Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 4284–4302
Computer Networks
www.elsevier.com/locate/comnet

Sampling large Internet topologies for simulation purposes

Vaishnavi Krishnamurthy [a], Michalis Faloutsos [a,*], Marek Chrobak [a], Jun-Hong Cui [b], Li Lao [c], Allon G. Percus [d]

[a] *Department of Computer Science and Engineering, University of California Riverside, Riverside, CA 92521, United States*
[b] *University of Connecticut, Storrs, United States*
[c] *U.C. Los Angeles, United States*
[d] *Los Alamos National Labs, United States*

Received 5 December 2006; received in revised form 23 May 2007; accepted 5 June 2007
Available online 27 June 2007

Responsible Editor: M. Smirnow

Abstract

In this paper, we develop methods to ‘‘sample’’ a small realistic graph from a large Internet topology. Despite recent activity, modeling and generation of realistic graphs resembling the Internet is still not a resolved issue. All previous work has attempted to grow such graphs from scratch. We address the complementary problem of shrinking an existing topology. 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 an Internet graph by as much as 70% while maintaining its important properties. Second, we show that our reduced graphs compare favorably against construction-based generators. Finally, we successfully validate the effectiveness of our best methods in an actual performance evaluation study of multicast routing. Apart from its practical applications, the problem of graph sampling is of independent interest.

Keywords: Graph modeling; Graph sampling; Graph properties

1. Introduction

Small graphs that resemble the Internet topology are required in 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 researchers 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 more accurate performance predictions for the Internet graphs of the future.

* Corresponding author. Tel.: +1 951 827 2480.
E-mail address: michalis@cs.ucr.edu (M. Faloutsos).

1389-1286/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2007.06.004

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 [26,27].

Interestingly, 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 currently in use grow a graph incrementally, an approach that we call *constructive*. This area has seen unprecedented activity since the discovery of skewed degree distributions in the Internet topology [17]. The generators either use ''biased'' or *preferential* growth policy [2,5,8,16,35] or force a power law degree distribution [1,24]. These constructive method produce synthetic graphs by focusing on matching their degree distributions with that of real Internet instances, and they often fail to match other topological properties, as multiply documented in [8,12,22,24,45].

In this paper, we address the following problem: we want to ''sample'' a real topology[1] to produce a smaller graph. We examine the topology at the inter-domain or Autonomous Systems (*AS*) level, as we explain later. The overarching goal of our approach is very practical: we want the simulations on the sampled graph and the initial larger graph to lead to the same conclusions.[2] This is a novel problem in the Internet modeling community, although some related work in other areas exists [26,27,32,29]. We call our approach of generating a graph *reductive*. Very informally, the intuition is that our approach should ensure that it does not''-destroy'' the existing properties, and it is reasonable to expect that a suitably refined and statistically fair reduction method should be able to accomplish this task. In contrast, the constructive methods face the challenge of first identifying and then reproducing 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 becomes 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

- the real Internet instance of the same size (thus ''reversing'' the evolution of the Internet), or
- 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 [23]. Studies [23] suggest that even though every Internet instance at the AS level has power law characteristics, there exists a variation 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. We use two types of metrics for comparing graphs. The first type includes various graph invariants, for example average degree. The second type of metrics is based on comparing the relative performance of two multicast routing protocols in the two graphs. It turns out that for these metrics the performance results on the large Internet is comparable with those on the smaller Internet instance. This simplifies the comparison: the metrics on the reduced graph should match the performance of the larger Internet. Note that in our study we consider the metrics that have been suggested, selected and widely used by research efforts that study the topology of complex networks as in [8,16,35,1,24,42,34] and in the related book [39].

The contribution of this paper is three-fold: (i) we provide efficient graph sampling algorithms, (ii) we compare our reduction methods against constructive methods, and (iii) we compare our graphs using network protocol simulation.

1.1. 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

[1] To make it more specific, the current Internet has more than 18,000 Autonomous Systems. The smallest available Internet instance (from 1997) has about 3300 nodes, but even this size is computationally expensive, if not prohibitive, for some types of simulations such as BGP simulations or flow level simulations [14,43,21].

[2] The outcome of a simulation study is usually: (a) an observed trend or a relationship between two model parameters, and (b) the order of protocols according to a performance metric. We want the simulation on the reduced graph to lead to the same outcome as the simulation on the larger graph. This provides an additional indication that the reduced graph will be useful in simulation studies.

methods and (c) exploration methods. Our work yields the following results:

- Our best methods reduce the graph size by up to 70% in the number of nodes, while faithfully preserving all desired topological properties. Our methods seem 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.

1.2. Comparison of reductive and constructive methods

We compare our best reduction methods with commonly used constructive methods and find that our methods match more accurately the properties of the real Internet instances. Among the constructive methods, we find Inet [24] to perform best. Inet takes as input the available real instances. However, Inet does not generate graphs with less than 3300 nodes, while our methods can produce arbitrarily small graphs.

1.3. Network protocol simulation

We successfully validate the effectiveness of our best methods in a real-world performance evaluation of multicast routing. The performance comparison leads to the same conclusions[3] using the reduced graphs as with the real Internet instance. For example, we see that the behavior of the multicast protocol in the real and the reduced graph (with our best method) is similar, as we see later in detail. We consider this to be supplementary evidence for the validity of our approach.

In carrying out this work, we develop a software tool with these graph reduction capabilities, which we provide as an aid towards efficient and realistic simulations. The tool has already been used successfully for research purposes in a few studies in our institution. Our tool will be publicly available for research purposes. Finally, note that this work is a more extended version of our earlier work [28]. This version has: (a) the complete proof in Section 4, which was omitted in the earlier version, (b) a more extensive set of experiments and plots, namely 12 more plots, and (c) a more comprehensive list of previous and related efforts.

1.3.1. Our work in perspective

We compare the performance of our reductive methods with constructive methods and find that our approach works favorably. 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.

We stress that the reductive approach should not be thought of as a "competitor" to the constructive generators. Rather, both approaches complement each other, and, comprehensive studies may want to employ both types of generators to produce a variety of graphs for simulations.

The rest of the paper is organized as follows. In Section 2, we present the necessary background. In Section 3, we describe our reduction methods. In Section 4, we provide analytical arguments for the preservation of the degree distribution's power law. Section 5 discusses the performance of our methods with respect to various topological metrics. In Section 6, we compare our reductive approach to the constructive methods. In Section 7, we use our best methods in multicast routing study. Section 8 concludes the paper and proposes some future work.

2. Background and metrics

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

2.1. Internet instances

The Internet is divided into autonomously administered domains or Autonomous Systems

[3] However, we agree with Floyd and Paxson's opinion that simulations should be used mainly for qualitative and trend-related conclusions [18].

(AS). In our study, we focus on the AS level topology. We model the Internet as an undirected graph whose nodes are AS's and whose edges are interdomain connections. We note here that there have been efforts [20,31,6] to model the Internet as a directed graph by including business relationships. Whether our reductive methods extend to the directed model is a topic for further research.

Our real data come from the Oregon Routeviews project [19]. This is the archival data used frequently 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 [12] archive which spanned only over three months. Internet instances are labeled in the format IYYMMDD, where YYMMDD represents the collection date, with year, month and day represented by the two last digits each. For example, the instance collected on 07 May 2001 is named I010507. We use real Internet instances [19] from November 1997 to March 2003 in our experiments.

Note that in our study we use a large number of Internet instances spanning 6 years of the topological evolution namely from 1997 to 2003. Because of that, we are confident that our results are relevant to a narrow window of the evolution of the topology. Here, we mainly work with the evolution over the time span 1997–2001, but we also examine instances obtained[4] in 2003.

2.2. Graph properties

There are several graph properties that are used to capture the nature of real graphs. We use most of the known metrics to compare the realism of reduced graphs [17,22,40]. These properties are considered necessary, but may not be sufficient to guarantee the realism of the produced graphs.

2.2.1. Average degree and its standard deviation

The average degree of a graph is defined as $2m/n$, where m is the number of links and n is the number of nodes. We use this as a metric to compare the density of the reduced graphs with the real Internet topology. It has been noted that the average degree increases over time, as the size of the Internet graph increases, growing from 3.42 in November 1997 to 3.93 in November 1999 (15% growth). At the same time, the size of the Internet approximately doubled (100% growth). To measure how the degrees are distributed around the average, we also examine the standard deviation of the degree.

