Evasiveness of Internet Topology

Çiğdem Gündüz*, Mehmet Balman†, Bülent Yener‡

Abstract

Internet topology generators aim at producing graphs that have similar properties with the "actual" Internet. Each generator focuses on different properties to define *metrics* and uses them to compare its output graph to that of the Internet.

This paper shows that most of the metrics are *evasive*: their exact values cannot be determined without visiting all links in the Internet graph. This indicates a fundamental difficulty with the topology generators since (i) due to its size and dynamics, a complete Internet graph cannot be obtained, thus the comparisons are made under incomplete information, and (ii) each generator focuses on different metrics, thus comparison of topology generators is not absolute.

This work provides a *meta-metric* called (γ, σ)-evasiveness to determine if a metric can be estimated with at least 1-σ accuracy by using γ percentage of data. The (γ, σ)-evasiveness of the metrics considered in the literature is computed over the actual internet data. It is shown that the data collection method has impact on the values of the metrics, thus motivating a through comparison of the internet topology generators and their metrics. Finally, based on these observations a *greedy* algorithm is proposed for deciding on *optimal* amount of data to be collected.

*Ç. Gündüz is with the Department of Computer Engineering, Bogazici University, Istanbul, TR-80815, TURKEY (email: gunduzc@boun.edu.tr)

†M. Balman is with the Department of Computer Engineering, Bogazici University, Istanbul, TR-80815, TURKEY (email: balman@cmpe.boun.edu.tr)

‡B. Yener is with the Department of Computer Science, Rensselaer Polytechnic Institute, NY (email: yener@cs.rpi.edu)

1 Introduction

The Internet topology can be considered either at the router level [13] or at the inter-domain level [2, 8, 11]. Obtaining an accurate map of the Internet topology is a time consuming and non-trivial task.

Due to the immense size of the Internet, only a limited amount of data can be collected within a short time period. Furthermore due to its dynamics, the data may become outdated in a short time. Thus, many topology generators are proposed to produce graphs that have similar topological properties with the actual Internet graph. Each generator focuses on different properties to define a set of *metrics* to be satisfied by the output graph. Evaluation of the generators is based on comparing the values of these metrics on the produced graphs to that on the "actual" Internet graph obtained by *measurements*. Therefore snapshots of the Internet in different time periods are taken and they are used to analyze its topology and to verify the data generator models.

As the size of the snapshots determines the values of metrics, one must determine at least how much data are enough to draw meaningful conclusions. It is obvious that the metrics converge to their actual values as the size of data increases. However the complexity of collecting data from the Internet, the memory requirements to store this data, and the computational and space complexity of the metric evaluation also increase with the data size. For these reasons it is important to determine minimum amount of data that must be collected so that results will be to generalized to the whole Internet.

In [7], Floyd and Paxson discuss the difficulties in simulating the Internet. Multiple administrative policies and variations of the traffic over different periods of time make the topology of the Internet heterogeneous and it becomes difficult to define a

typical Internet behavior. In this work we continue with explaining the difficulties of simulating the Internet from a topological perspective. In particular we show that most of the metrics are *evasive*: their exact values cannot be determined without visiting all edges in the Internet graph. The evasiveness of metrics can be a fundamental difficulty with the Internet generators since the exact value of these metrics on the Internet is not available and the generators try to produce graphs on which the metrics have similar values to that of the Internet.

This paper provides a *meta-metric* called (γ, σ)-evasiveness, to answer the following questions: (i) can a metric be estimated with 1-σ accuracy by using only γ percentage of the total data? (ii) how do the values of γ and σ change for different metrics? (iii) is the sampling method important in this task? (iv) how do the (γ, σ)-evasiveness of the metrics change over the graphs generated by internet topology generators? Based on the answers to these questions, a greedy algorithm is proposed for data collection to address the cost-accuracy trade off.

The rest of the paper is organized as follows: In Section 2, we review different data generation models, metrics to use in model evaluations, and methods to compare a metric for different models. We define (γ, σ)-evasiveness in Section 3. Data collection procedures and sampling processes are explained, and the observations on the changing behaviors of metrics on the "actual" Internet data are discussed in Section 4. In Section 5, we compare the data generated by different topologies in terms of (γ, σ)-evasiveness. We explain our greedy algorithm that decides the network size dynamically and its results in Section 6. Section 7 contains the conclusion and discusses possible future work.

2 Background and Related Work

Internet topology generators can be categorized into three groups– random, structural, and power law degree generators. The first widely used random model is developed by Waxman [18], in which links are added with the probability depending on the Euclidean distance of its vertices. After the work of [6], many simulators have been proposed for generating power law based topologies [1, 2, 9]. In contrast with the degree-based models, structural models connect smaller random graphs to form the larger structures.

To show the similarities and the differences between all these models as well as the "actual" Internet, many metrics are defined on the data they generate. In the literature, there are different ways to make comparison of the same metric computed for different models. One way is just to compare their values and observe how close the metric values are. In [9], Jin et al. make such comparisons for the power law exponents mentioned in [6]. In [11], average path length and clustering coefficients are also compared in this way.

Another way is to examine the correlations between different metrics for different models. Bu and Towsley [2] examine the pairwise correlations between clustering coefficient, characteristic path length, and the maximum degree. Such a comparison is based on the observation that the Internet graph shows the small world characteristics discussed in [17]. Remember that the correlation between the clustering coefficient and the median shortest path length is an indicator of a small world graph.

Zegura et al. use statistical methods to make pairwise comparisons [20]. They use Kolmogorov-Smirnov test to answer whether metrics that are computed on data generated by different models come from the same distribution or not. Moreover, they define *intermetric* probability as a similarity measure. They compare the diameter of different graphs using this approach.

In [16], it is proposed to compare the metrics of different models by using their growing quantities. They compute the metrics (expansion, resilience, and distortion) on different sized networks that are formed by taking all nodes and edges within h hops and distinguish different topologies by using the changing curves of the metrics as a function of h.

In all these comparisons, it is important to determine the appropriate size of a network such that the value of a metric should converge to its real value. In [14], Riley and Ammar compute the metric on a small graph initially. In each step, they compare the current value of the metric with its previous value and increment the size of the graph unless these two values are the same. Our greedy algorithm is similar to this approach but we com-

pare the current value of a metric with its prior values computed in K steps. Note that we propose to compare the metrics of different models by comparing the values of γ with a fixed value of σ in their (γ, σ)-evasiveness properties.

2.1 Topology Generators

In this section, we review most commonly used models briefly. First Waxman model and its variations are described, then transit-stub model as an example of structural models is explained, and last power based models are discussed.

