of 50 for the AS-graph which contains about 1000 nodes for the Thorup and Zwick (TZ) scheme. The authors note that the current degree distribution observed for the AS-graph, as well as the standard deviation of this distribution lead to close to optimum results for average stretch, and make them question the "existence of a certain link between the Internet topology and the analytical structure of the average TZ stretch function." [6] In contrast to this work our paper focuses on the study of the maximum, rather than the average stretch value. We also study methods that allow us to find out the minimum routing table necessary to achieve a given stretch value s.

We will assume, as in [4], [8], [7], [16] that the designer of the scheme is allowed to assign a poly-logarithmic name to each node (labeled model). Name-independent schemes [2], [3], [10], [1], are inherently harder that labeled schemes, but can many times be derived from the corresponding labeled scheme by using a low-stretch lookup service before routing on the low-stretch path. Our work fits well with the current research agenda of the IRTF/RRG group, which deals with similar scalability in routing issues. In fact, scalability is their top goal, while stretch is also an important goals. Our paper studies the tradeoff between these two goals.

II. Model and Definitions

In this section we formalize our network and routing model, and introduce some of the basic notation that will be used in later sections.

We view a communication network as a symmetric, weighted, finite graph $G = (V, E, \delta)$, $|V| = n$, the nodes representing computers/processors, the edges - bidirectional communication links, and δ - the edge weight function. Each node v is assigned a unique identifier, and can have arbitrary degree. The weights on the edges induce a *distance* between any two nodes u, v, given by the sum of the edge weights on the shortest path connecting them, and which we will denote by $d_G(u, v)$ or $d(u, v)$. Note that Bernoulli random graphs are un-weighted, and for these, the distance becomes the hop distance between two nodes.

A *routing scheme* RS is a distributed algorithm for message delivery between any two nodes in the network. A name dependent routing scheme consists of a *distributed data structure*, a *delivery protocol*, and a *routing label* for every node in the graph. The message is delivered via a sequence of transmission determined *uniquely* by the distributed data structure (with no centralized control and no randomization).

The length of the path traversed by a message from u to v according to the routing scheme R is denoted by $d_R(u, v)$. We can now formally define the *stretch factor* of the scheme R as $max_{u,v} \frac{d_R(u,v)}{d(u,v)}$.

A. Landmark-Based Routing Scheme

The routing scheme we study in this paper is based on maintaining information about a set of landmarks as well as the neighboring nodes. There are four aspects we need to

make explicit: the landmark selection procedure, the storage requirement, the labeling, and the actual routing.

Since the scheme is supposed to bound the necessary memory requirements at all nodes, for any possible graph, a fundamental realization is that neighborhoods need to be defined in terms of volume as well as radius. Thus, we define and denote by $B_t(v)$ the set of the t closest to v, breaking ties by increasing node identities. We use $N_i(v)$ to denote the set of nodes at distance at most i from v (in number of hops), and $\Gamma_i(v)$ to denote those nodes at distance exactly equal to i from v.

As in the Cowen [4] scheme, we opt for picking the landmark set LS at random, in a single stage. To ensure that with high probability any node $v \notin LS$ has a landmark in its $B_v(t)$, we recall the following known result:

Lemma II.1 *[Awerbuch et al. [3]] Let L be a set of landmark produced by marking a node as landmark with probability $cn/t \log n$ (where $c \geq 2$ is a fixed constant). Then with probability at least $p = 1 - \frac{1}{n^{c-1}}$, any node $v \notin LS$ has a landmark in its $B_t(v)$ and $|L| \leq \frac{2cn \ln n}{t}$.*

We denote by l_v the landmark assigned to v, and by r_v the distance (number of hops for the Bernoulli analysis) from v to its landmark. Each non-landmark node has label (v, l_v, p_{l_v}), as in the Cowen scheme. To route from a node u to another node v, u first checks if v is in its neighborhood/routing table. If so, u has the next hop info for v and uses it, otherwise, u has the next hop info for l_v and uses that.

The storage requirement has three components. First, every node needs to maintain the next hop information for the landmarks. The second routing component, which we will denote by RRT_1 serves to route from a landmark l_v to the node v: any node u that is in the path from l_v to v needs to remember the next hop info for v. This component is necessary because the volume based neighborhoods are non-symmetrical. In previous schemes [4], [8] remembering the neighborhood ensures RRT_1 is covered. We make the assumption that if a node w is on the shortest path[1] from l_v to v, then w needs to remember v. This is what we call a *landmark-based routing scheme*.

Finally, in order to achieve a given stretch s, every node u for which $sd(u, v) < d(u, l_v) + d(l_v, v)$ needs to remember the next-hop info for v. We denote this storage requirement by RRT_2, and this will be the focus of our space requirement analysis. It is not necessary that all nodes in the path from u to v remember the next hop info. If u' satisfies $sd(u', v) > d(u', l_v) + d(l_v, v)$, then $\frac{d(u,u')+d(u',l_v)+d(l_v,v)}{d(u,u')+d(u'v)} < s$, so the resulting path satisfies the stretch requirement.

III. Stretches Less than 3 in Bernoulli Random Graphs

In this section, we will show that stretches between 1 and 3, can be achieved with high probability in a large family of random graphs. The graphs we are considering here are the Bernoulli random graphs denoted by $G(n, p)$. The parameter n represents the number of nodes of a graph in this class. The parameter p is the probability that an edge exists between any two nodes in the graph (independent coin toss for each edge).