2.2.2. Degree distribution

Power laws have been used to approximate[5] the skewed degree distribution [17], which is empirically observed. 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 kth ranked node is the one with the kth highest degree. Degree exponent[6] is the slope of the log–log plot of the degree frequency versus degree. Note that the two power laws are theoretically equivalent, if the distributions are perfect power laws. In practice, they provide slightly different approximate views of the real degree distribution.[7] In this metric, we check the existence of power laws and then compare the value of the exponent of the power laws [36]. Power laws are approximations whose accuracy is typically quantified by the correlation coefficient. The correlation coefficient is a metric of the accuracy of the approximation and an coefficient of more than 97% is often considered a reasonable approximation.

2.2.3. Spectral analysis

Gkantsidis et al. [22] 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. It subsumes the clustering coefficient metric that was used before [8].

In more detail, spectral analysis examines the eigenvectors corresponding to the largest eigenvalues of the normalised transposed adjacency

[4] Note the reviewers: we will be happy to repeat our experiments with more recent instances, but we are confident that the results will not change in any significant way.

[5] Chen et al. [12] created a more complete Internet graph at the BGP level, but recent work by Siganos et al. [23] shows that the power laws hold with 99% correlation coefficient even with the new graph.

[6] We use the reverse cumulative distribution function of power law 2, which is more robust than the cumulative distribution function [23].

[7] The correlation coefficient of the power law fit was verified by the authors of [12], who use more metrics to examine the quality of the fit.

matrix of an entire topology. The plot depicts a number (say, 100) of the largest eigenvalues, which correspond loosely to the eigenvectors of the main clusters in the topology. It is found that the clustering properties (the corresponding plot) have not changed significantly despite the Internet growth [22].

2.2.4. Neighborhood function and hop-plot

The neighborhood function $G(h)$ is defined as the number of pairs of nodes within at most h hops from each other [17]. The neighborhood function is a measure of the fan-out of the graph. The distribution of distances and hop-plot does not change over time [23], which makes it a good metric for comparison. Here we plot the percentage of node pairs reachable from each other within h hops.

Note that more recently there have been some other metrics that have been proposed for the comparison of graphs [34], and a new approach for generating graphs [33] based on the correlations of the degree of neighboring nodes.

2.3. Graph generators

Early graph generators failed to match the skewed degree distribution [9,15,48,49]. Several recent generators build topologies with power law degree distribution in mind [1,5,8,24]. The pioneering Barabasi–Albert model [5] generates a graph through preferential attachment: in attaching new nodes to existing ones, it favors high-degree nodes. Mitzenmacher provides an overview of current methods to generate power law distributions [38].

To illustrate the advantages of our methods, we compare the topology obtained by reducing the AS level Internet topology using our best reduction method with similar graphs generated by Inet [24], Waxman, Barabasi and the modified GLP heuristic [8].

Note that other generators and studies focus at the topology of the Internet at the router level, where each node represents an Internet router [30]. The topologies of interest are different in nature with the ones we examine, since they model the Internet at a different level of granularity, where the nodes (routers) have physical limitations in their ability to have large degrees.

2.4. Graph reduction

The problem of graph reduction and sampling appears in other disciplines, although with different objectives. For example, graph sampling has been used in graph partitioning in the context of distributed computing [26,27]. Randomized graph sampling has been used to solve different graph problems such as min-cut approximation [25]. Despite some algorithmic similarities, these methods cannot be applied directly to our graph reduction problem. More recently, and after the conference version of our work [28], a datamining study [29] examines sampling of large graphs of multiple different origins.

2.5. Multicast routing

As an additional test of our reduction methods, we performed some specific network simulations, to see whether the experimental results are similar for the real and synthetic graphs. We first use our graphs in comparing two different multicast routing algorithms, namely, source based tree (SBT) and core based tree (CBT) [3,47]. Then, we compare the efficiency gain of multicast routing over unicast routing for reduced and real graphs to evaluate our graph reduction methods.

We choose to use a performance evaluation of multicast routing algorithms, which has been a research topic[8] [7,47,11]. The reason is that multicasting is more sensitive to the topological properties of the graph compared to point-to-point connections [9]. In a nutshell, multicast routing establishes a communication tree between a group of member nodes. There are two approaches for supporting many-to-many group communications, where every member is also a source of data. The Source Based Tree (SBT) approach creates a separate tree for each source. The Core Based Tree (CBT) approach creates a single bidirectional tree which carries the data packets of all sources.

We compare the two routing protocols using commonly used metrics. We do not argue that these are the most important metrics in a multicast simulation. However, these are metrics that have been used in multicast protocol evaluations [10,47,49]:

Delay ratio: For a distribution tree, we measure the average delay from a source to a receiver. This

[8] We stress again that our topologies are at the AS level and not at the router level. Many multicast protocols are designed for intra-domain operation and they would need to be evaluated with router level topologies. Note that inter-domain multicast routing is also of interest and several protocols have been proposed [46,4].

metric captures the end-to-end delay that each receiver experiences, which is important for real-time interactive applications. For simplicity, we assume that the delay of a path is proportional to the path length. The delay ratio is the average SBT delay over the average CBT delay.

Tree cost ratio: The tree cost is measured by the number of links in a multicast distribution tree. It quantifies the efficiency of the routing scheme. The tree cost ratio is the average SBT tree cost over the CBT tree cost.

Finally, we compare our graphs using *multicast efficiency* metric, which is normally measured by the ratio of the total number of multicast links in the distribution tree to the total number of unicast hops [11,13]. Chuang and Sirbu [13] proposed a power law relationship to express multicast efficiency in terms of the number of members (or group size). It was later proved by a more rigorous study [41,37] that Chuang–Sirbu power law is a reasonable approximation for multicast groups of small to moderate size. We use the multicast efficiency of SBT-based multicast routing algorithm and its power law exponent to validate the effectiveness of our reduction methods.

3. Graph reduction methods

This section presents our approach for sampling a real AS level Internet 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: we reduce the graph iteratively by removing a percentage of the graph in each stage. The input is the initial graph G with n nodes and m edges, the percentage s of nodes that has to be removed at every stage, and the total percentage P of nodes to be deleted from the graph. A stage consists of several steps, in which we remove either one edge or one vertex selected according to the method. After each stage, connected components are found and the largest connected component is retained. The procedure stops when the graph G reduces to a graph with 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 3–5% of the nodes at each stage was sufficient to achieve the desired reduction. We calculate the size of the remaining graph only at the end of each stage in order to speed up the process. 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-β): In this method, with probability β we execute DRVE and with probability $(1 - \beta)$ we execute DRE. In particular, DHYB-1 is DRVE, and DHYB-0 is DRE.

The hybrid method was motivated by our initial studies showing that DRVE and DRE had opposite performances with respect to our metrics, namely when one of them underestimated a metric's target value then the other overestimated it. We consider nine values of β 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 all the merged nodes become neighbors of the new node.

This method bears some similarities to the random matching method [27] and the edge coarsening method [26]. 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 and proofs

We now prove that two of our reduction methods, DRE and DRV, preserve the degree power law. More specifically, we show that if an original graph G satisfies the power law, then the reduced graph G' satisfies it too, with the same exponent.

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 $n = \sum_d n_d$, $m = \frac{1}{2}\sum_d dn_d$, 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:

$$n_d = Cnd^{-\alpha}, \quad (1)$$

where $C = \left(\sum_{d=1}^{n} d^{-\alpha}\right)^{-1}$ and $\alpha > 1$ is the degree exponent. We wish to show that a similar property holds in G'. (Of course, power laws are empirical and they are true only approximately. In fact, the value of $Cnd^{-\alpha}$ in (1) may not even be integer. For convenience, we write (1) as an equation. One can then think of the n_d's as an approximation to the degree sequence, not the sequence itself.)

4.1. 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 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 (most are, in fact, one-degree nodes), 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 heavily concentrated around m', and the two processes have asymptotically the same behaviors.

4.1.1. Informal argument

Our goal in this section 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. Suppose that n is very large and $1 \ll d \ll n$. Roughly speaking, nodes in G with degree around d/p, say between $\left(d - \frac{1}{2}\right)/p$ and $\left(d + \frac{1}{2}\right)/p$, end up in G' with an expected degree of d. This range covers $1/p$ different degrees, and for degrees c close to d/p the values n_c are close to $n_{d/p}$. Therefore, 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.

