Reducing Large Internet Topologies for Faster Simulations

V. Krishnamurthy[1], M. Faloutsos[1], M. Chrobak[1], L. Lao[2], J-H. Cui[3], A. G. Percus[4] ⋆

U.C. Riverside[1], UCLA[2], U. Connecticut[3], Los Alamos National Labs and UCLA IPAM[4]

Abstract. In this paper, we develop methods to "sample" a small realistic graph from a large real network. Despite recent activity, the modeling and generation of realistic graphs is still not a resolved issue. All previous work has attempted to grow a graph from scratch. We address the complementary problem of shrinking a graph. In more detail, this work has three parts. First, we propose a number of reduction methods that can be categorized into three classes: (a) deletion methods, (b) contraction methods, and (c) exploration methods. We prove that some of them maintain key properties of the initial graph. We implement our methods and show that we can effectively reduce the nodes of a graph by as much as 70% while maintaining its important properties. In addition, we show that our reduced graphs compare favourably against construction-based generators. Apart from its use in simulations, the problem of graph sampling is of independent interest.

1 Introduction

Small graphs that resemble the Internet topology are needed for conducting simulations of various network protocols. Real graphs can have prohibitively large sizes, especially for highly detailed simulations such as packet level simulations. To produce high confidence results, one averages the experimental results over many graphs of a given size. Running the experiments over a range of sizes allows reseachers to interpolate the results to graph sizes outside the tested range. In particular, it shows whether the performance of the tested protocols scales well with increasing size, leading to accurate performance predictions for the Internet graphs of the future.

Currently, all known models for graph generation incrementally grow a graph with desired properties. Our work follows the opposite approach: we wish to reduce real large Internet instances to produce small realistic topologies. This task can be thought of as *graph sampling*, and it has attracted attention in other settings [18] [19].

Among the existing Internet topology generators, none has yet been widely accepted as sufficiently accurate. These generators produce arguably realistic graphs, but they do not necessarily match all the known topological properties of the Internet. Most graph generators attempt to grow a graph, an approach that we call **constructive**. This area has seen unprecedented activity since the discovery of skewed degree distributions in the Internet topology [11]. The generators either use "biased" or *preferential* growth policy [2] [3] [5] [10] [23] or force a power-law degree distribution [1] [16]. The weakness of these constructive methods lies in their dependence on the principles of construction, and the choice of parameter values. Furthermore, several of them focus on matching the degree distribution, while they often fail to match other topological properties, as multiply documented [5] [7] [15] [16] [29].

In this paper, we address the following problem: we want to "sample" a real topology[1] to produce a smaller graph. The overarching goal of our approach is very practical: we want the

⋆ This work was supported by the NSF CAREER grant ANIR 9985195, and NSF grant IDM 0208950, and DARPA award FTN F30602-01-2-0535. NSF/NeTS 0435230. and NSF CCR 0208856 and funding from the LANL LDRD and UCDRD programs.

[1] To make it more specific, the current Internet has more than 14,000 nodes. The smallest available Internet instance (from 1997) has about 3,000 nodes, but even this size is computationally expensive,

simulations on the sampled graph and the initial larger graph to lead to the same conclusions. This is a novel problem in the Internet modeling community, although some related work in other areas exists [18] [19]. We call our approach of generating a graph **reductive**. Intuitively, our approach has the easier task of not "destroying" the existing properties, in contrast to the task of the constructive approach, which has to reproduce all the right properties.

How do we evaluate the success of our approach? Establishing criteria for the realism of a generated graph is an open ended problem. In the case of graph sampling, the question is more involved: which Internet instance should the reduced graph try to match? One can distinguish two objectives: we can either try to match the properties of: (a) the real Internet instance of the same size (thus "reversing" the evolution of the Internet), or (b) the initial instance (thus producing its small imitation.) If the properties do not change with size, then both goals are equivalent. However, no obvious time independent topological metrics seem to exist [28]. The findings of [28] suggest that even though every Internet instance at the AS-level has power-law characteristics, there are variations in the value of the slope. Thus we chose the first method above, and we compare the reduced graph with an *equal size* real Internet instance.

The contribution of this paper is twofold: (i) we provide efficient graph sampling algorithms, and (ii) we compare our reduction methods against constructive methods. In addition, we compare our graphs using network protocol simulation but this study could not be included here due to space limitations. This paper significantly extends a preliminary version of this work [21], while a more detailed version appears as a technical report [20].

Graph Sampling Algorithms. As our main contribution, we develop and quantify the performance of a number of reductive methods. We group these methods into three main categories: (a) deletion methods, (b) contraction methods and (c) exploration methods. Our work yields the following results.

- Our best algorithms successfully reduce the graph size by up to 70%, in the number of nodes, while preserving the desired topological properties. Our methods are statistically robust to the initial topology and the randomization seed.
- We show analytically that some of our methods will maintain the power-law of the degree distribution, if such a distribution exists in the initial topology.

