ANALYSIS OF BEACON TRIANGULATION IN RANDOM GRAPHS

A Thesis

by

GEETHA KAKARLAPUDI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2004

Major Subject: Computer Science

ANALYSIS OF BEACON TRIANGULATION IN RANDOM GRAPHS

A Thesis

by

GEETHA KAKARLAPUDI

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Dmitri Loguinov (Chair of Committee)	Riccardo Bettati (Member)
A. L. Narasimha Reddy (Member)	Valerie E. Taylor (Head of Department)

December 2004

Major Subject: Computer Science

ABSTRACT

Analysis of Beacon Triangulation in Random Graphs.. (December 2004)

Geetha Kakarlapudi, B.Eng., Mangalore University, Mangalore, India

Chair of Advisory Committee: Dr. Dmitri Loguinov

Our research focusses on the problem of finding nearby peers in the Internet. We focus on one particular approach, Beacon Triangulation that is widely used to solve the peer-finding problem. Beacon Triangulation is based on relative distances of nodes to some special nodes called *beacons*. The scheme gives an error when a new node that wishes to join the network has the same relative distance to two or more nodes. One of the reasons for the error is that two or more nodes have the same *distance vectors*. As a part of our research work, we derive the conditions to ensure the uniqueness of distance vectors in any network given the shortest path distribution of nodes in that network. We verify our analytical results for $G(n, p)$ graphs and the Internet. We also derive other conditions under which the error in the Beacon Triangulation scheme reduces to zero. We compare the Beacon Triangulation scheme to another well-known distance estimation scheme known as *Global Network Positioning* (GNP).

To my parents.

ACKNOWLEDGMENTS

I express my deep gratitude to Dr. Dmitri Loguinov for his invaluable guidance and suggestions throughout the course of my research. I am also thankful to Dr. Riccardo Bettati and Dr. A.L. Narasimha Reddy for serving on my committee. I am very grateful to Nagesh for his support and encouragement during the course of my research. I would like to thank Samartha G. Anekal from the Department of Chemical Engineering for his help through valuable discussions on some of the analytical results of the paper. I am also grateful to my friends Xiaoming Wang and Vivek Khariwal for their latex and formatting tips. Lastly, I thank my family for all their support and love.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

Estimating network distances is an important problem that arises in numerous distributed Internet applications. A few examples of such applications are content-delivery networks, overlay network construction and peer-to-peer applications. In content delivery networks, the performance can be greatly improved if we know the relative positions of clients and servers in the Internet and locate the nearest servers for streaming media content downloads. In peer-to-peer networks like Napster, Gnutella or Kaaza, we need to choose closest peer for faster data transfers. Hence a new node that wishes to join the network has to find an already existing node that is near to it.

We can estimate network distances based on network latency, bandwidth and packet loss rate. One can measure the network distance information using utilities like ping, traceroute etc., But it is impractical to have all nodes in the network measure distances independently because the traffic in the Internet will then increase tremendously and hence the method will not be scalable. This constraint lead to many alternate distance estimation methods that enable estimation of distances between hosts without directly measuring the distance between the hosts. Among such schemes, those that use special nodes called beacons or landmarks to determine coordinates offer an elegant solution for determining the network distances.

In landmark-based or beacon-based schemes, beacons are positioned at random locations in the network. Every beacon node measures distances to other beacons and all the hosts in the network. Any host can estimate the distance to other hosts

---

The journal model is *IEEE Transactions on Automatic Control.*

by measuring its distance to the beacon nodes. The main advantage of this method is its scalability. From now on we refer to all the beacon-based schemes with a common name *Beacon Triangulation*. Beacon Triangulation [1], [2], [3] is a simple scheme that provides a solution to this peer-finding problem. In this scheme, there are special nodes called beacons scattered all over the network. Any node can get its coordinates by measuring the distances to the beacons. A new node a will choose the nearest peer from the set of the nodes S it gets from the beacons as follows. First, the new node computes the distance vector D_a denoted by $< d_{a1}, d_{a2}, ..., d_{ak} >$, where d_{ai} is the to beacon i in the network. The dimension of the vector k is equal to the number of beacons. In Fig. 1, we illustrate the distance vectors of nodes. Then, the new node computes an distance metric M to each node i in S. The different beaconing schemes differ in the way they compute their distance metric. For example, in [1], the authors define an average distance metric, which is given by:

$$M_1 = \sum_{i=1}^{k} (d_{xi} - d_{yi})^2$$

Another scheme [2], uses a max-min distance metric given by:

$$M_2 = \frac{\min\limits_{1 \le i \le k} (|d_{xi} - d_{yi}|) + \max\limits_{1 \le i \le k} (|d_{xi} - d_{yi}|)}{2}$$

The new node a chooses the node with the lowest M as the nearest peer. In this method, a new node can choose a wrong neighbor if there are many nodes with the same M value, but which are not equally close to the new node. In this paper, we analyze how to reduce the error in choosing a wrong neighbor. We first characterize the probability of two nodes having the same M value and then derive the number of beacons needed to reduce the error in the scheme to zero.

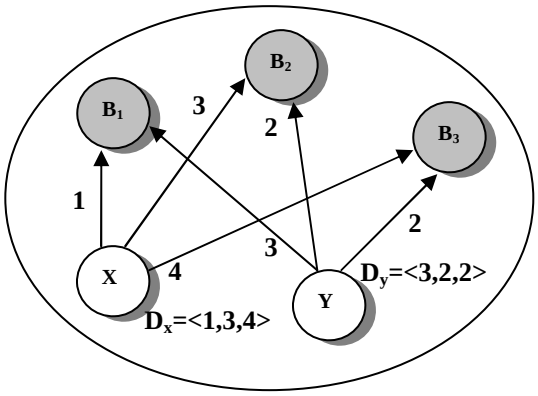


Fig. 1. Distance vectors used in Beaconing scheme.

A. Motivation and Problem Statement

A large number of distributed Internet applications can benefit from the knowledge of proximity between the hosts. For example, in content Delivery Networks (CDN), we need to replicate servers to improve latency for clients by redirecting the clients to the nearest mirrors to take advantage of CDNs. To solve the problem of network distance estimation, numerous solutions are proposed. Among such schemes, the ones that use beacons or landmark nodes provide a scalable and efficient solution to the problem and hence they are widely used.

We aim to solve one of the main problems associated with Beacon Triangulation. Consider the two scenarios illustrated in Fig. 2. In the first scenario that is shown at the top of the figure, the hosts $H_1...H_n$ use beacons $B_1...B_k$ in a star topology. We can see that the distance vectors of all the hosts are the same since they are at the same distance to the the center of the topology. But they are quite distant from each other. In the second scenario that is shown at the bottom of the figure, the hosts in the networks $N_1...N_p$ use beacons $B_1...B_k$ with a fully-meshed Internet core. Let us assume that the networks $N_1...N_p$ are i hops away from the core. We can see that the hosts X and Y have equal distance vectors and hence the same error metric D_E,

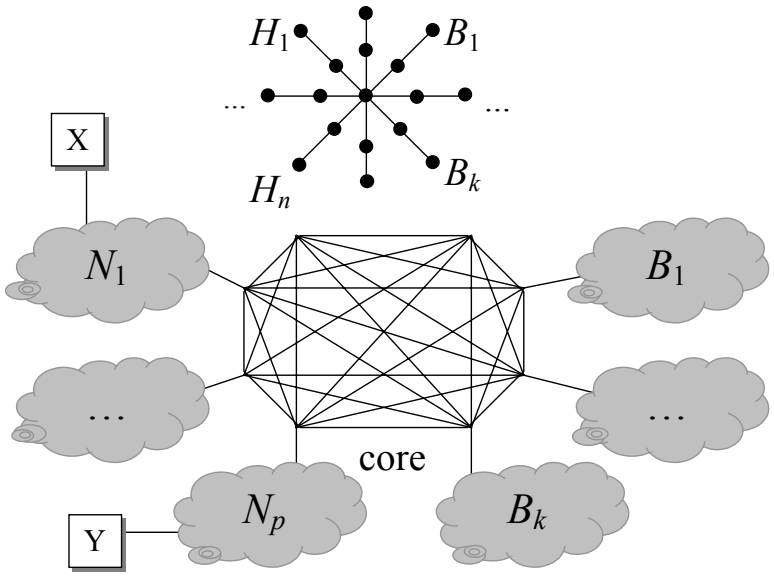


Fig. 2. Scenarios under which the beaconing schemes err. Figure source [4]

but they are $2i + 1$ hops away from each other.

In this research, we aim at solving the error that arises in Beacon Triangulation due to nodes having same distance vectors and distance metrics. We try to analytically model the underlying network and come up with the number of beacons that are necessary in order to ensure the uniqueness of distance vectors, which will improve the accuracy of the beaconing schemes. This problem can be stated as follows:

Given the shortest path distribution of nodes in a network, how many beacons are necessary in order to reduce the error in distance estimation to zero in beacon-based schemes?

Once we get an analytical model for ensuring the uniqueness of distance vectors, we test the model by simulating the following:

- Networks where the shortest path distribution is uniform.
- Networks where the shortest path distribution is non-uniform.
  - $G(n, p)$ Random Graphs

– The Internet.

We also analyze other conditions specific under which the error arises in Beacon Triangulation method other than the uniqueness of distance vectors and again verify the results with our simulations. Finally, we compare Beacon Triangulation scheme with another popular distance estimation method called Global Network Positioning.

B. Related work

In this section, we present the work related to network distance estimation. We can broadly categorize the related work in the area of network distance estimation into two types. The first type employs some kind of embedding strategies to map the Internet hosts into Euclidean space so that we can measure the network distances in a simple manner using coordinates-based approach. The second type employs some special nodes called *Tracers* or *Beacons* or *Landmarks* to measure the network distances in a scalable fashion.

Among the coordinate-based approaches, Global Network Positioning (GNP) [5] scheme is the most prominent one. In this scheme, the authors model the Internet as a Euclidean space where the hosts represent points in the space. The network distance between two hosts is obtained by evaluating a function on their coordinates. First, the coordinates of special and well distributed Landmark nodes are determined and then based on these coordinates; the positions of other hosts are measured relative to the Landmarks. This scheme provides a fast and scalable way to estimate the network distances.

Vivaldi [6] is another coordinate-based approach that assigns synthetic coordinates to hosts in a distributed fashion. Each node computes coordinates for itself without the help of any landmark nodes. The goal is to assign coordinates to hosts

in such a way that the euclidean distance between two hosts reflects the round trip latency between them. The authors find the best coordinates by simulating a network of physical springs in order to reduce the error between predicted distances and sampled latencies.

Shavitt *et al.* [7] propose a new embedding scheme called the *Big-Bang Simulation* (BBS) for estimating distances between nodes in a network. The scheme models the nodes of a network as a set of particles travelling in space under the effect of a potential force field. The scheme uses principles of Newtonian mechanics for embedding nodes in Euclidean space. The authors compare this model with other embedding schemes like GNP and also Tracer-based schemes like IDMaps [8] and show that it is more accurate with reasonable complexity than the other models.

Among the beacon-based approaches, IDMaps [8] architecture is a prominent one. It is a scalable architecture for the Internet which measures and disseminates distance information on the Internet. The main aim is to compute the distance between any given pair of IP address or any given pair of *Address Prefixes* (APs). The scheme employs tracers so that every AP is close to one or more tracers. The distance between any two APs can be calculated as the sum of distance between APs to the nearest tracer and the distance between the two tracers. The authors also provide some heuristics for selecting the number of tracers to be placed and locating the tracers for increasing the accuracy of distance measurement.

Theilmann *et al.* [9] propose Network Distance Maps for network distance estimation. The scheme employs a set of measurement servers called mServers. Each host is assigned to the closest mServer and the distance between any two hosts is estimated by the distance between their closest mServers. In order to reduce the network load and make the scheme scalable, the mServers are clustered and organized in a hierarchical manner. Each cluster elects a cluster representative such that the

distances from any host to a cluster representative should be similar to the distances to other cluster's hosts. The authors have shown that the hierarchical organization of mServers performs very well and at reasonable costs.