We now present a more careful argument. Consider a vertex $v \in G$ with $\deg(v) = k$. Each edge incident to v is preserved with probability p, and removed with probability $q = 1 - p$. This is simply a Bernoulli process with success probability p, so the probability that v's degree in G' is d is:

$$P[\deg_{G'}(v) = d] = \binom{k}{d} p^d q^{k-d} = \frac{1}{p} P_{dk},$$

where $P_{dk} = \binom{k}{d} p^{d+1} q^{k-d}$. Therefore, using (1), we have:

$$n'_d = \sum_{k \geqslant d} n_k \binom{k}{d} p^d q^{k-d} = \frac{Cn}{p} \sum_{k \geqslant d} k^{-\alpha} P_{dk}. \qquad (2)$$

From now on, instead of degree distribution, we will deal with degree *frequencies* $f_d = n_d/n$. This allows us to do the calculations for the limit case, with $n \to \infty$. (This is the third and final approximation.) We can then formulate the problem as follows: let the sequence,

$$f_d = Cd^{-\alpha}, \qquad (3)$$

for $d > 0$, represent the degree frequencies in the original graph, where $C = \left(\sum_{d=1}^{\infty} d^{-\alpha}\right)^{-1}$. By (2), the degree frequencies in the reduced graph are represented by:

$$f'_d = \frac{C}{p} \sum_{k \geqslant d} k^{-\alpha} P_{dk}. \qquad (4)$$

The rest of this section is devoted to the proof that f'_d satisfies a power law $f'_d \approx Cp^{\alpha-1} d^{-\alpha}$, where $x \approx y$ means that x, y are equal except for small order terms, that is $|x - y| = o(d^{-\alpha})$.

We introduce a random variable Y_d such that $P[Y_d = k] = P_{dk}$ for $k \geqslant d$. The expectation of Y_d is $E[Y_d] = \eta_d = (d + q)/p$ and variance $V[Y_d] = (d + 1)q/p^2$. (This can be derived by noticing that $Y_d = X_{d+1} - 1$, where X_{d+1} is a random variable with the negative binomial distribution.) By the Chebyshev's theorem [44], letting $\lambda = (d + 1)^{3/4}$, we get:

$$P[|Y_d - \eta_d| \geqslant \lambda] \leqslant \frac{V[Y_d]}{\lambda^2} \leqslant \frac{q}{p^2\sqrt{d}}. \qquad (5)$$

First, in the lemma below, we estimate the expectation of $Y_d^{-\alpha}$, and then we use this lemma to estimate the values of the f'_d.

Lemma 1. *For any constant exponent* $\alpha > 1$ *and probability* $p \in (0,1)$, *we have*:

$$E[Y_d^{-\alpha}] = \sum_{k \geqslant d} k^{-\alpha} P_{dk} \approx \left(\frac{d}{p}\right)^{-\alpha}.$$

Proof. Divide the value of $E[Y_d^{-\alpha}]$ into the contributions of the tails, and the "middle" part, that is $E[Y_d^{-\alpha}] = T + M$, where,

$$T = \sum_{d \leqslant k < \eta_d - \lambda} k^{-\alpha} P_{dk} + \sum_{k > \eta_d + \lambda} k^{-\alpha} P_{dk},$$

$$M = \sum_{\eta_d - \lambda \leqslant k \leqslant \eta_d + \lambda} k^{-\alpha} P_{dk}.$$

Using $Y_d \geqslant d$ and inequality (5), the tails' contribution is:

$$T \leqslant d^{-\alpha} P[|Y_d - \eta_d| > \lambda] \leqslant d^{-\alpha-1/2} q/p^2 = o(d^{-\alpha}).$$

Since $T = o(d^{-\alpha})$, to complete the proof, it is sufficient to show that:

$$M \approx (d/p)^{-\alpha}. \qquad (6)$$

We show the "$\leqslant$" and "$\geqslant$" estimates separately. By the definitions of M and λ, we have:

$$M \leqslant (\eta_d - \lambda)^{-\alpha} \sum_{\eta_d - \lambda \leqslant k \leqslant \eta_d + \lambda} P_{dk} \leqslant (\eta_d - \lambda)^{-\alpha}$$
$$= (d/p)^{-\alpha} - o(d^{-\alpha}),$$

and, using (5),

$$M \geqslant (\eta_d + \lambda)^{-\alpha} \sum_{\eta_d - \lambda \leqslant k \leqslant \eta_d + \lambda} P_{dk}$$
$$\geqslant (\eta_d - \lambda)^{-\alpha} \left(1 - q/(p^2\sqrt{d})\right)$$
$$= (d/p)^{-\alpha} + o(d^{-\alpha}).$$

This completes the proof of (6) and the lemma. □

To complete the proof of our main result, note that, from the definition (4) of f'_d, we have $f'_d = \frac{C}{p} \sum_{k \geqslant d} P_{dk} = \frac{C}{p} E[Y_d^{-\alpha}]$. Using the estimate for $E[Y_d^{-\alpha}]$ in Lemma 1, we conclude that $f'_d \approx (C/p)(d/p)^{-\alpha} = Cp^{\alpha-1} d^{-\alpha}$.

Theorem 1. *Suppose that in the original graph G the degree frequencies satisfy* $f_d = Cd^{-\alpha}$, *for some constant* $\alpha > 1$. *Let* G' *be the reduced graph obtained by removing each edge in G with probability* $p \in (0,1)$. *Then in* G' *the* (*expected*) *degree frequencies satisfy* $f'_d \approx Cp^{\alpha-1} d^{-\alpha}$.

In other words, Theorem 1 states (modulo the approximations described earlier in this section) that the DRE method preserves the power law: if the degree sequence in G is $n_d = Cnd^{-\alpha}$, then after the reduction the degree sequence satisfies the power law with the same exponent α.

4.2. DRV and power law preservation

An analogous argument as for DRE can also be applied to DRV. We only outline an informal explanation here. (Using analysis similar to that in the previous section, one can turn it into a formal argument.) 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/p, the values n_c are close to $n_{d/p}$, we might anticipate that for d not too small, $n'_d \sim n_{d/p} = Cp^{\alpha-1}n'd^{-\alpha}$, preserving the power law with the same exponent.

5. Graph reduction evaluation

In this section, we examine the performance of our sampling methods through empirical study. The starting point of the reduction in most of the experiments presented 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, and we will show the results from some more recent instances. 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 in the x-axis. For example, if we reduce by 70% the I010507 graph, we would end up with about 3300 nodes which is nearly the same size as the I980124 graph, which we plot at x-axis value. Thus, as described in the introduction, we evaluate the quality of our methods by comparing generated graphs to the real Internet instances of the same (or comparable) size. Each data point in our plots represents the *average of 50 runs* with different randomization seed.

Our 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 to topological metrics described in Section 2. 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 Tables 1–4, we present the top performing methods according to some metrics. Variations of DHYB and DRV are consistently present. We have not shown the performance of all the DHYB methods here as the graph becomes very congested. As the value of β in DHYB is increased, the metric values linearly increased between DRE and DRVE.

Table 1
Comparison of winners in average degree

Average percent deviation	Methods
Within 5%	DRVE, DHYB-0.8
Between 5% and 15%	DRV, DHYB-0.6, -0.5

Table 2
Comparison of winners in rank exponent

Average percent deviation	Methods
Within 3%	DHYB-0.1, -0.6, -0.5, -0.8
Between 3% and 5.5%	DRV

Table 3
Comparison of winners in degree exponent

Average percent deviation	Methods
Within 3%	DHYB-0.1, DRE, DHYB-0.5
Between 3% and 5.5%	DHYB-0.6, -0.8, DRV

Table 4
Comparison of winners in hop-plot

Average percent deviation	Methods
Within 6%	DHYB-0.8
Between 6% and 11%	DHYB-0.5, -0.6, DRV

In the rest of this section, we present our empirical results in more detail.

Test 1: average degree – Figs. 1–4 Fig. 1 shows how average degree varies for deletion, contraction and exploration methods. Fig. 2 shows average degree of hybrid methods with $\beta = 0.1$, 0.5, 0.6, 0.8. Even though DRVE follows the evolution of the average degree closely, when we quantitatively calculated the average of the percentage deviations at various points along the graph, we found that DHYB-0.8 is better with an average percentage deviation of 4.2% followed by DRVE with 5%. Also DHYB-0.8 has nearly the same value of average degree at the 70% reduction point as that of Internet. 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 Figs. 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

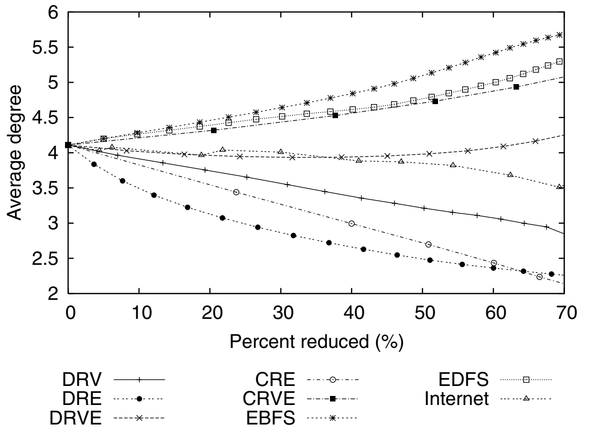


Fig. 1. Average degree comparison of deletion, contraction and exploration methods: starting instance I010507.