Comparison of Reductive and Constructive Methods. We compare our best reduction methods with commonly used constructive methods and find that our methods match more closely the properties of the real instances. We find that the best constructive generator is Inet [16], which takes as input the available real instances. Inet currently does not generate graphs with less than 3000 nodes. Therefore, we can confidently say that for really small graphs ($n < 3,000$), the reductive methods are the best choice.

It is worth noting that the reductive approach has two additional attractive properties:

- A "statistically fair" reduction may preserve many graph properties, including some that we have not used for our metrics, or even some properties that we have not yet identified.
- The reductive method is likely to extend to different types of graphs, for example, the policy-based Internet topology, or the Web graph.

Graph sampling can be used as a tool to provide insight into the topological properties and structure of the graph. Finally, sampling can also complement a visualization effort, when the sizes are too large for a meaningful graphical representation.

2 Background and Metrics

In this section, we introduce the topology model and several topological properties of the Internet, which we use to evaluate the realism of our graphs.

if not prohibitive, for some types of simulations such as BGP simulations or flow level simulations [8] [27] [14].

2.1 Internet Instances

The Internet is divided into autonomously administered domains or Autonomous Systems (AS). In our study, we focus on the AS level topology, and we model the Internet as an undirected graph whose nodes are AS's and whose edges are inter-domain connections. This has been a standard approach in the past literature and there continues to be significant activity in measuring and modeling the AS level Internet as an undirected graph. It is worth noting, however, that, more recently, there have been efforts [13] [22] [4] to model the Internet as a directed graph by including business relationships.

Our real data come from the Oregon Routeviews project [12]. This is the frequently used archival data by researchers in this area and the only data archive that has instances spanning over 5 years. This data is specifically chosen for our study as we need a wide range in the size of the Internet topology. This was the reason why we could not use the [7] archive which spanned only over three months. Each instance in this paper is named using its collection date, in the format IYYMMDD. For example, the instance collected on May 07, 2001 is named I010507. We use real Internet instances [12] from November 1997 to March 2003 in our experiments.

2.2 Graph Properties

Several graph properties have been proposed to capture the characteristics of real Internet graphs [11] [15] [26], and we adopt most of these metrics in our study. Needless to say, preserving these properties is necessary for the generated graphs to resemble the Internet topologies, but it may not be sufficient, for these topologies may share other properties that have not yet been discovered.

Average and Standard Deviation of Degree. The average degree of a graph is equal to $2m/n$, where m is the number of links and n is the number of nodes. The average degree represents the density of a graph. The average degree of Internet topologies is known to increase over time, as the size of the graph also increases. Another measure we examine is the standard deviation of the degree distribution, which can be thought of as a measure of the "diversity" of the nodes in the network.

Degree Distribution. It has been shown that power laws approximate well[2] the skewed degree distribution [11]. Here, we focus on power law 1, the *degree rank exponent* and power law 2, the *degree exponent*. Degree rank exponent is defined as the slope of log-log plot of the nodes' degrees versus their rank, where the k-th ranked node is the one with the k-th highest degree. Degree exponent[3] is the slope of the log-log plot of the degree frequency versus degree. The two power laws can be shown to be, in fact, equivalent. In practive, however, the actual degree distributions follow these power laws only approximately, and studying both laws provides slightly different approximate views of the real degree distribution[4]. In this metric, we check the existence of power-laws and then compare the value of the exponent of the power laws [24]. Power laws are approximations whose accuracy is typically quantified by the correlation coefficient. In addition, power laws often target the tail of a distribution, which is the focus of our analytical work.

Spectral Analysis. Gkantsidis *et al.* [15] characterize the clustering and spatial properties of a topology using spectral analysis of the adjacency matrix of a graph. Spectral analysis captures significant information about the clustering properties of the topology in a unique way. It subsumes the clustering coefficient metric that was used before [5].

[2] Chen *et al.* [7] created a more complete Internet graph at the BGP level, but recent work by Siganos *et al.* [28] shows that the power laws hold with 99% correlation coeffi cient even in this new graph.

[3] We use the reverse cumulative distribution function (RCDF) of power law 2, which is more robust than the cumulative distribution function (CDF)[28].

[4] The correlation coeffi cient of the power law fi t was verifi ed by the authors of [7], who use more metrics to examine the goodness of the fi t.

In more detail, spectral analysis examines the eigenvectors corresponding to the largest eigenvalues of the normalised transposed adjacency matrix of an entire topology. These vectors correspond loosely to the eigenvectors of the main clusters in the topology. The resulting plot depicts the 100 largest eigenvalues in order of magnitute, from largest to smallest. It is found that the clustering properties (the corresponding plot) have not changed significantly despite the Internet growth [15].

C. Graph Generators. Early graph generators failed to match the skewed degree distribution [6] [9] [31] [32]. Several recent generators build topologies with power-law degree distribution in mind [1] [3] [5] [16]. It is worth mentioning that the pioneering Barabasi-Albert model [3] generates a graph through preferential attachment: in attaching new nodes to existing ones, it favors high-degree nodes. Mitzenmacher provides an overview of methods to generate power law distributions [25].