2.1.1 Waxman Model

In this model [18], N nodes are randomly distributed in a plane and a link probability between any two nodes u and v is defined as:

$$P(u,v) = \alpha \cdot e^{-d \ / \ (\beta \cdot L)} \tag{1}$$

where d is the Euclidean distance between nodes u and v, and L is the maximum Euclidean distance between any two nodes. α and β are parameters such that $0 < \alpha, \beta \leq 1$. In this equation, α controls the number of edges. Increasing α results in dense graphs and decreasing β increases the ratio of shorter edges over longer ones. Some variations of Waxman model are also defined:

1. Doar and Leslie defines the link probability between nodes u and v as:

$$P(u,v) = \frac{k \cdot e}{N} \cdot \alpha \cdot e^{-d \ / \ (\beta \cdot L)} \tag{2}$$

   where N is the number of nodes, e is the desired average node degree and k is a constant which depends on α and β. Additional term $(k \cdot e)/N$ provides more control over the number of edges [4].

2. In the exponential model [19], the link probability depends on only one parameter which has the effect on the number of edges and given as follows:

$$P(u,v) = \alpha \cdot e^{-d \ / \ (L-d)} \tag{3}$$

2.1.2 Transit-Stub Model

Calvert et al. point out that routing domains in the Internet fall into either transit or stub domains. In a stub domain, only traffic between two routers that belong to that stub domain is carried. There is no such restriction for transit domains. Transit-stub model is proposed to reflect the differences between these different domains [3]. In this model, a random graph R is generated for all transit domains. A node in the graph R corresponds to a single transit domain and it is replaced by another random graph R_i at the second step. Next, each node in the graph R_i is connected to a number of random graphs considered as stub domains. The process is finished by adding extra edges between the nodes of transit and stub domains and between the nodes of different stub domains.

2.1.3 Power Law Graph Generators

The power law random graph model explained in [2] generates data according to node outdegree power law. This model first assigns degrees to N nodes depending on $f_d \propto d^O$, where f_d is the frequency of outdegree d and O is the outdegree exponent. The degrees are matched with the nodes and a graph is formed by connecting each node with randomly selected m nodes, where m is the outdegree of the corresponding node. After deleting self loops and merging duplicate links, a giant connected component is taken as the generated model.

Barabasi and Albert claim that incremental growth and preferential attachment cause power law distributions observed in the Internet and use these properties in their proposed model [1]. In their model, they begin with a small number of nodes. At each time t, they add a new node n with a specified number of links. The node n connects to other nodes such that it connects to the nodes with larger degrees with larger probabilities.

In addition to incremental growth and preferential attachment, Medina et al. report that node placement and connection locality are the other reasons on the origin of power laws [11]. They claim that nodes are skewed distributed in space, which is referred as node placement, and nodes have tendency to connect closer nodes, which is referred as connection locality. Their generator, BRITE, also makes use of these two findings to

generate the power law distributed networks.

In [9], Jin et al. show that the outdegree and the frequency grow exponentially over time. Their generator, Inet 2.0, takes November 1997 as its origin and generates data accordingly. First it computes how many months are needed to reach N nodes from November 1997. The number of months is used in the computation of the outdegree-frequency and the rank outdegree distributions. Then Inet 2.0 connects each node to its neighbors according to these distributions.

2.2 Metrics

Metrics can be distinguished in terms of the range of the information they provide. Local metrics, which are extracted from individual nodes, give information about a single node whereas global metrics reflect the properties of a whole network. A single global value can be extracted from the local values by using simple statistics such as mean, median, minimum or maximum.

2.2.1 Degree

Degree is the most simple and the trivial metric and it is defined as the number of the connections of a single node for an undirected graph. For directed graphs, indegree and outdegree are also defined. To evaluate the degree of node i, we should check all other nodes whether they are connected to node i or not. Thus the complexity of the degree computation for all graph is $\mathcal{O}(N^2)$, where N is the number of nodes in a graph.

Although degree of a node is a local property, the statistics on it give connectivity information of a whole graph. For example, the average node degree gives the number of connections that a typical node has. It is also possible to use the minimum and the maximum of node degrees to obtain a global value. All these statistics are computed in a $\mathcal{O}(N)$ time, thus the complexity remains $\mathcal{O}(N^2)$.

Faloutsos et al. state that metrics based on minimum, maximum and average values are not sufficient to describe the skewed distributed data and they propose to use the exponents of power laws as new metrics [6]. The exponents measure the tendency of a property. The exponents based on the connectivity of the nodes are given as follows:

1. **Outdegree exponent O:** Faloutsos et al. examine the frequency of the outdegrees. Frequency f_d of an outdegree d is defined as the total number of nodes with degree d. They observe $f_d \propto d^O$, where the outdegree exponent O is the slope of f_d versus d plot.

2. **Rank exponent R:** In [6], the rank exponent R is defined as the slope of a d_v versus r_v plot, where d_v and r_v are the outdegree and the rank of the node v respectively and it is observed that $d_v \propto r_v^R$. In the computation of rank values, degrees of all nodes should be sorted first (there are sorting algorithms with $\mathcal{O}(NlogN)$ complexity) and the overall complexity still remains $\mathcal{O}(N^2)$.

3. **Eigen exponent $\mathcal{E}$:** The connectivity information of a graph can be kept in an adjacency matrix. In [6], it is observed that $\lambda_i \propto I^{\mathcal{E}}$, where λ_i is an eigenvalue of the adjacency matrix. All eigenvalues are sorted in decreasing order and the value of i gives the order of λ_i. We use Jacobi transformations for the computation of eigenvalues, with complexity $\mathcal{O}(N^3)$.

2.2.2 Clustering Coefficients

Clustering coefficients are the local metrics that reflect the connectivity information in the neighborhood environment of a node [5]. It can be also thought that they provide the transitivity information [12], since it controls whether two different nodes are connected or not if they are connected to the same node.

Clustering coefficient C_i is defined as the percentage of the connections between the neighbors of node i and it is given as:

$$C_i = \frac{2 \cdot E_i}{k \cdot (k-1)} \tag{4}$$

where k is the number of neighbors of node i and E_i is the existing connections between its neighbors.

Clustering coefficient D_i is defined similar to C_i with an exception. It also considers node i and its connections in the computation of the clustering coefficient [5]. The formula of D_i is given as:

$$D_i = \frac{2 \cdot (E_i + k)}{k \cdot (k+1)} \tag{5}$$

The global clustering coefficients of C and D are computed as the averages of C_i and D_i respectively. In [2], clustering coefficient $C^{(2)}$ is computed by taking average over the clustering coefficients of all nodes, C_i, except the ones whose degrees are one.

In the computation of the clustering coefficient of a single node, we should check all edges in the worst case. This gives $\mathcal{O}(E)$ for a single node and $\mathcal{O}(E \cdot N)$ for a whole graph, where E and N are the numbers of the edges and the nodes in a graph respectively.