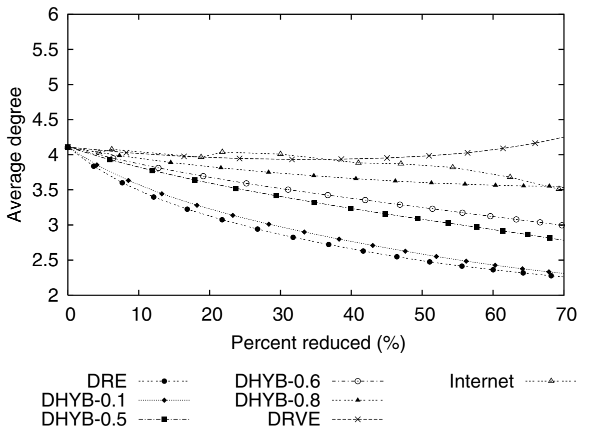


Fig. 2. Average degree comparison of hybrid methods: starting instance I010507.

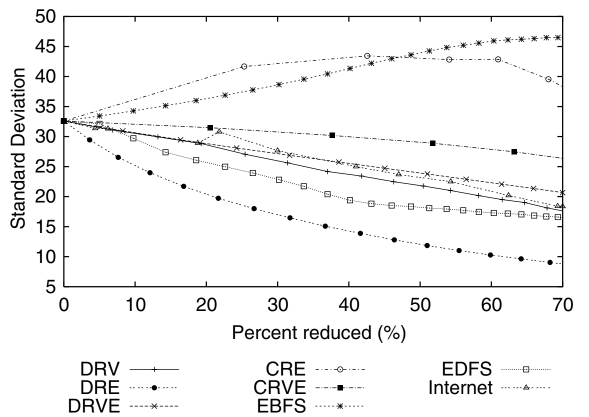


Fig. 3. Comparison of standard deviation of degree of deletion, contraction and exploration methods: starting instance I010507.

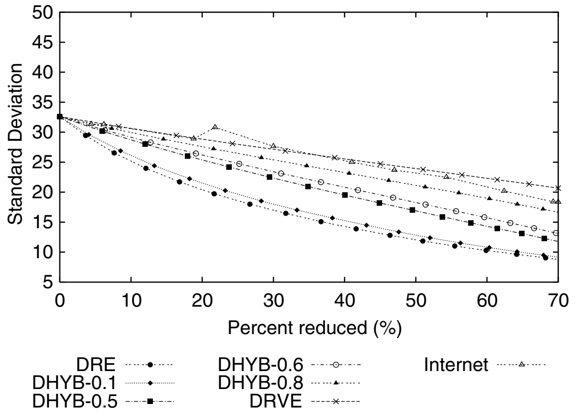


Fig. 4. Comparison of standard deviation of degree of hybrid methods: starting instance I010507.

Internet, so we conclude that they do not fare well in this metric comparison.

Figs. 3 and 4 show the test results for the standard deviation of the degrees. Note that here the best performing methods with respect to average degree also have similar degree distributions. In particular, DRVE and DHYB-0.8 are doing very well.

Summarizing Test 1, in terms of the approximating the average degree of the Internet, *DHYB-0.8 has the best performance.*

Test 2: exponent of rank power law – Figs. 5 and 6. The hybrid methods follow the variation of the Internet rank exponent very closely as is evident in Fig. 6. 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

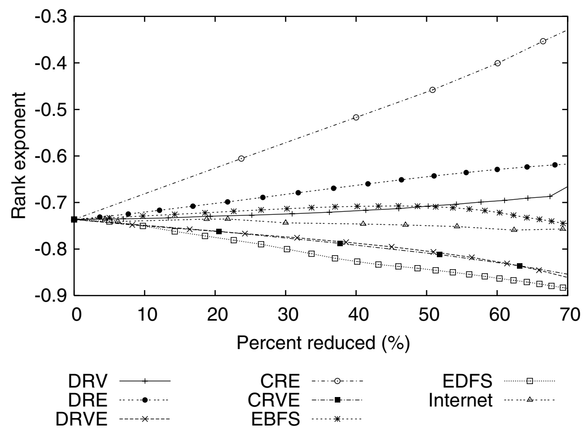


Fig. 5. Rank exponent comparison of deletion, contraction and exploration methods: starting instance I010507.

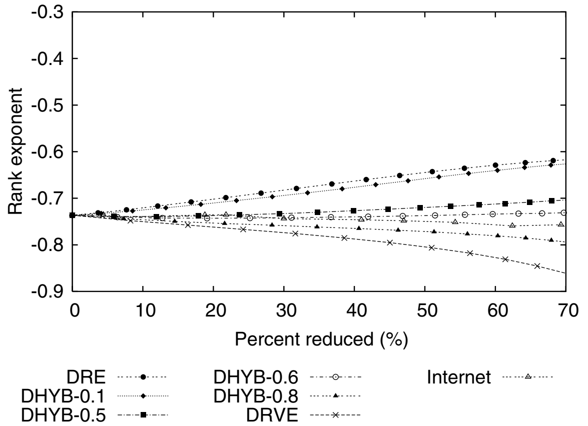


Fig. 6. Rank exponent comparison of hybrid methods: starting instance I010507.

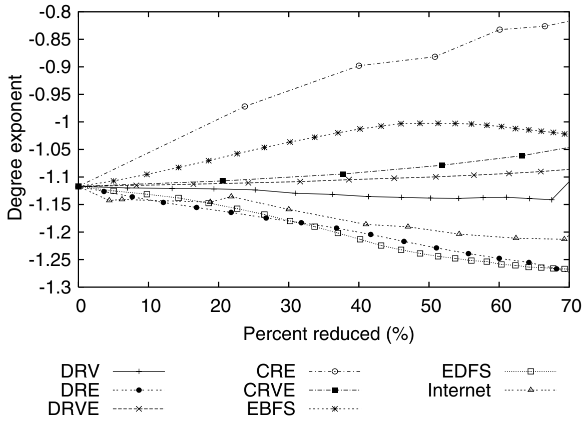


Fig. 7. Degree exponent comparison of deletion, contraction and exploration methods: starting instance I010507.

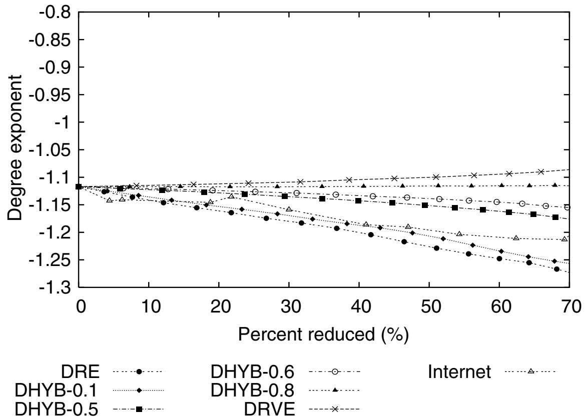


Fig. 8. Degree exponent comparison of hybrid methods: starting instance I010507.

the Internet line in Fig. 6. DRV was quite close, with an average percentage deviation of 5.2%.

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%, as we mentioned earlier. In our tests, DHYB-0.5, -0.6, -0.8 and DRV consistently maintained a high correlation coefficient. Even though it looks like EBFS performs equally well as DHYB-0.6, it turns out 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 Fig. 5 for rank exponent comparison, we exclude them from being viable solutions at this point. The table with all the correlation coefficients is not reported here due to lack of space.

Summarizing the results of Test 2, the hybrid methods match best the exponent of the Internet's rank power law, with *DHYB-0.6 begin the best method in this category*.

Test 3: exponent of degree power law – Figs. 7 and 8. The degree exponent of DHYB, DRV, and DRE is within 5.5% from the exponent of the Internet instance. In Fig. 7, we observe that DRE and DRV follow fairly closely the exponent of the real instances. Note that EDFS seems to perform well, but the correlation coefficient of its exponent is too low as we will see in the next test. In Fig. 8, we notice that DHYB-0.1 is very close to the Internet exponent, although for large reduction values it starts to deviate. As mentioned above, DHYB-0.5, -0.6 and -0.8 perform adequately in this test having a value within or close to 5% of the exponent of the Internet topologies.