To illustrate that the reduction methods generate graphs which resemble Internet topology better than the constructive methods, we compare the topology obtained by reducing the AS level Internet topology using our best reduction method with similar graphs generated by Inet [16], Waxman [30], Barabasi-Albert [3], and the modified GLP heuristic [5].

Finally, the problem of graph reduction and sampling appears in other disciplines, often with different goals. For example, graph sampling has been used in graph partitioning in the context of distributed computing [18] [19], and randomized graph sampling has been used to solve different graph problems, such as min-cut approximation [17]. To our knowledge, however, prior to our work, no research has been reported on using graph sampling for generating realistic network topologies.

3 Graph Reduction Methods

This section presents our approach for reducing a real AS level Internet topology to a smaller realistic topology. Our methods fall into three categories: (a) *deletion methods*, that remove edges or nodes from the graph, one by one, until a desired size is reached, (b) *contraction methods*, that contract adjacent nodes, step by step, until a desired size is reached, (c) *exploration methods*, that traverse a desired number of nodes according to a given exploration policy, and retain the subgraph induced by those nodes. For consistency of notation, we abbreviate the methods starting with the letter that indicates the category they belong to: D for deletion, C for contraction, and E for exploration.

3.1 Deletion Methods

Our deletion methods are embedded in the following framework: The input consists of an initial graph G with n nodes and m edges, and the total percentage P of nodes to be deleted. The graph is reduced iteratively in stages, where at each stage a small percentage s of nodes is removed (where s is a parameter that can be set by the user.) A stage consists of several steps, in which we remove either one edge or one vertex selected according to the specific method. After each stage, connected components are found and the largest connected component is retained. The procedure stops when the reduced graph has approximately $n(1 - P/100)$ nodes. By reducing a small percentage s of the graph in each iteration, we are able to meet the target size more accurately. In practice, a reduction of 3% to 5% of the nodes at each stage was sufficient to achieve the desired reduction. (The partition into stages was introduced for efficiency, for maintaning connected components incrementally, under node or edge deletion operations, is either very slow or cumbersome to implement.) The deletion methods we study are:

Deletion of Random Vertex (DRV): Remove a random vertex, each with the same probability.

Deletion of Random Edge (DRE): Remove a random edge, each with the same probability.

Deletion of Random Vertex/Edge (DRVE): Select a vertex uniformly at random, and then delete an edge chosen uniformly at random from the edges incident on this vertex.

Hybrid of DRVE and DRE (DHYB-w): In this method, with probability w we execute DRVE and with probability $(1 - w)$ we execute DRE. In particular, DHYB-1 is DRVE, and DHYB-0 is DRE. (This method was motivated by our initial studies showing that DRVE and DRE had opposite performances with respect to different metrics, namely when one of them underestimated a metric's target value then the other overestimated it.) We consider nine values of w in our experiments, ranging from 0 to 1.0 in increments of 0.1. For clarity, we show only a subset of those here.

3.2 Contraction Methods

These methods proceed by contracting adjacent nodes. The two methods below differ in the manner their connecting edge is chosen.

Contraction of Random Edge (CRE): Pick a random edge, uniformly, and contract its endpoints. The neighbors of the merged nodes become neighbors of the new node. (This method bears some similarities to the random matching method [19] and the edge coarsening method [18]. We also considered a generalization of the CRE method, where more neighbors contract all at once, but the results were not satisfactory and are not shown here.)

Contraction of Random Vertex/Edge (CRVE): Pick a random vertex, uniformly, and contract it with a uniformly-chosen random neighbor.

3.3 Exploration Methods

Here, we pick an initial node randomly, traverse the graph according to a given exploration method, until a desired number of nodes is visited. We then retain the subgraph induced by these nodes: all nodes that have been visited and the edges between them are retained in the final graph. We study two ways to explore a graph:

Exploration by Breadth First Search (EBFS): Randomly select a start node, and then do breadth-first search starting from that node, until the desired number of nodes have been visited.

Exploration by Depth First Search (EDFS): Randomly select a start node, and then do a depth-first search starting from this node (following a random yet non-traversed edge at each forward step), until the desired number of nodes have been visited.

4 Analysis

In this section, we prove that two of our reduction methods, DRE and DRV, preserve the degree power-law. More specifically, we show that if an original graph satisfies the power law, then the reduced graph satisfies it too, with the same exponent, for large degrees.

Let G denote the original graph with n vertices and m edges. By n_d we denote the number of nodes of degree d, and by d_{ave} the average degree. These quantities are related to each other by $n = \sum_d n_d$, $m = \frac{1}{2}\sum_d d n_d$, and $d_{\text{ave}} = 2m/n$.

The symbols n', m', n'_d, d'_{ave} denote the corresponding values in the reduced graph G'. Since our reduction methods are probabilistic, these symbols actually represent expected values of the corresponding random variables.

We assume that the degree sequence of G satisfies the power law in the following form: $n_d = C n d^{-\alpha}$, where $C = (\sum_{d=1}^{n} d^{-\alpha})^{-1}$ and α is the degree exponent. We wish to show that a similar property holds (approximately) in G'.