Guyton *et al.* [10] analyze different techniques of locating nearby replicated Internet servers. They simulated the triangulation scheme to test its effectiveness. They placed beacons randomly in the graph with 1180 clients trying to locate nearby servers. They use Hotz's [2] triangulation metric in their scheme. They find that the triangulation technique is highly portable.

Ratnasamy *et al.* [3] propose a distributed binning scheme that helps in determining network proximity information. In this scheme, the nodes partition themselves into bins such that the nodes within the same bin are closer to each other. The scheme relies on a few landmark nodes. Each node measures its distance to the landmark nodes based on the round trip time. They apply this scheme to construct overlays and for server selection. They show that the application's performance can be effectively enhanced by gaining topological information.

Internet Coordinate System (ICS) [11] is another model that facilitates the estimation of network distance between two arbitrary Internet nodes. The distances from hosts to m beacon nodes are represented as a distance vector. Using *Principal Component Analysis* (PCA), the distance data space is transformed into a cartesian coordinate system of small dimension. The transformation retains as much topological information as possible. It also minimizes the measurement overhead because the host need not make distance measurements to all the beacons.

Pias *et al.* [12] describe *Lighthouse* scheme which is a mechanism for locating the position of hosts in a scalable fashion. It differs from other beaconing techniques in the way it overcomes the issue of well-known pivot failures. It avoids a single set of pivots forming bottlenecks by having relative or multiple local-coordinate systems.

A host can determine its coordinates relative to any set of pivot nodes as long as it maintains a transition matrix. If any network topology changes occur, the hosts recalculate the coordinates from the transition matrix which is referred to as base changing.

Another scheme called *Internet Iso-bar* [13], is a scalable architecture that measures distances between nodes. It is based on the notion of clustering hosts based on a metric called correlation distance between a pair of nodes. The center of the cluster is chosen as the monitor for the cluster. The monitors measures the distance to other cluster monitors as well as the distance to all the hosts in the same cluster continuously. This information is use to predict network distances. The prediction is claimed to be simple, fairly accurate and stable with small overhead and timely information.

C. Organization of the Thesis

The thesis is divided into seven chapters. Chapter II defines the distance vectors and derives conditions for the uniqueness of distance vectors. We analyze networks with both uniform and non-uniform distribution of shortest paths and compare the performance of our model with our simulation results. In Chapter III, we discuss in detail, the performance of our model in graphs generated using the $G(n, p)$ model. We analyze how well our simulation data fits into our model. Chapter IV, we discuss BA model of network topology generation and analyze how our model works in such topologies. We also discuss the extension of our model to the Internet. We characterize the shortest path distribution of the real Internet. We compare the performance of our model with simulation results on the real Internet data. In Chapter V, we discuss the ambiguities in the distance vectors. In Chapter VI, we provide a brief overview

of *Global Network Positioning* scheme for network distance estimation. We analyze how accurate the scheme is in computing the distances when we vary the number of landmark nodes. In Chapter VII, we present the conclusion of our research work and provide some future recommendations.

CHAPTER II

UNIQUENESS OF DISTANCE VECTORS

In this chapter, we derive the conditions for ensuring the uniqueness of distance vectors. The knowledge of the shortest path distribution of nodes in a given network is critical to analyze the conditions for the uniqueness of distance vectors. We consider both uniform and non-uniform distribution of shortest paths. We derive an expression for the probability that the distance vectors of nodes have no conflicts. i.e. the probability that the distance vectors of all the nodes are unique. We denote this probability by $p_u(n)$. For our discussion, we denote the number of nodes by n and the number of beacons by k. The expression for $p_u(n)$ depends on the shortest path distribution to the beacons. We analyze the conditions under which the value of $p_u(n)$ tends to one for the different shortest-path distributions.

Let us assume that there are n nodes in a given network and k beacons are placed randomly in the network. The distance vector of node i is given by $< d_{i1}, d_{i2}, ..., d_{ik} >$ where, d_{i1} is the distance of node i from the 1^{st} beacon, and d_{ik} is the distance of node i from the k^{th} beacon. From now on, we assume that the maximum distance of any node to any beacon is d i.e. in the distance vector $< d_{i1}, d_{i2}, ..., d_{ik} >$, the maximum value of any d_{i1} is the diameter d. Hence the total number of unique distance vectors id given by $d \times d \times ...k$ times $= d^k$. Out of the total d^k unique combinations, we have only n distance vectors (one for each node). The probability of any of d^k distance vectors occurring in one of the n samples is not the same. This is because the distribution of distances from each node to a beacon is not uniform for a given network.

A. Uniform Distribution of Shortest Paths

In this section, we assume that the distribution of shortest paths to each beacon is uniform and the shortest paths to all the beacons are $i.i.d$ random variables.

1. Analytical Model

We will now obtain an expression for the probability of no conflicts in the distance vectors i.e. $p_u(n)$.

Lemma 1. *If the shortest paths to the beacons are uniformly distributed in the interval* $[1, d]$*, the probability that there are no conflicts due to equal distance vectors is given by:*

$$p_u(n) = \begin{cases} \prod_{i=1}^{n} \left(1 - \frac{i-1}{d^k}\right) & n < d^k \\ 0 & n \geq d^k \end{cases} \tag{2.1}$$

Proof. Let $D_i = (d_{i1}, \ldots, d_{ik})$ be the distance vector of a node i. Since we assume that the shortest paths to all beacons are $i.i.d$, diameter d is the same for all beacons. There are d^k unique distance vectors out of which we choose n vectors. We have to evaluate the probability of choosing n vectors such that all the n vectors are distinct. The probability that the first vector is unique is given by d^k/d^k. The probability that the second vector chosen is distinct from the first is $(d^k - 1)/d^k$. Similarly probability that i^{th} distance vector is distinct from the initial $i-1$ vectors given that the initial $i-1$ vectors are distinct is given by $(d^k - i + 1)/d^k$. Thus the probability that all the n vectors are distinct which is $p_u(n)$ is given by:

$$p_u(n) \quad = \quad \frac{d^k}{d^k} \times \frac{d^k - 1}{d^k} \times \ldots \times \frac{d^k - n + 1}{d^k}. \tag{2.2}$$

Simplifying this expression, we get (2.1). □

We next show an asymptotic expansion of model (2.1).

Lemma 2. *For the uniform distribution of shortest paths in* $[1, d]$ *and* $k = 2\log_d n - \log_d 2 + w$*, the probability of obtaining a non-conflicting set of distance vectors is asymptotically:*

$$p_u(n) \approx e^{-d^{-w}}. \tag{2.3}$$

Proof. First notice that $d^k \approx n^2$ is much larger than n. Then, using Taylor expansion $(1 - x) \approx e^{-x}$ and the fact that $n \ll d^k$, we get from (2.1):

$$p_u(n) \approx \prod_{i=1}^{n} e^{-(i-1)/d^k} = e^{-n(n-1)/2d^k}. \tag{2.4}$$

Select the number of beacons k to be $2\log_d n - \log_d 2 + w$, where w is possibly a function of n. From (2.4), observe that p_u can be now written as:

$$p_u(n) \approx \exp\left\{-\frac{n(n-1)}{n^2 d^w}\right\}, \tag{2.5}$$

which immediately leads to (2.3). □

We next obtain the number of beacons necessary to guarantee a non-conflicting set $\{D_i\}$.

Corollary 1. *Assuming that the diameter of the graph* $d = \Omega(\ln n)$ *and the number of beacons as in Lemma 2, the uniqueness of* $\{D_i\}$ *is guaranteed in almost every graph as long as* $w > 0$*. At the same time, almost no graph will have a unique set* $\{D_i\}$ *if* $w < 0$*:*

$$\lim_{n\to\infty} p_u(n) = \begin{cases} 1 & w > 0 \\ 0 & w < 0 \end{cases}. \tag{2.6}$$

From our model in (2.1), we compute the number of beacons needed for p_u to be equal to 1 for different values of n, assuming that d is $\lfloor \ln n \rfloor$. The Table I shows the

Table I. Beacons needed for different n to avoid conflicts

Nodes n	Beacons k
10^2	11
10^3	11
10^4	12
10^5	13
10^6	14

values of k needed to get $p_u(n)$ to 1 for different n from our simulations. We observe that we can reduce conflicts even for large number of nodes with manageable number of beacons. We verify the results from our simulations.

2. Simulations

For the purpose of our simulations, we generate uniformly distributed shortest paths to all the beacons using random number generator [14]. We form distance vectors with the generated shortest paths to the beacons. We calculate $p_u(n)$ as the percentage of number of times there are no equal distance vectors in 1000 runs of our simulation.

3. Discussion

In Fig. 3, we plot simulations of model (1) for $n = 10000$ and $d = \lfloor \ln n \rfloor$. From our Lemma 2 and Corollary 1, the number of beacons k should be greater than 9 for $p_u(n)$ to go to 1. For $w = 1$, k should be equal to 10 for $p_u(n)$ to be equal to 1. We

plot the variation of $p_u(n)$ as we increase k. We compare our simulation data with our model. We can see that the simulation data fits our model well.

In Fig. 4, we plot model (2.6) for positive and negative values of w. The value of d changes as $\lfloor \ln n \rfloor$. In Fig. 4(a), we plot $p_u(n)$ when w is positive and equal to 1. We observe that the curve eventually converges to 1. In Fig. 4(b), we plot $p_u(n)$ when $w < 0$, specifically when $w = -1$. We observe that as inferred by Corollary 1, the curve eventually converges to 0.

B. Non-uniform Distribution of Shortest Paths

We consider the case where the distribution of shortest paths is not uniform. Again, we assume that the shortest paths to all the beacons are $i.i.d.$ with d as the diameter.

1. Analytical Model

Let us assume that the probability of the shortest path hop count being i is given by p_i. The probability that two distance vectors are equal is given by $p_c = \left(\sum_{i=1}^{d} p_i^2\right)^k$. The probability that there are no conflicts is given by:

$$\begin{aligned} p_u(n) &= \prod_{k=1}^{n} (1 - p_c)^{n-k} \\ &= (1 - p_c)^{\frac{n(n-1)}{2}} \\ &= \left(1 - \left(\sum_{i=1}^{d} p_i^2\right)^k\right)^{\frac{n(n-1)}{2}} \end{aligned} \tag{2.7}$$

2. Simulations

In order to verify the model in equation (2.7), we simulate a network with non-uniform shortest path distribution. We construct a $G(n, p)$ random graph with n nodes for