We will study the radius of the volume based neighborhoods $R_t(v) = \max\{d(v, u) | u \in B_t(v)\}$, as well as distance based neighborhoods $N_i(v) = \{u | d(u, v) \leq i\}$ and $\Gamma_i(v) = \{u | d(u, v) = i\}$. We will show the following:

Theorem III.1 (Achieving Stretch 2 in Random Graphs) *Assume $2c_0 \cdot n^{3/4} \log n$ landmarks are selected, and assume the labels and next-hop info for all $n^{3/4}$ closest neighbors are remembered at each node. Also suppose $p > \frac{2 \log n}{n}$ and $np = o(n^{1/9})$. Then with probability at least $1 - o(n^{-1})$, we have a stretch of at most 2.*

The assumption $np > 2 \log n$ ensures the average degree is above the connectivity threshold, so that the graph is connected, while the assumption $np = o(n^{1/9})$ excludes very high density graphs.

To prove our result we will use a stronger version of the results regarding neighborhood size from Chung and Lu [9] which we state below. We provide the proof of the next two lemmas in the appendix.

Lemma III.2 (Upper-Bound Lemma) *Suppose $p > \frac{c \log n}{n}$ for a constant $c \leq 2$. Then with probability at least $1 - o(n^{-2})$, we have*

$$|\Gamma_i(x)| \leq \tfrac{11}{c}(np)^i \quad \forall 1 \leq i \leq n \tag{1}$$

$$|N_i(x)| \leq \tfrac{13}{c}(np)^i \quad \forall 1 \leq i \leq n \tag{2}$$

Lemma III.3 (Lower-Bound Lemma) *Suppose $p > \frac{c \log n}{n}$ for a constant $1 \leq c \leq 2$ and suppose that $np \leq n^{1/6}$. Then, for each vertex x in the giant component (in case $G(n, p)$ is not connected), and for each i satisfying $i_0 \leq i \leq \frac{2}{3} \frac{n}{\log(np)}$, with probability at least $1 - o(n^{-2})$, we have:*

$$|\Gamma_i(x)| \geq \tfrac{5}{c}(np)^{i - i_0} \tag{3}$$

where i_0 satisfies $i_0 \leq \lfloor \frac{2}{c} \rfloor + 1$.

Proof of Theorem III.1:

To show the desired stretch result, it is enough to bound the ratio $\frac{r}{R}$ by $\frac{1}{2}$, where r is an upper bound of the distance from a node x to its landmark l_x (as defined in Section II-A) and R is a lower-bound on the radius of $B_{n^{3/4}}(x)$ for any node x, i.e. $R = \min_x R_{n^{3/4}}(x)$. To obtain a bound on the ratio, we first bound r and R individually.

First, by choosing $c_0 = 3$ in Lemma II.1, we get that with probability at least $1 - o(n^{-2})$ every node x has a landmark within its closest $n^{1/4}$ neighbors. We will now use this to bound r_x, the largest distance from x to its landmark, using the fact that the sum of all rings at distance $1, 2, \ldots, r_x$ from

[1] when multiple shortest paths exist, we break ties consistently, for example by smallest node id from v to l_v

a node x is at most $n^{1/4}$ with high probability. This result is true for any node x, thus using the union bound we get that with probability at least $1 - o(n^{-1})$ the following is true for r (the upper bound on all distances):

$$n^{1/4} \geq |\Gamma_1(x)| + |\Gamma_2(x)| + \ldots + |\Gamma_r(x)|$$

Now, since the result in Lemma III.3 holds for a given node x with probability at least $1 - o(n^{-2})$, then with probability $o(n^{-1})$ it will hold for all x, so we can lower bound $|\Gamma_i(x)|$ as long as $i_0 \leq i \leq 2/3\frac{n}{\log(np)}$. If $1 \leq i < i_0$ we can use the trivial lower bound of 1. [Note: we check in the end that indeed $r \leq 2/3\frac{n}{\log(np)}$ to be able to ensure that the use of the lemma is legal.] Now, using Lemma III.3 we get:

$$\Sigma_{i=1}^{r}|\Gamma_i(x)| \geq i_0 + \frac{5}{c}((np)^0 + (np)^1 + \ldots + (np)^{r-i_0})$$

This result will give us a way to bound r. We have:

$$n^{1/4} \geq i_0 + \frac{5}{c}\left(\frac{(np)^{r+1-i_0} - 1}{(np) - 1}\right)$$

Taking the logarithm of both sides, we get:

$$\begin{aligned}\log(n^{1/4} - i_0) \quad \geq \quad & \log 5 - \log c \\ & + \log((np)^{r+1-i_0} - 1) - \log((np) - 1)\end{aligned}$$

Since $\log$ is an increasing function and $i_0 \geq 0$ the above equation implies that:

$$\frac{1}{4}\log n \geq \log 5 - \log c + \log((np)^{r+1-i_0} - 1) - \log(np)$$

Since $\log(x-1) \geq \log x - c_1$, for any constant $c_1 > 1$ and x large enough, we get:

$$\frac{1}{4}\log n - \log 5 + \log c + \log(np) + c_1 > (r + 1 - i_0)\log(np)$$

And thus, $r < i_0 + \frac{\frac{1}{4}\log n - \log 5 + \log c + c_1}{\log(np)}$ (note the necessary bound on r mentioned in the beginning of our proof does hold for large enough values of n).