2.2.3 Distance Between Nodes

The hop distance between nodes u and v is defined as the shortest path between them, taking the weight of each edge as a unit length. Note that, it is also possible to define distances in terms of physical distances between nodes [20]. In this work, only the metrics based on hop distances are considered. The *diameter* of a graph is the maximum of minimum distances between any two nodes and it determines the effective size of a network.

Closeness and *Betweenness* are local metrics that measure the connectedness of a network [12]. The closeness of node i is the average distance from node i to all other nodes. It reflects the centrality property of a single node and smaller values indicate that this node places close to the center of a network. The average path length is one of the global metrics defined as the average of the closeness values for all nodes [11]. In [2], the characteristic path length is defined as the median of all closeness values. Betweenness of a node i is the total number of the shortest paths that pass through node i. The higher value of a node indicates that the node has greater flow, thus it controls the traffic in a network.

Eccentricity of node i is another local metric and it is defined as the minimum number of hops required to reach at least 90 per cent of reachable nodes from node i. By taking average over all nodes, it also reflects the size of a network.

In [6], exponent $P(h)$, which is the indicator of connectiveness of the graph, is defined as the number of pairs within h hops. Faloutsos et al. state that $P(h) \propto h^{\mathcal{H}}, h \ll \delta$, where $\mathcal{H}$ is the *hop-plot exponent* and δ is the diameter of a network. In that work, the effective hop diameter δ_{ef} is defined as:

$$\delta_{ef} = \frac{N^2}{N + 2 \cdot E}^{1/\mathcal{H}} \quad (6)$$

In the computation of all these metrics, hop distances between all the nodes should be computed. By using breadth-first search, hop distances from node i to all other nodes can be computed in $\mathcal{O}(E)$ time. Thus the computational complexity of each metric is $\mathcal{O}(E \cdot N)$.

3 (γ, σ)-Evasiveness

There are more than a dozen metrics proposed in the literature along with several topology generators to model the Internet graph. Some of these generators value some metrics more than the others. In this work we examine which metrics are hard to compute accurately by defining an *approximate* evasiveness notion, and to compare the graph generators with respect to this new metric.

Graph evasiveness (also known as elusiveness) considers the following problem. Given an input graph G suppose we are to decide if G has a certain property P by asking, to an oracle $\mathcal{O}$, whether or not edge (u, v) belongs to G. In a graph with N nodes there are at most $N(N-1)/2$ edges that can be used as a query to the oracle $\mathcal{O}$. If the decision about P can only be made using *exactly* $N(N-1)/2$ queries then the property P is said to be *evasive*. In other words if P can only be decided by checking all the edges of G then it is an evasive property. Thus, evasiveness of graphs is used for determining the worst case complexity of computing some graph properties [10].

It is conjectured by Karp that every nontrivial monotone graph property is evasive [15]. A property is *monotone* if insertion of new edges to a graph with property P does not destroy the property and P is *nontrivial* if it holds for some graphs with N nodes and it does not hold for some other with the same number of nodes. Planarity, 2-connectivity, connectivity are examples of such properties.

In this work we introduce a new concept called (γ, σ)-evasiveness by relaxing the strict or exact evasiveness definition as follows:

Definition 1 *Given a graph G with vertex set V and edge set E, a property P is called (γ, σ)-evasive*

if it can be "computed" by making at least $\lceil \gamma |E| \rceil$ *queries with an error margin of at most* $\pm\sigma$ *for* $0 \leq \gamma \leq 1$ *and* $\sigma \geq 0$.

Note that in this definition the graph is given but computation of a property has (γ, σ)-evasiveness The (γ, σ)-evasiveness is an approximation to the exact evasiveness with two differences: in exact evasiveness (i) the next query can be chosen based on the current answer from the oracle, and (ii) there is no margin of error - the property P exists or not.

In (γ, σ)-evasiveness a property (metric) P is called monotone if for $\gamma' > \gamma$ property P still holds. In other words additional information will not change the fact that metric P can be computed with an error margin of at most $\pm\sigma$. Similarly, if the variance of a metric P, which is computed over say K graphs $G = (V, E)$ with $\gamma|E|$ queries with an error margin of at most $\pm\sigma$, is non-zero then P is said nontrivial.

As we shall show later in this paper, evasiveness of the metrics used by the topology generators varies significantly on the Internet graph from that on the generated topologies [1].

4 Emprical Observations

4.1 Actual Data Collection

Data collection procedure collects path information between different IP addresses and constructs IP router graph. We use IP addresses of the traceroute gateways found in *www.traceroute.org* and *www.tracert.com*. We obtained the pairwise paths from the traceroute commands executed between these servers. We processed these paths to eliminate the redundancies in order to identify node and edges sets of the graph.

Finally we constructed 10 different Internet graphs at the router level, where the total node count is 6-7K and the total edge count is 11-15K.

4.2 Sampling Methods

In this work, we take samples from each collected data and observe the values of metrics on them, depending on their sizes. To this end, we use two different sampling methods. *Hop sampling* creates a subnetwork of the original topology by randomly selecting an initial node at the edge of the network and growing it from that node according to the specified hop count h. All nodes and edges visited within h hops are taken to form a subnetwork. This technique is called "ball growing" in [16]. In our experiments, we select hop count value starting with one and incrementing it until the sampling graphs nearly converge to their original topology.

In *path sampling*, we grow a subnetwork by using the paths between traceroute servers that are used in data collection. Given the set of traceroute servers $S = \{S_i \mid i = 1, ..., K\}$ and an empty subnetwork H, it works as follows:

1. Choose an initial source server S_u randomly
2. Choose a destination server S_v similarly, such that $v \neq u$ and $S_u \rightarrow S_v$ is not chosen before
3. Run traceroute to get path $P(u, v) = S_u \rightarrow S_v$
4. $H = H \bigcup P(u, v)$
5. Set S_v as the new source (new S_u) and go to step 2 unless all paths are visited

In step 3, we need to get path $P(u, v) = S_u \rightarrow S_v$ and this run has been already done during data collection. In step 5, we make the selection of the source server deterministic while keeping the destination server still randomly chosen to ensure connectivity. In this algorithm, in each iteration a new path is added, therefore a new subnetwork is formed. On the other hand, we compute the metrics on these subnetworks only at certain points, at the first time that the following percentages of the total nodes $5\%, 10\%, ..., 95\%$ are covered.

4.3 Metrics Used

We select less costly metrics to make them efficiently computable in terms of time and memory. For example we discard the eigenvalue exponent since increasing the size of networks increases the eigenvalue computational time rapidly and requires larger amount of memory. We do not compute the outdegree exponent and the rank exponent, since they are defined on directed graphs and we keep our graphs as undirected.