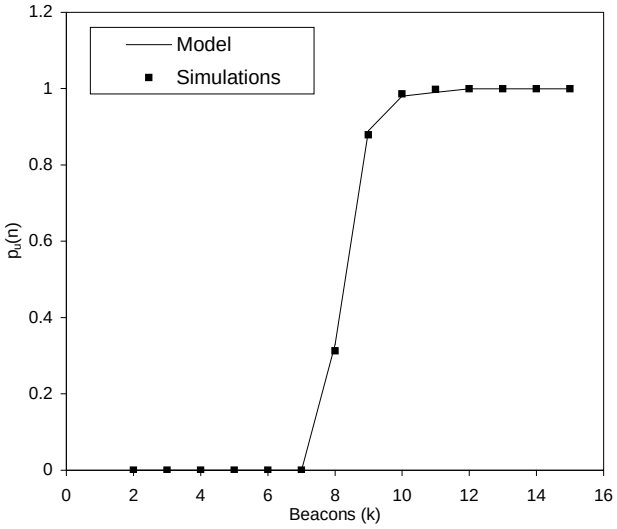
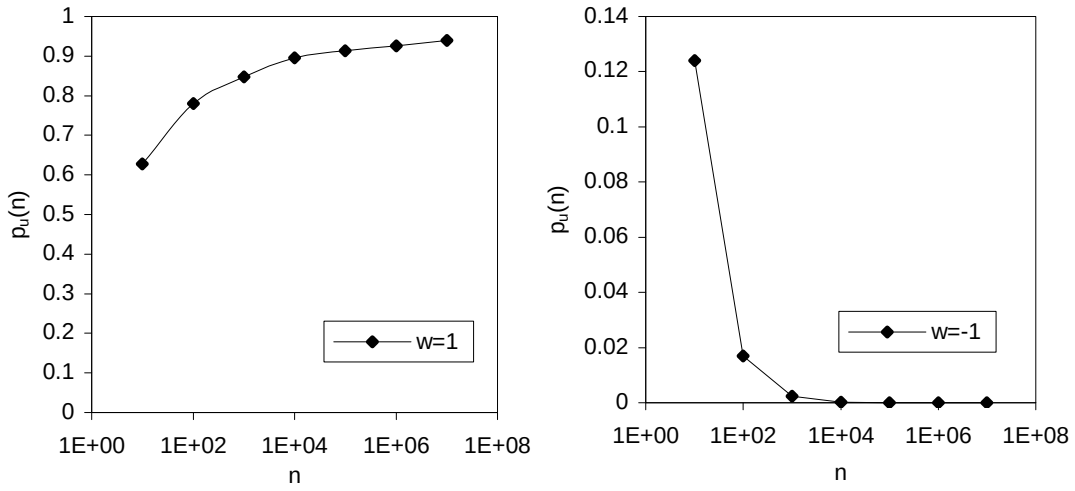
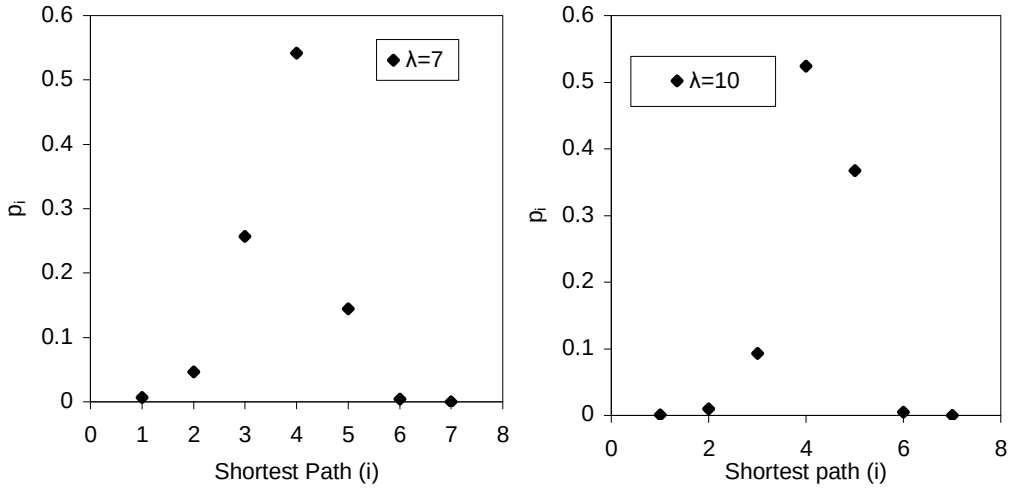


Fig. 3. Probability $p_u(n)$ versus the number of beacons k for $n = 10000$ from model (2.1) with $d = \lfloor \ln n \rfloor$.



(a) (b)

Fig. 4. Percentage of no conflicts $p_u(n)$ versus n for model (2.6): (a) Percentage of no conflicts $p_u(n)$ versus n for $w > 0$ $(w = 1)$; (b) Percentage of conflicts $p_u(n)$ versus n for $w < 0$ $(w = -1)$.



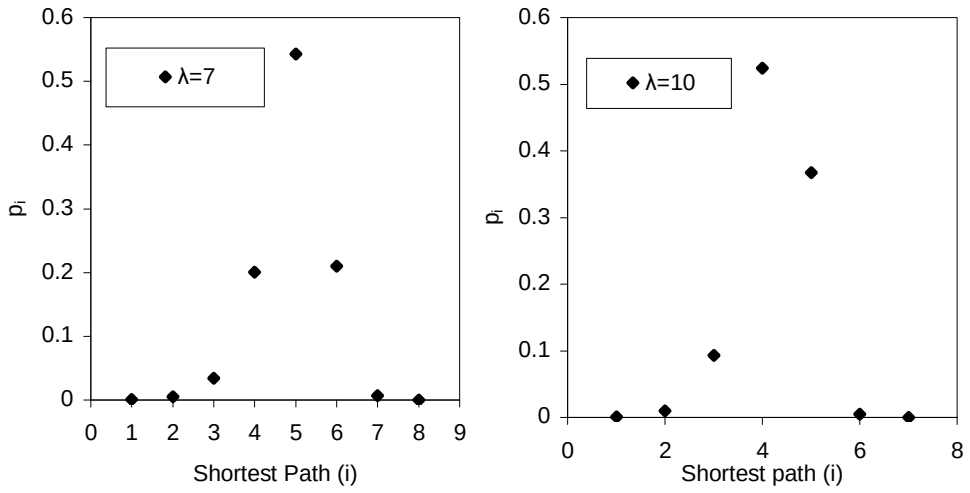
(a) (b)

Fig. 5. Shortest path distribution for a $G(n,p)$ random graph: (a) $n = 1000, \lambda = 7$; (b) $n = 1000, \lambda = 10$.

the purpose of our simulations. Each edge is included in the graph with independent probability p between any two nodes. After we construct the graph, we compute all pair shortest paths using Djikstra's algorithm to get the shortest path distribution of the network. We fix the graph once and randomly vary beacon positions each time to check for a conflict in the distance vectors. We repeat this over thousands of runs and calculate the percentage of no conflicts in the distance vectors i.e. $p_u(n)$.

3. Discussion

In Fig. 5 and Fig. 6, we plot the shortest path distribution in a $G(n,p)$ graph with different values of n and p. The average degree of the graph λ is equal to the product np. We observe that the distribution is gaussian-like. Using the distribution of shortest paths from simulations, we obtain p_i values and we plug these values into the model in (2.7) to calculate the value of $p_u(n)$. We compare the value of $p_u(n)$ that

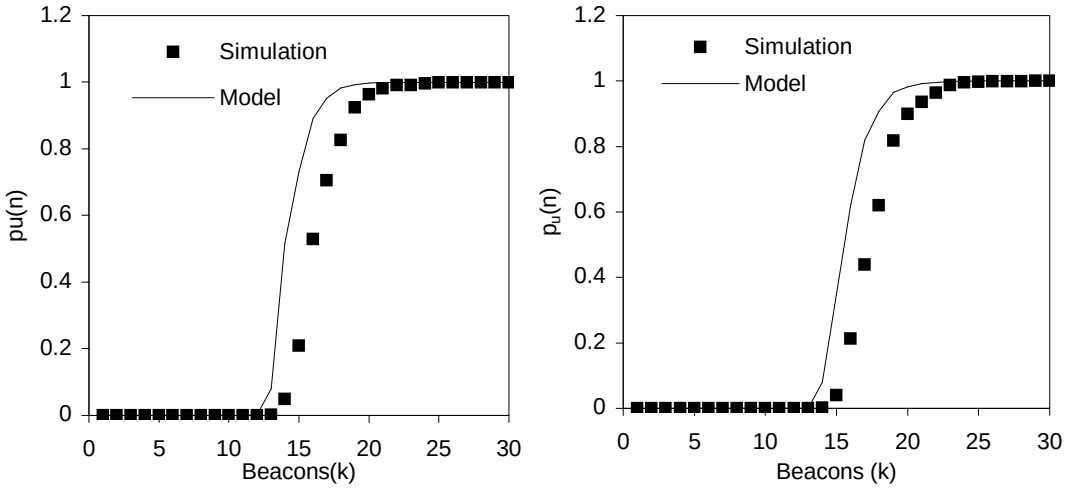


(a) (b)

Fig. 6. Shortest path distribution for a $G(n,p)$ random graph: (a) $n = 10000, \lambda = 7$; (b) $n = 10000, \lambda = 10$.

we get from the model with the one from simulations. We compare the two values of $p_u(n)$ from model and simulations for $n = 1000$ and $n = 10000$ for two different values of p.

We plot the simulation results in Fig. 7 and Fig. 8 for different values of n and p. In Fig. 7(a), we plot the value of $p_u(n)$ versus k for $n = 1000$ and $p = 0.007$ from our model and simulations. We observe that the values of $p_u(n)$ from simulation results are a bit off from the model at the transition of $p_u(n)$ from 0 to 1. But the values of k at which there is a transition of $p_u(n)$ from 0 to 1 from our model and simulations are almost the same. From Fig. 7(a), we observe that from our model, we need 26 beacons to ensure that the value of $p_u(n)$ is 1 and from our simulations, we need 28 beacons which is quite close to that from our model. In Fig. 7(b), we plot the value of $p_u(n)$ versus k for $n = 1000$ and $p = 0.01$ from our model and simulations. The results are very similar to that seen in Fig. 7(a). From the plot, we can see that we

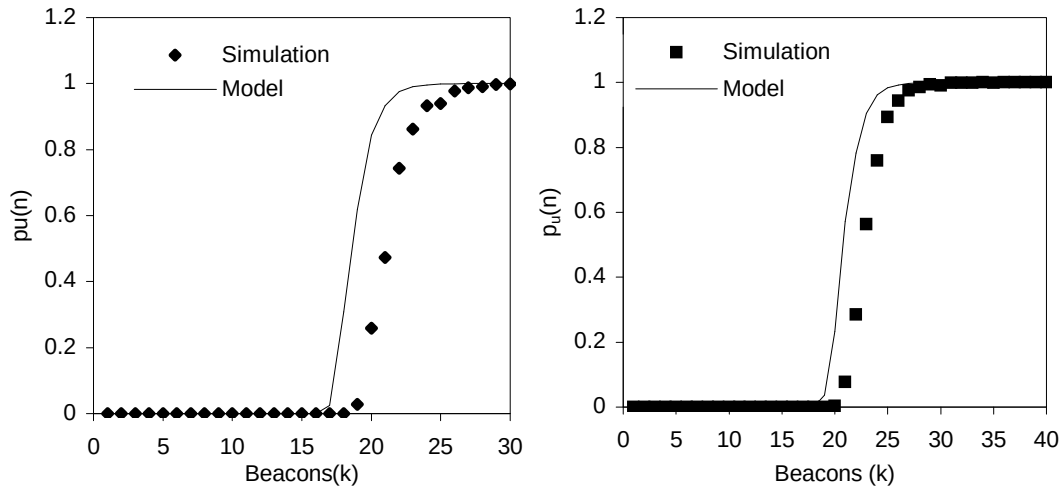


(a) (b)

Fig. 7. Percentage of conflicts $p_u(n)$ vs number of beacons k: (a) $n = 1000, \lambda = 7$; (b) $n = 1000, \lambda = 10$.

need 29 beacons to ensure that the value of $p_u(n)$ is 1 and from our simulations, we need 30 beacons which is very close to that from our model. From Fig. 7, We also observe that as the average degree of the graph λ decreases, we need lesser number of beacons to ensure the uniqueness of the distance vectors.

Now, let us analyze the results for $n = 10000$. In Fig. 8, we plot the value of $p_u(n)$ versus k for $n = 10000$ and $p = 0.001$ from our model and simulations. The plot is very similar to Fig. 7 in its characteristics. From Fig. 8(a), We observe that from our model, we need 30 beacons to ensure that the value of $p_u(n)$ is 1 and from our simulations, we need 31 beacons which is very close to that from our model. As we increase the value of p to 0.001, we can see from the Fig. 8(b), we need 36 beacons to ensure that the value of $p_u(n)$ is 1 and from our simulations, we need 37 beacons which is very close to that from our model. Again, similar to our observations in Fig. 8, we notice that as the average degree of the graph λ decreases, we need lesser



(a) (b)

Fig. 8. Percentage of conflicts $p_u(n)$ vs number of beacons k: (a) $n = 10000, \lambda = 7$; (b) $n = 10000, \lambda = 10$.

number of beacons to ensure the uniqueness of the distance vectors.

C. Chapter Summary

We analyze the conditions for uniqueness of distance vectors in the case of both uniform and non-uniform distributions of shortest paths of nodes in any given network. As the number of nodes increase, we need more beacons to ensure the uniqueness of distance vectors. In the case of non-uniform shortest path distribution, we also observe that as the average degree of the graph increases, we need more beacons to ensure the uniqueness of distance vectors.