DRE and Power Law Preservation. The DRE method, as implemented in our experiments, removes edges at random, one at a time, and retains the largest connected component. This process, in its raw form, is not amenable to analytical studies, as the degree distribution of the eliminated nodes depends heavily on (unknown) topological properties of G. To facilitate the analysis, we will approximate DRE by another process that is easier to analyze. This approximation will proceed in several steps.

First, we ignore the fact that DRE removes the nodes outside the largest connected component, and study instead the degree distribution among *all* the vertices of G, after the edges are deleted. Thus, throughout this section, G' has the same vertex set as G and $n' = n$. This simplification is justified by experimental results showing that the nodes eliminated by DRE have very low degree, so this simplification should not affect the asymptotic behavior of the degree distribution.

Let $p = m'/m$. We can think of p as a probability of an edge being retained in the graph. Thus, in the second approximation, instead of removing edges one by one, we will flip a coin for each edge, independently, and remove each with probability $q = 1 - p$. Although this does not guarantee that the resulting graph will have exactly m' edges, its expected number of edges is very heavily concentrated around m', and the two processes have asymptotically the same behaviors.

Informal argument. Our goal is to show that in G' the degrees satisfy $n'_d = C'n'd^{-\alpha}$, for some constant C'. The general idea of our proof is summarized as follows. Roughly speaking, nodes in G with degree between $(d-\frac{1}{2})/p$ and $(d+\frac{1}{2})/p$ end up in G' with an expected degree of d. Since this range covers $1/p$ different degrees, and for degrees c close to d the values n_c are close to n_d, we might anticipate that for d not too small, we should get $n'_d \sim \frac{1}{p}n_{d/p} = Cp^{\alpha-1}n'd^{-\alpha}$, preserving the power law with the same exponent.

Theorem 1. *For any fixed exponent $\alpha > 1$ and probability $p \in (0, 1)$, given a graph with degree distribution given by $n_d = Cnd^{-\alpha}$, the degree distribution of the graph reduced through the process above will approximately follow a power-law: $n'_d \approx Cp^{\alpha-1}d^{-\alpha}$.*

The formal statement of the above theorem and its proof are omitted due to space limitations.

DRV and Power Law Preservation. An analogous argument as for DRE can also be applied to DRV. We only outline an informal explanation here: Let $n' = pn$. We can think about DRV as removing each vertex in G, independently, with probability $q = 1 - p$. Then, roughly speaking, a fraction p of the nodes with degree between $(d - \frac{1}{2})/p$ and $(d + \frac{1}{2})/p$ end up in G' with an expected degree of d. Other nodes are either deleted or their new degrees are not d. Since this range covers $1/p$ different degrees, and for degrees c close to d, the values n_c are close to n_d, as long as d is large enough, we should get $n'_d \sim n_{d/p} = Cp^{\alpha-1}n'd^{-\alpha}$, preserving the power law with the same exponent.

5 Graph Reduction Evaluation

We examine the performance of our sampling methods in practice. The starting point of the reduction in most of the experiments in this paper is the AS level Internet topology I010507 collected on 07/05/2001. (However, we have experimented with other topologies with similar results.) The I010507 graph has 10,966 nodes and 22,536 edges, thus an average degree of 4.11. The "Internet curve" shown in all the graphs represents the value of actual Internet instances of size corresponding to the value on the x-axis. If we reduce the I010507 graph by 70% we end up with about 3,300 nodes which is roughly the same size as the I980124 graph. Each data point in our plots represents the *average of 50 runs* with different random seed.

Our experimental results show that among all the methods DHYB-0.8 seems to have the least deviation from the Internet's topological properties. Thus it is the best method with respect

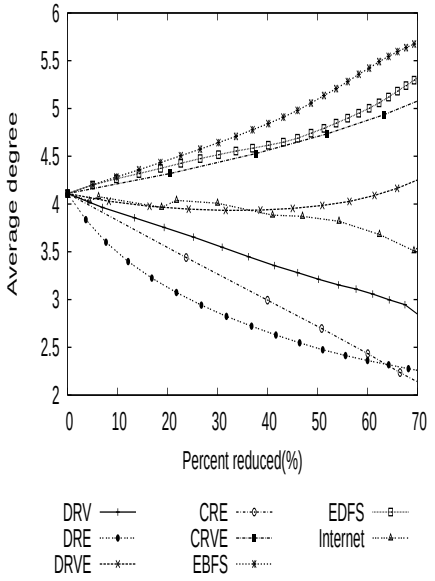

Fig. 1. Average degree comparison of Deletion, Contraction and Exploration Methods.