Similarly, we can bound R using Lemma III.2. First, recall that we only consider neighborhoods up to size $n^{3/4}$, thus we need to consider at most an R s.t.:

$$|\Gamma_1(x)| + |\Gamma_2(x)| + \ldots + |\Gamma_R(x)| \geq n^{3/4}$$

or in other words $|N_R(x)| \geq n^{3/4}$. From Lemma III.2 with probability at least $1 - o(n^{-2})$: $|N_R(x)| \leq \frac{10}{c}(np)^R$ for a given x, and using union bound, this result holds in fact *for all* x with probability $1 - o(n^{-1})$. Thus we get the following bound on R:

$$R\log(np) \geq \frac{3}{4}\log n - \log 10 + \log c$$

We want to show that $\frac{r}{R} \leq \frac{1}{2} \Leftrightarrow 2r \leq R$. It is enough then to argue (based on the bounds on r and R) that:

$$\begin{aligned}& 2(i_0\log(np) + \frac{1}{4}\log n - \log 5 + \log c + c_1) \\ & \leq \frac{3}{4}\log n - \log 10 + \log c \Leftrightarrow \\ & 2i_0\log(np) + \log c + 2c_1 + \log 10 \leq \frac{1}{4}\log n + 2\log 5 \Leftrightarrow \\ & \frac{2i_0\log(np) + \log c + 2c_1 + \log 10}{\log n} \leq \frac{1}{4} + \frac{2\log 5}{\log n}\end{aligned}$$

The only contributions on the LHS that does not tend to zero as $n \to \infty$ is the first term. Using the hypothesis assumption that $np = o(n^{1/9})$ and $c = 2$ we can bound the $\frac{2i_0\log(np)}{\log n}$ term by $\frac{2}{9}$. Since this is clearly less than $\frac{1}{4}$, and given that the other terms are negligible, we conclude that the above inequality holds.

It remains to show why this bound on $\frac{r}{R}$ induces a stretch of at most 2. Let x be the destination node, and y the source node. If the routing from x to y is done via a non-direct path then $D = d_G(x,y) > R$ (otherwise this destination node would be in the neighborhood of x). The routing cost $d_R(x,y)$ of the non-direct path is bounded by $r + (r + D)$, where r represents the bound from the destination node y to l_y, and $r + D$ represents a bound from x to the landmark l_y (using triangle inequality). Thus the stretch cannot exceed $\frac{r+r+D}{D} \leq \frac{R+D}{D} \leq 2$.

■

Note that the argument regarding the inter-dependence of the stretch s and the ratio $\frac{r}{R}$ exposed in the proof of Theorem III.1 can be expanded. In general, in order to guarantee a stretch value s we need to have $\frac{r}{R} \leq \frac{s-1}{2}$. This ratio gets arbitrarily close to 0 as s approaches the optimum stretch 1.

Since in Lemma III.3 i_0 is a constant, $\frac{r}{R}$ is bounded away from 0 as long as $\frac{\log(np)}{\log n}$ is bounded away from 0 and thus we cannot guarantee stretch values arbitrarily close to 1 without imposing strict restrictions on how high the average degree np can be. In general we can show the following tradeoff:

Theorem III.4 *Suppose $np \geq 2\log n$. If $np = o(n^{\tilde{c}})$ then we can achieve stretches $s > 1 + 2\tilde{c}$ while storing $\tilde{O}(n^x)$ bits at each node, where $x \geq \frac{2}{s+1} + \epsilon(\tilde{c})$, where $\epsilon(\tilde{c}) \to 0$ as $\tilde{c} \to 0$*

The proof is similar with the proof in the previous section as is omitted due to space constraints. Notice the implications for the memory requirement, which increases steeply as $s \to 1$, going above n for $s \approx 1.2$ for $n = 100000$. For $s \in [1.5, 3]$, however, the memory requirement is quite scalable. Note also that the upper bound on the average degree must decrease as $s \to 1$.

IV. Greedy Algorithm for Landmark Selection

In this section we present a simple greedy scheme for landmark selection that takes a desired stretch s and a budget L on the number of landmarks as input, and produces a set of at most $O(L\log n)$ landmarks that achieve stretch s. Our scheme produces routing tables that use no more total space than the optimum *landmark-based scheme* (as defined in

Section II-A) for achieving stretch s with L landmarks. This may be a valuable tool for obtaining near-optimum stretch-space tradeoffs for specific graphs. Note that the analysis does not depend on any assumptions regarding the graph, so the algorithm can be applied to any general network topology.

Let us denote by $M_{v,l}$ the number of nodes w that would need to keep v in either their RRT_1 (i.e. w is on the shortest path[1] from v to l_v) or RRT_2 (i.e. for which the stretch is above s) given that $l_v = l$ (i.e. l is v's designated landmark). Let T_i be the total memory requirement at step i. Let $v_{\mathcal{L}}$ be the *best* landmark of node v from the set $\mathcal{L}$, i.e. the landmark that is the argument of $\min_{l\in\mathcal{L}} M_{v,l}$.

We consider the memory requirement for storing landmark information separately. We will study the remaining total memory requirement (i.e. for RRT_1 and RRT_2) of the following greedy algorithm for choosing landmarks:

Input: Graph G, initial landmark l_1, parameter L
Output: Landmark set $\mathcal{L}$ containing $L\log n$ elements
$\mathcal{L} \leftarrow l_1$;
$T_1 = \sum_{v\neq l_1} M_{v,l_1}$;
for $i = 2$ *to* $2L\log n$ **do**
  $T_i = T_{i-1}$;
  foreach $x \in V - \mathcal{L}$ **do**
    $N = 0$;
    foreach $v \in V - \mathcal{L}$ *and* $v \neq x$ **do**
      $N = N + \min(M_{v,x}, M_{v,v_{\mathcal{L}}})$
    end
    if $N < T_i$ **then**
      $T_i = N$;
      $l_i = x$;
    end
  end
  $\mathcal{L} \leftarrow l_i$;
end

Algorithm 1: Greedy Landmark Selection

Note: l_1 can also be chosen via the same Greedy method assuming we start with 0 landmarks and $T_0 = n^2$.

Theorem IV.1 *The above Greedy algorithm is guaranteed to produce a landmark set with memory requirement less than or equal to the total memory requirement of an optimal set of size L. The Greedy will select $2L\log n$ landmarks (by construction).*

Greedy Performance: In the following proof we will track the memory requirements of an optimal choice of L landmarks (the OPT) compared to the memory requirement of our algorithm (GREEDY). In particular we will compare the number of "entries" utilized by these solutions.