In this work, the following metrics are used: (i) the average node degree (ii) the average clustering

[1] For the rest of this paper evasiveness will refer to (γ, σ)-evasiveness unless a distinction is made explicitly.

coefficients C, $C^{(2)}$ and D (iii) the hop diameter of a graph (iv) the hop-plot exponent (v) the effective hop diameter (vi) the maximum betweenness (vii) the minimum closeness, the average path length and the characteristic path length (viii) the average eccentricity

4.4 Observations on Metrics

For each one of the 10 graphs we run both sampling methods 20 times to compute the metrics over a data set of 200 runs.

In Figures 1-6, we plot the changing behaviors of the metrics for both sampling methods. In these graphs, x-axis indicates how many percentage of data are visited in each iteration. The y-axis in these plottings gives the ratio between the values computed in each iteration and the exact metric values. For each iteration, a boxplot is given. In the boxes, the lower quartile, median, and upper quartile values of data are shown as lines. The extent of the rest of the data is also shown at the lines extending from each end of the box.

In all these graphs, we check the first point where the median values shown in boxes lie within the ± 10 per cent error margin for this point and all its successor points. This gives us at least how many percentage of data are visited when the metrics converge to their exact values within ± 10 error margin, in other words this gives us the γ values for $(\gamma, 0.10)$-evasiveness. In the x-axis, these points are also given with their percentage values. For example, in the average degree graph in Figure 1, 21% shows that the average degree is computed with 10 per cent error first with 21 per cent of data. At all the successor of this point, error margin gets smaller.

Moreover, in all these graphs, we also check the variances. The longer boxes and lines indicate the greater variability of data.

In Figures 1 and 2, we give the metrics related to the connectivity of a node and its neighbors. From these graphs, we see that *path sampling* method causes metrics to change linearly as subnetworks grow. This means that these metrics do not converge their exact values unless all nodes are visited. For example, the average degree and the clustering coefficient C are (0.85,0.10) and (0.90,0.10)-evasive respectively. On the other hand, the metrics that are computed on subnetworks created by *hop sampling*, "converge" faster than are those computed by *path sampling*. The average degree and the clustering coefficient C are (0.21,0.10) and (0.57,0.10)-evasive respectively. These values are more tolerable compared to those for *path sampling*. We conclude that to have the connectivity information, it is better to use *hop sampling* in data collection. In these graphs, we also see that clustering coefficient $C^{(2)}$ is not as computationally efficient as the other two clustering coefficients in both sampling methods. Thus it is better to use clustering coefficients C and D instead of $C^{(2)}$ to obtain the connectivity information between the neighbors of a typical node.

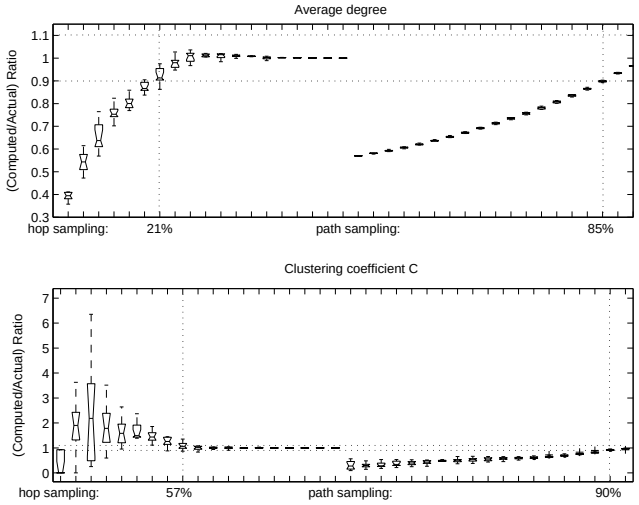

Figure 1: The changing behaviors of the average degree and the clustering coefficient C.

Another observation on the graphs in Figures 1 and 2 is that, the values of clustering coefficients of C and $C^{(2)}$ are increasing when a subgraph grows by using *path sampling* but they are decreasing when it grows by using *hop sampling*. The reason is that we begin with a sparse network with *path sampling*, and it becomes denser as increasing the size. On the other hand subnetworks of smaller sizes are denser and they become sparser in *hop sampling*.

It is also interesting to observe that the average degree and the clustering coefficients computed on the subnetworks obtained by *hop sampling* show

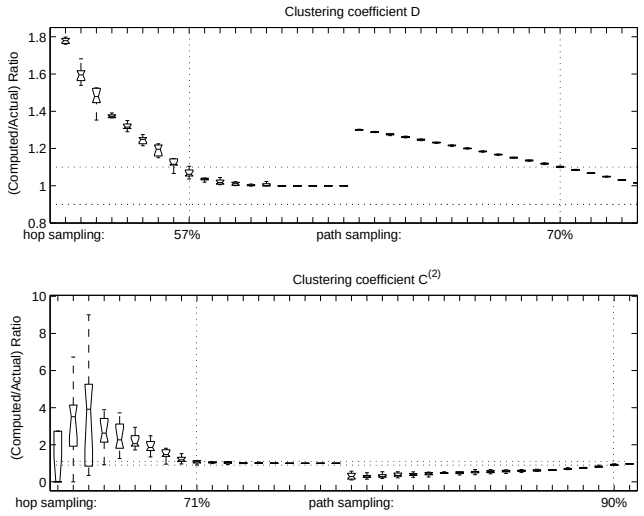

Figure 2: The changing behaviors of the clustering coefficients, D and $C^{(2)}$.

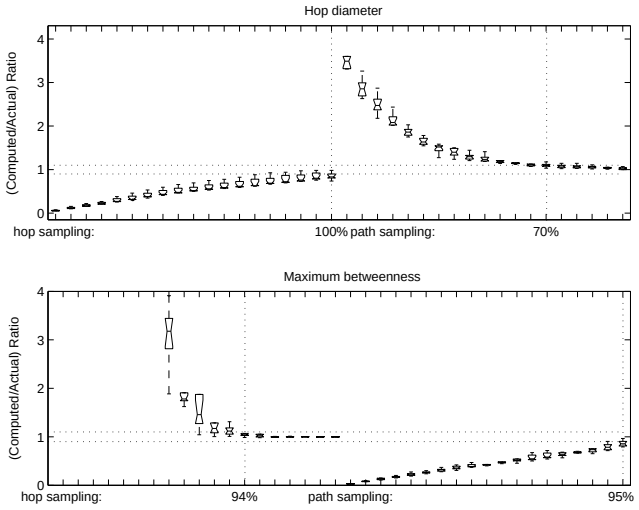

Figure 3: The changing behaviors of the hop diameter and the maximum betweenness.

greater variability than are those obtained by *path sampling*. It indicates that the randomness of selecting the initial nodes has greater effects when we use *hop sampling*, and one must consider this effect in analyzing these metrics. As subnetworks converge to their original graph, variance decreases as expected.