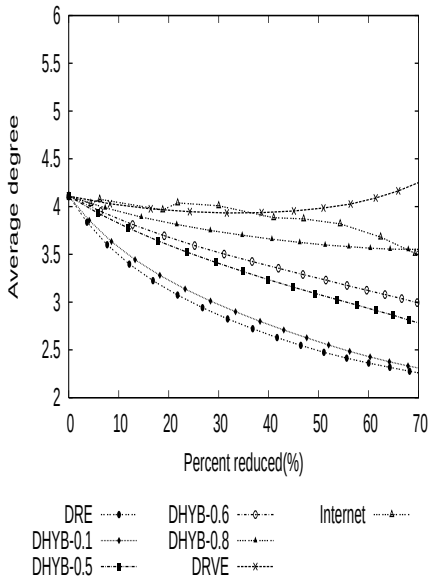

Fig. 2. Average degree comparison of Hybrid Methods.

to topological metrics described in Section II. Among the non-hybrid methods, DRV performs well. Recall that the DHYB method combines edge deletions DRE and DRVE. It is interesting to see how well the random node removal DRV worked in practice. In Table 1 we present the top performing methods according to some metrics. Variations of DHYB and DRV are consistently present in every column. We have not shown the performance of all the DHYB methods here as the graph becomes very congested. As the value of w in DHYB is increased, the metric values increased between DRE and DRVE.

Test 1: Average Degree and its Deviation. *DHYB-0.8 has the best performance.* Figure 1 shows how the average degree varies for the deletion, contraction and exploration methods. Figure 2 shows the average degree of the hybrid methods with $w = 0.1, 0.5, 0.6, 0.8$. DRVE follows the evolution of the average degree fairly closely up to 50% reductions, but then it diverges quickly. DHYB-0.8 stays close to the Internet curve in the whole range, and it has nearly the same value of average degree at the 70% reduction point.

We also found that DHYB-0.8 is better in terms of the average deviation, with an average percentage deviation of 4.2%, followed by DRVE with 5%. These methods are followed by DHYB-0.6 and DRV with average percent deviations of 11% and 12.2% respectively. We have selected only methods whose average degree decreased under graph reduction, mirroring the trend in the real Internet data observed in Figures 1 and 2. With the exception of one data point, the Internet's average degree constantly decreased with decreasing size. All the other methods are farther away from the Internet, so we conclude that they do not fare well in this metric comparison.

Test 2: Exponent of Rank Power Law. *DHYB-0.6 is the best method.* The hybrid methods follow the variation of the Internet rank exponent more closely than the other methods, as is evident in Figures 3 and 4. In fact, the average percent deviations was within 3% for DHYB-0.6, DHYB-0.5 and DHYB-0.8. DHYB-0.8 and DHYB-0.5 are the second best methods after DHYB-0.6 which lie above and below the Internet line. DRV was quite close, with an average percentage deviation of 5.2%.

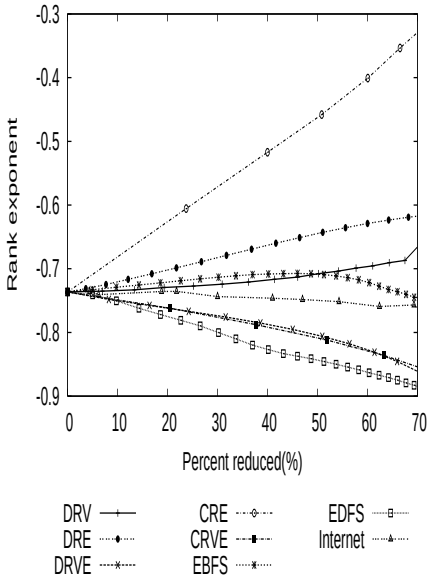

Fig. 3. Rank exponent comparison of Deletion, Contraction and Exploration Methods.

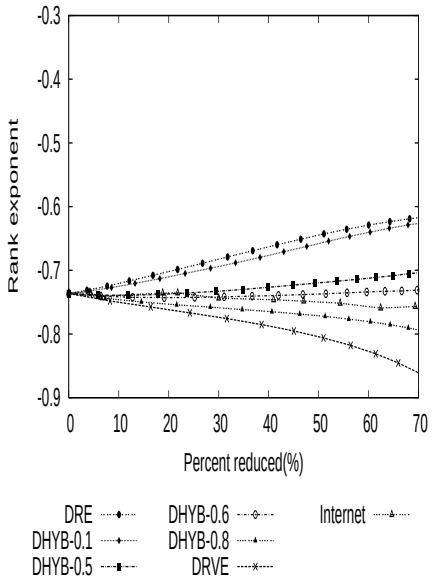

Fig. 4. Rank exponent comparison of Hybrid Methods.

Test 3: Correlation Coefficient of Rank Power Law. *DHYB-0.5, 0.6, 0.8 and DRV consistently maintained a high correlation coefficient*. In addition to having an exponent value closer to that of the Internet, the methods should also have a high correlation coefficient, preferably above 97%. Even though it looks like EBFS performs equally well as DHYB-0.6, it has a smaller correlation coefficient (below 96%). A similar trend is seen in CRVE which follows DRVE very closely. The other methods have correlation coefficient above 96% except CRE whose correlation coefficient drops steadily from 90% (at 25% reduction) to 61% (at 70% reduction). Even though we include EBFS, CRVE and CRE in Figure 3 for degree exponent comparison, we exclude them from being viable solutions at this point.