We also determined the correlation coefficient of degree power law. As it turns out, 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. Note that EDFS closely resembles DRE in its performance but its correlation coefficient is below 96% and thus was not considered as one of the best methods.

Summarizing the results of Test 3, *DHYB-0.1, -0.5, -0.6, -0.8, DRE and DRV all match best the exponent of the Internet's degree power law.*

We evaluate the methods based on their average percent deviations from the Internet with respect to the four metrics we examined so far. Tables 1–4, we conclude that DHYB-0.8, -0.6, -0.5 and DRV are the best 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 for each topology. Thus, we focus now only on the 70% reduction point; we experimented with the other reduction points and the results were very similar. For the hop-plot and spectral analysis test discussed below, 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 4: hop-plot – Fig. 9. DHYB-0.8's hop-plot resembles the Internet's hop-plot well. In Fig. 9, we see that DHYB-0.8 is the one that resembles the Internet better than the other methods (not all methods are shown). In a more detailed examination, we find that the average percentage difference was lowest at 6% for DHYB-0.8, whereas the other methods (even the ones not shown in the figure) had a value close to 10%, refer Table 4.

Test 5: spectral analysis – Fig. 10. The spectral analysis of the four best methods, DHYB-0.8, -0.6, -0.5 and DRV, have been plotted in Fig. 10. Recall that the spectral behavior of the Internet topology is consistent over time [22]. 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 very closely, and significantly better than 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.

Test 6: robustness to initial instance – Figs. 11 and 12. We further investigated the stability of each method with respect to the input Internet instance. We took the most recent AS level Internet topology I030313 with 15,026 nodes and 31,200 edges, thus an average degree of 4.15. We applied the seven

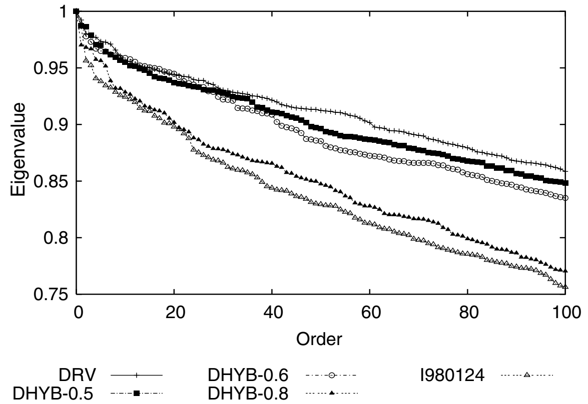


Fig. 10. Spectral analysis of 70% reduced DHYB-0.5, -0.6, -0.8, DRV and Internet instance I980124.

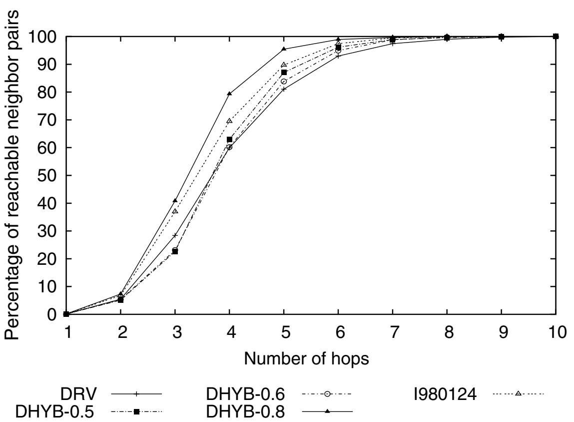


Fig. 9. Percentage of reachable neighbor pairs versus number of hops of 70% reduced DHYB-0.5, -0.6, -0.8, DRV and Internet instance I980124.

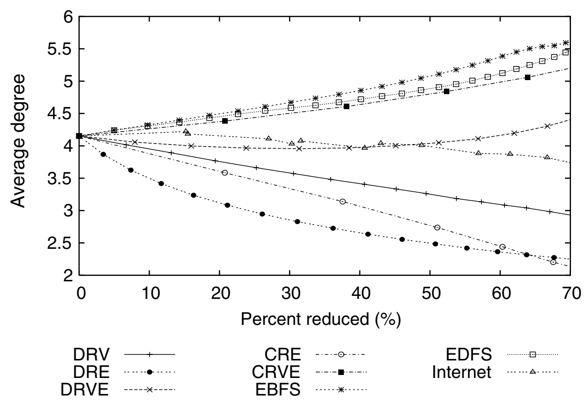


Fig. 11. Average degree comparison for deletion, contraction and exploration methods: starting instance for reduction is I030313.

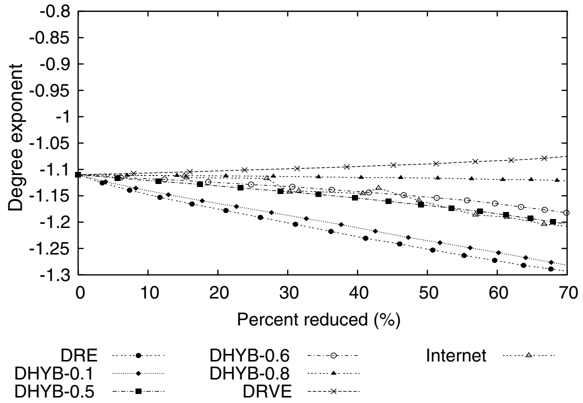


Fig. 12. Degree exponent comparison for hybrid methods: starting instance for reduction is I030313.

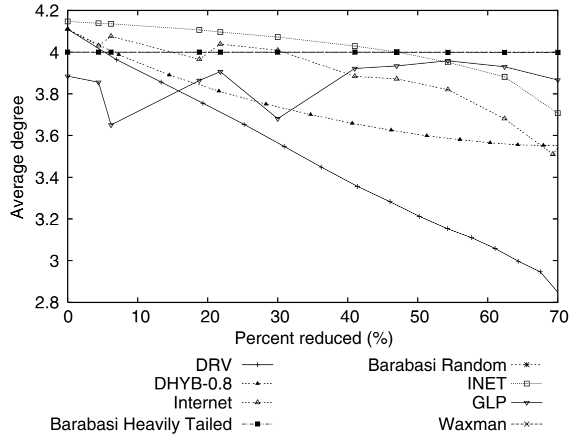


Fig. 13. Average degree comparison of reduction methods with constructive methods.

methods and confirmed that the trends exhibited by the methods are very similar to the trends obtained with starting instance I010507. We show only two of the six graphs here due to space constraints. Fig. 11 shows the average degree comparison of deletion, contraction and exploration methods, and Fig. 12 shows the degree exponent of the reverse cumulative distribution for the comparison of hybrid methods. Similar results were obtained when tested with seven different topology instances from the year 1999 up to 2003 but the plots are not shown here due to lack of space. Summarizing, *all the methods are insensitive to the choice of the initial instance*.

6. Reductive versus constructive methods

In this section, we compare our reduction methods with existing well-known constructive generators. Using the same metrics as in Section 5, we have compared the topologies reduced by our best reduction methods, namely DHYB-0.8 and DRV, with topologies from other topology generators like Inet, Waxman, Barabasi and Generalized Linear Preference (GLP) [8].

We have generated topologies by reducing the Internet instance I010507 using DHYB-0.8 and DRV. We have also generated topologies using Inet, Waxman, GLP, Barabasi method (with heavily tailed and random node placement options) of sizes similar to the various reduction points. For brevity, we show results only for five selected metrics.

Test 7: average degree – Fig. 13. 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 [19] that we use, and forces this degree distribution. DHYB-0.8 is the next best method. DRV does not follow the variations in the Internet but decreases in value linearly, unlike GLP which varies haphazardly with no specific pattern. Both Barabasi and Waxman generates topologies with an average degree of four independent of the size of the graph. Overall, *in terms of the average degree, Inet and DHYB-0.8 are closest to the Internet*.

Test 8: exponent of rank power law – Fig. 14. The hybrid method follows the variation of the Internet rank exponent very closely, as is evident in Fig. 14. 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

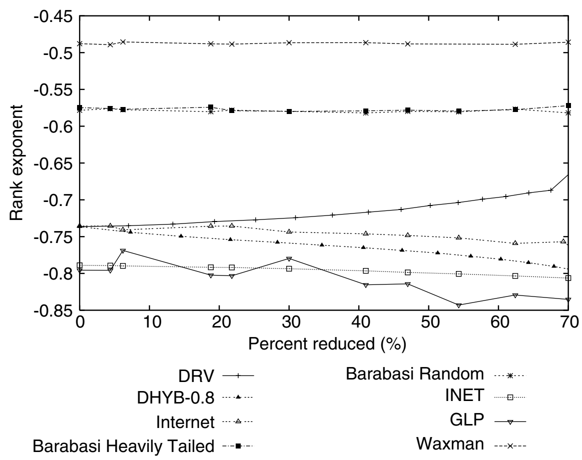


Fig. 14. Rank exponent comparison of reduction methods with constructive methods.