In Figures 3-6, the metrics related to the shortest paths between every two nodes are shown. In Figure 3, we see that *path sampling* performs better than *hop sampling* on the diameter of the network, since hop count value determines the size of the network in *hop sampling*. For the maximum betweenness, we see that both sampling methods do not perform well, it is (0.95,0.10)-evasive and cannot be estimated within error margin ± 10 unless 95 per cent of the graph is visited. Thus we draw a conclusion that maximum betweenness is not a good metric in characterizing the network topology, it changes as the network changes and its exact value cannot be approximated.

Figure 4 again shows that it is better to use *hop sampling* to approximate the exact values closer. From the graphs of both the hop diameter and the effective hop diameter, one can say that in *hop sampling* the length of the shortest paths are shorter than are those obtained by *path sampling*. Note that the effective hop diameter show small variability in both sampling methods. This is a plus for the effective hop diameter.

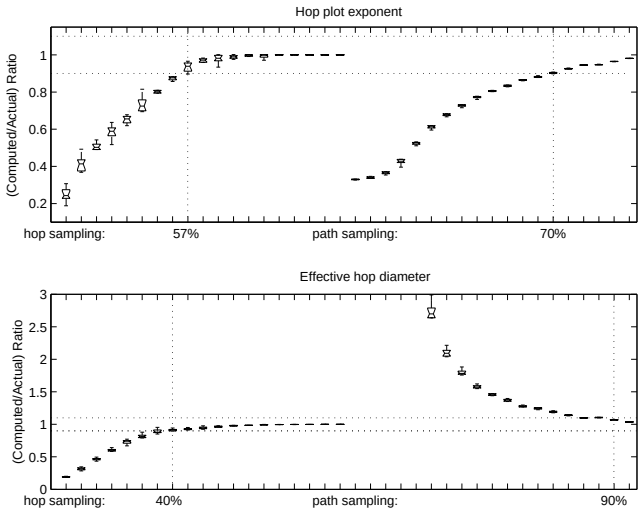

Figure 4: The changing behaviors of the hop plot exponent and effective hop diameter.

In Figures 5 and 6, the statistics on the closeness values and the average eccentricity are given and their changing behaviors are very similar. In *hop sampling*, they are (0.57,0.10)-evasive and show smaller variability, which make advantageous to use them.

In summary, to approximate the exact values of

Hop sampling	Path sampling
1. Average degree (21%)	1. Clustering coefficient D (70%)
2. Effective hop diameter (40%)	Hop plot exponent (70%)
3. Clustering coefficient C (57%)	Hop diameter (70%)
Clustering coefficient D (57%)	2. Characteristic path length (80%)
Hop plot exponent (57%)	Average path length (80%)
Characteristic path length (57%)	Average eccentricity (80%)
Average path length (57%)	3. Average degree (85%)
Minimum closeness (57%)	Minimum closeness (85%)
Average eccentricity (57%)	4. Effective hop diameter (90%)
4. Clustering coefficient $C^{(2)}$ (71%)	Clustering coefficient C (90%)
5. Maximum betweenness (94%)	Clustering coefficient $C^{(2)}$ (90%)
6. Hop diameter (100%)	5. Maximum betweenness (95%)

Table 1: Ranking of metrics for two different data collection methods. The values in parenthesis indicate the γ values (i.e. percentage of the actual data had to be used) for computing each metric with $\sigma = 0.10$. The values are average values computed over 200 runs.

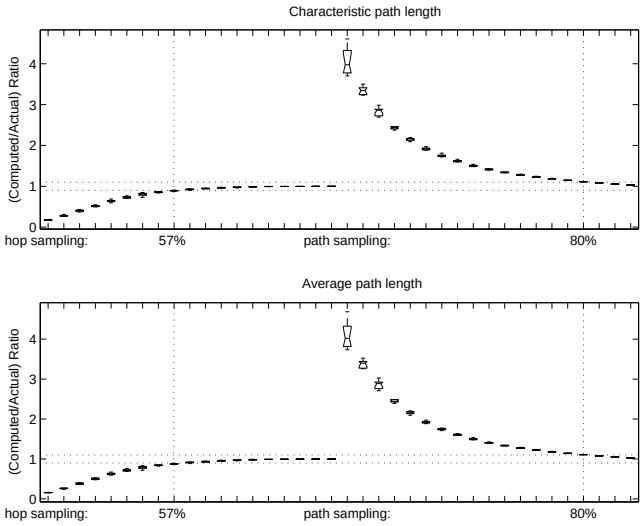

Figure 5: The changing behaviors of the characteristic and the average path lengths.

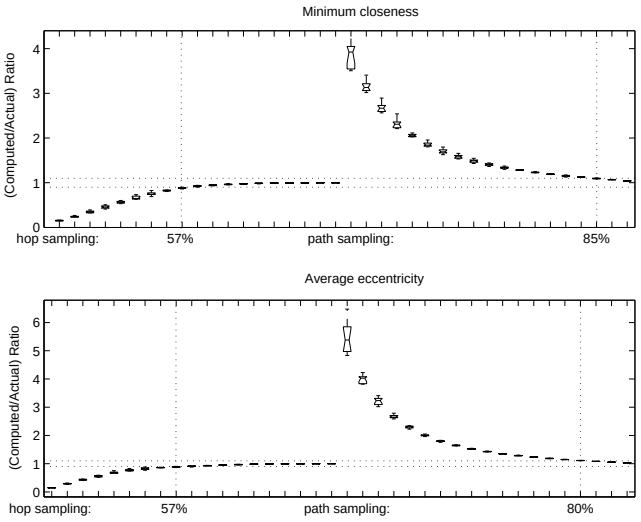

Figure 6: The changing behaviors of the minimum closeness and the average eccentricity.

the metrics on a small portion of data, we should use the data collection procedure similar to *hop sampling* instead of *path sampling*. The only exceptional case is the hop diameter. This observation is actually inconsistent with the data collection procedure, where networks grow by adding paths between two traceroute servers. We conclude that collecting data by crossing different traceroute servers is not efficient and the exact value of a metric cannot be obtained on these sampled data.

Secondly, we see that most useful metrics in terms of computational efficiency are the average node degree and the effective hop diameter, which are (0.21,0.10) and (0.40,0.10)-evasive. Thus we can say that these metrics are superior to the others. Similarly, we can say that all metrics are superior to the maximum betweenness since it is (0.95,0.10)-evasive for both methods and it does not provide any information of its real value unless almost all of the graph is visited.

Finally, we observe that subnetworks created by different sampling methods show different properties. The graph formed by *path sampling* is initially sparser and contains longer paths and it becomes denser and contains shorter paths as the data size increases. However, the graph formed by *hop sampling* begins with a denser graph and has shorter paths and it becomes sparser and contains longer paths as it grows. In Table 1 we show a summary and comparison of the metrics and the sample methods.