Test 4: Exponent of Degree Power Law. *DHYB, DRV, and DRE are successful in this test:* their degree exponent is within 5.5% from the the exponent of the Internet instance. Among the hybrid methods, DHYB-0.5, 0.6 and 0.8 perform well, having a value within or close to 5% of the exponent of the Internet topologies. (Figures not included due to space limitations.)

Test 5: Correlation Coefficient of Degree Power Law. *The correlation coefficient of the best methods namely DHYB-0.1,0.5,0.6,0.8, DRE and DRV are above 97% in all the cases.*

Range	**Average Degree**	**Rank Exponent**	**Degree Exponent**
Best group	DRVE,DHYB-0.8	DHYB-0.1,0.6,0.5,0.8	DHYB-0.1,DRE, DHYB-0.5
Second group	DRV,DHYB-0.6,0.5	DRV	DHYB-0.6,0.8,DRV

Table 1. The best methods, based on average percent deviation from the target value. For average degree, “best” is within 5% and “second best” within 5-15%. For the other two metrics, “best” is within 3%, and “second best” within 3-5.5%.

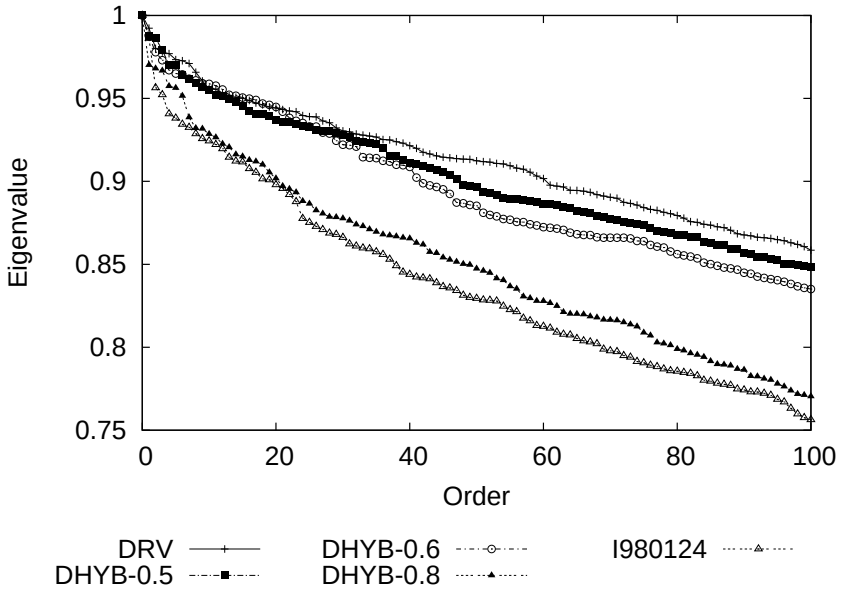

Fig. 5. Spectral Analysis of 70% reduced DHYB-0.5, 0.6, 0.8, DRV, and Internet instance I980124.

We evaluate the methods based on their average percent deviations from the Internet with respect to the four metrics we examined so far. From Table 1, we conclude that DHYB-0.8, 0.6, 0.5 and DRV are the leading methods, and from now on we will use only those four methods in the remaining experiments.

In the following two tests, we need to generate a plot for every topology, unlike the previous tests where we had a single value corresponding to a topology. Thus we chose to show only the 70% reduction point; we had similar results for the other reduction points also. We could maintain successfully the above mentioned topological properties up to the 70% reduction point using our methods: DHYB-0.5, 0.6, 0.8 and DRV. For the hop-plot and spectral analysis test, we reduce the I010507 topology by 70% using DHYB-0.5, 0.6, 0.8 and DRV. The reduced graph now has about 3290 nodes and its performance is compared with the I980124 Internet topology having 3291 nodes.

Test 6: Spectral Analysis. *DHYB-0.8 gives best results.* The spectral analysis results of DHYB-0.8, 0.6,0.5 and DRV (the selected best methods) are shown in Figure 5. Recall that the spectral behavior of the Internet topology is consistent over time [15]. So we have reason to believe that the reduction method whose spectral behavior matches I980124 is the best method. As we can see DHYB-0.8 follows the I980124 topology closely, outperforming the other methods.

Considering the results of all tests above, the best of the methods we tested is DHYB-0.8, as it maintains an average percent deviation from the target values close to or below 6% with respect to the four topological metrics, and in several tests it outperforms all other methods. Among the non-hybrid methods, DRV seems to the best method maintaining an average percent deviation close to or below 5% for the power-law metrics and within 15% for average degree.

Robustness to Input Internet Instance. *All the methods are insensitive to the initial instance.* We further investigated the stability of each method with respect to the input Internet instance. We tested our seven reduction methods on the most recent AS level Internet topology I030313 with 15,026 nodes and 31,200 edges. The results were very similar to those reported for the instance I010507. Similar results were also obtained for other Internet instances.