CHAPTER III

$G(N, P)$ GRAPH

In this chapter, we focus on graphs generated using $G(n, p)$ model. We get an expression for the shortest path distribution in $G(n, p)$ graphs. We use this distribution to derive the conditions for the uniqueness of distance vectors. Consider a random graph generated using $G(n, p)$ model with n nodes and probability p. Let us recall that a $G(n, p)$ random graph is one which has n nodes and each edge is included in the graph with independent probability p between any two vertices. The average degree of the graph λ is given by the product np.

A. Analytical Model

In [15], the authors derive an expression relating to the probability of hopcount in $G(n, p)$, which is given by $P(h_n > i) = e^{-c\frac{\lambda^i}{n}}, c > 0$. Using this result on hopcount, we can obtain the hopcount probability distribution as

$$p_i = \left(e^{-c\frac{\lambda^{i-1}}{n}} - e^{-c\frac{\lambda^i}{n}}\right), c > 0. \tag{3.1}$$

.

From [2.7], the probability of no conflicts is given by

$$p_u(n) = \left(1 - \left(\sum_{i=1}^{d} p_i\right)^k\right)^{\frac{n(n-1)}{2}}. \tag{3.2}$$

Substituting the hopcount distribution for $G(n, p)$ from (3.1) in (3.2), we obtain

the probability of no conflicts for a $G(n,p)$ graph as

$$p_u(n) = \left(1 - \left(\sum_{i=1}^{d} \left(e^{-c\frac{\lambda^{i-1}}{n}} - e^{-c\frac{\lambda^{i}}{n}}\right)^2\right)^k\right)^{\frac{n(n-1)}{2}} \tag{3.3}$$

Let us now analyze the asymptotic behavior of $p_u(n)$, $\lim_{n\to\infty} p_u(n)$ which is given by

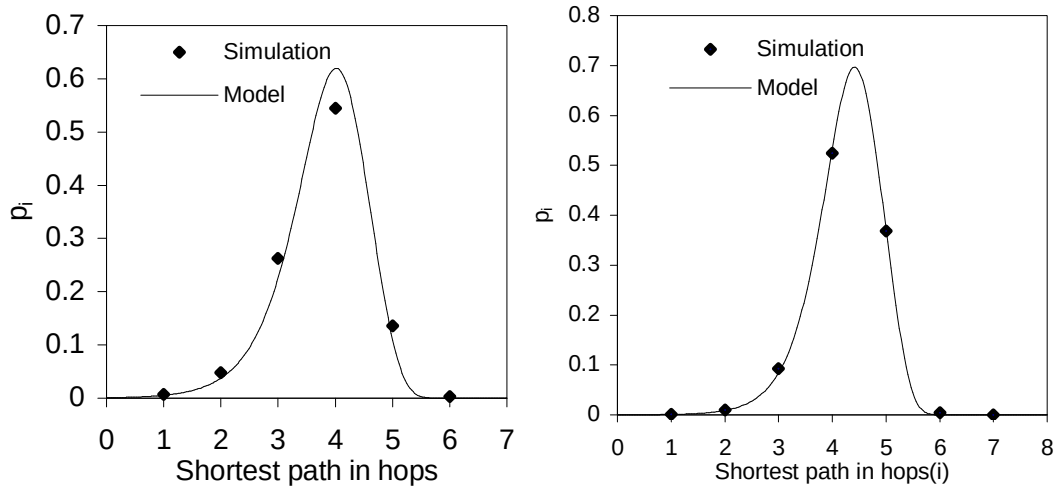
$$\lim_{n\to\infty} \left(1 - \left(\sum_{i=1}^{d} \left(e^{-c\frac{\lambda^{i-1}}{n}} - e^{-c\frac{\lambda^{i}}{n}}\right)^2\right)^k\right)^{\frac{n(n-1)}{2}} \tag{3.4}$$

Since the model in (3.4) is very cumbersome to simplify, we analyze the behavior of the model by plotting the value of $p_u(n)$ for increasing values of n and for different values of k and for the shortest path distribution that we have from (3.1). For the purpose of our calculations, we assume the diameter of the graph as $\log n$ [16]. We simulate a $G(n,p)$ network and compare the results from our model and simulations in the network in the following subsections.

B. Simulations

In this section, we first describe our simulation setup for $G(n,p)$ graphs. In order to build an undirected random $G(n,p)$ graph of n nodes, we choose a probability p of having an edge between two nodes. We represent the graph by its adjacency matrix. To place an edge between two nodes, we generate a random number between 0 and 1. If the random number is less than or equal to p, we join the nodes by placing a 1 in the adjacency matrix at the corresponding location. If the random number is greater than p, we place a 0 in the adjacency matrix. Thus, we construct the adjacency matrix for the graph. The nodes are numbered from $0..n-1$.

In order to check the connectivity of the graph, We do a Breadth First Search



(a) (b)

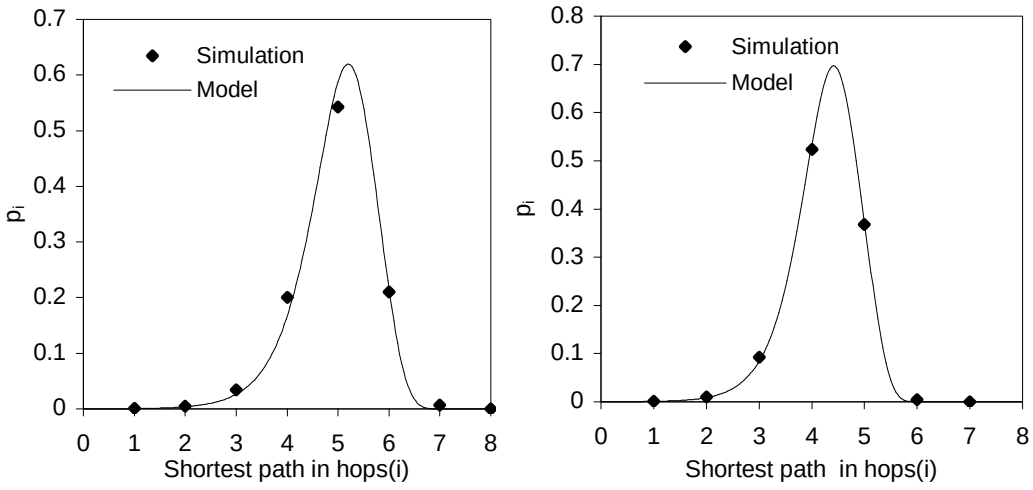
Fig. 9. Shortest path distribution fit: (a) $n = 1000$, $\lambda = 7$; (b) $n = 1000$, $\lambda = 10$.

(BFS) on the graph starting from the node 0. We mark all the nodes that are visited by the BFS algorithm. If the total number of nodes that are marked is greater than fifty percent of the total nodes n, we proceed further. We fix the number of beacons k. We choose the beacon positions in the graph randomly and we also make sure that they lie in the connected component of the graph that we consider for our simulations.

For the purpose of our simulations, we consider two values of n, 1000 and 10000 and two values of λ, 7 and 10. We choose these values to observe how the model behaves for different values of n and λ. In order to obtain the value of the constant c, we fit a curve with shortest path distribution expression in (3.1) to the distribution of shortest paths from our simulations.

C. Discussion

We plot the distribution of shortest paths from our simulations and from (3.1) in Fig. 9 and Fig. 10. We observe that the shortest path data from our simulations fits well

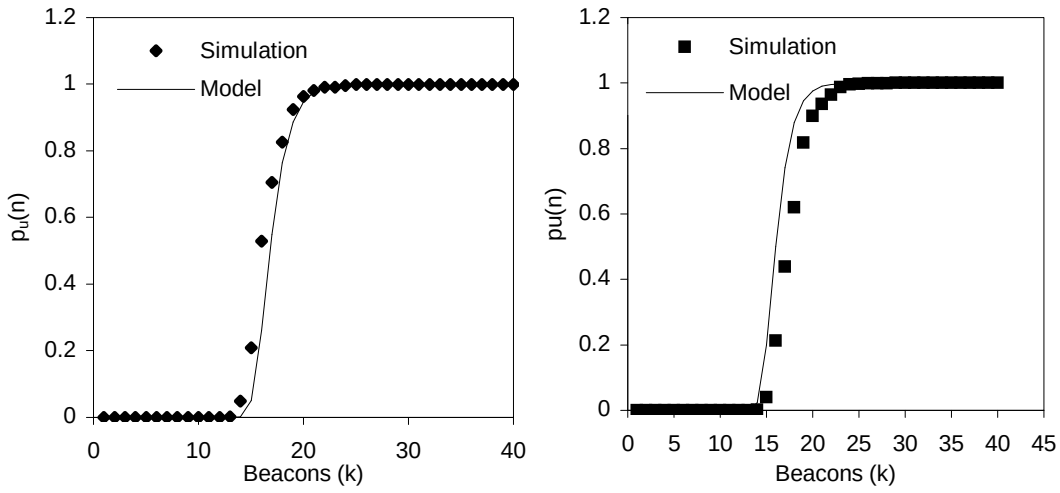


(a) (b)

Fig. 10. Shortest path distribution fit: (a) $n = 10000$, $\lambda = 7$; (b) $n = 10000$, $\lambda = 10$.

to the model in (3.1). From the shortest path distribution fits from Fig. 9 and Fig. 10, we obtain the values of c for different combinations of n and λ values. We use the value of c thus obtained, in calculating the values of $p_u(n)$.

We plot the values of $p_u(n)$ from our model in (3.3) and simulations for different values of n and λ in Fig. 11 and Fig. 12. In Fig. 11(a), we plot $p_u(n)$ versus k from our model and simulations for $n = 1000$ and $p = 0.007$. We can clearly see that our simulation data is off from our model at the transition of $p_u(n)$ from 0 to 1. But the transition of $p_u(n)$ from 0 to 1 in the plot for our model and simulation data occurs for almost the same values of k. To observe this, let us fix a small threshold $\epsilon = 0.01$ and look at the number of beacons needed from our model such that $p_u(n) \geq 1 - \epsilon$. For $n = 1000$ and $\lambda = 7$, we need 23 beacons to ensure that $p_u(n) \geq 1 - \epsilon$, i.e, to ensure that $p_u(n) \geq 0.99$. Next, we look at the number of beacons needed from our simulation data such that $p_u(n) \geq 0.99$ for the same n and λ values. we need 23 beacons to ensure that $p_u(n) \geq 0.99$ which is the same as what we have from our



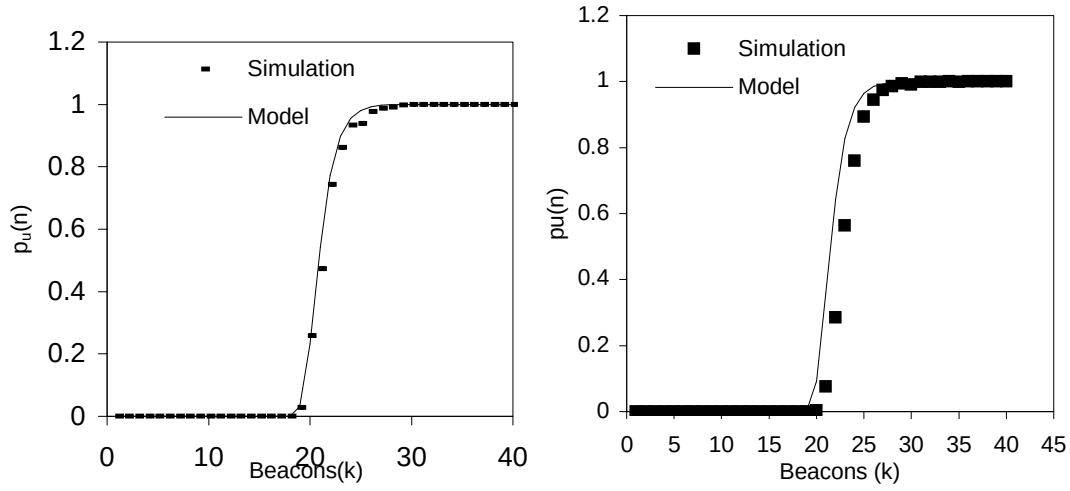
(a) (b)

Fig. 11. Percentage of conflicts: (a) $n = 1000$, $\lambda = 7$; (b) $n = 1000$, $\lambda = 10$.

model. In Fig. 11(b), we plot $p_u(n)$ versus k from our model and simulations for $n = 1000$ and $p = 0.01$.We observe the same characteristics in the figure as in the case of the Fig. 11(a)

We analyze the same for $n = 10000$ and for two different values of λ for the same threshold value of 0.01. We can see from the Fig. 12(a) that we need 27 beacons to ensure that $p_u(n) \geq 0.99$. Next, we look at the number of beacons needed from our simulation data such that $p_u(n) \geq 0.99$ for the same n and λ values. we need 29 beacons to ensure that $p_u(n) \geq 0.99$ which is fairly close to what we have from our model. In Fig. 12(b), we plot $p_u(n)$ versus k from our model and simulations for $n = 10000$ and $\lambda = 10$.We observe the same characteristics in the figure as in the case of the Fig. 12(a).