5 Comparison of Generators

In this section, we examine the evasiveness properties on different data generated by different models and compare them with those of the "actual" Internet. We use three different topologies to model the power law data generators. The first one is the first explained model in Subsection 2.1.3. The out-degree exponent is taken as -2.48, which is stated as the exponent for router level data in [6]. In this paper, we call this model as *Power* model. The second one is Inet 2.0 [9] that generates data at inter-domain level.

BRITE topology generator [11] provides us to generate data based on Barabasi-Albert model as well as to model Waxman topology. We use default values set by BRITE in our data generation. For Barabasi-Albert model, incremental growth type is set and two is selected as the number of links added per new node. For Waxman model, BRITE topology generator selects the parameters as $\alpha = 0.15$, $\beta = 0.2$. For both model, router level data generation option is selected.

We compare these models and the "actual" Internet data regarding evasiveness. We generate 50 different graphs with node count 6K. For each graph, we run *hop sampling* method three times to create subnetworks. The hop count values, at which the whole graphs are covered are given in Table 2. In this table, it is obvious that the maximum hop count value for the Internet data differs from those of the model generators.

In Figures 7 and 8, the changing behaviors of the average degree and the clustering coefficients are compared for different models. In these figures, we see that the average degree and the clustering coefficient D for data generators are at least (0.90,0.10)-evasive. However these metrics are (0.21,0.10) and (0.57,0.10)-evasive on the "actual" Internet data. Therefore the average degree and the clustering coefficient D can be used to distinguish the data generators from the "actual" Internet data. Moreover it is not appropriate to compute the clustering coefficient $C^{(2)}$, since neither it distinguishes any graph among the others nor it converges its real value when a small portion of data is used.

Actual	20
Power	10
Barabasi	7
Waxman	8
Inet	5

Table 2: The hop count values, at which subnetworks nearly converge to their original networks

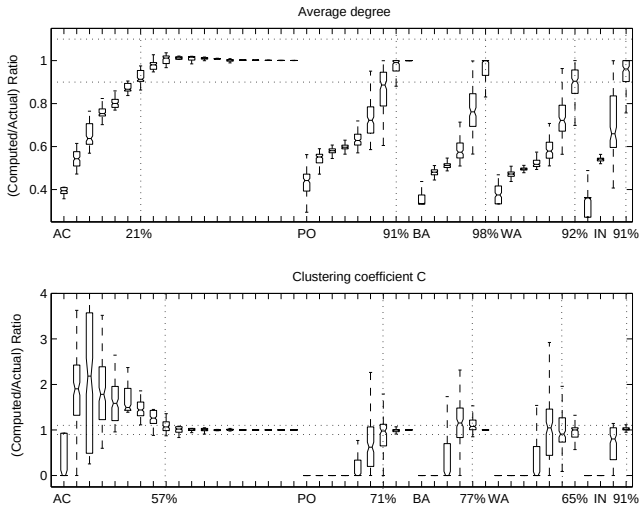

Figure 7: The changing behaviors of the average degree and the clustering coefficients C on different models.

In Figure 9, we see that although the hop diameter cannot be approximated using a small portion of data for both *Power* and *Inet* models, it is nearly (0.30,0.10)-evasive for *Barabasi* and *Waxman* models. Thus latter models are distinguished regarding to their γ values for the hop diameter.

In Figure 10, we see that effective hop diameters

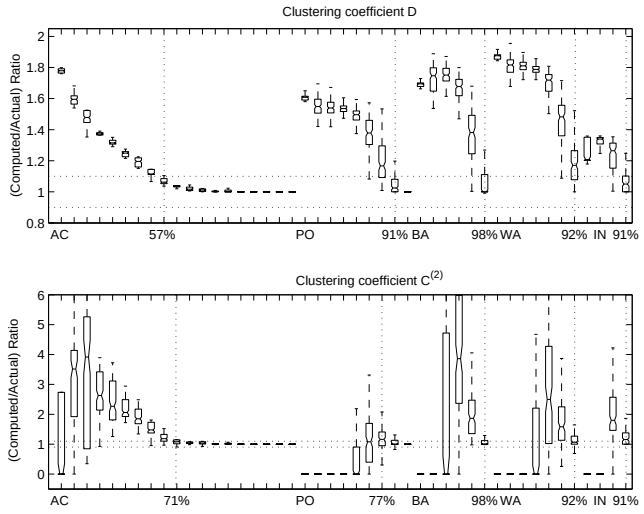

Figure 8: The changing behaviors of the clustering coefficients D and $C^{(2)}$ on different models.

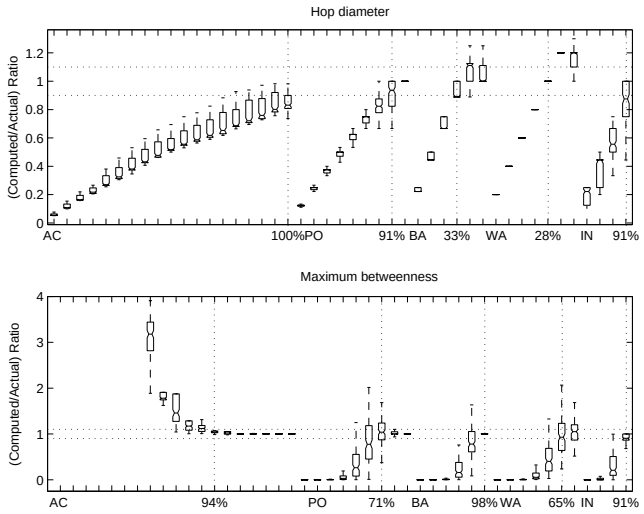

Figure 9: The changing behaviors of the hop diameter and the maximum betweenness on different models.

for both the Internet and all the models are at most $(0.50, 0.10)$-evasive. Thus it is possible to compute this metric effectively for all models. Moreover, the effective hop diameters are (0.02,0.10) and (0.05,0.10)-evasive for *Power* and *Waxman* models and it can be used to distinguish these metrics from the others.

In Figure 10, we see that the hop plot exponent and the effective hop diameter are (0.49,0.10)-evasive for *Inet* graphs, where all other metrics for *Inet* graphs are (0.91,0.10)-evasive. Remember that *Inet* model differs in all other models such that it generates data at inter-router level, where all others produces router level data. The hop plot exponent and the effective hop diameter for this model has smaller γ values like the other models regardless of this difference.