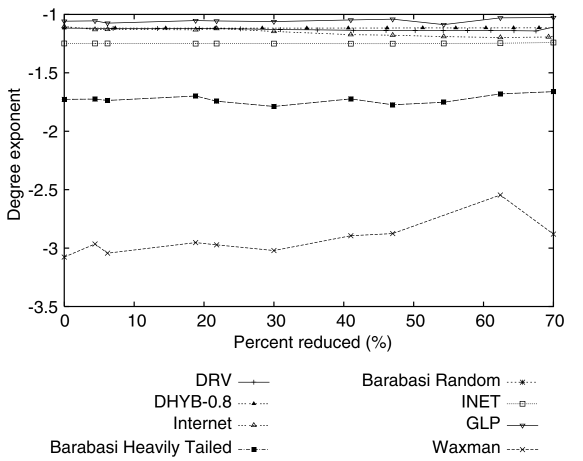


Fig. 15. Degree exponent comparison of reduction methods with constructive methods.

value for the exponent irrespective of the size of the graph. DRV has values higher than the Internet and is the next best method. For Barabasi model, 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 Barabasi topologies. Overall, we determined that, *in this category, DHYB-0.8 is the best method.*

Test 9: exponent of degree power law – Fig. 15. As we can infer from the plot in Fig. 15, most of the methods perform well, except for Waxman and Barabasi. 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,

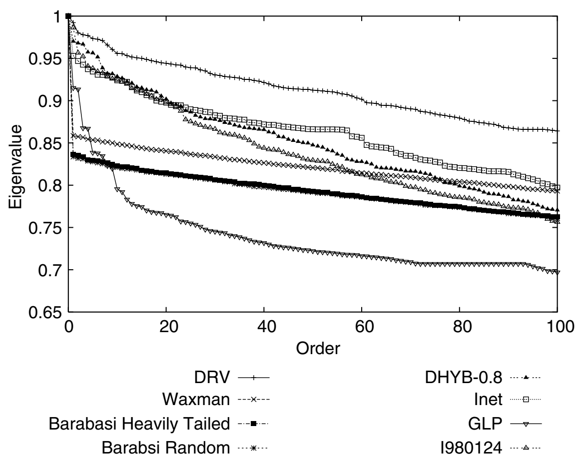


Fig. 16. Spectral analysis of reduction methods versus constructive methods.

respectively. Overall, *DYB-0.8 and DRV seem to be best*.

Test 10: spectral analysis – Fig. 16. Synthetic generators like Barabasi, GLP [8,36] have not only smaller eigenvalues compared to the Internet but also a slope value that is very different from the AS level Internet topology [22]. Gkantsidis et al. [22] 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. The eigenvalues of Inet does not decrease gradually unlike the Internet but instead exhibits sharper trends. Thus, overall, in this category, *DHYB-0.8 seems to be the best*.

6.1. Conclusion

We find that DHYB-0.8 is the best method followed by Inet and DRV. We note that Inet does not generate graphs below 3000 nodes. This could be related to the fact that Inet actually uses all the available instances from the RouteViews archive [19] in order to calibrate its intended graph metrics. The smallest instance (collected on 15 November 1997) in the archive has 3037 nodes.

With this limitation, DHYB-0.8 seems the best choice for small graphs. We actually create and use such a small topology with DHYB-0.8, when compare topologies using performance metrics in the next section. More specifically, we have reduced the I980124 graph by about 50%, thus generating a topology with 1500 nodes. We show that even using this small graph, we can obtain realistic simulation conclusions.

7. Simulation of multicast routing

As an ultimate test, we use our reduced AS level graphs in network simulations, which have been conducted in actual research studies [47,7,11]. We choose to use multicast routing for this purpose since it is more "sensitive" to the topology than, say unicast connections, and for this reason it has been used before in topological studies [9]. We would like to stress again that our purpose is not to evaluate the multicast algorithms, but to test the topological properties of the graphs.

We compare the performance of two multicast algorithms, namely, Source Based Tree (SBT) and Core Based Tree (CBT), on real and reduced graphs and evaluate the multicast efficiency metric on these graphs. Our results lead to the following conclusions:

Observation 1: The performance of the large and small real Internet instances are similar for the particular experiments and metrics that we use.
Observation 2: Our reduction methods produce graphs that yield the multicast performance closer to real Internet instances than most other topology generators tested.

Recall from Section 2 that we use cost ratio and delay ratio to compare two approaches for supporting one-to-many group communications. SBT creates a tree for each source, while CBT creates a single tree for the whole group. In the CBT approach, we choose as the core one of the member nodes uniformly. Note that the selection of the core is a separate research issue, outside the scope of this paper. In addition, we use the Chuang–Sirbu power law (described in Section 2) to verify our reduction methods. For each metric, we conduct two series of experiments.

Series A: In this series, we vary the group size of a multicast group. Since the size of the multicast group is an important parameter in multicast studies, we first compare the performance of reduced graphs with the performance of a real instance I980124 of the same size (of approximately 3200 nodes), as the multicast group size increases. The initial graph for the reduction is I010507 with 10,966 nodes and it is reduced by 70% using our best methods DHYB-0.8 and DRV. As a reference, we also conduct the same experiments for synthetic graphs with comparable size, which are generated by several commonly used topology models (i.e., INET, Barabasi, GLP, and Waxman). We mention earlier that we want to generate graphs of sizes smaller than the real Internet size available in any data archive. For this purpose, in this series we reduce the I980124 by 50% using DHYB-0.8 (contains about 1600 nodes) and compare its performance with other topologies. For this series, we vary the group size from 0.125% to 32% of the graph size, or from about 4 to about 1000 nodes. For each group size, we repeat the experiment 100 times.
Series B: In this series, we fix the group size to 16 members, the size of a large teleconference, and change the percentage of graph reduction from 10% to 70%. We select a series of Internet instances at regular time intervals of increasing size from I980124 to I010507, and produce a set of graphs with the same size from each of the topology generators mentioned above. Then, we compare the trends of the metric variation on real graphs and generated graphs. Thus, in Figs. 18 and 19, the Internet line consists of multiple real instances.

The delay ratio of DHYB-0.8 and DRV is very close to the delay ratio of the Internet instances. In Fig. 17, we plot the average end-to-end delay ratio versus the size of the multicast group. We observe that the delay ratio of the graphs generated by DHYB-0.8, DRV and GLP have average percent deviations within 5% of the delay ratio of Internet I980124, whereas the remaining graphs deviate from this real graph by more than 10%. To study how well our reduction methods preserve the properties of the original graph, we also plot the delay ratio of the Internet instance I010507. In this case, DHYB-0.8 performs better with an average percentage deviation of 1.8% followed by DRV with 8.8%. The 95% confidence intervals for all data points are less than 0.01.

Furthermore, we plot the average end-to-end delay ratio versus the percentage of graph reduction in Fig. 18. This graph leads to similar conclusions as Fig. 17: DHYB-0.8, DRV and GLP are consistently closer to the real graphs than the other methods.

The multicast efficiency of DHYB-0.8 and DRV matches the efficiency of Internet instance I980124

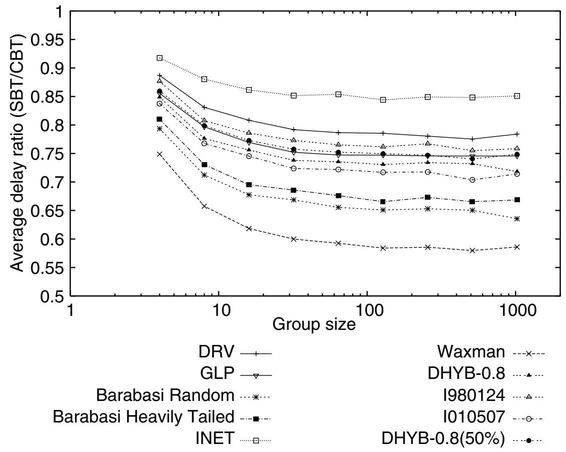


Fig. 17. Series A: Delay ratio of 70% reduced graphs, synthetic graphs, and Internet instances I980124 and I010507.