In Table. II, we compare the number of beacons needed for different n to make $p_u(n)$ equal to one for $\lambda = 10$ from our model. We can see that simulation data is just a few beacons off from the model. In Table. III, we compare the number of beacons



(a) (b)

Fig. 12. Percentage of conflicts: (a) $n = 10000$, $\lambda = 7$; (b) $n = 10000$, $\lambda = 10$.

needed for different n to make $p_u(n)$ equal to one for $\lambda = 10$ from simulation data for a $G(n, p)$ graph and for uniform shortest path distribution case. We can observe that in the case of uniform shortest path distribution, we need very few beacons compared to the non-uniform shortest path distribution in $G(n, p)$ graph.

Table II. Beacons needed for different n to make $p_u(n)$ equal to one for $\lambda = 10$ from our model and simulation

n	Model	Simulation
1000	22	24
10000	27	29

From our analysis, we observe that our model in (3.3) can be used to get an estimate on the number of beacons needed to ensure that no two distance vectors are same with fairly reasonable accuracy. Table. IV gives the number of beacons k

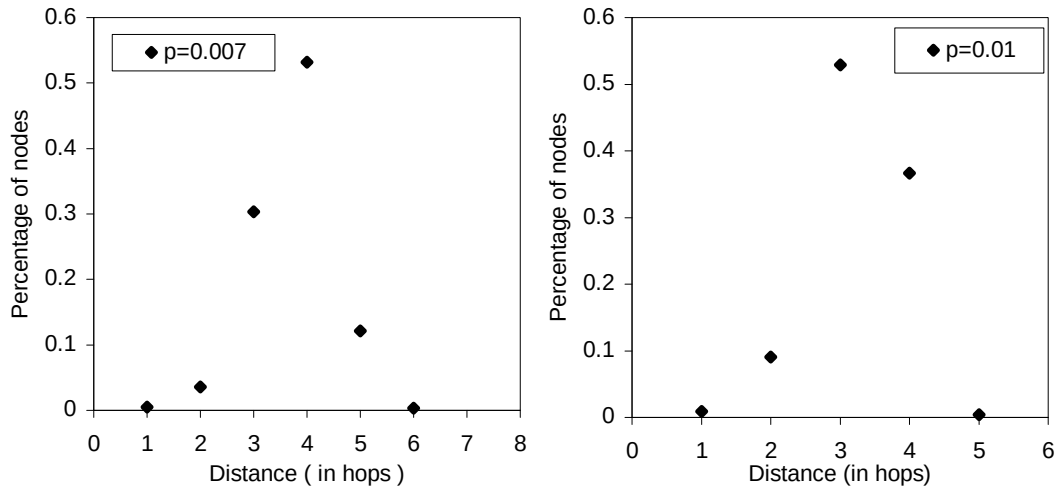
Table III. Beacons needed for different n to make $p_u(n)$ equal to one for $\lambda = 10$ from our simulation data for $G(n,p)$ graph and uniform distribution case

n	$k(G(n,p))$	kUniform
10^2	19	11
10^3	25	11
10^4	30	12
10^5	36	13
10^6	14	14

needed to make $p_u(n)$ equal to 1 for different n and λ values. For each value of n, we choose a value for p such that the average degree λ is equal to 5 and 10. From the Table. IV, we can clearly see that, by fixing beacons a fairly few number of beacons even in large networks, we can increase the accuracy of the beaconing schemes by ensuring that no two distance vectors are equal.

Table IV. Beacons needed for different n to make $p_u(n)$ equal to one for $\lambda = 5$ and $\lambda = 10$ from our model

n	k for $\lambda = 5$	k for $\lambda = 10$
10^2	16	19
10^3	22	25
10^4	25	30
10^5	29	36
10^6	35	41
10^7	37	45

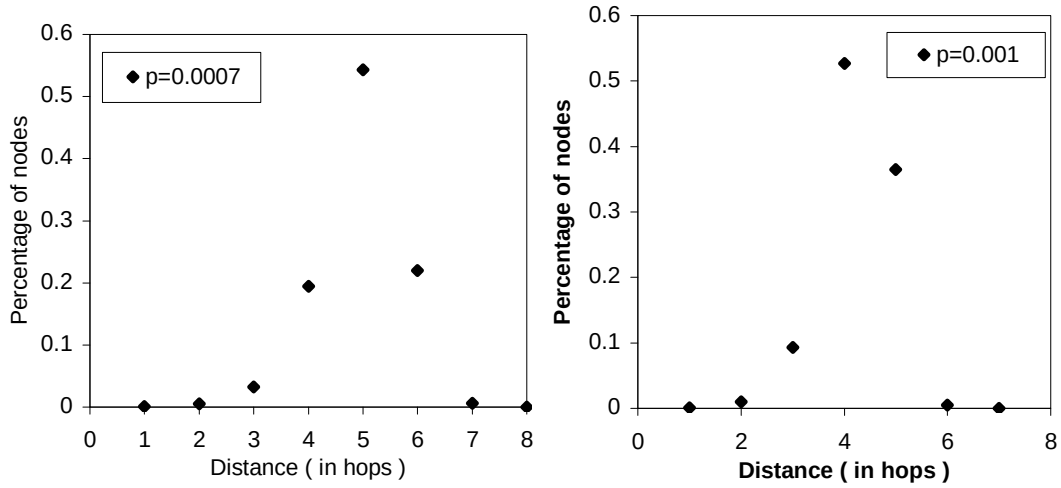


(a) (b)

Fig. 13. Shortest path distribution of nodes with equal distance vectors: (a) $n = 1000$, $\lambda = 7$; (b) $n = 1000$, $\lambda = 10$.

Now, we analyze how the nodes with the same distance vectors are related. We analyze the distribution of shortest paths between the nodes with the same distance vectors. By observing any pattern in the shortest path between two nodes with the same distance vectors will give us an idea of how inaccurate or accurate can a beaconing scheme get if two or more nodes have the same distance vectors.

We plot the shortest path distribution of nodes with equal distance vectors data from our simulations for different values of n and λ in Fig. 13 and Fig. 14. In Fig. 13(a), for $n = 1000$ and $\lambda = 7$, we notice that the diameter of the graph is 6. We also see that around 30% of the nodes with the same distance vectors are 3 hops away and around 53% of the nodes with the same distance vectors are 5 hops away from each other. There are just 0.4% of the nodes that are immediate neighbors that have the same distance vectors. In Fig. 13(b), for $n = 1000$ and $\lambda = 10$, we notice that the diameter of the graph is 7. We also see that around 53% of the nodes with



(a) (b)

Fig. 14. Shortest path distribution of nodes with equal distance vectors: (a) $n = 10000$, $\lambda = 7$; (b) $n = 10000$, $\lambda = 10$.

the same distance vectors are 3 hops away and around 36% of the nodes with the same distance vectors are 4 hops away. There are just 0.04% of the nodes that are immediate neighbors that have the same distance vectors.

We observe similar trend for $n = 10000$ in Fig. 14. In Fig. 14(a), for $n = 10000$ and $\lambda = 7$, we notice that the diameter of the graph is 8. We also see that around 54% of the nodes with the same distance vectors are 5 hops away and around 22% of the nodes with the same distance vectors are 6 hops away. A very small percentage of nodes (0.05%) with equal distance vectors are immediate neighbors. n Fig. 14(b), for $n = 10000$ and $\lambda = 10$, we notice that the diameter of the graph is 7. We also see that around 52% of the nodes with the same distance vectors are 4 hops away and around 36% of the nodes with the same distance vectors are 5 hops away. A very small percentage of nodes (0.09%) with equal distance vectors are immediate neighbors.

Hence we can see from our simulations that a major percentage of the nodes with the same distance vectors are not very close to each other and hence they can contribute to significant error in the beaconing schemes if we do not choose the right closest peer node.

D. Chapter Summary

We deal with $G(n, p)$ graphs in detail. We derive expressions for the shortest path distribution in a $G(n, p)$ graph and the probability of no conflicts in distance vectors. We compare our model results with our simulations. We also observe a pattern among the nodes with the same distance vectors which leads us to the conclusion that uniqueness of distance vectors is of key importance to measure network distances accurately.

CHAPTER IV

THE INTERNET

In this chapter, we measure the effectiveness of our model in Internet-like topologies and the Internet. First, we generate a network topology using BA model and verify how our model works with the BA graphs. Next, we extend our discussion to the real Internet. We study the shortest path distribution in the Internet and verify the distribution with real Internet data. We also verify if our model of $p_u(n)$ for non-uniform distribution of shortest paths can be extended to the Internet.

A. Internet-like Topologies

There are many topology generators that generate Internet-like graphs by imitating the properties of the Internet. Some of the topology generators like [17] generate random topologies. Some of them generate topologies to reflect hierarchical properties of the Internet [18, 19]. A few of the topology generators produce graphs with similar degree-related properties as the Internet [20, 21, 22]. We consider the BA model [23] which produces a graph with power-law degree distribution. We extend our experiments to a BA graph and study its shortest path distribution. We apply the shortest path distribution from our simulations to our model to obtain the probability of uniqueness in distance vectors. We compare our model with simulations to test the accuracy of the model in Internet-like topologies.

1. Overview of the BA Model

BA model produces graphs with power-law degree distribution. In the model, the networks grow incrementally and as the network grows, new nodes are attached preferentially attached to the nodes with higher degree. Due to such incremental growth

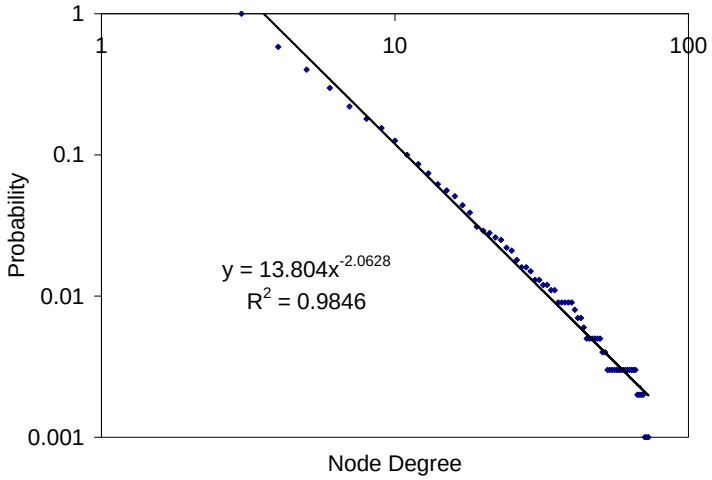


Fig. 15. CCDF of node degree distribution of a BA graph for $n = 100$, $m = 3$ and $m_0 = 2$.

and preferential attachment, the BA model is shown to generate networks with power-law degree distribution.

In BA model, we start with a small number of nodes m_0 that are connected arbitrarily. At each step, we do the following:

- Add a new node with $m \leq m_0$ edges that are connected to already existing nodes in the graph.
- Each of the m neighbors is picked up randomly with a probability proportional to its degree. For example, a new node picks a node i with degree k_i as its neighbor with probability $p(k_i) = k_i / \sum_j k_j$. In this manner, a new node chooses m neighbors.