In Figures 11 and 12, metrics related to the closeness property and the average eccentricity are examined. We see that the *Inet* model can be distinguished among others regarding to all these metrics. The other generation models are distinguished from the "actual" Internet data in terms of γ values, the metrics for these models converge more quickly than those for the "actual" Internet data. Although the actual data show very similar curves for all these four metrics, the generated models have similar values for only the characteristic and the average path length (the median and the mean of the closeness values). For *Power*, *Barabasi*, and *Waxman* models, the minimum closeness and the average eccentricity graphs are different for the ones shown in Figure 11. In Table 3 we provide a summary of the comparision.

6 Greedy Algorithm

As the results in Section 4.4 show, some metrics can be estimated by using only small portion of the network if *hop sampling* is used to form the subgraphs. This is motivated us to implement an algorithm that automatically decides the minimum size of the data, on which a given metric is computed accurately within some error margin.

Initially, our algorithm randomly selects an initial node N_0 as a starting point. In each iteration, *hop sampling* is performed, (i.e., the subgraph is expanded by taking all nodes within the distance of specified hop count from the starting point N_0).

At time t, our algorithm computes a metric M_t on the subgraph and decides to stop according to the absolute difference between the current value of a metric, M_t, and the average of its previous K values, $M_{t-K} \cdots M_{t-1}$. The difference D_t is given in the following equation:

Actual (Measured)	Power Law	BRITE(Barabasi-Albert)	BRITE(Waxman)	Inet
Average degree (21%)	91%	98%	92%	91%
Effective hop diameter (40%)	5%	33%	2%	49%
Clustering coefficient C (57%)	71%	77%	65%	91%
Clustering coefficient D (57%)	91%	98%	92%	91%
Hop plot exponent (57%)	71%	33%	65%	49%
Characteristic path length (57%)	16%	33%	8%	91%
Average path length (57%)	16%	33%	8%	91%
Minimum closeness (57%)	41%	33%	27%	91%
Average eccentricity (57%)	5%	7%	8%	91%
Clustering coefficient $C^{(2)}$ (71%)	77%	98%	92%	91%
Maximum betweenness (94%)	71%	98%	65%	91 %
Hop diameter (100%)	91%	33%	28%	91%

Table 3: Summary of comparison of Internet topology generators to actual Internet topology with respect to evasiveness. The numbers indicate the average γ values (computed over 200 runs) for computing each metric with $\sigma = 0.10$.

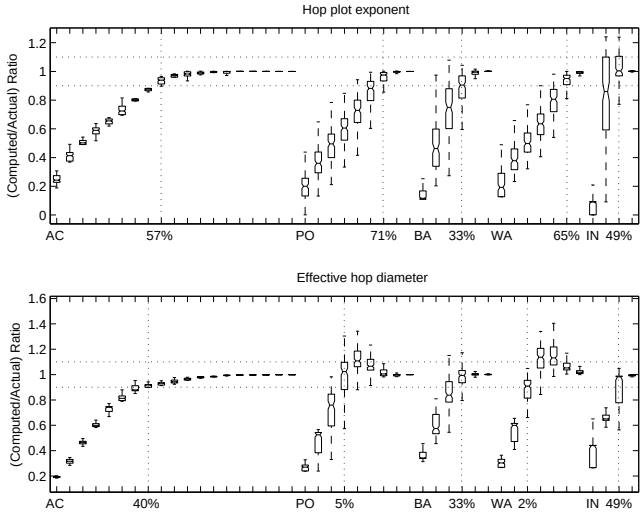

Figure 10: The changing behaviors of the hop plot exponent and effective hop diameter on different models.

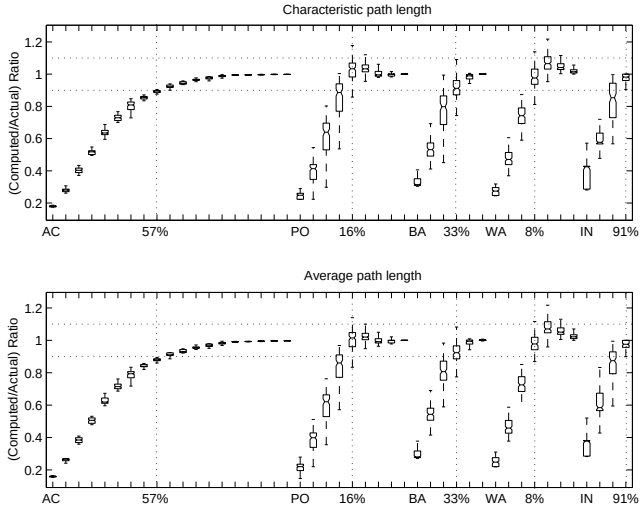

Figure 11: The changing behaviors of the characteristic and the average path lengths on different models.

$$D_t = \left| M_t - \frac{\sum_{i=1}^{K} M_{t-i}}{K} \right| \tag{7}$$

where K is the user parameter. Our algorithm continues iteratively and stops if

$$\frac{D_t}{M_t} < T \tag{8}$$

where T is the threshold parameter. When our algorithm stops, the final subgraph is assigned as the sufficient data to compute the metric M accurately. Note that increasing T reduces the data size, at which our algorithm stops.

6.1 Results of Greedy Algorithm

We apply our greedy algorithm on data obtained by *hop sampling*, but note that our proposed algorithm can be also applied on different sampling algorithms as well as synthetic data generators. To test our algorithm we create ten different subnet-

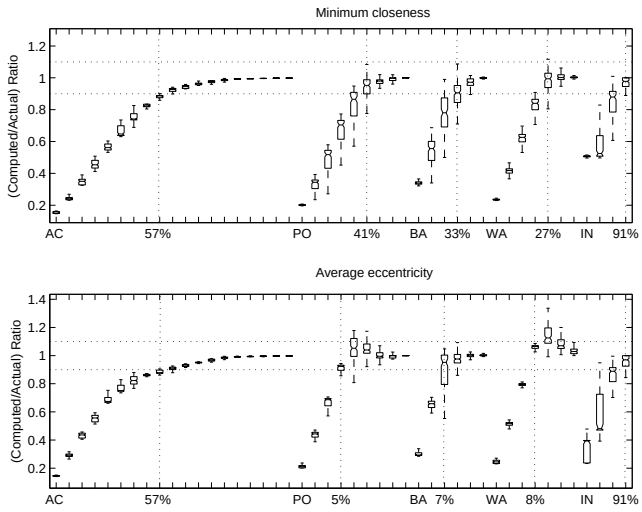

Figure 12: The changing behaviors of the minimum closeness and the average eccentricity on different models.

works by using *hop sampling* for each graph and our results are given as the averages over them. We run our algorithms for different parameters, K and T, and the results for each metric are given in Table 4. In this table, *computed/actual* is the ratio of the estimated and the exact metric values and *nodes*% is the percentage value, where our algorithm stops and reports the estimated metric value. In this table, we see that decreasing the threshold parameter T results in more accurate but costlier results. We observe that our algorithm stops in appropriate points such that the results are similar to our observations in Section 4.4. The only exception is the maximum betweenness, our algorithm stops earlier although the value of this metric does not converge. From Table 4, we see that the average degree can be estimated with 86 per cent accuracy only by using 23 percent of data and it is very efficient to use this metric. If we want to use at most 50 per cent of data to compute a metric, even the most clustering coefficient gives at most 70 per cent accuracy (when $K = 2$ and $T = 0.01$). When our algorithm determines to stop for the hop diameter metric, nearly 90 per cent data are covered, which is similar to our observations. For the other metrics based on the shortest paths between nodes, our algorithm stops when 40-50 nodes are visited and metrics are computed with 80-90 per cent accuracy. Note that the results show large variability. This means that the selection of the initial point is very important and this is consistent with our observation in Section 4.4.