Each pair of nodes (w, v) represents a potential memory "entry", i.e. w need to remember v in reverse routing table (either RRT_1 or RRT_2) unless either v is a landmark or v's landmark l is such that $d(w, l) + d(l, v) \leq sd(w, v)$. If the entry is not needed, we will say that a landmark covers the entry.

Lemma IV.2 *The number of covered entries only increases during GREEDY.*

Proof: Although the specific entries may get covered/uncovered, this lemma is true because landmarks, once added are never deleted (thus entries covered by the first case remain covered), and a node changes landmarks only if this improves its number of covered entries (thus entries covered by the second case are exchanged with a larger set of covered entries that were not previously covered). ■

Now let us analyze what we can say about the magnitude of the decrease during consecutive stages of GREEDY. Let $D_i = |T_i - \text{OPT}|$ if $T_i > OPT$ ($D_i = 0$ otherwise). There are at least D_i entries that could be covered by OPT but remain uncovered by GREEDY after i landmarks are added to GREEDY. Since OPT covers these entries using L landmarks, there exists one landmark in OPT that is not currently used by GREEDY and which could cover at least $\frac{1}{L}D_i$ new elements in GREEDY. By construction, GREEDY will pick a new landmark that covers at least this many elements in the next stage, so that $|T_{i+1} - \text{OPT}| \leq \left(1 - \frac{1}{L}\right) D_i$.

This argument holds for any of the $L\log n$ stages. Also initially $D_1 = |T_1 - \text{OPT}| \leq n^2$, thus at the end of the algorithm:

$$D_{2L\log n} \leq \left(1 - \frac{1}{L}\right)^{2L\log n} D_1 < 1$$

Since the difference can only be a non-negative integer, the above shows that after $2L\log n$ landmark are added GREEDY covers at least as many elements as OPT.

■

Note that although the above greedy routine ensures the RRT_1 and RRT_2 requirements are no more than those of the landmark-based optimal scheme, we will be using $O(\log n)$ more landmarks.

V. Simulations and Real Network Data Results

In this section, we present simulation results to verify the theoretical studies shown in previous sections and extensions to power-law graphs and real network graphs. We focus on the scalability of routing table size (RRT_2 in particular) as the total number of nodes increases while keeping the maximum stretch to less than or equal to 2.

A. Random Graphs

Random graphs are generated based on [17]. With a total of n nodes, each pair of nodes has a probability of p that these two nodes are directly connected by an edge. We explored several schemes of landmark construction based on random graphs.

1) Scheme 1: Random Landmark Selection: In this scheme, landmarks are randomly picked among the nodes as in Section III. The simulation constitutes the following steps.
1. Randomly select $n^{2/3}$ nodes as landmarks in a random graph with a total of n nodes;
2. For each node u, choose the landmark that is the closest

to this node L_u;
3. For each source node u, route to all destination nodes v by first going to v's landmark node L_v and then from L_v to v. If $d(u, L_v) + d(L_v, v) > 2d(u, v)$, i.e., stretch between u and v is greater than 2, v is placed in u's routing table;
4. Find the maximum routing table size among all nodes.

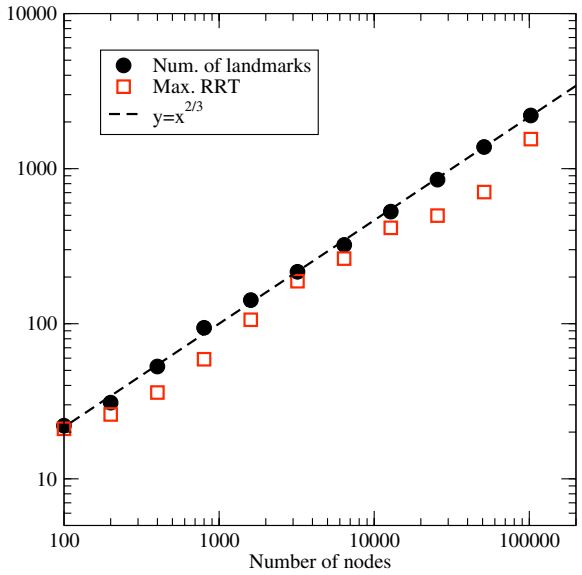

Fig. 1. Random Landmark Selection scheme on random graphs: max RRT scales as $n^{2/3}$ using random selection of $n^{2/3}$ landmarks.

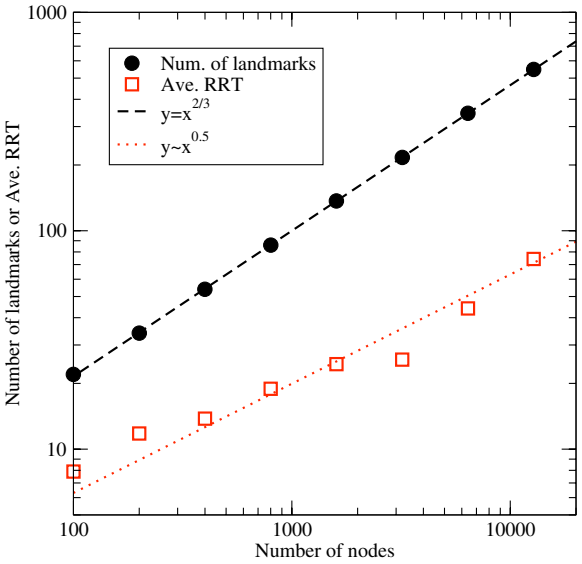

Fig. 2. Greedy Landmark Selection on random graphs: ave RRT scales as $n^{1/2}$ using greedy selection of $n^{2/3}$ landmarks.

Figure 1 shows that when the number of landmarks increases as $n^{2/3}$, the maximum RRT[2] size also increases as $n^{2/3}$, consistent with the theoretical studies laid out in Section III.