Fig. 15 shows the plot of the Complementary Cumulative Distribution Function (CCDF, which is $1 - CDF$) of the degree distribution of nodes in a BA graph of 100 nodes (log-log scale).

2. Simulations and Discussion

We simulate a BA graph and compute the shortest path distribution. Fig. 16 shows the shortest path distribution in a BA graph for $n = 100$ and $n = 1000$. For the purpose of our simulations, we fix the value of m at 3 and m_0 at 2. We observe that the shortest path distribution in a BA graph is very similar to the one in $G(n, p)$ random graph.

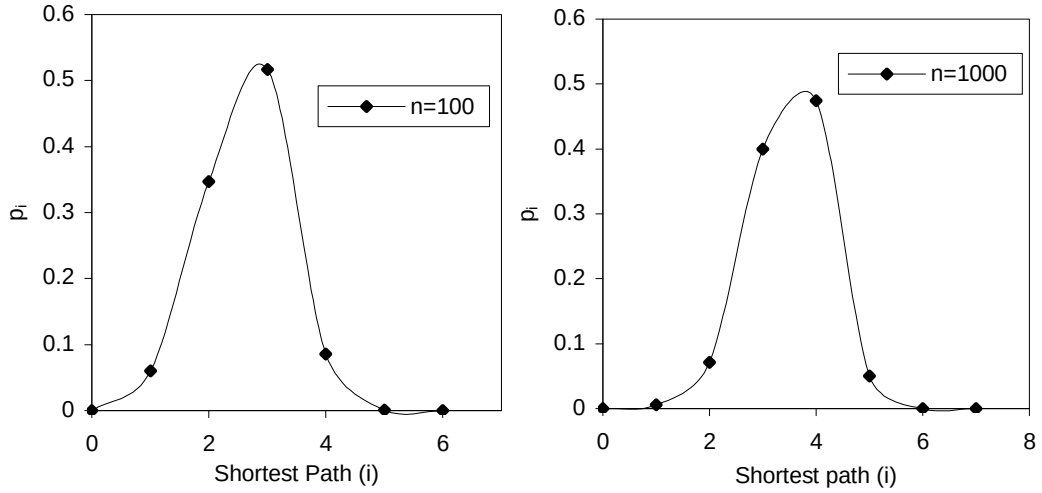
We plot the values of $p_u(n)$ from our model in (3.3) and simulations for different values of n in Fig. 16(a) and Fig. 16(b). We can see from the plot that the simulation data is way off from our model. We observe a similar trend in the plots that we observed with our real data from the Internet. This shows that our model is not accurate enough to model the Internet data. This is due the fact that we do not take into account the dependencies in distance vectors in our model. These dependencies may be a prominent factor in the Internet and Internet-like topologies. Modelling such dependencies is an interesting direction to explore in future.

B. The Real Internet

In this section, we discuss the effectiveness of our model in the real Internet. First, we study the shortest path distribution in the Internet and verify the distribution with real Internet data. Then, we verify if our model of $p_u(n)$ for non-uniform distribution of shortest paths can be extended to the Internet.

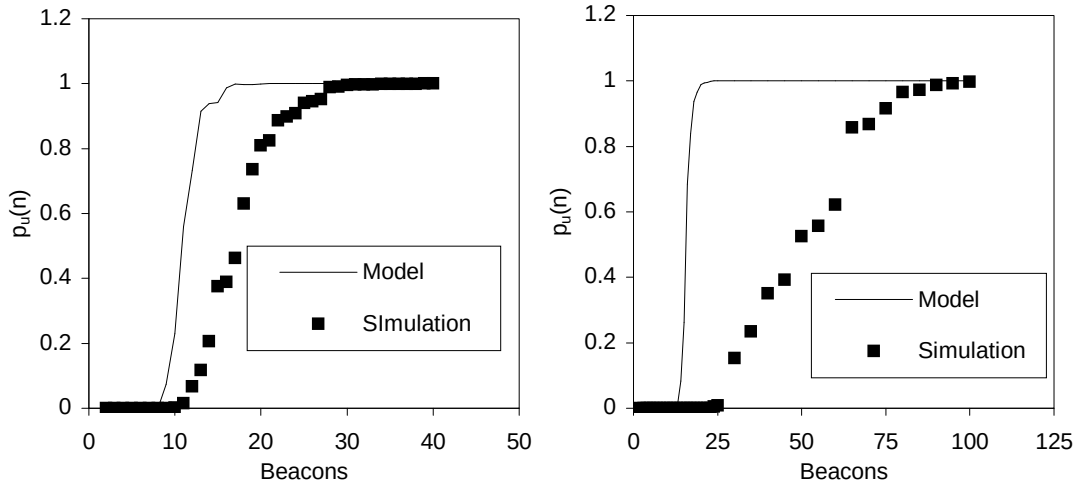
1. Shortest Path Distribution

A lot of work has been done to model the distribution of shortest paths in the Internet. Dorogovtsev et al. [24] obtain the expression for shortest path length distribution. The authors describe the DGM model which analytically produces a Gaussian distrib-



(a) (b)

Fig. 16. Shortest path distribution in a BA graph; (a) $n = 100$; (b) $n = 1000$.



(a) (b)

Fig. 17. Probability $p_u(n)$ versus number of beacons k in a BA graph. (a) $n = 100$; (b) $n = 1000$.

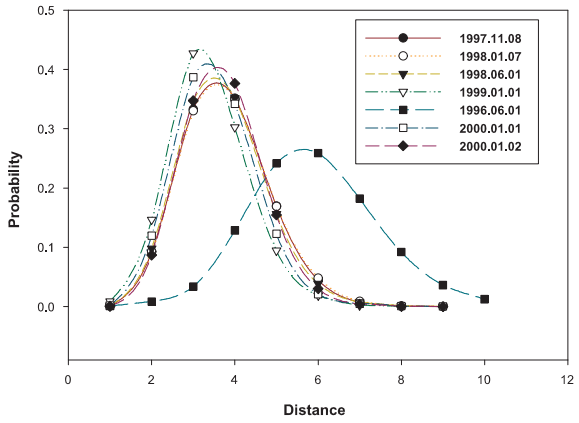


Fig. 18. Shortest path distribution in the Internet.

ution similar to the distance distribution seen in the real Internet. For large networks, the distribution tends to a Gaussian of width $\sqrt{\ln n}$ centered at $\ln n$. They show that at large t, the distribution takes the form

$$P(\ell, t) = \frac{1}{\sqrt{2\pi\,(2^2/3^3)\,t}} \exp\left[-\frac{\left(\ell - \bar{\ell}(t)\right)^2}{2\,(2^2/3^3)\,t}\right]$$

where $P(\ell, t)$ = The probability of the shortest-path length being equal to ℓ at time t. The Fig. 18 shows the shortest path distribution at the router-level in the Internet.

We can see similar router level graphs in [25, 26, 27]. The width of the Gaussian for a 10000-node network, 1.1, is very close to the width of the Internet inter-domain distance distribution, 0.9, but the average distance is slightly higher-4.8 instead of 3.6 [28]. As noted in [24], simulation-based measurements of the distance distribution in the BA model also produce similar Gaussians, [29].

In Fig. 19, we fit Gaussian curve to the Internet's shortest path distribution from the year 2000.

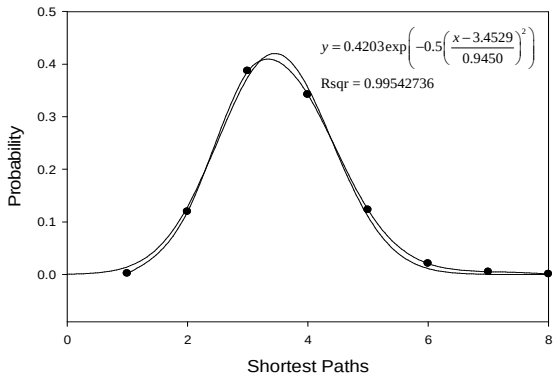


Fig. 19. Gaussian fit to the shortest path distribution in the year 2000.

2. Simulations and Discussion

For the purpose of our simulations, we use the NLANR's [30] Active Measurement Project (AMP) data. The AMP data provides a very good source for network analysis. There are around 150 AMP monitors deployed throughout many campuses in the United States and also some locations in other countries. These monitor take site-to-site measurements like Round Trip Time (RTT), packet loss, throughput and topology. We consider the traceroute measurements taken between the different AMP monitors. From these measurements, we calculate a shortest path matrix which has the shortest path (hop count) between all the AMP monitors. We use this shortest path matrix to get distance vectors.

The shortest path distribution for the AMP monitors is shown in Fig. 20. From the figure, we can see that the distribution is Gaussian-like as we expect in the Internet. We choose beacons randomly among the AMP monitors and calculate the distance vectors. From the distance vectors, we compute the value of $p_u(n)$ for different values of k. We plot the results in Fig. 21. We can see the simulation results are really off our model. This may be due to the fact that the model is not accurate enough to be applicable to the Internet and also the number of the AMP monitors

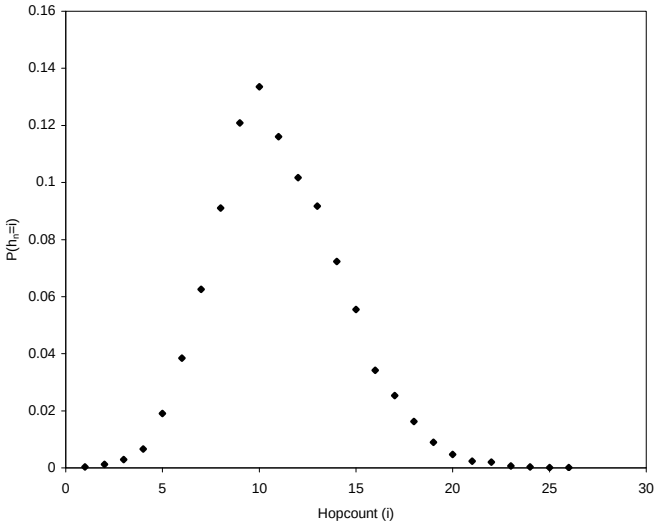


Fig. 20. Shortest path distribution for NLANR AMP data.

that we use for simulations are just 140. We can check the model using Internet topology generators for large n and see if the simulations come close to the model.

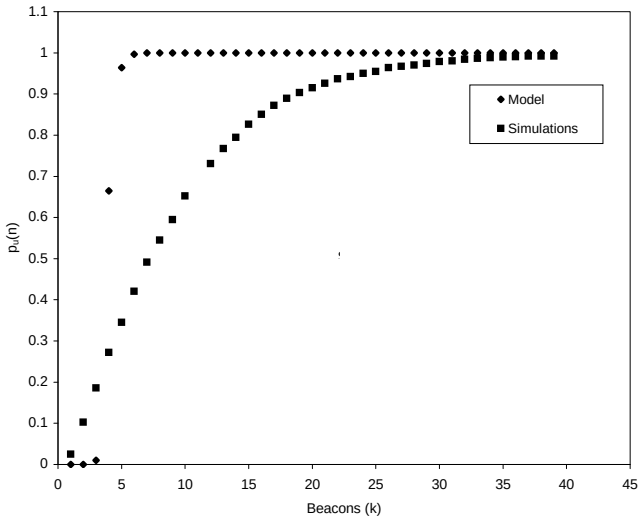


Fig. 21. Probability $p_u(n)$ versus number of beacons k for NLANR AMP Internet data.

CHAPTER V

AMBIGUITIES IN DISTANCE VECTORS

We have seen the conditions for the uniqueness of distance vectors. Uniqueness condition ensures that no two distance vectors are equal. If two nodes have equal distance vectors, then their distance metric M will not be the same and hence there will be no ambiguity between the nodes. Thus uniqueness of distance vectors is a necessary condition for resolving ambiguity. However, there can be instances where two nodes have different distance vectors but the same M values. Hence the uniqueness of distance vectors is not a sufficient condition for resolving ambiguity. In this chapter, we explore the reasons that give rise to ambiguities in distance vectors.