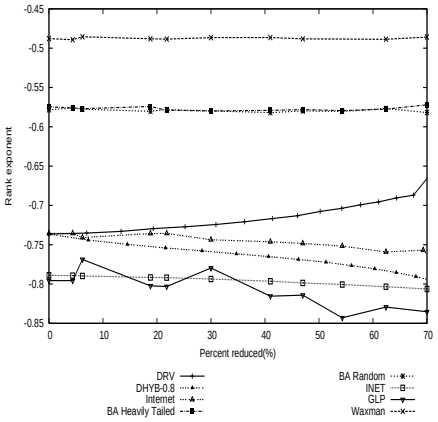

Fig. 6. Rank exponent comparison of Reductive and Constructive Methods.

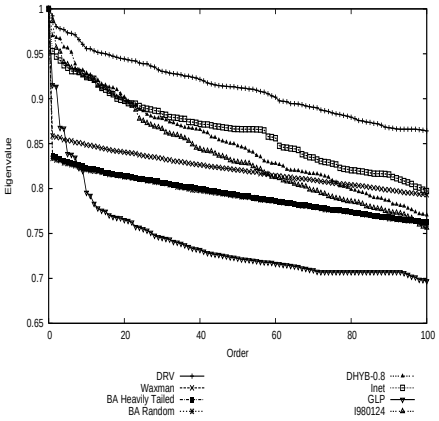

Fig. 7. Spectral Analysis of Reductive and Constructive Methods.

6 Reductive Methods versus Constructive Methods

We compare our reduction methods with existing well-known constructive generator: Inet [16], Waxman [30], BA (Barabasi-Albert) [3], and GLP (Generalized Linear Preference) [5]. Using the same metrics as in Section 5, we compare the topologies reduced by our best reduction methods, namely DHYB-0.8 and DRV (starting from instance I010507) with topologies from these other topology generators. For brevity, we show results only for two selected metrics.

Test 8: Average Degree. *Inet follows closely the variations in the Internet's average degree.* The behavior of Inet is not surprising as this generator predicts the average degree using real Internet instances from the same data archive [12] that we use, and forces this degree distribution. DHYB-0.8 is the next best method. DRV doesn't follow the variations in the Internet but decreases in value linearly, unlike GLP which varies haphazardly with no specific pattern. The BA and Waxman generators produce topologies with an average degree of 4 independent of the size of the graph.

Test 9: Exponent of Rank Power Law. *DHYB-0.8 is the best method.* The hybrid method follows the variation of the Internet rank exponent very closely as is evident in Figure 6. We recall from the previous section that the average percent deviation of DHYB-0.8 with respect to this metric was within 3%. Inet maintains a constant value for the exponent irrespective of the size of the graph. DRV has values higher than the Internet and is the next best method. In the BA generator, both the node placement options (random and heavily tailed) generate topologies with similar values. Similar to the previous test, the exponent value is independent of the size for both Waxman and BA topologies.

Test 10: Exponent of Degree Power Law. *DYB-0.8 and DRV seem to be best.* Most of the methods perform well, except for those by Waxman and BA. In particular, DHYB-0.8 and DRV have values close to the Internet. They are followed by GLP and Inet, which fall above and below the Internet respectively.

Test 11: Spectral Analysis. *DHYB-0.8 seems to be the best.* Synthetic generators like BA and GLP have not only smaller eigenvalues compared to the Internet but also a slope value that is very different from the AS level Internet topology [15]. Gkantsidis *et al.* [15] claim that these generators fail to reproduce the strong clusters that are present in the Internet. On the other hand, our methods DHYB-0.8 and DRV have a higher eigenvalue and a distribution very similar to the Internet (Figure 7). The eigenvalues of Inet doesn't decrease gradually unlike the Internet but instead exhibits sharper trends.

Summary. We find that DHYB-0.8 is the best method followed by Inet and DRV. Inet, however, does not generate graphs below 3000 nodes. (This could be related to the fact that Inet

uses the available instances from the RouteViews archive [12] in order to calibrate its intended graph metrics, and the smallest instance (collected on 15th Nov 1997) in the archive has 3,037 nodes.) Given this restriction, we believe that DHYB-0.8 is the best choice for small graphs ($nodes < 3,000$). We used such small topologies, including the I980124 graph reduced to 1500 nodes, to evaluate our generation techniques in multicast simulations. We show that even using such small graphs, we can obtain realistic simulation conclusions. (Results not included due to space limitations.)

7 Conclusion

The goal of this paper has been to propose and study methods for sampling Internet-like graphs. We propose and evaluate the performance of three types of reduction methods with multiple methods of each type. Our work leads to the following conclusions.

How can I sample a real network? We conclude from our experiments that DHYB-0.8 is the best among our methods for the Internet sampling, and that it also compares favorably to graph generation methods proposed previously in the literature. DRV is nearly as good, which, given DRV's simplicty, is an interesting result in its own right.