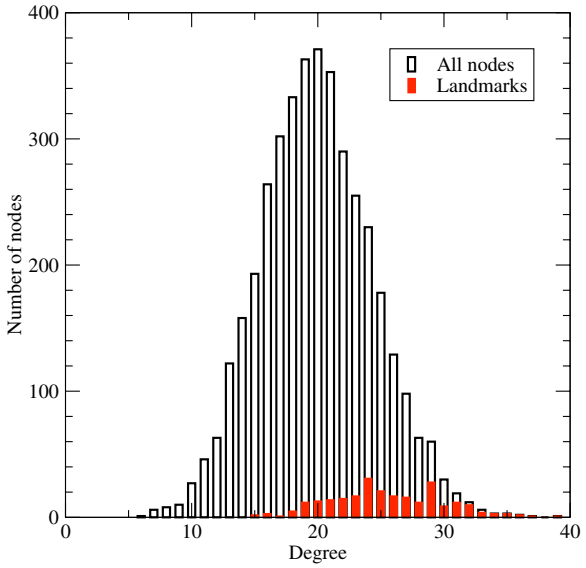

Fig. 3. Histogram of node degree and the degree distribution of landmarks chosen by Greedy Landmark Selection scheme on a random graph with 4000 nodes total and 252 landmarks.

2) Scheme 2: Greedy Landmark Selection: The Greedy Landmark Section algorithm iteratively picks each landmark that minimizes the RRT size. When applied to random graphs, we found that the RRT size is much reduced compared to random selection of landmarks, $n^{1/2}$ rather than $n^{2/3}$. The results on Greedy Landmark Selection are summarized in Figure 2.

The smaller RRT size from Greedy Landmark Selection demonstrates that Greedy algorithm is more efficient in selecting landmarks. Analysis of the landmarks chosen from this scheme show that most of the landmarks chosen are high degree nodes, as seen in Figure 3. This result suggests that an efficient choice of landmarks is by picking those nodes with higher degrees than others.

3) Scheme 3: Highest-degree Landmark Selection: The simulation results based on Greedy Landmark Selection prompted us to experiment with a heuristic approach – picking the landmarks based on the degree of the nodes:

1. Rank the degree of all n nodes and pick $n^{2/3}$ of the nodes with the highest degrees as landmarks.

Steps 2-4 are the same as those in the Random Landmark Selection. This scheme is intuitive and easy to implement. Our simulation results, shown in Figure 4, show that the average RRT scale as $n^{1/2}$, almost identical to the results based on Greedy Landmark Selection.

B. Power-law Graphs

Previous studies have shown that real networks behave more like power-law graphs [6] (otherwise known as scale-

[2]in this section RRT refers to RRT_2

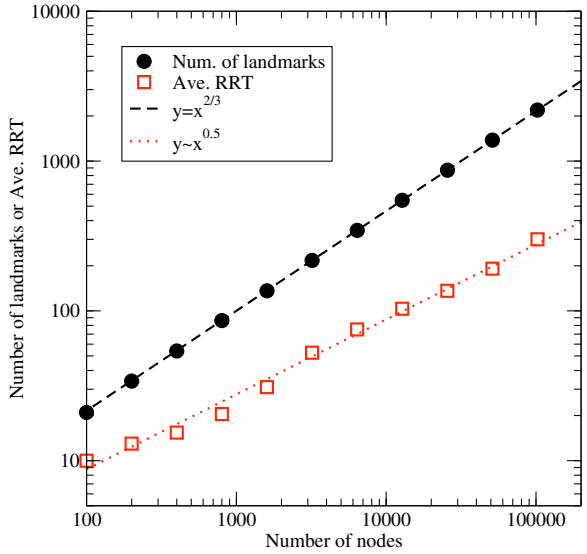

Fig. 4. A heuristics method, Highest-degree Landmark Selection, on random graphs: average RRT scales as $n^{1/2}$ by choosing $n^{2/3}$ of highest-degree nodes as landmarks.

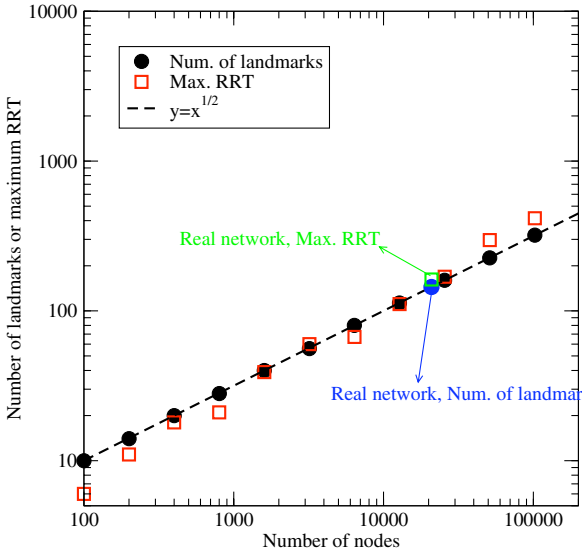

Fig. 5. Highest-degree Landmark Selection on power-law graphs and AS graph from real network: max RRT scales as $n^{1/2}$ using $n^{1/2}$ of highest-degree nodes as landmarks.

free graphs) rather than random graphs. The degrees of the nodes follow power-law distributions with a small fraction of the nodes having a high degree of connections with others. It is important to analyze the scaling behavior of our technique on these networks.

The power-law graphs are generated according to [18] where several seed nodes are all connected to each other. With the addition of each new node, it is connected to one or a few of the existing nodes with a probability that is proportional to the degree of the existing nodes.

Because of the heavy-tailed nature of the power-law graphs, it is natural to apply the scheme with Highest-degree Landmark Selection, which has been shown above to generate near-optimum landmarks for random graphs. The simulation results plotted in Figure 5 show that the scaling in power-law graphs is much more favorable than in random graphs. Even when the number of landmarks is reduced to $n^{1/2}$ (compared to $n^{2/3}$ in random graphs), the maximum RRT size is also limited to $n^{1/2}$. Note that stretch is also limited to 2 in these simulations on power-law graphs.