A. Distance Metric M

Recall that different beaconing schemes differ in the distance metric that they define. For example, in [1], the authors define an average distance metric that is given by:

$$M = \sum_{i=1}^{k} (d_{xi} - d_{yi})^2 \tag{5.1}$$

Ng *et al.* [5], define other average distance metrics, a normalized distance metric and a logarithmic distance metric which are given by:

$$M = \sum_{i=1}^{k} \left(\frac{x_i - y_i}{x_i} \right)^2$$

$$M = \sum_{i=1}^{k} (\log x_i - \log y_i)^2$$

Another scheme [2], uses a max-min distance metric given by:

$$M = \frac{\min\limits_{1\leq i\leq k}(|d_{xi} - d_{yi}|) + \max\limits_{1\leq i\leq k}(|d_{xi} - d_{yi}|)}{2}$$

The reasons for the ambiguity in distance vectors depend on the particular type of the distance metric. We focus on the average distance metric (5.1) in analyzing the reasons for ambiguities in distance vectors.

B. Distribution of M Values

Ambiguity arises when two nodes have the same M values without their distance vectors being the same.

Let $D_i = (d_{i1}, d_{i2}, ..., d_{ik})$ be the distance vector of a node i and $D_j = (d_{j1}, d_{j2}, ..., d_{jk})$ be the distance vector of node j. Let $D_n = (d_{n1}, d_{n2}, ..., d_{nk})$ be the distance vector of a new node that wishes to join the network. Ambiguity arises if M values for both i and j are the same i.e.

$$(d_{i1} - d_{n1})^2 + ... + (d_{ik} - d_{nk})^2 = (d_{j1} - d_{n1})^2 + ... + (d_{jk} - d_{nk})^2$$

To analyze the total number of such ambiguities, we need to look at the distribution of M values . Let X_i be a random variable that denotes the shortest paths to a beacon i. Let us assume that each of X_i s are i.i.d. In a $G(n, p)$ graph as well as the real Internet, each of the X_i s follows Gaussian distribution as we have seen in the previous chapters. The difference of any two X_i s is a zero-mean distribution. Fig. 22 shows the plot of the difference in the distances to beacon 3 for $n = 10000, p = 0.00034$. We can see that the mean is close to zero from the figure. To consider the distribution for each of the M values , we need to look at the distribution of the sum of the squares of the differences in the shortest paths.

Recall the definition of chi-Squared distribution [31] which is given by $\chi^2 = \sum_{i=1}^{r} Y_i^2$, where each of the Y_is is normally distributed with mean 0 and variance 1. In our case, the difference in the shortest paths are normally distributed with mean 0. But we do not have variance exactly equal to 1. Hence, we consider non-central chi-Squared distribution that is defined as follows:

If X_is are independent variates with normal distribution having means μ_i and variances σ_is for $i = 1, ..., n$, then $\frac{\chi^2}{2} = \sum_{i=1}^{n} \left(\frac{x_i - \mu_i}{2\sigma_i^2} \right)$ obeys gamma distribution with $\alpha = \frac{n}{2}$, i.e., $P(y)dy = \frac{1}{\Gamma\left(\frac{n}{2}\right)} e^{-y} y^{\left(\frac{n}{2}-1\right)} dy$. where $y = \frac{\chi^2}{2}$.

Let us denote the probability of M being equal to k as $P(M = k)$. Since M follows the chi-square distribution, we can write the gamma distribution equivalent as

$$P(M = k) = \frac{1}{\Gamma\left(\frac{k}{2}\right)} e^{\frac{-2y}{\sigma^2}} \left(\frac{2y}{\sigma^2} \right)^{\frac{k}{2}-1} \tag{5.2}$$

C. Simulations

We plot the distribution of M values from our simulations in Fig. 23 for $n = 1000$.

We analyze the shortest path distribution of nodes with the least M value for different values of n and p. In Fig. 24, we plot the shortest path distribution of nodes with the least M value for $n = 1000$ and $p = 0.01$. We observe that most of the nodes with the same M value are 3 and 4 hops away.

In Fig. 25, we plot the shortest path distribution of nodes with the least M value for $n = 10000$ and $p = 0.001$. We observe that most of the nodes with the same M value are 4 and 5 hops away.

From the plots, we can see that M metric that we use is most of the times not very accurate in determining the actual distance.

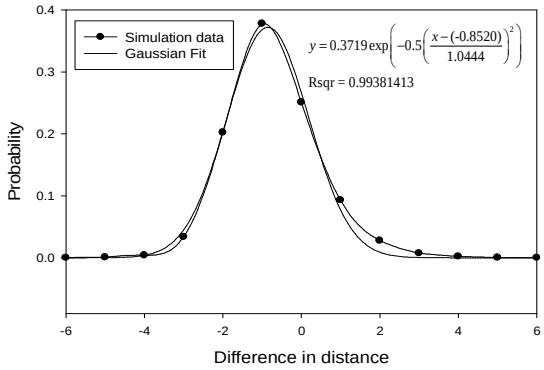


Fig. 22. Distribution of the difference in shortest paths.

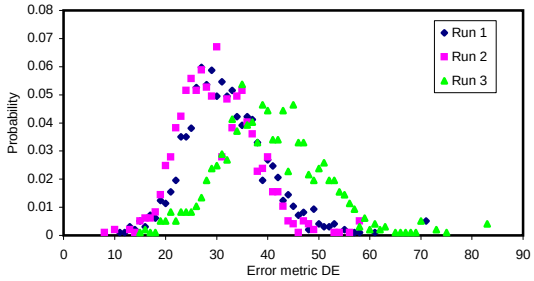


Fig. 23. Distribution of the M values.

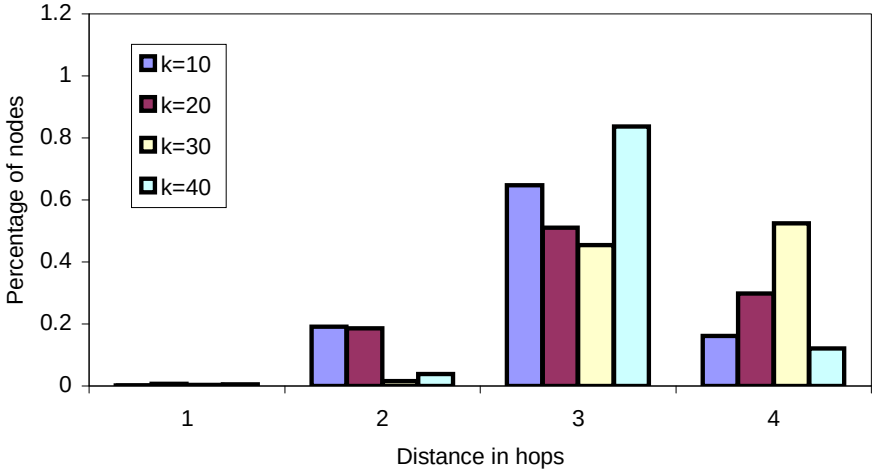


Fig. 24. Ambiguities in distance vectors in a $G(n,p)$ graph for $n = 1000$ and $p = 0.01$.

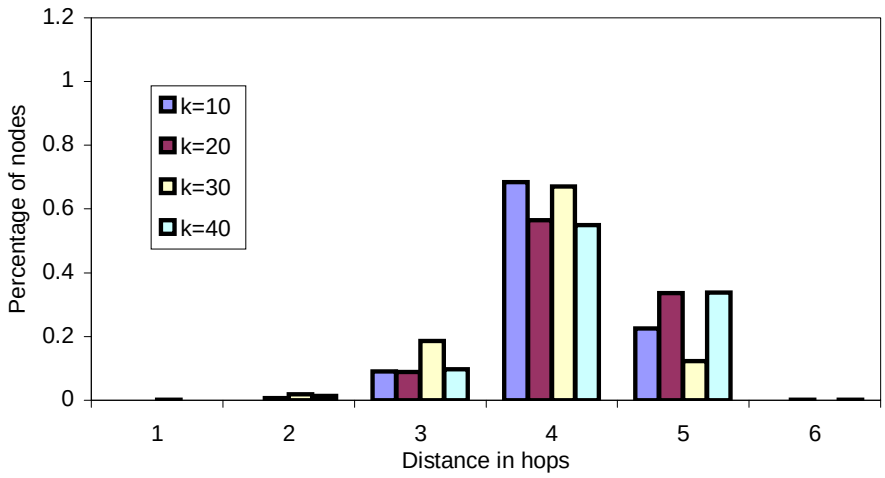


Fig. 25. Ambiguities in distance vectors in a $G(n,p)$ graph for $n = 10000$ and $p = 0.001$.

CHAPTER VI

GLOBAL NETWORK POSITIONING

In this chapter, we discuss in detail about Global Network Positioning (GNP) [5], a popular scheme used for network distance estimation. We give a brief overview of how the scheme works. We present some results on the accuracy and efficiency of the scheme.

A. Overview of GNP

GNP scheme models the Internet as an n-dimensional Euclidean space. The model defines a coordinate system where the hosts represent points in the space. The network distance between two hosts is obtained by evaluating a function on their coordinates. For efficiency in mapping the hosts to points, initially, the coordinates of certain special and well distributed *landmark* nodes are determined. Based on these coordinates, the positions of other hosts are measured relative to the landmarks. Thus the authors define a two-part architecture. The first part deals with the *landmark operations* which compute the coordinates of the landmarks. The second part of the architecture defines the *ordinary host operations* where the coordinates of the hosts are computed based on their relative distance to the landmarks. This scheme provides a fast and scalable way to estimate the network distances.

B. Simulations

In this section, we attempt to verify the accuracy of the GNP scheme. First, we generate a graph according to the $G(n, p)$ model and obtain the actual shortest path matrix by performing an all-pair shortest path computation on the graph. The output

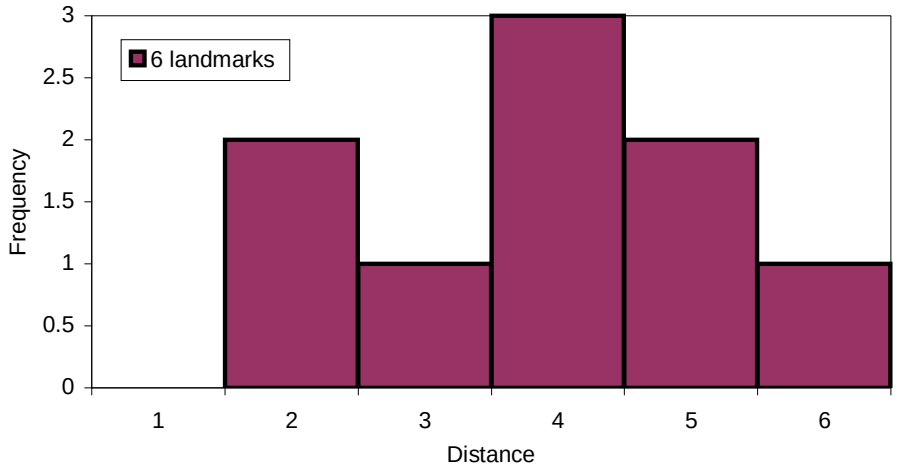


Fig. 26. Predicted distances between nodes from GNP scheme when the actual distance = 1 for $n = 1000$ and 6 landmark nodes.

shortest path matrix is given as an input to the GNP software. The GNP software then computes the predicted distances and outputs the predicted distance matrix. We perform simulations for $n = 1000$ and vary the number of landmark nodes.

We verify the accuracy of the GNP scheme in two ways. First, we analyze the predicted distances between nodes given by the GNP sofware when the actual distance between the nodes is 1 from the shortest path matrix (i.e. when the nodes are immediate neighbors). In this way, we verify how accurately the GNP scheme identifies the nearest peers. Next, we analyze how the actual distances between the nodes are when the GNP scheme predicts them to be the nearest neighbors. We plot our results in Figures. 26, 27, 28 and 29.