How much can I reduce a real network? We are able to reduce a graph successfully by approximately 70% in terms of the number of nodes. Beyond 70% we often find that the statistical confidence coefficient is low.

Provable reduction performance. We show analytically that DRV and DRE respect an initial power-law degree distribution.

Simulation speedup. The speedup depends on the complexity of simulations. Given a 70% reduction in size, an $O(n^2)$ or $O(n^3)$ simulation will decrease by a factor of about 11 or 37, respectively. Furthermore, smaller graphs will require less memory which can decrease the simulation time further.

References

1. W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. *STOC*, 2000.
2. R. Albert and A. Barabasi. Statistical mechanics of complex networks. *Review of Modern Physics*, 2002.
3. A. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 8, October 1999.
4. G. Battista, M. Patrignani, and M. Pizzonia. Computing the types of the relationships between autonomous systems. *IEEE INFOCOM*, 2003.
5. T. Bu and D. Towsley. On distinguishing between Internet power law topology generators. *Infocom*, 2002.
6. K. Calvert, E. Zegura, and M. Doar. Modeling Internet topology. *IEEE Trans on Communication*, pages 160–163, December 1997.
7. Q. Chen, H. Chang, R. Govindan, S. Jamin, S. J. Shenker, and W. Willinger. The origin of power laws in Internet topologies revisited. *INFOCOM*, 2002.
8. X. A. Dimitropoulos and G. F. Riley. Creating realistic BGP models. *11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 2003.
9. M. Doar. A better model for generating test networks. *Proc. Global Internet, IEEE*, Nov. 1996.
10. A. Fabrikant, E. Koutsoupias, and C.H.Papadimitriou. Heuristically optimized trade-offs: A new paradigm for power laws in the internet (extended abstract). *STOC*, 2002.
11. M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. *ACM SIGCOMM*, pages 251–262, Sep 1-3, Cambridge MA, 1999.
12. National Laboratory for Applied Network Research. Online data and reports. Supported by NSF, http://www.nlanr.net, 1998.
13. Lixin Gao. On inferring automonous system relationships in the Internet. *In Proc. Global Internet*, November 2000.

14. G.F.Riley. On standardized network topologies for network research. *Simulation Conference, 2002. Proceedings of the Winter*, 1:664 –670, 2002.
15. C. Gkantsidis, M. Mihail, and E. Zegura. Spectral analysis of Internet topologies. *IEEE INFOCOM*, 2003.
16. C. Jin, Q. Chen, and S. Jamin. Inet: Internet topology generator. *Techical Report UM CSE-TR-433-00*, 2000.
17. D. Karger. Randomization in graph optimization problems: A survey. *Optima*, 58:1–11, 1998.
18. G. Karypis. Multilevel hypergraph partitioning. *Technical Report, Department of Computer Science, University of Minnesota: 02-025*, 2002.
19. G. Karypis and V. Kumar. A fast and high quality scheme for partitioning irregular graphs. *Technical Report, Department of Computer Science, University of Minnesota: 95-035*, 1995.
20. V. Krishnamurthy, M. Faloutsos, M. Chrobak, L. Lao, J.H. Cui, and A. G. Percus. Reducing large internet topologies for faster simulations, 2005. UC Riverside, Technical Report.
21. Vaishnavi Krishnamurthy, Junhong Sun, Michalis Faloutsos, and Sudhir Tauro. Sampling internet topologies: How small can we go? In *International Conference on Internet Computing, Las Vegas*, 2005.
22. L.Subramanian, S.Agarwal, J.Rexford, and R.Katz. Characterizing the Internet hierarchy from multiple vantage points. *Proc. IEEE INFOCOM*, 2002.
23. A. Medina, A. Lakhina, I. Matta, and J. Byers. Brite:an approach to universal topology generation. *MASCOTS*, 2001.
24. A. Medina, I. Matta, and J. Byers. On the origin of powerlaws in Internet topologies. *ACM SIGCOMM Computer Communication Review*, 30(2):18–34, April 2000.
25. M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1(2), 2004.
26. C. R. Palmer, P. B. Gibbons, and C. Faloutsos. Anf: A fast and scalable tool for data mining in massive graphs. *SIGKDD*, 2002.
27. G. F. Riley, M. H. Ammar, R. M. Fujimoto, K. Perumalla, and D. Xu. Distributed network simulations using the dynamic simulation backplan. *International Conference on Distributed Computing Systems 2001 (ICDCS'01)*, 2001.
28. G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos. Power-laws of the Internet topology. *IEEE/ACM Trans. on Networking*, August 2003.
29. H. Tangmurankit, R. Govindan, S. Jamin, S. J. Shenker, and W. Willinger. Network topology generators: Degree-based vs structural. *SIGCOMM*, 2002.
30. B. M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, December 1988.
31. E. Zegura, K. Calvert, and S. Bhattacharjee. How to model an Internetwork. *IEEE INFOCOM*, 1996.
32. E. W. Zegura, K. L. Calvert, and M. J. Donahoo. A quantitative comparison of graph-based models for Internetworks. *TON*, 5(6), December 1997.