C. AS Graph from Real Network Data

To verify our results on real network topography, we carried out simulations on the current AS graph acquired from CAIDA [19] with a total of 20906 number of nodes. By applying the Highest-degree Landmark Selection to the AS graph, we found that results match very well with our simulations on power-law graphs (see Figure 5).

These simulation results on the AS graph are quite significant. The existing routing scheme requires that each node stores the routing information to all other nodes in the network. By switching to compact routing, each node only needs to store the routing information to a very small fraction of the nodes: the landmarks and RRTs. In this particular case with the AS graph, we found that the memory required is less than 2% of what is needed currently. This large gain in memory is accomplished by relaxing the stretch, which is bound to 2 with our approach. The simulation results on power-law graphs demonstrate that such scheme is scalable to much larger network sizes.

VI. Conclusions

In this paper we explore ways to lower the bound of the maximum stretch to values below 3, while maintaining sublinear average and maximum routing table size. We use the stretch value 2 as an illustrative example throughout the paper, but the results can be generalized to achieve any stretch less than 3.

Through theoretical proofs and simulations, we demonstrate the bounds for stretch and routing table size, using three landmark selection schemes on three types of graphs: random graphs, power-law graphs, and AS graph from real network.

Our results show: 1). Stretch less than 3 can be achieved with high probability in a large family of random graphs while maintaining sublinear memory size on each node. We presented the memory vs. stretch and average-degree vs. stretch tradeoffs specific to this family of graphs in Lemma III.4. 2). Selecting high degree nodes as landmarks is efficient for reducing memory consumption while keeping stretch bounded by a stretch value $s < 3$. 3). Compact routing scheme applying to AS graph from real network can reduce the maximum routing table size at each node to 1.5% (0.7% on average) of

the total number of nodes on the entire network and achieve maximum stretch bounded by 2.

These results are encouraging for reducing routing table size and increasing scalability of the network.

REFERENCES

[1] I. Abraham, C. Gavoille, D. Malkhi, N. Nisan, and M. Thorup, Compact Name-Independent Routing with Minimum Stretch, *16th ACM Symposium on Parallelism in Algorithms and Architectures*, 2004.

[2] M. Arias, L. J. Cowen, K. A. Laing, R. Rajaraman, and O. Taka Compact Routing with Name Independence, *15th ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 184–192, 2003.

[3] B. Awerbuch and A. Bar-Noy and N. Linial and D. Peleg, Compact Distributed Data Structures for Adaptive Routing, *Proc. 21st ACM Annual Symposium on Theory of Computing*, pages 479–489, 1989.

[4] L. J. Cowen, Compact Routing with Minimum Stretch *Proc. ACM Annual Symposium on Discrete Algorithms*, 1999.

[5] C. Gavoille and M. Gengler. Space-Efficiency for Routing Schemes of Stretch Factor Three. *Journal of Parallel and Distributed Computing*, 61(5):679–687, 2001.

[6] D. Krioukov, K. Fall, and X. Yang Compact Routing on Internet-Like Graphs, *IEEE INFOCOM*, 2004.

[7] M. Thorup, U. Zwick, Approximate Distance Oracles, *Proc. 33 rd Ann. ACM Symp. on Theory of Computing*, 2001.

[8] M. Thorup, U. Zwick, Compact Routing Schemes, *ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, 2001.

[9] F. Chung and L. Lu, The Diameter of Random Sparse Graphs, *Advances in Applied Mathematics*, 2001.

[10] I. Abraham and D. Malkhi, Name Independent Routing for Growth Bounded Networks, *ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, 2005.

[11] N. Santaro and R. Khatib Labelling and Implicit Routing in Networks, *The Computer Journal*, 28:5-8, 1985.

[12] G. N. Fredrickson and R. Janardan, Designing Networks and Compact Routing Tables, *Algorithmica*, 3:171-190, 1988.

[13] I. Abraham and D. Malkhi, Compact Routing on Euclidean Metrics, *23rd ACM Symposium on Principles of Distributed Computing*, 2004.

[14] I. Abraham, C. Gavoille, and D. Malkhi, Routing with Improved Communication-space Trade-off. *18th International Symposium on Distributed Computing (DISC)*, 2004.

[15] D. Peleg and E. Upfal, A Trade-off between Space and Efficiency for Routing Tables, *STOC*, 1988.

[16] T. Eliam, C. Gavoille, D. Peleg, Compact Routing Schemes with Low Stretch Factor, *Symposium on Principles of Distributed Computing*, 1998.

[17] B. Bollobas, Random Graphs, *Academic Press*, 1985.

[18] R. Albert, A. Barabasi, Statistical Mechanics of Complex Networks, *Reviews of Modern Physics, Vol 74*, 2002.

[19] www.caida.org

VII. APPENDIX

We will need the following lemma (obtained via application of Chernoff bounds) which we quote from Chung and Lu [9]:

Lemma VII.1 (Binomial Distribution) *$B(t,p)$ denotes the binomial distribution with probability p in a space of size t.*

- *Suppose X dominates $B(t,p)$. For $a > 0$, we have* $\Pr(X < tp - a) \leq e^{-\frac{a^2}{2tp}}$
- *Suppose X is dominated by $B(t,p)$. For $a > 0$, we have* $\Pr(X > tp + a) \leq e^{-\frac{a^2}{2tp} + \frac{a^3}{(tp)^3}}$

A. Proof idea for Lemma III.2

Since we closely follow the approach in [9] for this proof we only mention the change in a parameter needed to show the stronger probability bound necessary for proving Theorem III.1.