7 Conclusion

For analyzing the Internet and comparing it with different models, it is important to define proper metrics. The metrics converge to their actual values as the data size increases. But it is not possible to collect the whole Internet data and the cost of the computation becomes high, increasing the data size. Thus the exact values of the metrics must be estimated by using only a small portion of data. In this work, we define (γ, σ)-evasiveness to assess whether it is possible to estimate the real value of a metric in $\pm\sigma$ error margin or not. The γ value indicate the percentage of data and it gives us a measure to compare different models.

- Our experiments on the "actual" Internet data show that sampling method is very important in approximating the values of metrics with a small portion of data. We propose to use *hop sampling* method in data collection. We also see that the most appropriate metrics in terms of computational efficiency are the ones that converge their exact values quickly, in other words they have smaller γ values in (γ, σ)-evasiveness and we propose to use γ value to assess the metrics.
- In this work, we also compare different data generators, namely *Power*, *Inet*, *Barabasi*, and *Waxman* models, regarding the evasiveness property of the metrics. γ values of a metric for different models with a fixed σ value provides us to distinguish different models.
- All the observations in our experiments motivates us to implement our greedy algorithm. It determines the minimum size of the network for a specified metric by checking its rates of changes. It is seen that, it gives promising results in the average but it shows great variability and one must be aware of this variability.

References

[1] A. L. Barabasi and R. Albert, "Emergence of Scaling in Random Networks", *Science*, vol. 286, pp.

	K=2, T=0.02		K=2, T=0.01		K=3, T=0.05	
	$\frac{computed}{actual}$	nodes %	$\frac{computed}{actual}$	nodes %	$\frac{computed}{actual}$	nodes %
Average degree	0.86	0.23	0.94	0.44	0.90	0.29
Clustering coefficient C	1.10	0.74	1.06	0.86	1.11	0.71
Clustering coefficient D	1.45	0.08	1.32	0.25	1.38	0.11
Clustering coefficient $C^{(2)}$	1.04	0.85	1.05	0.91	2.42	0.71
Hop diameter	0.79	0.89	0.79	0.89	0.70	0.88
Hop plot exponent	0.80	0.41	0.90	0.70	0.84	0.44
Effective hop diameter	0.87	0.35	0.93	0.54	0.90	0.38
Maximum betweenness	6.11	0.59	3.61	0.76	6.52	0.58
Characteristic path length	0.86	0.47	0.93	0.73	0.88	0.49
Average path length	0.88	0.53	0.93	0.73	0.87	0.52
Average eccentricity	0.86	0.41	0.91	0.61	0.87	0.40
Minimum closeness	0.81	0.52	0.87	0.69	0.89	0.62

Table 4: The ratios of the estimated and exact metric values and the percentages of the nodes covered when our algorithm decides to stop.

509–512, 1999.

[2] T. Bu and D. Towsley, "On Distinguishing between Internet Power Law Topology Generators", in *Proceedings of the IEEE INFOCOM*, 2002.

[3] K. L. Calvert, M. B. Doar and E. W. Zegura, "Modeling Internet Topology", *IEEE Transactions on Communications*, vol. 35, pp. 160–163, 1997.

[4] M. Doar and I. M. Leslie, "How Bad is Naive Multicast Routing?", in *Proceedings of INFOCOM*, pp. 82–89, 1993.

[5] S. N. Dorogovtsev and J. F. F. Mendes, "Evolution of Networks", *Advances in Physics*, cond–mat/0106144, 2002.

[6] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On Power-Law Relationships of the Internet Topology", in *Proceedings of ACM SIGCOMM*, pp. 251–262, 1999.

[7] S. Floyd and V. Paxson, "Difficulties in Simulating the Internet", *IEEE/ACM Transactions on Networking*, vol. 9, pp. 392–403, 2001.

[8] R. Govindan and A. Reddy, "An Analysis of Internet Inter-Domain Topology and Route Stability", in *Proceedings of the IEEE Infocom*, pp.851–858, Kobe, Japan, April 1997.

[9] C. Jin, Q. Chen, and S. Jamin, "Inet: Internet Topology Generator", *Technical Report CSE-TR443 -00*, Department of EECS, University of Michigan, 2000.

[10] L. Lovasz and N. Young, "Lecture notes on evasiveness of graph properties", Technical Report TR 317-91, Princeton University, 1994.

[11] A. Medina, I. Matta, and J. Byers, "On the Origin of Power Laws in Internet Topologies", *ACM Computer Communications Review*, vol. 30, no. 2, pp. 18–28, 2000.

[12] M. E. J. Newman, "Who is the Best Connected Scientist? A Study of Scientific Coauthorship Networks", *Phys.Rev.*, cond–mat/0011144, 2001.

[13] J. Pansiot and D. Grad, "On Routes and Multicast Trees in the Internet", *ACM Computer Communication Review*, vol. 28, no. 1, pp. 41–50, 1998.

[14] G F. Riley and M. H. Ammar, "Simulating Large Networks - How Big is Big Enough?", At http://citeseer.nj.nec.com/495035.html.

[15] A L. Rosenberg, "On the Time Required to Recognize Properties of Graphs: a Problem,", *SIGACT News*, vol. 5, pp. 15–16, 1973.

[16] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "Network Topology Generators: Degree-Based vs. Structural.", in *Proceedings of the ACM SIGCOMM*, 2002.

[17] D. J. Watts and S. H Strogatz, "Collective Dynamics of 'Small-World' Networks", *Nature*, vol. 393, pp. 440–442, 1998.

[18] B. M. Waxman, "Routing of Multipoint Connections", *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1617–1622, 1988.

[19] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, " How to Model an Internetwork", *Proceedings of INFOCOM*, pp. 594–602, 1996.

[20] E. W. Zegura, K. L. Calvert, and M. J. Donahoo, "A Quantitative Comparison of Graph–based Models for Internet Topology", *IEEE/ACM Transactions on Networking*, vol. 5, pp. 770–783, 1997.