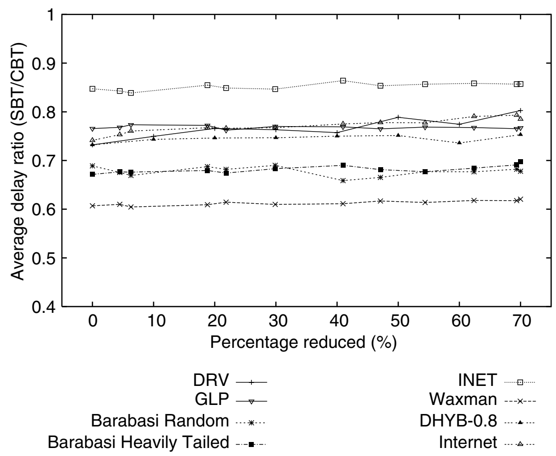


Fig. 18. Series B: Delay ratio versus the percentage of graph size reduction for reduced graphs, synthetic graphs, and Internet instances of various sizes; starting instance I010507.

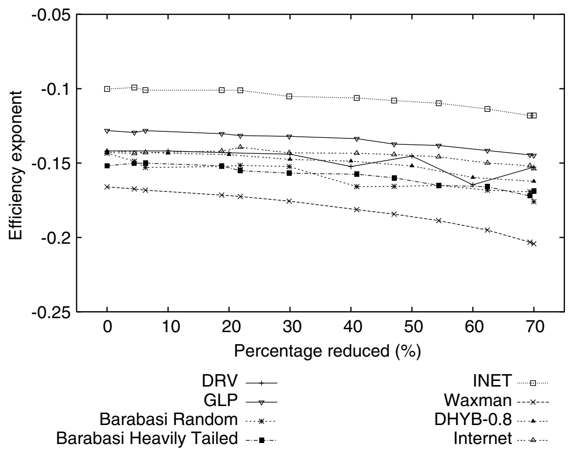


Fig. 19. Series B: Efficiency exponent versus the percentage of graph size reduction for reduced graphs, synthetic graphs, and Internet instances of various sizes; starting instance I010507.

very well. We present the efficiency exponents and correlation coefficients of Chuang–Sirbu power law in Table 5. From this table, we observe again that DRV, DHYB-0.8 and GLP are the best methods, since their exponent matches better the exponent of a real instance of the same size (I980124). INET and Barabasi topologies also follow I980124 closely with average percent deviations of approximately 6–8%.

In [37], it is shown that the efficiency exponent is not constant, but slowly increases with the graph size. We calculate the efficiency exponents for reduced graphs with different sizes and plot the results in Fig. 19. The result is consistent with previous observations, in that DHYB-0.8, DRV and GLP have the closest multicast performance as the real Internet instances. We also found that DHYB-0.8 and Internet have a steady decrease in efficiency exponent as the graphs are further reduced, whereas DRV does not exhibit this trend. Therefore, we conclude that DHYB-0.8 reduced graphs follow Internet more closely than DRV.

Table 5
Series A: Multicast efficiency comparison for reduction and constructive methods, with I980124

Methods	Exponent	Correlation coefficient
I010507	−0.138	0.994
I980124	−0.150	0.995
DHYB-0.8	−0.158	0.997
DRV	−0.149	0.995
INET	−0.118	0.991
DHYB-0.8 (50% red. I980124)	−0.174	0.997
GLP	−0.145	0.999
Barabasi random	−0.176	0.999
Barabasi heavily railed	−0.169	0.998
Waxman	−0.204	0.993

7.1. Tree cost ratio

Our experiments with the tree cost ratio also show similar results as the other two metrics, that is, *DHYB-0.8 and DRV work well*. The plots are not shown due to space limitations.

7.2. Conclusion

Graphs from our best methods can be used in simulation studies and the results will be similar to that of more computationally intensive simulations on the initial real graph.

Since the methods chosen by the graph metrics perform well according to the performance metrics, we are left to believe that the graph metrics manage to capture key topological properties.

8. 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.

The reduced graphs produce realistic simulation conclusions. We show that for DHYB-0.8 and DRV, the reduced graphs and the initial graph produce very similar results in our experiment.

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 accelerate by a factor of about 11 or 37, respectively. Furthermore, smaller graphs will require less memory which can decrease the simulation time further.

We have used our method in our lab for simulations with satisfactory results. The observed reduction in the simulation time was significant, especially for computationally intensive multicast applications.

8.1. Future work

First, we have made only a first step in proving bounds on the performance of our reductive methods. We believe that more analysis can pose many interesting theoretical problems and lead to results that can provide novel intuition about the topology. Second, we would like to experiment with a wider range of real simulation studies to validate our confidence in the realism of simulations on the reduced graphs. Finally, the proposed sampling methods are based on primarily random selection of nodes or edges. One could envision a more involved selection process that will take into consideration node properties.

Acknowledgements

This material is based upon work supported by the National Science Foundation ANI-0721889, and a INTEL and CISCO grant.

References

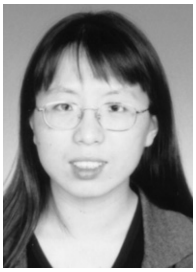
[1] W. Aiello, F. Chung, L. Lu, A random graph model for massive graphs, STOC (2000).
[2] R. Albert, A. Barabasi, Statistical mechanics of complex networks, Review of Modern Physics (2002).
[3] A. Ballardie, Core Based Trees (CBT): an architecture for scalable inter-domain multicast routing, ACM SIGCOMM (1993).
[4] A. Ballardie, Core Based Trees (CBT) multicast border router specification, 1997, Internet-draft: draft-ietf-idmr-cbt-br-spec-01.txt.
[5] A. Barabasi, R. Albert, Emergence of scaling in random networks, Science 8 (1999).
[6] G. Battista, M. Patrignani, M. Pizzonia, Computing the types of the relationships between autonomous systems, IEEE INFOCOM (2003).
[7] T. Billhartz, J.B. Cain, E. Farey-Goudreau, D. Fieg, S.G. Batsell, Performance and resource cost comparisons for CBT and PIM multicast routing protocols, IEEE Journal of Selected Areas in Communications 15 (3) (1997) 304–315.
[8] T. Bu, D. Towsley, On distinguishing between Internet power law topology generators, Infocom (2002).
[9] K. Calvert, E. Zegura, M. Doar, Modeling Internet topology, IEEE Transactions on Communications (1997) 160–163.
[10] Kenneth L. Calvert, Ellen W. Zegura, Michael J. Donahoo, Core selection methods for multicast routing, in: Proceedings of the International Conference on Computer Communications and Networks (ICCCN), 1995.
[11] R.C. Chalmers, K.C. Almeroth, Modeling the branching characteristics and efficiency gains in global multicast trees, in: Proceedings of IEEE INFOCOM'01, April 2001.
[12] Q. Chen, H. Chang, R. Govindan, S. Jamin, S.J. Shenker, W. Willinger, The origin of power laws in Internet topologies revisited, INFOCOM (2001).
[13] J.C.-I. Chuang, M.A. Sirbu, Pricing multicast communication: a cost-based approach, Telecommunication Systems 17 (3) (2001) 281–297.
[14] X.A. Dimitropoulos, G.F. Riley, Creating realistic bgp models, in: Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2003.
[15] M. Doar, A better model for generating test networks, in: Proceedings of the Global Internet, IEEE, November 1996.
[16] A. Fabrikant, E. Koutsoupias, C.H. Papadimitriou, Heuristically optimized trade-offs: a new paradigm for power laws in the internet (extended abstract), STOC (2002).
[17] M. Faloutsos, P. Faloutsos, C. Faloutsos, On power-law relationships of the Internet topology, in: Proceedings of the ACM SIGCOMM, Cambridge, MA, September 1–3, 1999, pp. 251–262.
[18] S. Floyd, V. Paxson, Difficulties in simulating the internet, IEEE/ACM Transactions on Networking (2001).
[19] National Laboratory for Applied Network Research, Online data and reports. Supported by NSF, 1998, http://www.nlanr.net.
[20] L. Gao, On inferring autonomous system relationships in the Internet, in: Proceedings of the Global Internet, November 2000.
[21] G.F. Riley, On standardized network topologies for network research, in: Simulation Conference 2002. Proceedings of the Winter, vol. 1, 2002, pp. 664–670.