In the original proof the authors define a parameter λ which they chose to be equal to $\sqrt{5 \log n}$ to obtain the probability of $1 - o(n^{-1})$. We pick instead $\lambda' = \sqrt{7 \log n}$ we can get a probability of $1 - o(n^{-2})$, and modify the constants in the statement of Lemma III.2 appropriately for the results to hold.

B. Proof of Lemma III.3 [Modified from [9]]

For proving Lemma III.3 more changes need to me made to the original proof in [9], so we present the complete proof here.

Proof: We first prove the following statement:
Claim: With probability at least $1 - o(n^{-2})$ there exists a $i_0 \leq \lfloor \frac{2}{c} \rfloor + 1$ satisfying $|\Gamma_{i_0}(x)| \geq d$, where $d = \frac{20}{c}$.

Let $k = \lfloor \frac{2}{c} \rfloor$. Since x is in the giant component, $|\Gamma_k(x)| \geq 1$. There exists a path $xx_1 \ldots x_k$ satisfying $x_j \in \Gamma_j(x)$ for $1 \leq j \leq k$. We write $x_0 = x$. Let $f(x_j)$ denote the number of vertices y, for which $x_j y$ forms an edge but y is not one of those vertices $x_0, x_1, \ldots, x_k$. We compute the probability that $f(x_j) \leq d$ as follows:

$$\begin{aligned} Pr[f(x_j) \leq d] &= \Sigma_{i=0}^{d} C_{n-k-1}^{l} p^l (1-p)^{n-l} \\ &\leq \Sigma_{i=0}^{d} \frac{(np)^l}{l!} e^{-(n-l-k-1)p} \\ &\leq (np)^d e^{-(n-d-k-1)p} \Sigma_{i=0}^{d} \frac{1}{l!} \\ &\leq (c \log n)^d e^{-c(1-\frac{d+k+1}{n}) \log n} e \\ &= o(n^{-c+\epsilon}) \end{aligned}$$

Here, $f(x_j)$ are independent random variables. The probability that $f(x_j) \leq d$ for all $j \leq k$ is at most $o((n^{-c+\epsilon})^{k+1}) = o(n^{-2})$ if ϵ is small enough. Therefore, with probability at least $1 - o(n^{-2})$, there is an index $1 \leq i_0 \leq k+1$ satisfying $f(x_{i_0+1}) \geq d$. Hence, $|\Gamma_{i_0}| \geq d$ (we can be shown by contradiction that the nodes represented by $f(x_{i_0+1})$ cannot have appeared in layers other than $\Gamma_{i_0}, \Gamma_{i_0+1}$ or Γ_{i_0-1}).

From Lemma III.2 it follows that $|N_i(x)| \leq n^{\frac{3}{4}}$ with probability $1 - o(n^{-2})$ for all $1 \leq i \leq \frac{\log n}{\log(np)}$.

Next, we need to prove the following claim for $i = i_0 + 1$:
Claim: Let us denote by I the inequality $|\Gamma_i(x)| \leq \frac{1}{8}|\Gamma_{i_0}(x)|(n - |N_{i_0}(x)|)p$. Assuming that $pn \leq n^{1/6}$ the following holds:

$$\begin{aligned} Pr[I] &= e^{-|\Gamma_{i_0}(x)|(n-|N_{i_0}(x)|)p/8} \\ &\leq e^{-dc(1-n^{-1/4}) \log n/8} \\ &= o(n^{-dc/9}) = o(n^{-2}) \end{aligned}$$

The first inequality holds using Chernoff bounds with $\beta = 1/2$, and noting that the probability a node not in N_{i_0} becomes attached to a node in Γ_{i_0} is at most $1/2|\Gamma_{i_0}|p$. Using $d = \frac{20}{c}$ we obtain a probability at most $o(n^{-2})$ that $|\Gamma_i(x)| \leq \frac{1}{4}|\Gamma_{i_0}(x)|(n - |N_{i_0}(x)|)p$ [or with $d \geq \frac{10}{c}$ that $|\Gamma_i(x)| \leq \frac{1}{8}|\Gamma_{i_0}(x)|(n - |N_{i_0}(x)|)p$].

Combining our two claims, with probability at least $1 - o(n^{-2})$ we have: $|\Gamma_{i_0+1}(x) \geq \frac{1}{2}|\Gamma_{i_0}(x)|(n - |N_{i_0}(x)|)p \geq$

$\frac{1}{3}dnp$. The $\frac{1}{2}$ becomes $\frac{1}{3}$ to compensate for the small difference that is introduced by the $o(n)$ contribution of $N_{i_0}(x)$

A similar argument can be made for $i = i_0 + 2$. In [9] the proof is based on the claim that $|\Gamma_i(x)|$ dominates a random variable with distribution $B(t,p)$ where $t = |\Gamma_{i_0+1}(x)|(n - |N_{i_0+1}(x)|)$, which in general is not true. $|\Gamma_i(x)|$ may be smaller due to some vertices being linked to the previous layer by more than one edge. Thus $B(t,p)$ will double count some vertices. If we add a term that generally will be larger than the double-counting error, we can fix the above statement. Let D_i represent the contribution of extra vertices. By definition $|\Gamma_i(x)| + D_i$ dominates a random variable with distribution $B(t,p)$.