C. Discussion

In Fig. 26, we plot the distribution of the measured distances between nodes from the GNP scheme when they are actually immediate neighbors for 1000-node network when 6 landmark nodes are used. We observe that the predicted distances of most of

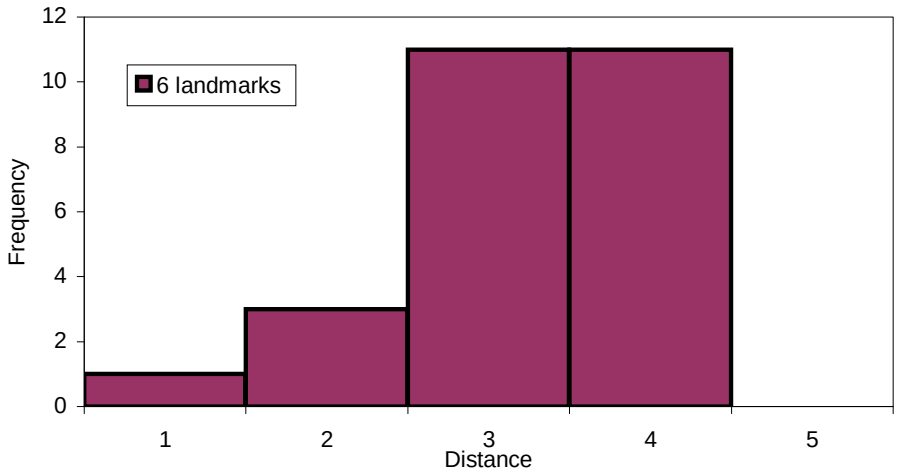


Fig. 27. Actual distances between nodes when the predicted distance from GNP scheme = 1 for $n = 1000$ and 6 landmark nodes.

the nodes that are actually neighbors are around 3 and 4 hops. In Fig. 27, we plot the distribution of the actual distances between nodes when the GNP scheme predicts them to be immediate neighbors for the same 1000-node network when 6 landmark nodes are used. We observe that the actual distances of most of the nodes that are predicted as neighbors are around 3 and 4 hops.

In Fig. 28 and Fig. 29, we compare the above results by varying the number of landmark nodes. We choose three different number of landmark nodes, 10, 15 and 20. From the two plots, we can observe that, as we increase the number of landmarks, the accuracy of prediction in the GNP scheme increases.

We can see that when the distance between nodes is actually one, GNP predicts the distance to be more than 2 hops away more than 80% of the time for 6 landmark nodes.

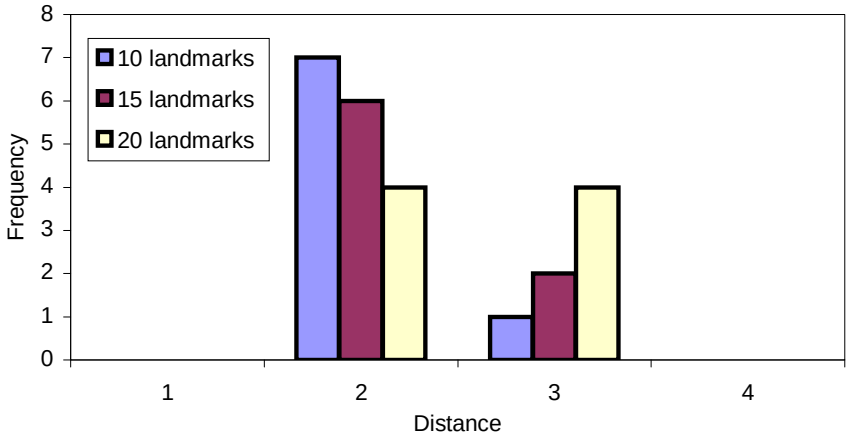


Fig. 28. Predicted distances between nodes from GNP scheme when the actual distance = 1 for $n = 1000$ and varying number of landmark nodes.

D. Chapter Summary

We provide an overview of the GNP scheme. We verify the accuracy of the scheme by comparing the predicted distance measurements from the GNP scheme to the actual distances. We conclude that, increase in the landmark nodes decreases the accuracy of the scheme.

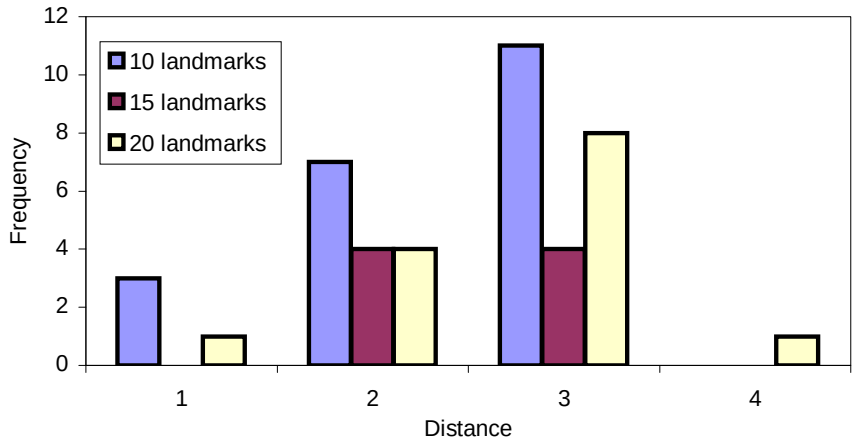


Fig. 29. Actual distances between nodes from GNP scheme when the predicted distance = 1 for $n = 1000$ and varying number of landmark nodes.

CHAPTER VII

CONCLUSION

The shortest path distribution of a given network is critical to determining the conditions for the uniqueness of distance vectors. We have seen that our model works well with uniform distribution of shortest paths. In the case of non-uniform distribution of shortest paths, we do not consider the dependencies in the distance vectors. Still our model is very close with simulations in predicting the number of beacons needed to ensure uniqueness of distance vectors. Our model is not accurate enough in the case of real Internet data and also Internet-like topologies since we do not consider the dependencies in the distance vectors of nodes.

Increasing the number of beacons will increase the probability of uniqueness of distance vectors. However, accuracy of predicting the network distances decreases when we increase the number of beacons. We analyze the performance of GNP scheme. We observe that as we increase the number of landmark nodes in GNP, the accuracy of prediction decreases.

REFERENCES

[1] N. Shankar, C. Komareddy, and B. Bhattacharjee, "Finding close friends over the Internet," in *International Conference on Network Protocols*, Washington, D.C., November 2001.

[2] S. M. Hotz, "Routing information organization to support scalable interdomain routing with heterogeneous path requirements," in Tech. Report, Computer Science Department, University of Southern California, Los Angeles, California, 1994.

[3] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-aware overlay construction and server selection," in *IEEE INFOCOM*, New York, NY, 2002.

[4] D. Loguinov, "Application layer multicast," in *CPSC 689 Lecture slides*, Department of Computer Science, Texas A&M University, 2002.

[5] T.S.E. Ng and H. Zhang, "Towards global network positioning," in *ACM SIGCOMM Internet Measurement Workshop*, San Francisco, California, November 2001., pp. 25–29.

[6] R. Cox, F. Dabek, F. Kaashoek, J. Li, and R. Morris, "Practical, distributed network coordinates," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 1, pp. 113–118, January 2004.

[7] Y. Shavitt and T. Tankel, "Big-bang simulation for embedding network distances in euclidean space," in *Technical Report, E.E.-Systems Department*, Tel-Aviv University, Isreal, July 2002.

[8] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "Idmaps: a global internet host distance estimation service," *IEEE/ACM Transactions on Networking (TON)*, vol. 9, no. 5, pp. 525–540, October 2001.

[9] W. Theilmann and K. Rothermel, "Dynamic distance maps of the internet," in *IEEE INFOCOM*, Portland, OR, USA, March 2000, vol. 1, pp. 275–284.

[10] J.D. Guyton and M.F. Schwartz, "Locating nearby copies of replicated internet servers," *ACM SIGCOMM Computer Communication Review*, vol. 25, no. 4, pp. 288–298, October 1995.

[11] H. Lim, J.C. Hou, and C. Choi, "Constructing internet coordinate system based on delay measurement," in *2003 ACM SIGCOMM conference on Internet Measurement*, Miami Beach, FL, USA, 2003, pp. 129–142.

[12] M. Pias, J. Crowcroft, S. Wilbur, S. Bhatti, and T. Harris, "Lighthouses for scalable distributed location," in *Peer-to-Peer Systems II, Second International Workshop, IPTPS*, Berkeley, CA, USA, February 2003, pp. 278–291.

[13] Y. Chen, K. Lim, R. H. Katz, and C. Overton, "On the stability of network distance estimation," in *ACM SIGMETRICS Practical Aspects of Performance Analysis Workshop (PAPA)*, 2002.

[14] M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator," *ACM Trans. on Modeling and Computer Simulation*, vol. 8, no. 1, pp. 3–30, January 1998.

[15] P. van Mieghem, G. Hooghiemstra, and R.W. vander Hofstad, "A scaling law for the hopcount in the internet," in Technical Report, Delft University of Technology, The Netherlands, 2000.

[16] F. Chung and L. Lu, "The diameter of random sparse graphs," *Advances in Applied Math*, pp. 257–279, 2001.

[17] B.M. Waxman, "Routing of multipoint connections," *Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, December 1998.

[18] K.I. Calvert, M.B. Doar, and E.W. Zegura, "Modeling Internet topology," *IEEE Communications Magazine*, vol. 35, no. 6, pp. 160–163, June 1997.

[19] M. Doar, "A better model for generating test networks," in *IEEE Global Telecommunications Conference*, London, UK, November 1996, pp. 86–93.

[20] A. Medina, I. Matta, and J. Byers, "On the origin of power laws in Internet topologies," *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 2, pp. 18–28, April 2000.

[21] P. Barford, A. Bestavros, J. Byers, and M. Crovella, "On the marginal utility of deploying measurement infrastructure," in *Technical Report Computer Science Technical Report 2000-018*, Boston University, July 2000.

[22] W. Aiello, F. Chung, and L. Lu, "Random graph model for massive graphs," in *32nd Annual ACM Symposium on Theory of Computing*, Portland, OR, May 2000, pp. 171–180.

[23] R. Albert and A.L. Barabsi, "Topology of Evolving Networks: Local Events and Universality," *Physical Review Letters*, vol. 85, no. 24, pp. 5234–5237, December 2000.

[24] S.N. Dorogovtsev, A.V. Goltsev, and J.F.F. Mendes, "Pseudofractal scale-free web," *Physical Review E*, 65(06):066122, 2002.

[25] A. Vazquez, R. Pastor-Satorras, and A. Vespignani, "Internet topology at the router and autonomous system level," Los Alamos Archive, cond-mat/0206084, 2002.

[26] A. Vazquez, R. Pastor-Satorras, and A. Vespignani, "Large-scale topological and dynamical properties of the Internet," *Physical Review E*, vol. 65, no. 6 2, pp. 066130/1–066130/12, June 2002.

[27] A. Broido, E. Nemeth, and K. Claffy, "Internet expansion, refinement and churn," *European Transactions on Telecommunications*, vol. 13, no. 1, pp. 33–51, Jan/Feb 2002.

[28] D. Krioukov, K. Fall, and X. Yang, "Compact routing on internet-like graphs," in Tech. Report IRB-TR-03-010, Intel Research, 2003.

[29] A. Krzywicki, "Defining statistical ensembles of random networks," in *Workshop on Discrete Random Geometries and Quantum Gravity, cond-mat/0110574*, Utrecht, Netherlands, 2001.

[30] "NLANR's Active Measurement Project (AMP)," "http://watt.nlanr.net/".

[31] M. Evans, N. Hastings, and Brian Peacock, *Statistical Distributions, 2nd ed.*,, Wiley & Sons, Inc, New York, 1993.

VITA

Name: Geetha Kakarlapudi

Address: 301 Harvey R. Bright Building, Computer Science Department, Texas A&M University, College Station, TX 77843-3112

Education:

- Master of Science, Computer Science, Texas A&M University, December 2004
- Bachelor of Engineering, Computer Science and Engineering, Mangalore University, June 2001

Work Experience:

- Graduate Research Assistant, Department of Horticulture, Texas A&M University System, May 2003 - September 2004
- Senior Software Development Engineer, Novell Software Development Inc, Bangalore, India, July 2001 - August 2002