[22] C. Gkantsidis, M. Mihail, E. Zegura, Spectral analysis of Internet topologies, IEEE INFOCOM (2003).
[23] G. Siganos, M. Faloutsos, P. Faloutsos, C. Faloutsos, Power-laws of the Internet topology, IEEE/ACM Transactions on Networking (2003).
[24] C. Jin, Q. Chen, S. Jamin, Inet: Internet topology generator, Technical Report UM CSE-TR-433-00, 2000.
[25] D. Karger, Randomization in graph optimization problems: a survey, Optima 58 (1998) 1–11.
[26] G. Karypis, Multilevel hypergraph partitioning, Technical Report, Department of Computer Science, University of Minnesota, 02-025, 2002.
[27] G. Karypis, V. Kumar, A fast and high quality scheme for partitioning irregular graphs, Technical Report 95-035, Department of Computer Science, University of Minnesota, 1995.
[28] V. Krishnamurthy, M. Faloutsos, M. Chrobak, L. Lao, J.H. Cui, A.G. Percus, Reducing large internet topologies for faster simulations, in: Proceedings of the IFIP Networking, Waterloo, Canada, 2005.
[29] Jure Leskovec, Christos Faloutsos, Sampling from large graphs, in: Proceedings of the ACM SIGKDD'06, 2006, pp. 631–636.
[30] L. Li, D. Alderson, W. Willinger, J. Doyle, R. Tanaka, S. Low, A first principles approach to understanding the Internet's router technology, in: Proceedings of the ACM SIGCOMM, 2004.
[31] L. Subramanian, S. Agarwal, J. Rexford, R. Katz, Characterizing the Internet hierarchy from multiple vantage points, in: Proceedings of the IEEE INFOCOM, 2002.
[32] D. Magoni, J.J. Pansiot, Internet topology modeler based on map sampling, in: Proceedings of the International Symposium on Computers and Communications, July 2002, pp. 1021–1027.
[33] P. Mahadevan, D. Krioukov, K. Fall, A. Vahdat, Systematic topology analysis and generation using degree correlations, in: Proceedings of the ACM SIGCOMM, Pisa, Italy, 2006.
[34] P. Mahadevan, D. Krioukov, B. Huffaker, X. Dimitropoulos, K.C. Claffy, A. Vahdat, The internet AS-level topology: three data sources and one definitive metric, ACM SIGCOMM Computer Communication Review 36 (1) (2006).
[35] A. Medina, A. Lakhina, I. Matta, J. Byers, Brite: an approach to universal topology generation, MASCOTS (2001).
[36] A. Medina, I. Matta, J. Byers, On the origin of powerlaws in Internet topologies, ACM SIGCOMM Computer Communication Review 30 (2) (2000) 18–34.
[37] P. Van Mieghem, G. Hooghiemstra, R. van der Hofstad, On the efficiency of multicast, IEEE/ACM Transactions on Networking 9 (6) (2001).
[38] M. Mitzenmacher, A brief history of generative models for power law and lognormal distributions, Allerton (2001).
[39] Mark Newman, Albert-Laszlo Barabasi, Duncan J. Watt, The Structure and Dynamics of Networks, Princeton Press, 2006.
[40] C.R. Palmer, P.B. Gibbons, C. Faloutsos, Anf: a fast and scalable tool for data mining in massive graphs, SIGKDD (2002).
[41] G. Phillips, S. Shenker, H. Tangumanarunkit, Scaling of multicast trees: comments on the Chuang–Sirbu scaling law, in: ACM Computer Communication Review, Proceedings of ACM SIGCOMM '99 Conference, vol. 29(4), 1999, pp. 41–51.
[42] M. Faloutsos Q. Yang, G. Siganos, S. Lonardi, Evolution versus intelligent design: comparing the topology of protein–protein interaction networks to the internet, in: Proceedings of the LSS Computational Systems Bioinformatics Conference (CSB'06), Stanford, CA, August 2006, pp. 299–310.
[43] G.F. Riley, M.H. Ammar, R.M. Fujimoto, K. Perumalla, D. Xu, Distributed network simulations using the dynamic simulation backplan, in: Proceedings of the International Conference on Distributed Computing Systems 2001 (ICDCS'01), 2001.
[44] C. Scheideler, Probabilistic methods for coordination problems, HNI-Verlagsschriftenreihe, vol. 78, University of Paderborn, 2000.
[45] H. Tangmurankit, R. Govindan, S. Jamin, S.J. Shenker, W. Willinger, Network topology generators: degree-based vs. structural, SIGCOMM (2002).
[46] D. Thaler, D. Estrin, D. Meyer, Border gateway multicast protocol (BGMP): protocol specification, 1998, Internet-draft: draft-ietf-idmr-gum-02.txt.
[47] L. Wei, D. Estrin, The trade-offs of multicast trees and algorithms, in: Proceedings of the International Conference on Computer Communications and Networks (ICCCN), 1994.
[48] E. Zegura, K. Calvert, S. Bhattacharjee, How to model an Internetwork, IEEE INFOCOM (1996).
[49] E.W. Zegura, K.L. Calvert, M.J. Donahoo, A quantitative comparison of graph-based models for Internetworks, IEEE/ACM Transactions on Networking 5 (6) (1997).

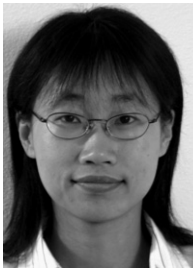
Vaishnavi Krishnamurthy graduated from University of California, Riverside with a Masters in Electrical Engineering. She then worked as a Research Consultant at University of Washington, Seattle in the Dept of Biostatistics where she helped develop front end and back end applications to collect and store clinical data from various hospitals across US and Canada. She then moved to NJ where she took up a software developer position at Vonage in the Database Systems team.

Michalis Faloutsos is a faculty member at the Computer Science Department in the University of California, Riverside. He got his bachelor's degree at the National Technical University of Athens and his M.Sc. and Ph.D. at the University of Toronto. His interests include, Internet protocols and measurements, multicasting, cellular and ad-hoc networks. With his two brothers, he co-authored the paper on powerlaws of the Internet topology (SIGCOMM'99), which is in the top 15 most cited papers of 1999. His work has been supported by several NSF and DAPRA grants, including the prestigious NSF CAREER award. He is actively involved in the community as a reviewer and a TPC member in many conferences and journals.

Marek Chrobak received his Ph.D. in Computer Science from Warsaw University in 1985. Since 1987 he has been a faculty member at the Department of Computer Science and Engineering, University of California, Riverside. His research interests include algorithm design and analysis, theory of computation, combinatorial optimization, and computational biology.



Jun-Hong Cui received her B.S. degree in Computer Science from Jilin University, China in 1995, her M.S. degree in Computer Engineering from Chinese Academy of Sciences in 1998, and her Ph.D. degree in Computer Science from UCLA in 2003. Currently, she is an assistant professor in the Computer Science and Engineering Department at University of Connecticut. Her research interests cover the design, modeling, and performance evaluation of networks and distributed systems. Recently, her research mainly focuses on exploiting the spatial properties in the modeling of network topology, network mobility, and group membership, scalable and efficient communication support in overlay and peer-to-peer networks, algorithm and protocol design in underwater sensor networks. She is actively involved in the community as an organizer, a TPC member, and a reviewer for many conferences and journals. She is a guest editor for ACM MCCR (Mobile Computing and Communications Review) and Elsevier Ad Hoc Networks. She co-founded the first ACM International Workshop on UnderWaterNetworks (WUWNet'06), and she is now serving on the WUWNet steering committee. She also serves as a TPC co-chair for the Wireless Ad-hoc and Sensor Networks Track of ICCCN 2007. She is a member of ACM, ACM SIGCOMM, ACM SIGMOBILE, IEEE, IEEE Computer Society, and IEEE Communications Society.



Li Lao received her B.S. degree from Fudan University, China in 1998. She received her M.S. and Ph.D. degrees in Computer Science from University of California, Los Angeles in 2002 and 2006, respectively. She joined Google Inc in April 2006. Her research focuses on multicasting, overlay network management, multicast modeling and performance evaluation.

Allon G. Percus is a scientist at Los Alamos National Laboratory, and former Associate Director of the Institute for Pure and Applied Mathematics at UCLA. He received his B.A. in physics from Harvard in 1992 and his Ph.D. from the University of Paris, Orsay in 1997. His main research interests are in discrete optimization and statistical physics. He is a co-founder of the Extremal Optimization methods, and his work on the stochastic traveling salesman problem has resulted in the most precise numerical estimates to date for asymptotic tour lengths. He has edited a volume on Computational Complexity and Statistical Physics, published by Oxford University Press in 2006.