Note that for a vertex to be double-counted two edges from Γ_{i_0+1} need to connect to the same vertex outside of $N_{i_0+1}(x)$. There are $C^2_{|\Gamma_{i_0+1}(x)|}$ possible ways to pick the conflicting edges, and the probability of a conflict is p^2. Thus, the expected value of the number of double-counted vertices is $\leq C^2_{|\Gamma_{i_0+1}(x)|}p^2(n - N_{i_0+1}(x))$. Using a Chernoff bound, we can claim that w.h.p. that the total contribution of the double-counted vertices $D_i \leq 2C^2_{|\Gamma_{i_0+1}(x)|}p^2(n - N_{i_0+1}(x))$. More specifically, with probability at least $1 - o(^{-2})$ we have:

$$\begin{aligned} D_i &\leq 2C^2_{|\Gamma_i(x)|}p^2(n - N_i(x)) \\ &\leq \left(\frac{|\Gamma_i(x)|^2}{2}\right)2p^2n \\ &\leq |\Gamma_i(x)|p^2n\left(\frac{11}{c}\right)(np)^i \\ &\leq |\Gamma_i(x)|\frac{(np)^2}{n}\left(\frac{11}{c}\right)n^{2/3} \\ &= \frac{11}{c}|\Gamma_i(x)|(np)^2n^{-1/3} \end{aligned} \tag{4}$$

For the fourth inequality above we use the fact that $i \leq \frac{2}{3}\log n/\log(np)$ and thus $(np)^i \leq n^{2/3}$.

Therefore, with high probability we have: $Pr\left(|\Gamma_i(x)| + D_i < tp - \lambda\sqrt{tp}\right) < e^{-\frac{\lambda^2}{2}}$.

More specifically, with probability $1 - o(n^{-2}) - e^{-\frac{\lambda^2}{2}}$

$$\begin{aligned} |\Gamma_{i_0+2}| &\geq tp - \lambda\sqrt{tp} - D_{i_0+1} \\ &\geq |\Gamma_{i_0+1}|(n - n^{3/4})p - \lambda\sqrt{|\Gamma_{i_0+1}|np} - D_{i_0+1} \\ &\geq |\Gamma_{i_0+1}|(np)(1 - n^{-\frac{1}{4}} - \frac{11}{c}n^{-\frac{1}{3}}np - \frac{\lambda}{\sqrt{|\Gamma_{i_0+1}|np}}) \\ &\geq \frac{1}{3}d(np)^2(1 - n^{-\frac{1}{4}} - \frac{11}{c}n^{-\frac{1}{6}} - \frac{3\lambda}{\sqrt{(np)^2}}) \\ &\geq \frac{1}{3}d(np)^2(1 - n^{-\frac{1}{4}} - n^{-\frac{1}{7}} - \frac{3\lambda}{\sqrt{(np)^2}}) \end{aligned}$$

In the above we use the upper bound lemma for $|\Gamma_{i_0+1}(x)|$ and we omitted the parameter x since it is understood from context.

By induction on $i \geq i_0 + 2$, we can show that with probability at least $1 - o(n^{-1}) - 2ie^{-\frac{\lambda^2}{2}}$ [the 2 factor in from of the exponential term accounts also for the probability error introduced by the D_i term] the following holds:

$$|\Gamma_i(x)| \geq \frac{d}{3}(np)^{i-i_0}\Pi_{j=2}^{i-i_0}\left(1 - n^{-1/4} - n^{-1/7} - \frac{3\lambda}{\sqrt{(np)^2}}\right)$$

The base case $i = i_0 + 2$ is already done. For the inductive hypothesis, assume the statement holds for $|\Gamma_i|$. We use again the corrected argument that $|\Gamma_{i+1}| + D_i$ dominates $B(t,p)$ for $i_0 + 2 \leq i \leq \frac{2}{3}\log n/\log(np)$. We get, following the same reasoning as in the $i = i_0 + 2$ case above:

$$\begin{aligned} |\Gamma_{i+1}| &\geq tp - \lambda\sqrt{tp} - D_i \\ &\geq |\Gamma_i|(n - n^{\frac{3}{4}})p - \lambda\sqrt{|\Gamma_i|np} - D_i \\ &\geq |\Gamma_i|(np - n^{\frac{3}{4}}p - \lambda\frac{\sqrt{np}}{\sqrt{|\Gamma_i(x)|}}) - D_i \\ &\geq |\Gamma_i|np(1 - n^{-\frac{1}{4}} - n^{-\frac{1}{7}} - \frac{\lambda}{\sqrt{|\Gamma_i|np}}) \\ &\geq |\Gamma_i|np(1 - n^{-\frac{1}{4}} - an^{-\frac{1}{3}} - \frac{3\lambda}{\sqrt{(np)^{i-i_0}}} \end{aligned}$$

The last inequality holds by applying the induction hypothesis, bounding $|\Gamma_i(x)|$ for large enough n and we omitted x for brevity.

Once the induction step is completed, we can conclude the proof by picking an appropriate value λ.

Take $\lambda' = \sqrt{5\log n}$ combined with Lemma III.2. Since $i \leq \log n$, we have:

$$1 - o(n^{-2}) - (i - i_0)e^{-\lambda^2/2} \geq 1 - o(n^{-2}) - in^{-2.5} = 1 - o(n^{-2})$$

Note that $\Pi_{j=2}^{i-i_0}\left(1 - n^{-1/4} - n^{-1/7} - \frac{3\lambda}{\sqrt{(np)^2}}\right) \geq 1 - in^{-1/4} - in^{-1/7} - \Sigma_{j=2}^{i-i+0}\frac{3\lambda}{\sqrt{(np)^j}}$. Thus, we probability $1 - o(n^{-2})$ we have:

$$\begin{aligned} |\Gamma_i(x)| &\geq \frac{d}{3}(np)^{i-i_0}(1 - in^{-1/4} - in^{-1/7} - \Sigma_{j=2}^{i-i_0}\frac{3\lambda}{\sqrt{(np)^j}}) \\ &\geq \frac{d}{3}(np)^{i-i_0}(1 - in^{-1/4} - in^{-1/7} - \frac{3\lambda}{np}\frac{1}{1 - (np)^{-1/2}}) \\ &\geq \frac{d}{3}(np)^{i-i_0}(1 - O(\frac{1}{\sqrt{\log n}})) \\ &\geq \frac{d}{4}(np)^{i-i_0} = \frac{5}{c}(np)^{i-i_0} \end{aligned}$$

■